

**APLIKASI TRANSLATOR MESSENGER
BERBASIS JAVA DAN GOOGLE TRANSLATE API**

SKRIPSI

**Diajukan kepada Fakultas Teknik
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Pendidikan**



Oleh :

NOVITA ANINDITA

NIM. 08520241003

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2013**

PERSETUJUAN

SKRIPSI

APLIKASI TRANSLATOR MESSENGR BERBASIS JAVA DAN GOOGLE TRANSLATE API

Oleh:

NOVITA ANINDITA

NIM. 08520241003

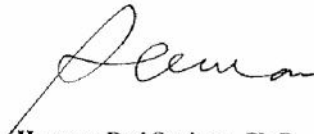
Telah disetujui untuk diuji Jurusan Pendidikan Teknik Elektronika, Program Studi Pendidikan Teknik Informatika, Fakultas Teknik Universitas Negeri Yogyakarta untuk memenuhi sebagian dari syarat guna memperoleh gelar Sarjana Pendidikan.

Pada tanggal 16 Januari 2013

Menyetujui,

Dosen Pembimbing

Tugas Akhir Skripsi



Herman Dwi Surjono, Ph.D.

NIP. 19640205 198703 1 001

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Novita Anindita

NIM : 08520241003

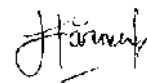
Program Studi : Pendidikan Teknik Informatika

Fakultas : Fakultas Teknik

Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat pernyataan, pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali sebagai acuan atau kutipan dengan mengikuti tata cara penulisan karya ilmiah yang berlaku.

Tanda tangan dosen penguji yang terdapat dalam lembar pengesahan adalah asli, apabila terbukti tanda tangan dosen penguji palsu, maka saya bersedia untuk memperbaiki dan mengikuti yudisium satu tahun lagi.

Yogyakarta, 16 Januari 2013
Yang menyatakan,



Novita Anindita
NIM. 08520241003

PENGESAHAN

LAPORAN TUGAS AKHIR SKRIPSI
APLIKASI TRANSLATOR MESSENGER
BERBASIS JAVA DAN GOOGLE TRANSLATE API

Disusun oleh:

Novita Anindita
NIM. 08520241003

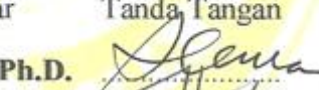


Telah Dipertahankan di depan Dewan Penguji Tugas Akhir Skripsi

Pada tanggal Januari 2012

Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar


Sarjana Pendidikan Teknik Informatika

SUSUNAN DEWAN PENGUJI TUGAS AKHIR SKRIPSI

Jabatan	Nama Lengkap dan Gelar	Tanda Tangan	Tanggal
Ketua Penguji	Herman Dwi Surjono, Ph.D.		4/3 2013
Sekretaris	Handaru Jati, Ph.D.		15/03 2013
Penguji Utama	Adi Dewanto, M.Kom.		6/3 2013

Yogyakarta, Januari 2013
Dekan FT UNY



Dr. Moch. Bruri Triyono
NIP. 19560216 198603 1 003 

MOTTO :

Keajaiban adalah nama lain dari kerja keras. #toTheBeautifulYou

Teruslah bermimpi karena dengan bermimpi kita punya arah kemana kita kan menuju. Tak peduli seberapa tua kita, tak peduli bagaimana keadaan kita. Hidup tanpa mimpi bagai air mengalir tanpa arah. Mimpi bukan sekedar angan-angan tapi keinginan pencapaian dengan kerja keras.

Hidup itu adalah manfaat. Selagi kita hidup, berusahalah untuk selalu bermanfaat bagi semua. Selalu membantu sekitar walaupun hanya hal kecil. Hidup itu aksi reaksi. Jika kita baik terhadap semua orang tentu semua orang akan baik kepada kita.

Selagi kita mampu, kelilingilah dunia. Bukan untuk berwisata, tapi ambillah setiap pelajaran hidup yang kita temui.

“Apapun yang terjadi dalam hidup ini, yakinkanlah bahwa ini cara Allah mencintaimu.” #MyQuote

PERSEMBAHAN :

Kupersembahkan karya kecil ini untuk:

1. *Kedua orang tuaku, atas do'a yang dipanjatkan disetiap sholat.*

2. *Kakak-kakak'ku tercintah.*

3. *Penghuni KOMBRÉ squad (ada permintaan disuruh nempel*

dipersembahan → nyyohi) ☺

ABSTRAK

APLIKASI TRANSLATOR MESSENGER BERBASIS JAVA DAN GOOGLE TRANSLATE API

Oleh:

Novita Anindita
08520241003

Tujuan dari penelitian ini adalah menghasilkan aplikasi messenger berupa perangkat lunak Aplikasi Translator Messenger. Kemudian menguji kelayakan aplikasi tersebut sebagai aplikasi yang memudahkan menerjemahkan bahasa atau *translator* dalam komunikasi *chatting*. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Java dan diintegrasikan dengan Google Translate API. Fungsi dari Google Translate API sebagai penerjemah bahasa secara otomatis dalam *messenger*.

Penelitian ini menggunakan metode penelitian dan pengembangan (*Research and Development*). Adapun langkah-langkah metode yang digunakan yaitu: analisis kebutuhan, desain, pengkodean, dan pengujian. Pengujian *alpha* dilakukan oleh 2 orang ahli rekayasa perangkat lunak. Sedangkan pengujian *beta* dilakukan oleh responden sebanyak 16 orang mahasiswa. Adapun teknik analisis yang digunakan adalah teknik analisis deskriptif kuantitatif. Deskriptif kuantitatif digunakan untuk menyatakan hasil perhitungan kelayakan yang berupa persentase.

Hasil analisis dari penelitian menunjukkan bahwa perangkat lunak Aplikasi Translator Messenger sudah valid dari segi *correctness* dan *reliability*. Pada pengujian *alpha*, perangkat lunak memperoleh persentase sebesar 90% dari segi *functionality* dan 85% dari segi *usability*. Sedangkan kelayakan perangkat lunak menurut pengguna sebesar 89,69% dari segi *functionality* dan 83,45% dari segi *usability*.

Kata kunci: Aplikasi, Messenger, Translator.

KATA PENGANTAR



Puji syukur kehadiran Allah SWT atas berkat dan karunia-Nya, sehingga penyusun dapat menyelesaikan laporan Tugas Akhir Skripsi ini. Pembuatan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan, Fakultas Teknik, Universitas Negeri Yogyakarta.

Penyusun menyadari bahwa laporan ini tidak akan tersusun dengan baik tanpa adanya bantuan dari berbagai pihak. Oleh karena itu, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu penyusun selama penelitian di lokasi maupun dalam penyusunan laporan ini. Ucapan terimakasih penyusun sampaikan kepada:

1. Kedua orang tuaku yang selalu memberi semangat dan do'a yang berlimpah.
2. Prof. Dr. Rochmat Wahab M.Pd., MA, selaku Rektor Universitas Negeri Yogyakarta.
3. Dr. Moch. Bruri Triyono, selaku Dekan Fakultas Teknik UNY.
4. Muhammad Munir, M.Pd., selaku Ketua Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY.
5. Dr. Ratna Wardani, selaku Ketua Program Studi Pendidikan Teknik Informatika Fakultas Teknik UNY.
6. Handaru Jati, Ph.D., selaku Koordinator Tugas Akhir Skripsi.
7. Herman Dwi Surjono, Ph.D., selaku Dosen Pembimbing Tugas Akhir Skripsi.
8. Teman-teman Prodi PT. Informatika UNY kelas E angkatan 2008 atas dukungan dan kerjasama yang diberikan.

9. Semua pihak yang tidak dapat penyusun sebutkan satu persatu yang telah membantu penyusun hingga tersusunnya laporan tugas akhir skripsi ini.

Berbagai upaya telah penyusun lakukan untuk menyelesaikan tugas akhir ini, namun penyusun menyadari bahwa dalam penyusunan laporan tugas akhir skripsi ini masih jauh dari kesempurnaan. Oleh karenanya, penyusun mohon maaf apabila dalam penyusunan laporan ini banyak kesalahan. Penyusun berharap semoga laporan ini berguna dan mendatangkan banyak manfaat bagi penyusun serta bagi para pembaca.

Yogyakarta, Januari 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PERNYATAAN	iii
HALAMAN PENGESAHAN	iv
MOTTO	vi
PERSEMBAHAN	vii
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
BAB I. PENDAHULUAN	
A. Latar Belakang.....	1
B. Identifikasi Masalah.....	3
C. Batasan Masalah	4
D. Rumusan Masalah.....	4
E. Tujuan Penelitian	4
F. Manfaat Penelitian	5
G. Spesifikasi Produk	6
BAB II. KAJIAN PUSTAKA	
A. Kajian Teori	7
1. <i>Messenger</i>	7
2. <i>Java</i>	11
3. Google Translate API	13
4. UML.....	15
5. Use Case.....	16
6. Class Diagram.....	17

7. Sequence Diagram	18
8. Activity Diagram	18
B. Unjuk Kerja	19
C. Kelayakan Aplikasi.....	20
D. Penelitian yang Relevan	26
E. Kerangka Berpikir	27
 BAB III. METODE PENELITIAN	
A. Desain Penelitian	29
B. Subjek dan Objek Penelitian.....	34
C. Tempat dan Waktu Penelitian.....	35
D. Responden	35
E. Teknik Pengumpulan Data	35
F. Instrumen Penelitian	37
G. Teknik Analisis Data	42
 BAB IV. HASIL PENELITIAN DAN PEMBAHASAN	
A. Hasil Penelitian	44
1. Analisis	44
2. Desain	45
3. Pengkodean	58
4. Pengujian	62
B. Deskripsi Program	63
C. Pembahasan	65
 BAB V. KESIMPULAN DAN SARAN	
A. Kesimpulan	75
B. Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN	80

DAFTAR TABEL

Tabel 1. Tabel Perbandingan Translator.....	14
Tabel 2. Rumus jumlah error perangkat lunak	23
Tabel 3. Nilai MTBF Microsoft Word	24
Tabel 4. Sub Karakteristik <i>Functionality</i>	25
Tabel 5. <i>List</i> pengujian <i>black-box</i>	37
Tabel 6. Pengujian berdasarkan <i>Functionality</i>	39
Tabel 7. Pengujian berdasarkan <i>Usability</i>	40
Tabel 8. Kuesioner pengujian <i>beta</i>	41
Tabel 9. Deskripsi Aktor.....	48
Tabel 10. Deskripsi <i>use case</i>	48
Tabel 11. Skenario <i>use case Login</i>	49
Tabel 12. Skenario <i>use case Sign up</i>	49
Tabel 13. Skenario <i>use case chatting</i>	50
Tabel 14. Skenario <i>use case sign out</i>	51
Tabel 15. Pengujian black-box	65
Tabel 16. Daftar kelas pada program.....	66
Tabel 17. Hasil pengujian <i>functionality</i>	70
Tabel 18. Skala kelayakan	71
Tabel 19. Hasil pengujian <i>usability</i>	71
Tabel 20. Hasil pengujian <i>beta</i>	72

DAFTAR GAMBAR

Gambar 1. Indikator Mc Call menurut Pressman (2010)	21
Gambar 2. Langkah-langkah metode penelitian.....	30
Gambar 3. Tabel identitas	46
Gambar 4. Tabel mes	46
Gambar 5. Use case sistem	47
Gambar 6. Activity diagram Chatting	52
Gambar 7. Activity diagram <i>Sign Up</i>	53
Gambar 8. Sequence Diagram <i>Login</i>	53
Gambar 9. Sequence Diagram <i>Chatting</i>	54
Gambar 10. Sequence Diagram <i>Sign Up</i>	54
Gambar 11. Class Diagram package chat.....	55
Gambar 12. Rancangan <i>interface chatser</i>	56
Gambar 13. Rancangan <i>interface login</i>	56
Gambar 14. Rancangan <i>interface menu utama</i>	57
Gambar 15. Rancangan <i>interface chatting</i>	57
Gambar 16. Rancangan <i>interface sign up</i>	58
Gambar 17. Halaman <i>Chatser</i>	59
Gambar 18. Halaman <i>Login</i>	59
Gambar 19. Halaman menu utama	60
Gambar 20. Halaman chatting.....	61
Gambar 21. Halaman Chatting user lawan.....	61

Gambar 22. Halaman Sign Up.....	62
Gambar 23. Penjumlahan baris kode.....	66
Gambar 24. Tampilan FindBugs	67

DAFTAR LAMPIRAN

Lampiran 1. SK Skripsi	81
Lampiran 2. Source code	82
Lampiran 3. Validasi Instrumen	132
Lampiran 4. Validasi Aplikasi.....	135
Lampiran 5. Hasil User	143

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi informasi telah mengubah cara berkomunikasi zaman dahulu secara konvensional menjadi komunikasi digital pada masa sekarang. Penggunaan teknologi membuat semua pekerjaan menjadi lebih mudah dibandingkan dengan cara konvensional. Tentunya hal itu berdampak pada informasi yang semakin cepat dan mudah diperoleh. Tuntutan zaman yang semakin berkembang mengharuskan semua pekerjaan dapat diselesaikan menggunakan hasil perkembangan teknologi informasi secara cepat, tepat, dan relevan.

Di dunia komunikasi, perkembangan teknologi yang semakin maju dapat dilihat dengan semakin berkembangnya berbagai fitur-fitur untuk lebih memudahkan pengguna dalam berkomunikasi. Zaman dahulu, jarak yang jauh menjadi kendala utama dalam berkomunikasi. Namun sekarang, jarak yang jauh bukanlah menjadi kendala utama lagi. Menurut Onong Uchyana Effendi (1993), komunikasi adalah proses penyampaian pesan oleh seseorang kepada orang lain untuk memberitahu, mengubah sikap, pendapat, atau perilaku baik secara langsung ataupun tidak langsung (melalui media). Sedangkan menurut Harold Laswell (1999), komunikasi adalah gambaran mengenai siapa, mengatakan apa, melalui media apa, kepada siapa dan apa efeknya.

Salah satu aplikasi komunikasi yang banyak digunakan adalah aplikasi *messenger*. Aplikasi *messenger* merupakan *tool chatting* yang mudah digunakan untuk berkomunikasi jarak jauh maupun jarak dekat dan lebih murah dibandingkan dengan mengirim pesan singkat. *Messenger* yang sering digunakan seperti Yahoo Messenger, Nimbuz, Google Talk, dan Skype. Yahoo Messenger dapat diakses jika pengguna mempunyai akun Yahoo, begitu juga dengan Google Talk, pengguna harus mempunyai akun Gmail. Berbeda dengan Yahoo dan Google, Nimbuzz dan Skype dapat diakses tanpa harus memiliki akun email tertentu. Aplikasi-aplikasi tersebut gratis bagi siapa saja. Penggunaan *messenger* pun terbilang cukup mudah, pengguna hanya perlu mengetikkan pesan yang akan dikirim. Namun pengguna harus *online* untuk dapat mengakses aplikasi-aplikasi tersebut.

Aplikasi-aplikasi *messenger* sudah menjadi tren dalam masyarakat, baik kalangan remaja, mahasiswa maupun pebisnis. Bagi kalangan pebisnis, aplikasi *messenger* sudah menjadi bagian dari pekerjaannya. Di era modern, para pebisnis tidak lagi harus bertatap muka untuk menjalin negoisasi ataupun kontrak kerja. Terkadang penggunaan aplikasi messenger dirasa sudah cukup mewakili untuk menjalin komunikasi secara mudah dan efisien. Aplikasi tersebut dapat digunakan untuk berbagai tujuan seperti bertransaksi dengan client, membeli peralatan baru, dan berkonsultasi. Tentunya pihak lawan bicaranya bukan hanya dari dalam negeri saja. Terkadang tidak sedikit, para pebisnis melakukan negoisasi dengan orang luar negeri. Perbedaan bahasa adalah masalah yang sering terjadi. Tentunya hal ini membuat komunikasi

menjadi kurang optimal bahkan dapat menyebabkan salah paham dalam berkomunikasi.

Sebenarnya, masalah untuk menerjemahkan bahasa dapat menggunakan fasilitas Google Translate. Namun penggunaannya dirasa kurang efisien dan efektif karena pengguna atau *user* harus berkali-kali membuka dan menutup Google translate untuk menerjemahkan bahasa.

Dari permasalahan di atas, dirasa perlunya suatu aplikasi yang dapat membantu pengguna *messenger* untuk menerjemahkan bahasa tanpa kesulitan. Sebuah aplikasi yang merupakan gabungan dari *messenger* dan *translator* bahasa. Oleh karena itu, Aplikasi Translator Messenger dibangun untuk menjawab permasalahan tersebut. Aplikasi ini dibuat agar memudahkan pengguna atau *user* dalam berkomunikasi melalui *messenger* dengan bahasa yang berbeda. Pengembangan aplikasi tersebut diharapkan dapat menjadi solusi dalam berkomunikasi bagi pengguna yang memiliki perbedaan bahasa

B. Identifikasi Masalah

Dari latar belakang masalah diatas maka dapat diidentifikasi masalahnya sebagai berikut:

1. Terdapat kesulitan dalam berkomunikasi terhadap orang-orang yang memiliki perbedaan bahasa.
2. Kurang efisien menerjemahkan bahasa menggunakan web translator terhadap para pengguna *messenger* yang ingin menerjemahkan bahasa.

3. Belum maksimalnya penggunaan *translator* untuk menerjemahkan bahasa pada aplikasi *messenger*.

C. Batasan Masalah

Berdasarkan latar belakang dan identifikasi masalah diatas, maka dilakukan pembatasan pada pembuatan aplikasi *messenger* hanya berbasis *text* dan bersifat *local area* dengan menggunakan bantuan bahasa pemrograman Java dan Google Translate API sebagai penerjemah bahasa.

D. Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana cara membuat Aplikasi Translator Messenger Berbasis Java dan Google Translate API?
2. Bagaimana unjuk kerja Aplikasi Translator Messenger Berbasis Java dan Google Translate API?
3. Bagaimana kelayakan penerapan Aplikasi Translator Messenger Berbasis Java dan Google Translate API berdasarkan indikator *correctness*, *reliability*, *functionality*, dan *usability*?

E. Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Mengetahui cara membuat Aplikasi Translator Messenger Berbasis Java dan Google Translate API.

2. Mengetahui unjuk kerja Aplikasi Translator Messenger Berbasis Java dan Google Translate API.
3. Mengetahui kelayakan penerapan Aplikasi Translator Messenger Berbasis Java dan Google Translate API berdasarkan indikator *correctness*, *reliability*, *functionality*, dan *usability*.

F. Manfaat Penelitian

1. Manfaat Praktis

- a. Bagi pengusaha, diharapkan dapat membantu mempermudah dalam pengorderan barang, penyampaian usul, berbagi pendapat, atau menanggapi client bisnis yang memiliki perbedaan bahasa.
- b. Bagi dunia pendidikan, dapat digunakan para pelajar untuk saling bertukar pikiran mengenai pelajaran tanpa terhalang oleh perbedaan bahasa dan tidak mempunyai memahami suatu bahasa.
- c. Bagi peneliti, dapat menjadi pembelajaran dalam membuat suatu software.

2. Manfaat Teoritis

- a. Diharapkan penelitian ini dapat menjadi referensi guna mengembangkan hasilnya.
- b. Diharapkan dapat membantu memunculkan ide aplikasi *messenger* yang lebih efektif, efisien dan sempurna.

G. Spesifikasi Produk

Produk Aplikasi Translator Messenger adalah aplikasi *messenger* yang dapat menerjemahkan bahasa sesuai dengan kemampuan bahasa yang dikuasai *user*. Penggunaan bahasa harus bersifat formal sesuai dengan tatanan bahasa yang baku. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan terintegrasi dengan Google Translate API sebagai penerjemah bahasa secara otomatis. Bahasa yang disediakan sesuai dengan pilihan bahasa yang terdapat pada Google Translate.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. *Messenger*

Messenger adalah suatu *feature*/program dalam Internet untuk berkomunikasi langsung sesama pemakai Internet yang sedang *online* (yang sedang sama-sama menggunakan Internet). Seseorang mengirim pesan dengan teks kepada orang lain yang sedang *online*, kemudian orang yang dituju membalas pesan dengan teks, demikian seterusnya.

Messenger memungkinkan komunikasi instan antara sejumlah pihak secara bersamaan, dengan mengirimkan informasi secara cepat dan efisien. Dalam kasus-kasus tertentu *messenger* melibatkan fitur tambahan, yang membuatnya bahkan lebih populer, yaitu untuk melihat pihak lain, misalnya dengan menggunakan *web-cam* untuk berbicara langsung secara gratis melalui internet.

a. Sejarah *Instant Messaging* (*Messenger*)

Perkembangan *Instant Messaging* diawali ketika mulai maraknya penggunaan teknologi secara *online* pada awal tahun 1990 ketika orang-orang mulai meluangkan banyak waktu untuk mengakses Internet. Para pengembang piranti lunak menciptakan sebuah *software chat room*, dimana suatu grup atau perorangan dapat melihat serta mengirimkan pesan kepada setiap orang yang ada pada “*room*” tersebut. Penyedia jasa *online* Quantum

Link menawarkan fitur untuk dapat saling berkirim pesan dengan sesama pengguna komputer yang sedang online. Penerapan *instant messaging* ini memiliki dasar yang tidak jauh berbeda, yaitu sebuah *chat room* yang ditujukan untuk dua orang.

Menurut Megdi Khosrow (2005) dalam bukunya yang berjudul *Encyclopedia of Information Science and Technology* menyebutkan, *instant messaging* ini mulai meledak di Internet pada November 1996, yaitu ketika Mirabilis memperkenalkan ICQ yang merupakan sebuah *instant messaging* yang dapat dipergunakan secara gratis bagi setiap orang. Pada tahun 1997, AOL menjadi pionir dalam komunitas *online*. Hal ini dikarenakan AOL memberikan kemampuan bagi pengguna untuk dapat melakukan komunikasi dalam waktu yang sama dalam sebuah *chat room* dan *instant messaging*. Model ICQ inilah yang kemudian menjadi dasar yang penting dalam penggunaan *instant messaging* yang beredar sekarang ini.

Dewasa ini, perkembangan *instant messaging* telah mengalami kemajuan yang amat pesat. Penggunaan *instant messaging* yang awalnya hanya dapat diakses oleh para penggunanya melalui komputer, sekarang ini telah dapat diakses melalui telepon genggam. Dengan kemampuan yang dimiliki telepon genggam untuk mengakses internet, maka para pengguna *instant messaging* dapat mengakses dunia maya kapan saja dan dimana saja.

b. Penggunaan *Instant Messaging (Messenger)*

Penggunaan *Instant Messaging* ini dapat dilakukan melalui langkah-langkah berikut ini:

1. Jika komputer belum memiliki piranti lunak *instant messaging*, maka pengguna harus mengunduh piranti lunak *instant messaging* dulu.
2. Setelah piranti lunak berhasil diunduh, selanjutnya dilakukan penginstallan.
3. Setelah piranti lunak *instant messaging* ada di komputer, bukanlah piranti tersebut.
4. Setelah terhubung dengan server, pengguna dapat mengetik nama dan password. Jika ini pertama kalinya terhubung, pengguna harus melakukan pendaftaran terlebih dahulu untuk memperoleh *account* yang dapat digunakan sebagai langkah awal penggunaan fasilitas *instant messaging*. Ketika *server* melakukan verifikasi terhadap nama dan password, maka pengguna sudah tercatat sebagai member/ pengguna fasilitas *instant messaging*.
5. Setelah itu, komputer secara otomatis akan mengirimkan data kepada *server* berisi informasi koneksi yang ada pada komputer yang digunakan (berupa *IP address* dan nomor *port*).
6. Lalu *server* akan membuat file secara sementara yang akan mengkoneksikan informasi dan *list* kontak. Lalu akan dilakukan penyingkronisasian dengan para pengguna yang ada dalam kontak *list* dan akan menandai para pengguna yang sedang *online*.
7. Kemudian pengguna dapat mengklik nama pengguna lain yang sedang *online* hingga sebuah halaman akan terbuka sebagai tempat untuk menuliskan pesan. Pada tahap ini percakapan dimulai.

8. Ketika percakapan telah selesai, pengguna dapat menutup halaman percakapan . Pada tahap ini pengguna dapat mengakhiri koneksi serta keluar dari aplikasi *instant messaging*. Pada saat yang bersamaan, *account* akan mengirimkan pesan pada *server* untuk mengakhiri sesi penggunaan *instant messaging*. *Server* akan mengirimkan pesan *logged off* kepada setiap pengguna lain.

c. Struktur *Instant Messaging (Messenger)*

Menurut Naomi Baron (2010), secara struktural, komunikasi melalui komputer dapat didefinisikan dalam dua parameter dasar yaitu, sinkron dan asinkron.

1) Sinkron

Pada komunikasi sinkron, transmisi dasarnya seketika, dan mediator diasumsikan hadir secara fisik untuk membaca dan menanggapi pesan. Pada tipe asinkron, setiap proses pengiriman dan penerimaan pesan akan dilakukan sinkronisasi. Contoh proses komunikasi dengan sinkron yaitu, *chat* pada *instant messaging*.

2) Asinkron

Pada komunikasi asinkron, setiap proses yang dilakukan tidak perlu adanya sinkronisasi. Contoh proses komunikasi asinkron yaitu, email, blog. Namun dalam prakteknya, pengguna komunikasi yang terjadi sering melintasi jalur kategori. Sebagai contoh, email kontemporer (teknologi asynchronous) sering mengirimkan pesan dalam waktu singkat, dan banyak pengguna membalas segera. Sebaliknya, dalam kasus *instant messaging (IM)*, meskipun

teknologi ini dirancang untuk bekerja serentak, sering digunakan dengan tipe asinkron.

d. Arsitektur

Kebanyakan aplikasi messenger berjalan berdasarkan arsitektur *client-server*. Fungsi *client-server* adalah untuk mengatur jalur komunikasi dari *messenger*. Pihak yang meminta layanan disebut *client* dan yang memberi layanan disebut *server*. Pengguna menjalankan aplikasi messenger sebagai *client* sedangkan fungsi *server* menunggu permintaan *client* untuk melakukan suatu fungsi misalnya, bertukar pesan. Pesan yang dipertukarkan sesama *client* berupa teks.

Ada beberapa pula, aplikasi *messenger* yang menggunakan arsitektur *peer-to-peer*. Cara kerja dari arsitektur ini, *client* terlebih dahulu menghubungi *server* untuk mengetahui lokasi pihak *client* yang akan dihubungi. Setelah mengetahui lokasinya, *client* akan menghubungi secara langsung pihak *client* yang akan dituju.

2. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal.

Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda

Kelebihan-kelebihan java menurut Wahana (2010) antara lain:

a. *Multiplatform*

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/sistem operasi computer. Pemrogram cukup menulis sebuah program Java dan dikompilasi sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas *operating system* Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris.

b. OOP (Object Oriented Programming - Pemrogram Berorientasi Objek)

OOP artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu

bahasa pemrograman termudah, bahkan untuk fungsi-fungsi yang *advance* seperti komunikasi antara komputer sekalipun.

c. *Library* Lengkap

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Pengumpulan *spam* otomatis

Java memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung

Java banyak digunakan sekarang ini karena java merupakan bahasa pemrograman yang baru. Sehingga lebih *uptodate* dibandingkan dengan bahasa pemrograman yang lain.

3. Google Translate API

Google Translate adalah salah satu translate bahasa *online* paling terkenal dan paling banyak digunakan orang di seluruh dunia saat ini, banyak manfaat dari jasa Google Translate untuk menterjemahkan bahasa ke bahasa lain dengan mudah tanpa harus membuka kamus atau lainnya.

Tabel 1. Tabel Perbandingan Translator

Service	URL	Underlying MT	Language Pairs
Babelfish	http://babelfish.yahoo.com	Systran	38
Freetranslation	www.freetranslation.com	SDL	19
Google Translate	http://translate.google.com	Google	54
Online-tranlator	www.online-tranlator.com	PROMT	24
Reverso	www.reverso.net	Reverso	19
Worldlingo	http://www2.worldlingo.com	Worldlingo	225

Google Translate merupakan layanan aplikasi yang disediakan oleh Google.Inc , yang berfungsi untuk membantu menterjemahkan suatu teks atau halaman web dari suatu bahasa ke bahasa lain. Menurut Wikipedians (2001), Google melakukan penerjemahkan dengan pendekatan yang disebut penerjemahan berdasar statistik. Penerjemahan demikian merupakan hasil penelitian Franz-Josef Och yang telah memenangkan kontes DARPA untuk kecepatan mesin terjemahan pada tahun 2003.

Menurut Juju Dominikus(2009), kelebihan Google Translate adalah (1)Google translate memiliki lisensi. (2)Google translate juga support banyak bahasa. (3)Update database google translate dilakukan secara otomatis.

Kelebihan lain menurut Michael Miller (2009), Google translate adalah lebih akurat dibanding dengan yang lain. Sedangkan kekurangannya adalah Google Translate menggunakan translate berbasic *word by word translation* yang tidak selalu memperhatikan stuktur bahasa. Sehingga membuat kalimat yang diterjemahkan terkadang menjadi sedikit kacau.

4. UML

UML (Unified Modeling Language) adalah metode pemodelan visual untuk merancang sebuah software berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma object oriented. UML adalah sebuah bahasa standar untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat sehingga merupakan salah satu proses implementasi pengembangan software.

UML terbagi menjadi beberapa jenis diagram yang masing-masing memiliki aturan-aturan tertentu dalam penyusunannya. Diagram-diagram tersebut tersusun atas sejumlah elemen grafis yang saling membentuk satu kesatuan dalam pemodelan software. Masing-masing diagram UML mempersatukan berbagai sudut pandang terhadap sistem dan mendefinisikan apa yang dikerjakan oleh system. Diagram visual dalam UML:

- a. Diagram use case (use case diagram).
- b. Diagram aktivitas (activity diagram).
- c. Diagram sekuensial (sequence diagram).
- d. Diagram kolaborasi (collaboration diagram).
- e. Diagram kelas (class diagram).
- f. Diagram statechart (statechart diagram).
- g. Diagram komponen (component diagram).
- h. Diagram deployment (deployment diagram).

5. *Use case*

Use case adalah abstraksi dari interaksi antara sistem dan actor, gambar dari beberapa atau seluruh aktor dan use case dengan tujuan mengenali interaksi mereka dalam suatu sistem. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Komponen-komponen use case:

a. Actor

Actor hanya berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case.

b. Use Case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

c. Connector

Merupakan penghubung antara actor dengan use case. Ada beberapa jenis connector, yaitu :

1. Asosiasi

Asosiasi mengidentifikasi interaksi antara setiap actor tertentu dengan setiap use case tertentu. Tiap asosiasi menjadi dialog yang harus dijelaskan dalam use case naratif. Asosiasi bisa berarah jika komunikasi satu arah dan dapat juga tidak berarah jika komunikasi terjadi ke dua arah.

2. Generalisasi

Generalisasi untuk mendefinisikan relasi antara dua actor atau menambahkan sifat dari yang lainnya. Digambarkan menggunakan garis bermata panah kosong dari yang menurunkan mengarah ke yang diturunkan.

3. Dependensi

Dependensi terdiri dari dua jenis yaitu :

- a. Include, mengidentifikasi hubungan antar dua Use Case di mana yang satu memanggil yang lain. Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis. Arah mata panah sesuai dengan arah pemanggilan (umumnya mengarah ke kanan).
- b. Extend, terjadi jika pemanggilan memerlukan adanya kondisi tertentu. Digambarkan serupa dengan dependensi <<include>>, kecuali arah panah berlawanan menjadi <<extend>>

6. Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

7. Sequence diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.

8. Activity diagram

John Satzinger (2010) dalam bukunya *System Analysis and Design in a Changing World* menyatakan bahwa activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna atau system, orang yang melakukan masing-masing aktivitas dan aliran sekuensial dari aktivitas-aktivitas tersebut. Komponen activity diagram:

a. Activity

Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.

b. Transition

Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari activity ke activity.

c. Decision

Notasi yang menandakan kontrol cabang aliran berdasarkan decision point.

d. Synchronization Bars

Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan (pararel).

B. Unjuk Kerja

Unjuk kerja dilakukan untuk mengetahui *performance* fungsionalitas dari suatu produk perangkat lunak. Beberapa unjuk kerja menurut Bach (2005), Microsoft Window Logo sebagai berikut:

1. Fungsi primer/utama

Fungsi primer/utama adalah apabila produk dapat diasosiasikan sesuai dengan tujuan produk yang sebenarnya. Unjuk kerja dalam kategori fungsi primer yaitu menentukan kinerja fungsionalitas suatu produk yang dihasilkan. Misalnya, produk Microsoft Word harus mempunyai fungsi menambahkan teks ke dokumen. Hal ini sangat penting karena Microsoft Word merupakan alat pengolah data dalam bentuk teks. Contoh lain adalah pengecekan fungsionalitas *toolbar* yang ada pada Microsoft Word.

2. Fungsi kontribusi

Setiap fungsi yang memberikan kontribusi untuk *utilitas* produk, tetapi tidak termasuk dalam fungsi primer. Meskipun fungsi berkontribusi tidak utama, *inoperability* mereka mungkin menjadi dasar untuk menilai unjuk kerja dari suatu produk. Misalnya, pengguna secara teknis mampu melakukan fungsionalitas "*Undo*" sebagai fungsi tambahan untuk mengembalikan keadaan sebelumnya.

Kesimpulannya fungsi primer untuk mengetahui kesesuaian fungsionalitas tujuan dari produk yang dihasilkan. fungsi kontribusi adalah mengetahui kesesuaian fungsionalitas fungsi tambahan untuk mendukung kinerja dari produk yang dihasilkan.

Untuk mengkategorikan fungsi primer dan fungsi kontribusi suatu produk, perlu dilakukan beberapa langkah sebagai berikut:

a. Identifikasi tujuan produk

Identifikasi tujuan produk meliputi:

- 1) Meninjau produk dan menentukan layanan mendasar dari produk perangkat lunak yang dihasilkan. Memuat fungsi utama dari produk.
- 2) Menulis secara singkat hasil dari tinjauan produk untuk menjelaskan tujuan produk.

b. Identifikasi fungsi.

Identifikasi fungsi meliputi:

- 1) Mengamat produk dan menemukan cara kerja produk.
- 2) Buatlah garis besar dari semua fungsi utama.
- 3) Buatlah daftar fungsi kontribusi.
- 4) Deskripsikan setiap fungsi produk.

C. Kelayakan Aplikasi

Dalam setiap pengembangan perangkat lunak harus diperhatikan beberapa kualitas *software* agar aplikasi yang dibuat dapat dikatakan layak

dan tepat guna. Teori-teori kualitas software menurut McCall dan ISO dijelaskan sebagai berikut:

1. McCall

Dalam buku Pressman (Pressman, 2010), Mc. Call menyebutkan beberapa indikator-indikator kelayakan software yaitu sebagai berikut:



Gambar 1. Indikator Mc Call menurut Pressman (2010)

- a. *Correctness*
- b. *Reliability*
- c. *Efficiency*
- d. *Integrity*
- e. *Usability*
- f. *Maintainability*
- g. *Flexibility*
- h. *Testability*
- i. *Portability*
- j. *Reusability*
- k. *Interoperability*

2. ISO 9126 Model

Menurut Pressman (2010), ISO 9126, terdapat enam indikator untuk menentukan kualitas suatu software.

- a. *Functionality*
- b. *Reliability*
- c. *Usability*
- d. *Efficiency*
- e. *Maintainability*
- f. *Portability*

3. *Correctness*

Correctness adalah tingkat kebutuhan program terhadap kebutuhan yang dispesifikasikan dan memenuhi tujuan consumer. Pada indikator *correctness*, terdapat tiga sub-faktor kualitas yaitu:

- 1) *Completeness*: Kelengkapan fitur yang sesuai dengan konteks perangkat lunak yang dibangun.
- 2) *Consistency*: Penggunaan desain seragam dan teknik dokumentasi seluruh protocol pengembangan perangkat lunak.
- 3) *Traceability*: Kemampuan untuk melacak representasi desain atau komponen program kembali.

Menurut Pressman (2010), pengukuran indikator *correctness* dapat diukur menggunakan cacat per KLOC (*defects per KLOC*). KLOC merupakan jumlah baris kode yang dibuat sedangkan cacat adalah bug yang ada pada

perangkat lunak. Hasil dari pengukuran tersebut berupa jumlah error yang terjadi pada perangkat lunak yang dikembangkan. Pada buku Pressman (2010) juga disebutkan perkiraan jumlah maksimal error dari suatu perangkat lunak yaitu sebagai berikut:

Tabel 2. Rumus jumlah error perangkat lunak

No	Metode	Rumus
1	Watson – Felix Model	$E = 5.2 \times (\text{KLOC})^{0.91}$
2	Bailey-Basili Model	$E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$
3	Boehm Simple Model	$E = 3.2 \times (\text{KLOC})^{1.05}$
4	Doty Model(untuk KLOC>9)	$E = 5.28 \times (\text{KLOC})^{1.047}$

4. *Reliability*

Reliability adalah tingkat kemampuan program yang diharapkan dapat menampilkan fungsi yang dimaksud dengan presisi yang ditetapkan. Pada indikator ini, terdapat lima sub-faktor kualitas yaitu:

- 1) *Accuracy*: Ketepatan perhitungan dan ketepatan kontrol
- 2) *Consistency*: Penggunaan desain seragam dan teknik dokumentasi seluruh protocol pengembangan perangkat lunak.
- 3) *Error tolerance*: Toleransi kerusakan yang terjadi ketika sebuah program menemukan kesalahan.
- 4) *Modularity*: Independensi fungsional komponen program.
- 5) *Simplicity*: Sejauh mana program dapat dipahami tanpa kesulitan.

Menurut Wohlin (2003), menghitung jumlah kesalahan dapat digunakan untuk memprediksi *reliability* suatu perangkat lunak. Hal ini dikarenakan adanya

kemungkinan-kemungkinan eksternal yang terjadi ketika perangkat lunak tersebut digunakan dalam *real-condition*. Menurut Scoot Speaks (2004), penghitungan reliability dapat menggunakan rumus sebagai berikut:

$$MTBF = T/R$$

MTBF = Mean time between failure

T = lamanya waktu proses pengujian perangkat lunak

R = jumlah kesalahan yang terjadi selama proses

Menurut QuickSilver Controls (2005), pada Microsoft terdapat perhitungan MTBF yang dilakukan pada salah satu aplikasi Microsoft yaitu Microsoft Word sebagai berikut:

Tabel 3. Nilai MTBF Microsoft Word

No	Hour	Failure
1	17H	1
2	23H	8
3	34N	17

5. Usability

Usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan masukan dan mengartikan keluaran oleh program. Pada indikator ini, terdapat dua sub-faktor kualitas yaitu:

- 1) *Operability*: Kemudahan pengoperasian program.
- 2) *Traceability*: Kemampuan untuk melacak representasi desain atau komponen program kembali.

Pengukuran indikator *usability* dapat diuji menggunakan kuesioner. Kuesioner yang digunakan merupakan adaptasi dari kuesioner buatan IBM yang sudah banyak digunakan sebagai instrument untuk pengujian perangkat lunak.

6. *Functionality*

Kemampuan menutupi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan *user*. Pada indikator ini, terdapat empat sub-faktor kualitas yaitu:

- 1) *Suitability*: kemampuan perangkat lunak untuk menyediakan fungsi yang memiliki tugas-tugas tertentu.
- 2) *Accuracy*: kemampuan software untuk menyediakan hasil yang sesuai.
- 3) *Interoperability*: kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
- 4) *Security*: kemampuan perangkat lunak untuk melindungi informasi dan data dari akses yang tidak sah.

Padayachee, Kotze, Merwe (2010) menyebutkan sub karakteristik dari karakteristik kualitas ISO 9126 dapat digunakan untuk menguji kualitas suatu sistem. Pengujiannya menggunakan instrumen yang sesuai dengan sub karakteristik dari kriteria indikator *functionality*. Dibawah ini adalah penjelasan sub karakteristik dari *functionality*:

Tabel 4. Sub Karakteristik *Functionality*

Karakteristik	Sub Karakteristik	Keterangan
Functionality	Suitability	'Can software perform the tasks required?'

	Accurateness	'Is the result as expected?'
	Interoperability	'Can the system interact with another system?'
	Compliance	'Is the system compliant with standards?'
	Security	'Does the system prevent unauthorized access?'

D. Penelitian yang Relevan

Penelitian sebelumnya tentang aplikasi messenger atau chatting oleh para peneliti dapat dijadikan rujukan dalam membangun Aplikasi Translator Messenger. dari pengkajian penelitian sebelumnya, diharapkan dapat memberi gambaran yang harus dilakukan oleh peneliti. Beberapa penelitian yang dijadikan rujukan oleh peneliti sebagai berikut:

1. Merancang dan Membangun Aplikasi Chat Messenger untuk Android oleh Bakhtiar Wijayanto (2012). Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan PHP (*HyperText Preprocessor*). Aplikasi ini diimplementasikan berbasis android. Dari penelitian, dihasilkan bahwa aplikasi ini dapat dijalankan pada telepon genggam berbasis android dengan spesifikasi minimal froyo.
2. Aplikasi Chatting dengan Fasilitas FTP dan IP Address Blocking untuk Client-Server Multiple Connections oleh Tommy Budianto (2010). Aplikasi ini dibangun menggunakan Visual Basic 6.0 dan Winsock API. Visual basic 6.0 digunakan untuk membangun interface aplikasi. Sedangkan Winsock API digunakan sebagai pengganti client-server untuk multiple connections. Dari hasil penelitian, dapat disimpulkan bahwa aplikasi ini telah berhasil membuat aplikasi chatting model client-server

dengan system managemen terpusat. Jadi setiap client yang ingin bergabung harus mendapatkan persetujuan terlebih dahulu dari server.

3. Implementasi Aplikasi Chatting Menggunakan Server Jabber oleh Maestro Pakpahan, Adian Fatchur Rochim, dan Rizal Isnanto (2011). Pada penelitian in, peneliti hanya menguji implementasi chatting dengan Jabber sebagai pengganti client-server pada aplikasi chatting. Jabber adalah sebuah protocol yang terbuka untuk pertukaran pesan dan *presence* yang *real-time* antar dua pengguna di dalam jaringan Jabber. Dari hasil penelitian, peneliti berhasil melakukan pengujian penggunaan Jabber pada apliakasi chatting.

E. Kerangka Berfikir

Aplikasi messenger sebagai salah satu *tool* komunikasi yang sangat digemari masyarakat. Komunikasi dapat dikatakan berjalan dengan baik jika pesan dan maksud tujuan seseorang dapat tersampaikan kepada orang lain. Terkadang, komunikasi menjadi tidak berjalan dengan baik jika terdapat perbedaan bahasa antara orang yang saling berkomunikasi. Sebagai salah satu solusi, peneliti akan mengembangkan Aplikasi Translator Messenger untuk memudahkan pengguna atau *user* berkomunikasi berbasis teks dalam bahasa yang berbeda.

Pada pengembangan aplikasi tersebut, peneliti menggunakan bahasa pemrograman Java dan Google Translate API sebagai penerjemah. Sebagai sebuah produk software, diharapkan aplikasi tersebut memenuhi standar

kelayakan perangkat lunak. Kelayakan perangkat lunak dapat dianalisis menggunakan standar kualitas *software* (*quality software*).

Pada penelitian ini, peneliti menggunakan empat faktor kualitas *software* untuk menganalisis kualitas aplikasi yang dikembangkan. Standar kualitas software yang digunakan adalah standar kualitas software menurut McCall dan ISO 9126 model. Keempat faktor tersebut adalah *correctness*, *reliability*, *usability* (McCall) dan *functionality* (ISO 9126 model). Pemilihan keempat faktor indikator tersebut berdasarkan dari tujuan dikembangkannya Aplikasi Translator Messenger.

BAB III

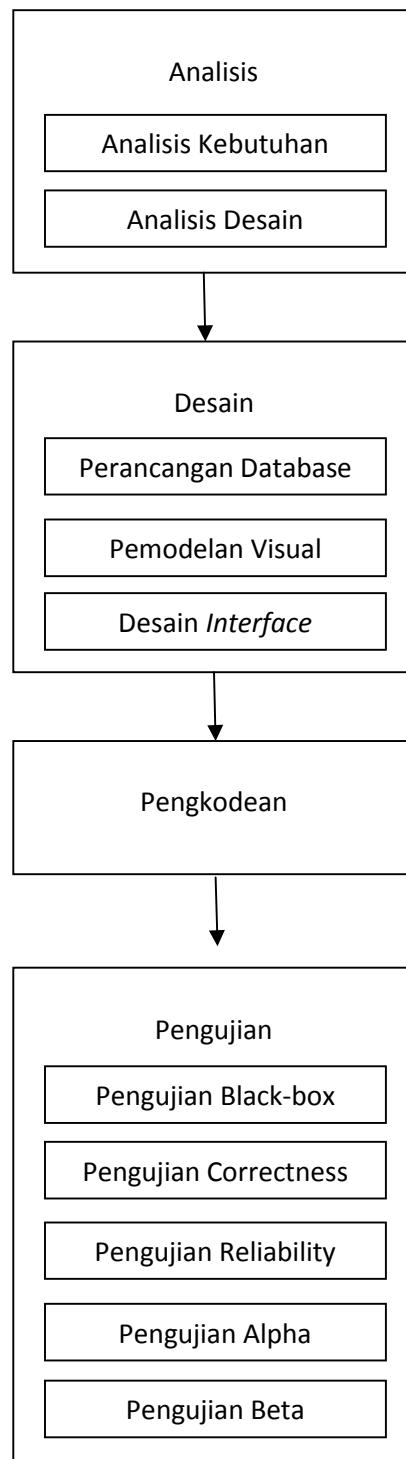
METODE PENELITIAN

A. Desain Penelitian

Metode yang digunakan dalam penelitian ini menggunakan metode Penelitian dan Pengembangan (*Research and Development*). Metode Penelitian dan Pengembangan (*Research and Development*) adalah metode yang digunakan untuk menghasilkan suatu produk perangkat lunak dan menguji kelayakan perangkat lunak yang dibangun. Menurut Sujadi (2003), Penelitian dan Pengembangan (*Research and Development*) adalah suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru, atau menyempurnakan produk yang telah ada, yang dapat dipertanggungjawabkan.

Pada penelitian ini, peneliti membangun perangkat lunak dengan menggunakan metode adaptasi ***Waterfall Model*** menurut Pressman (2001). Model fase *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun suatu software.

Selanjutnya peneliti membuat skema metode yang akan digunakan sesuai dengan adaptasi ***Waterfall Model***. Skema yang dibuat merupakan gambaran sederhana dari langkah-langkah yang akan dilakukan oleh peneliti. Berikut ini adalah gambar metode penelitian yang digunakan untuk mempermudah dalam pengembangan Aplikasi Translator Messenger:



Gambar 2. Langkah-langkah metode penelitian

Langkah-langkah metode penelitian tersebut dapat dijelaskan sebagai berikut:

1. Analisis (*Analysis*)

Analisis merupakan tahapan pertama dalam pengembangan suatu perangkat lunak. Pada tahap ini, analisis yang dilakukan oleh peneliti adalah analisis kebutuhan dan analisis desain perangkat lunak.

a. Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan melakukan observasi terhadap aplikasi-aplikasi *messenger* dan *tool translator* yang sudah ada. Peneliti melakukan observasi untuk mengetahui cara kerja *messenger* dan membandingkan *translator* yang banyak diminati sebagian besar *user*.

Dalam penelitian ini, peneliti mengobservasi Yahoo Messenger sebagai acuan *messenger* untuk membangun Aplikasi Translator Messenger. Sedangkan *translator* yang diobservasi oleh peneliti adalah Google Translate. Peneliti sengaja memilih Google Translate karena *translator* tersebut sering digunakan oleh sebagian besar *user*.

b. Analisis Desain

Analisis desain dilakukan untuk menentukan desain yang dibutuhkan untuk mengembangkan Aplikasi Translator Messenger. Pada tahap analisis desain, peneliti akan analisis desain kebutuhan user dan analisis batasan desain.

2. Desain (*design*)

Tahap desain dimulai setelah tahap analisis selesai dilakukan. Pada tahap ini, peneliti mulai merancang desain *messenger* yang sesuai dengan analisis desain kebutuhan user dan batasan desain. Desain yang dilakukan peneliti meliputi tiga tahapan yaitu:

a. Perancangan *database*

Database digunakan untuk menyimpan data-data user. Perancangan *database* dilakukan dengan membuat *database* dan tabel yang akan digunakan pada Aplikasi Translator Messenger.

b. Pemodelan visual

Pemodelan visual pada penelitian ini menggunakan UML sebagai notasi pemodelan. Pemodelan yang dirancang adalah use case diagram, activity diagram dan sequence diagram.

c. Desain *interface*(antarmuka)

Desain *interface* dilakukan untuk merancang antarmuka aplikasi tahap awal. Desain ini yang nantinya dijadikan acuan untuk membuat *interface* sebenarnya pada aplikasi.

3. Pengkodean (*Code*)

Pada tahap pengkodean, peneliti mulai membangun aplikasi sesuai dengan acuan pada desain yang sudah dibuat. Aplikasi Translator Messenger dibangun menggunakan bahasa pemrograman Java.

4. Pengujian (*Testing*)

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean menurut Pressman (2002). Pada pengujian, peneliti menggunakan beberapa pengujian, yaitu:

a. Pengujian *black-box*

Tahap kedua adalah pengujian *black-box*. *Black-box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji. Pada pengujian ini, peneliti hanya menguji kesesuaian output terhadap input yang diberikan dan fungsionalitas Aplikasi Translator Messenger. Pengujian *black-box* dilakukan pada tahap pengodean karena pengujian ini dilakukan langsung oleh peneliti.

b. Pengujian *Correctness*

Pengujian *correctness* dilakukan menggunakan cacat per KLOC menurut Pressman (2010). Pengujiannya dengan cara menghitung jumlah cscst yang selanjutnya dibagi dengan jumlah baris kode yang dibuat.

c. Pengujian *Reliability*

Pengujian *reliability* menggunakan MTBF (*Mean time between failure*). Pengujian ini dilakukan untuk memprediksi tingkat kehandalan perangkat lunak yang telah dibuat.

d. Pengujian *Alpha*

Pengujian *alpha* adalah pengujian yang dilakukan sebagai uji produk untuk mengetahui kelemahan-kelemahan produk perangkat lunak yang

dibangun. Pengujian ini meliputi indikator *functionality* dan *usability*. Pengujian ini dilakukan oleh ahli(*expert*) perancang perangkat lunak. Dari hasil pengujian, peneliti akan mendapatkan saran-saran mengenai kelemahan-kelemahan perangkat lunak Aplikasi Translator Messenger. Tindak lanjut dari saran-saran tersebut sebagai revisi awal untuk penyempurnaan perangkat lunak yang dibangun.

e. Pengujian *Beta*

Pengujian *beta* dilakukan sebagai akhir pengujian produk perangkat lunak. Pengujian ini juga meliputi indikator *functionality* dan *usability*. Pengujian ini dilakukan oleh beberapa responden yang dipilih sebagai *user* untuk mencoba fungsionalitas perangkat lunak Aplikasi Translator Messenger. Responden akan menilai produk dari segi *correctness*, *reliability*, *functionality* dan *usability*. Dari hasil penilaian akan diketahui kelayakan dari perangkat lunak Aplikasi Translator Messenger.

B. Subjek dan Objek Penelitian

a. Subjek Penelitian

Subjek penelitian ini adalah mahasiswa Pendidikan Teknik Informatika Universitas Negeri Yogyakarta.

b. Objek Penelitian

Objek penelitian adalah Aplikasi Translator Messenger Berbasis Java dan Google Translate API sebagai perangkat lunak yang digunakan untuk memudahkan komunikasi banyak bahasa berbasis text.

C. Tempat dan Waktu Penelitian

a. Tempat Penelitian

Penelitian dilakukan di area fakultas teknik Universitas Negeri Yogyakarta.

b. Waktu Penelitian

Pelaksanaan penelitian dimulai Februari 2012 sampai dengan selesai.

D. Responden

Pada penelitian ini, peneliti mengambil validasi ahli (*expert judgement*) sebanyak 2 orang dosen Pendidikan Teknik Informatika Universitas Negeri Yogyakarta. Sedangkan responden yang diambil sebanyak 16 orang mahasiswa Pendidikan Teknik Informatika Universitas Negeri Yogyakarta sebagai *user*.

E. Teknik Pengumpulan Data

Pada penelitian ini, teknik pengumpulan data yang dipilih peneliti adalah kuesioner. Menurut Sutopo (2006), kuesioner (angket) merupakan teknik pengumpulan data yang dilakukan dengan cara memberikan seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya, dimana peneliti tidak langsung bertanya jawab dengan responden. Sedangkan menurut Hamid Darmadi (2011), kuesioner adalah suatu kumpulan pertanyaan dan pernyataan yang telah disusun sedemikian rupa untuk dijawab oleh responden dalam rangka mengumpulkan data sesuai

dengan tujuan penelitian tertentu. Dalam penelitian kuantitatif penggunaan kuesioner paling sering ditemui karena jika dibuat secara intensif dan teliti kuesioner mempunyai keunggulan jika dibandingkan dengan alat pengumpul data.

Jenis kuesioner dapat dibeda-bedakan dari berbagai sudut pandang:

Dipandang dari cara menjawabnya, maka ada:

- a. Kuesioner terbuka, yang memberi kesempatan kepada responden untuk menjawab dengan kalimat sendiri.
- b. Kuesioner tertutup, yang sudah disediakan jawabannya sehingga responden tinggal memilih.

Dipandang dari jawaban yang diberikan ada:

- a. Kuesioner langsung, yaitu responden menjawab tentang dirinya
- b. Kuesioner tidak langsung, yaitu jika responden menjawab tentang orang lain

Dipandang dari bentuknya maka ada:

- a. Kuesioner pilihan ganda, yang dimaksud adalah sama dengan kuesioner tertutup.
- b. Kuesioner isian yang dimaksud adalah kuesioner terbuka.
- c. Check list sebuah daftar dimana responden tinggal membubuhkan tanda check pada kolom yang sesuai.
- d. Rating scale (skala bertingkat) yaitu sebuah pertanyaan diikuti oleh kolom-kolom yang menunjukkan tingkatan-tingkatan.

Ditinjau dari kuesioner yang dibuat, peneliti menggunakan kuesioner check list yang bersifat tertutup dan langsung.

F. Instrumen Penelitian

Instrumen penelitian dibuat berdasarkan indikator menurut Mc Call dan ISO 9126. Beberapa indikator menurut Pressman (2010) yang dipakai adalah indikator *correctness*, *reliability*, *usability* (Mc Call) dan *functionality* (ISO 9126). Namun untuk indikator *correctness* sebelum dilakukan pengujian alpha, terlebih dahulu diuji melalui analisis *correctness* menurut Pressman (2010). Analisis *correctness* dihitung berdasarkan error/KLOC dari perangkat lunak yang dibuat. Pada penghitungan LOC kode program, peneliti menggunakan LOC counter sedangkan FindBugs digunakan untuk mendeteksi error/bug dari program.

1. Black-box Testing

Pengujian *black-box* dilakukan oleh peneliti sendiri untuk menguji fungsionalitas perangkat lunak sebelum dilakukan pengujian *alpha*.

Tabel 1. List Pengujian *Black-box*

No	Use Case	Hasil Pengujian
Login		
1	Menampilkan pesan error	
Sign Up		
2	Menampilkan pesan error	
3	Menampilkan konfirmasi penyimpanan data sukses	
Chatting		
4	Mengirim pesan sesuai dengan tujuan user	
5	Menampilkan output pesan	
6	Menutup tab chat	
Sign Out		

7	MenampilkAn pesan sign out	
8	Menutup aplikasi secara sempurna	
OS		
9	Pengujian pada OS Windows	
10	Pengujian pada OS Mac	

2. *Alpha Testing*

Pengujian *alpha* dilakukan oleh *expert judgement*(ahli) yang sudah dipilih oleh peneliti. Pada tahap ini, produk diuji untuk mengetahui kelemahan-kelemahan produk yang dikembangkan. Pengujian ini dilakukan oleh ahli aplikasi yaitu beberapa dosen yang berkompeten dalam aplikasi yang dikembangkan.

Penilaian yang diperoleh akan menjadi unjuk kerja aplikasi yang sudah dibangun. Dari hasil validasi akan didapat saran-saran dari validator sebagai bahan penyempurnaan kelemahan-kelemahan aplikasi. Pada penelitian ini, peneliti akan menggunakan skala *Linkert*. Skala *Linkert* mempunyai gradasi nilai dari sangat positif sampai sangat negatif. Skala yang digunakan yaitu sangat setuju diberi skor 5, setuju diberi skor 4, ragu-ragu diberi skor 3, tidak setuju diberi skor 2 dan sangat tidak setuju diberi skor 1. Instrument ini digunakan pada tahap pengujian alpha dan beta. Pengujian perangkat lunak yang dilakukan adalah sebagai berikut:

Tabel 5. Pengujian Berdasarkan *Functionality*

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Tab chat memuat nama sesuai dengan nama user yang dipilih pada list user					
2.	Terdapat pesan user lawan yang telah sign out jika tab chat masih dalam kondisi aktif					
3.	Translator Messenger dapat dihubungkan dengan Google Translate API sebagai penerjemah bahasa					

Tabel 6. Pengujian Berdasarkan *Usability*

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Cara penggunaan sistem ini sangat simpel					
2.	Sistem ini sangat mudah dipelajari					
3.	Translator Messenger memudahkan komunikasi bahasa yang berbeda					
4.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini					

3. *Beta Testing*

Pada pengujian *beta*, peneliti mengambil responden mahasiswa Pendidikan Teknik Informatika Universitas Negeri Yogyakarta. Hasil dari pengujian ini sebagai hasil akhir untuk menilai kelayakan perangkat lunak yang dibuat.

Tabel 6. Kuesioner Pengujian *Beta*

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>					
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai <i>user</i>					
3.	Terdapat tempat input pada tab login untuk memasukkan <i>username</i> dan <i>password</i>					
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih					
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan					
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda					
7.	Tampilan sistem ini sangat memudahkan					
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>					

9.	Cara penggunaan sistem ini sangat <i>simpel</i>					
10.	Sistem ini sangat mudah dipelajari					
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda					
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini					

G. Teknik Analisis Data

Teknik analisis data yang digunakan dalam penelitian ini adalah deskriptif kuantitatif. Deskriptif kuantitatif adalah teknik analisis data yang dinyatakan dengan angka – angka atau persentase.

Data kuantitatif yang berwujud akan dihitung melalui porses perhitungan. Proses perhitungan tersebut akan dijelaskan sebagai berikut:

1. Dijumlahkan, dibandingkan dengan jumlah yang diharapkan dan diperoleh persentase;
2. Dijumlahkan, diklasifikasikan sehingga merupakan susunan urutan data (array) untuk selanjutnya dibuat tabel, maupun diproses lebih lanjut menjadi perhitungan pengambilan kesimpulan ataupun untuk kepentingan visualisasi datanya

Data kuesioner yang telah terkumpul akan dianalisis berdasarkan deskriptif kuantitatif. Data kuesioner akan dinilai melalui perhitungan dengan rumus sebagai berikut:

$$DP = \frac{n}{N} \times 100 \%$$

Keterangan :

DP = Deskriptif Persentase (%)

n = Skor empirik (Skor yang diperoleh)

N = Skor ideal (Ditentukan berdasarkan spesifikasi sistem)

Dari hasil perhitungan akan diperoleh persentase yang dapat mengukur tingkat kelayakan Aplikasi Translator Messenger dari segi *Functionality*, dan *Usability*.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

1. Analisis

Analisis merupakan tahap awal dari sebuah pengembangan perangkat lunak. Pada tahap ini, peneliti menganalisis hal-hal yang diperlukan untuk membangun perangkat lunak Aplikasi Tranlator Messenger. Analisis yang dilakukan peneliti meliputi analisis kebutuhan dan analisis desain.

a. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan-kebutuhan yang harus ada pada Aplikasi translator Messenger. kebutuhan-kebutuhan tersebut untuk memenuhi syarat dari tujuan dibangunnya aplikasi tersebut. Pada tahap ini, peneliti menganalisis kebutuhan *messenger* yang dapat menerjemahkan bahasa secara otomatis. Kebutuhan tersebut dapat ditanggulangi dengan pengintegrasian menggunakan *translator*.

Analisis kebutuhan yang dihasilkan adalah perlu adanya sistem pertukaran pesan yang merupakan fungsional dasar dari *messenger*. Pertukaran pesan dapat dilakukan dengan membuat arsitektur *client server* pada aplikasi yang dikembangkan. Penggunaan *client server* bertujuan sebagai pengatur komunikasi dalam hal ini pertukaran pesan antar client. Sedangkan untuk menerjemahkan diperlukan *translator* berupa Google Translate API yang akan diintegrasikan pada *messenger* yang dikembangkan.

b. Analisis Desain

Analisis desain dilakukan untuk menyesuaikan kebutuhan agar perangkat lunak Aplikasi Translator Messenger yang dibangun sesuai dengan harapan. Adapun beberapa analisis desain yang dibutuhkan sebagai berikut:

1. Analisis desain kebutuhan user

Berdasarkan kebutuhan *user*, Aplikasi Translator Messenger mampu menerjemahkan berbagai bahasa untuk memudahkan komunikasi dua arah berbasis text. Sehingga perlu adanya sebuah desain *list* untuk menampung pilihan bahasa yang diinginkan.

2. Analisis batasan desain

Pembatasan desain Aplikasi Translator Messenger hanya berbasis text dan bersifat local. Hal ini dikarenakan *server* yang akan dibuat hanya bersifat pengganti.

2. Desain

Desain sistem dilakukan setelah analisis kebutuhan selesai. Tujuan dari desain sistem untuk membuat gambaran sistem perangkat lunak Aplikasi Translator Messenger yang akan dibuat.

Desain sistem dalam penelitian ini meliputi:

a. Perancangan *Database*

Database merupakan salah satu peranan penting dalam membangun sebuah perangkat lunak. Fungsi *database* untuk menyimpan data-data yang ada dalam aplikasi sehingga dapat digunakan kembali jika diperlukan. Pada

pembuatan Aplikasi Translator Messenger, terlebih dahulu peneliti merancang *database* yang akan digunakan untuk menjalankan sistem aplikasi tersebut.

Database Aplikasi Translator Messenger meliputi:

1) Tabel identitas

Tabel identitas adalah tabel untuk menampung data *user* yang melakukan pendaftaran diri sebagai *user* baru.

Field	Type
<u>username</u>	varchar(20)
firstname	varchar(20)
lastname	varchar(20)
password	varchar(50)
email	varchar(30)
gender	enum('L','P')

Gambar 3. Tabel identitas

2) Tabel mes

Tabel mes adalah tabel untuk menampung *user* yang sedang online.

Field	Type
<u>username</u>	varchar(30)
idsocket	int(20)
userLanguage	varchar(30)

Gambar 4. Tabel mes

b. Pemodelan Visual

Pemodelan visual didefinisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik menurut Boggs (2002). Ada beberapa notasi visual, diantaranya adalah:

1. Notasi Booch.

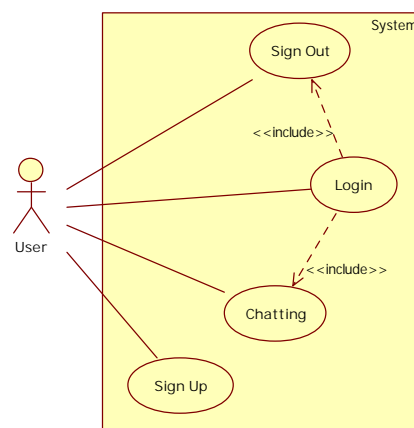
2. Notasi OMT(*Obyek Management Technology*).
3. Notasi UML(*Unified Modeling Language*).

Pada penelitian ini, pemodelan visual yang digunakan adalah notasi UML(*Unified Modeling Language*). Adapun beberapa UML yang digunakan oleh peneliti sebagai berikut:

1) *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari sistem Aplikasi Translator Messenger. *Use case* yang tercipta pada perangkat lunak Aplikasi Translator Messenger sebanyak 4 *use case* yaitu:

1. *use case login*
2. *use case sign up*
3. *use case chatting*
4. *use case sign out*.



Gambar 5. Use case system

a) Definisi aktor

Berikut ini adalah deskripsi aktor pada perangkat lunak Aplikasi

Translator Messenger:

Tabel 8. Deskripsi Aktor

No	Use case	Deskripsi
1	User	Orang yang dapat menggunakan aplikasi translator messenger

b) Definisi *Use Case*

Berikut ini adalah deskripsi *use case* pada perangkat lunak Aplikasi

Translator Messenger:

Tabel 9. Deskripsi *use case*

No	<i>Use case</i>	Deskripsi
1	<i>Login</i>	<i>Use case</i> yang berfungsi untuk validasi <i>user</i> dengan cara pengecekan terhadap <i>username</i> dan <i>password user</i> yang sesuai dengan data yang sudah tersimpan sebelumnya di dalam <i>database</i> .
2	<i>Sign Up</i>	<i>Use case</i> yang berfungsi untuk melakukan pendataan data <i>user</i> yang mendaftarkan diri sebagai <i>user</i> baru.
3	<i>Chatting</i>	<i>Use case</i> yang berfungsi untuk melakukan komunikasi antar <i>user</i> .
4	<i>Sign Out</i>	<i>Use case</i> yang berfungsi untuk keluar dari Aplikasi Translator Messenger.

c) Definisi skenario

Berikut ini skenario setiap *use case* yang sudah dijelaskan sebelumnya:

1. Skenario *use case Login*

Tabel 10. Skenario *use case Login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan <i>username, password</i> dan memilih bahasa	
	2. Mengecek valid tidaknya data masukkan
	3. Masuk ke Aplikasi Translator Messenger
Skenario Alternatif	
1. Memasukkan <i>username, password</i> dan memilih bahasa	
	2. Mengecek valid tidaknya data masukkan
	3. Menampilkan pesan login tidak valid
4. Memasukkan <i>username, password</i> dan memilih bahasa yang valid	
	5. Mengecek valid tidaknya data masukkan
	6. Masuk ke Aplikasi Translator Messenger

2. Skenario *use case Sign Up*Tabel 11. Skenario *use case Sign up*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan data pengguna sesuai kolom yang ada	
	2. Menyimpan data pengguna ke database
	3. Menampilkan pesan sukses disimpan
Skenario Alternatif	
1. Memasukkan data	

pengguna sesuai kolom yang ada	
	2. Mengeluarkan pesan bahwa data masukkan belum diisi semua
3. Memperbaiki data masukkan yang belum diisi	
	4. Menyimpan data pengguna ke <i>database</i>
	5. Menampilkan pesan sukses disimpan

3. Skenario use case *Chatting*

Tabel 12. Skenario *use case chatting*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih lawan user chatting dengan cara mengklik nama <i>user</i> yang diinginkan pada List User	
	2. Mengecek valid tidaknya nama <i>user</i>
	3. Menampilkan tab chat baru
4. Mengetikkan kalimat yang ingin ditulis ke dalam <i>input text</i> yang disediakan.	
	5. Menampilakn tab baru pada lawan <i>user</i> yang diajak <i>chatting</i> .
	6. Menampilkan percakapan kalimat pada <i>output text</i> yang disediakan.
	7. Mengkosongkan isi <i>input text</i> .
Skenario Alternatif	
1. Memilih lawan <i>user chatting</i> dengan cara mengklik nama <i>user</i>	

yang diinginkan pada <i>List User</i>	
	2. Mengecek valid tidaknya nama user
	3. Menampilkan pesan bahwa nama user yang diklik adalah <i>user</i> sendiri.
4. Mengklik nama <i>user</i> lain yang diinginkan pada <i>List User</i>	
	5. Mengecek valid tidaknya nama <i>user</i>
	6. Menampilkan tab chat baru
7. Mengetikkan kalimat yang ingin ditulis ke dalam <i>input text</i> yang disediakan.	
	8. Menampilakn tab baru pada lawan <i>user</i> yang diajak <i>chatting</i> .
	9. Menampilkan percakapan kalimat pada <i>output text</i> yang disediakan.
	10. Mengkosongkan isi <i>input text</i> .

4. Skenario user case *Sign Out*

Tabel 13. Skenario *use case sign out*

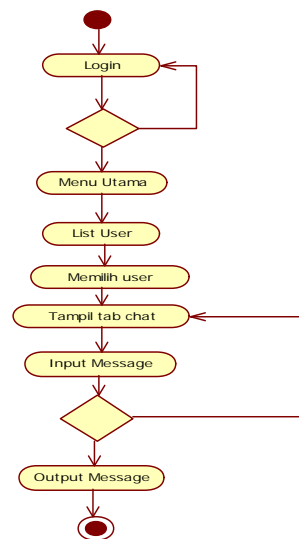
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengklik menu File dan memilih <i>Sign Out</i>	
	2. Keluar dari aplikasi

2) Activity Diagram

Activity diagram menggambarkan diagram alir aktivitas system perangkat lunak yang dibangun. Diagram ini memuat cara sistem bekerja dan memilih *decision* yang mungkin terjadi.

a) Activity Diagram *Chatting*

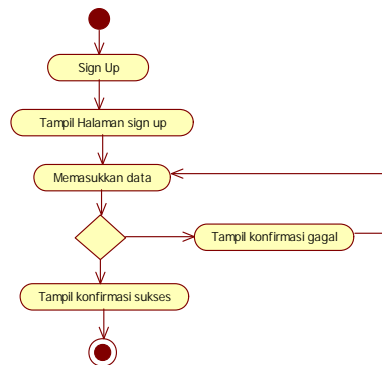
Activity diagram ini menggambarkan cara sistem Aplikasi Translator Messenger bekerja agar *user* dapat mengakses aplikasi untuk melakukan *chatting*. Pada activity diagram *chatting* dimulai dari login user hingga pilihan aktivitas untuk dapat mengirim pesan.



Gambar 6. Activity diagram Chatting

b) Activity Diagram *Sign Up*

Activity diagram ini menggambarkan cara sistem Aplikasi Translator Messenger menampilkan form *sign up* dan menyimpan data sebagai reaksi dari perintah *user*. Pada activity diagram *sign up* dimulai dari mengisi data user pada *interface* yang disediakan.



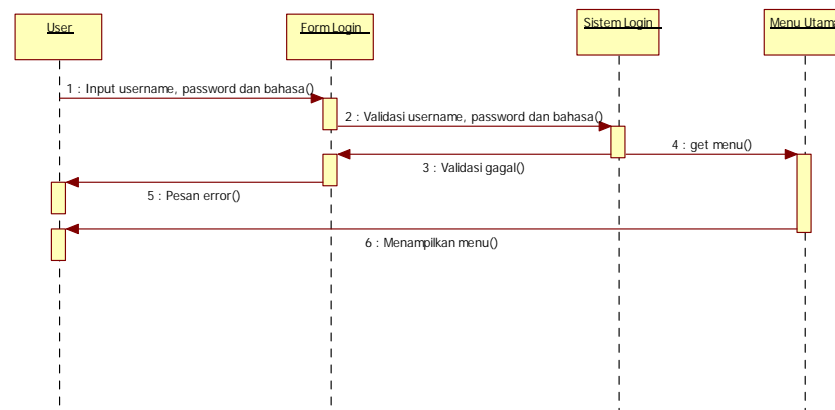
Gambar 7. Activity diagram *Sign Up*

3) Sequence Diagram

Sequence diagram bertujuan untuk menggambarkan interaksi antar objek di dalam dan di sekitar sistem.

a) Sequence Diagram *Login*

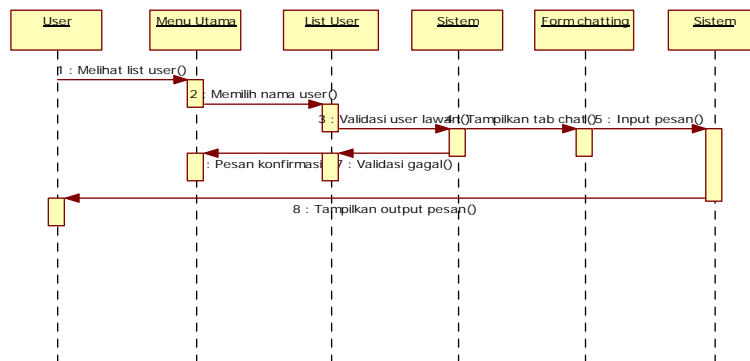
Pada sequence diagram *Login* dimulai dari user melakukan input username dan password pada tab yang disediakan. Selanjutnya sistem akan melakukan validasi username dan password user. Jika valid sistem secara langsung akan masuk ke halaman menu utama dan jika gagal akan muncul perintah error.



Gambar 8. Sequence Diagram *Login*

b) Sequence Diagram *Chatting*

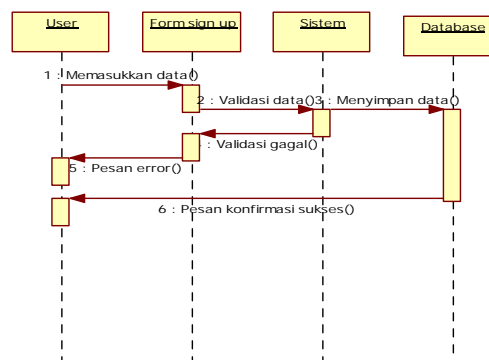
Pada sequence diagram *chatting* dimulai dari user memilih user lain sebagai lawan bicara. Selanjutnya sistem akan melakukan validasi dan menampilkan tab *chatting* baru. Pada tab *chatting* user dapat mulai mengetikkan pesan yang diinginkan.



Gambar 9. Sequence Diagram *Chatting*

c) Sequence Diagram *Sign Up*

Pada sequence diagram *sign up* dimulai dari user memasukkan data diri yang akan divalidasi oleh sistem. Sistem akan menampilkan konfirmasi pesan jika data sudah tersimpan di database.



Gambar 10. Sequence Diagram *Sign Up*

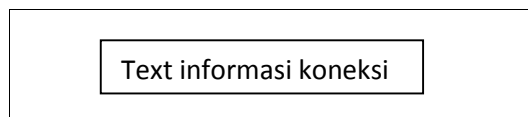
c. Desain *Interface*

Desain *interface* merupakan desain tampilan awal dari suatu perangkat lunak. Peranan *interface* dalam perangkat lunak sangatlah penting. Proses komunikasi tidak akan berjalan dengan baik jika user tidak dapat memahami *interface* dari suatu perangkat lunak.

Aplikasi Translator Messenger memiliki beberapa rancangan *interface* yang akan dijelaskan sebagai berikut:

1) Chatser

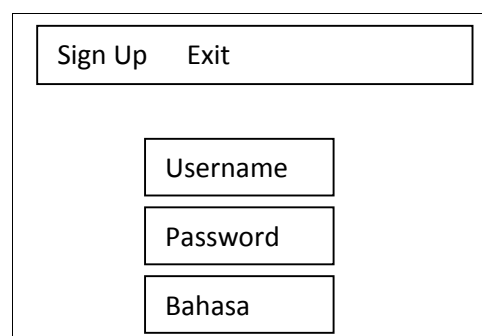
Gambar di bawah ini adalah rancangan desain interface server buatan Aplikasi Translator Messenger.

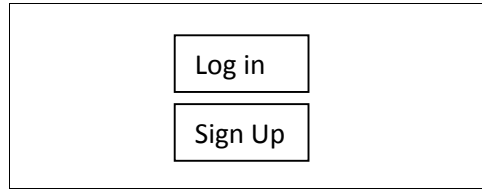


Gambar 12. Rancangan *interface chatser*

2) Form Login

Gambar di bawah ini adalah rancangan form login sebagai halaman awal aplikasi. Form login berfungsi sebagai validasi user agar dapat masuk ke Aplikasi Translator Messenger.



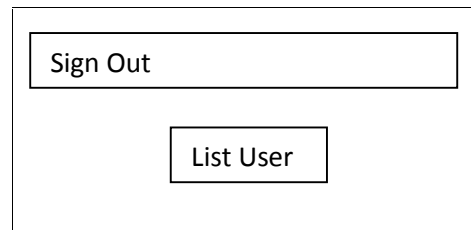


Gambar 13. Rancangan *interface login*

3) Form Menu Utama

Gambar di bawah ini merupakan rancangan desain menu utama aplikasi.

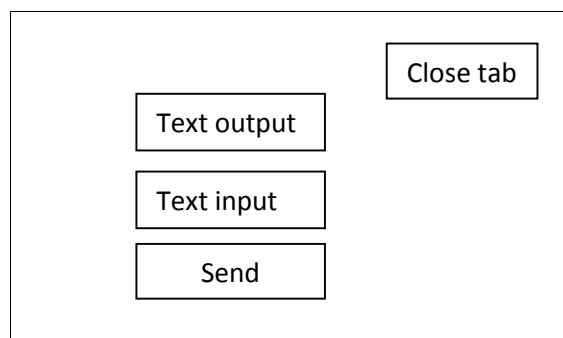
Pada menu utama aplikasi hanya berisi list user.



Gambar 14. Rancangan *interface menu utama*

4) Form Chatting

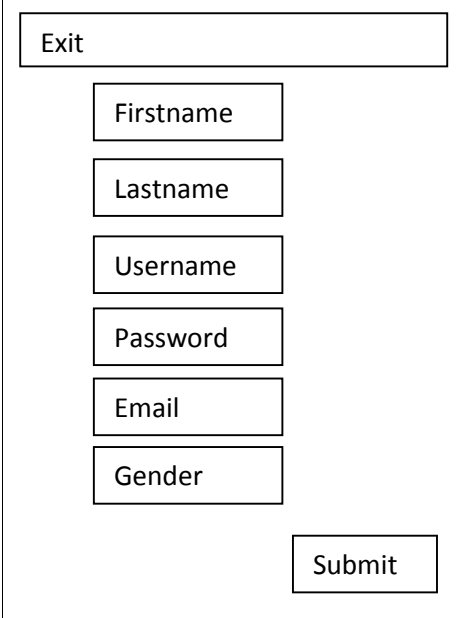
Gambar di bawah ini merupakan rancangan desain form chatting.



Gambar 15. Rancangan *interface chatting*

5) Form Sign Up

Gambar di bawah ini merupakan rancangan desain form sign up. Form sign up difungsikan sebagai tempat mendaftarkan diri sebagai user baru.



The image shows a wireframe of a sign-up form. It consists of a large outer rectangle containing several smaller rectangular input fields and buttons. At the top left is an 'Exit' button. Below it, the input fields are stacked vertically: 'Firstname', 'Lastname', 'Username', 'Password', 'Email', and 'Gender'. At the bottom right of the form is a 'Submit' button.

Gambar 16. Rancangan *interface sign up*

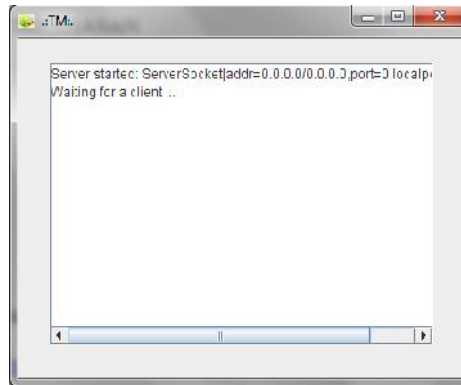
3. Pengkodean

Pengkodean dilakukan setelah proses analisis dan design selesai. Pada proses ini, pengkodean aplikasi mulai dibangun menggunakan bahasa pemrograman yang telah ditetapkan yaitu bahasa pemrograman Java.

Hasil dari pengkodean perangkat lunak Aplikasi Translator Messenger dibagi menjadi beberapa form. Penjelasan di bawah ini akan menjelaskan form yang ada di dalam Aplikasi Translator Messenger.

a. Chatser

From chatser pada Aplikasi Translator Messenger berfungsi sebagai server buatan. Chatser akan menunggu client terkoneksi ke server untuk melakukan permintaan layanan.



Gambar 17. Halaman *Chatser*

b. Form Login

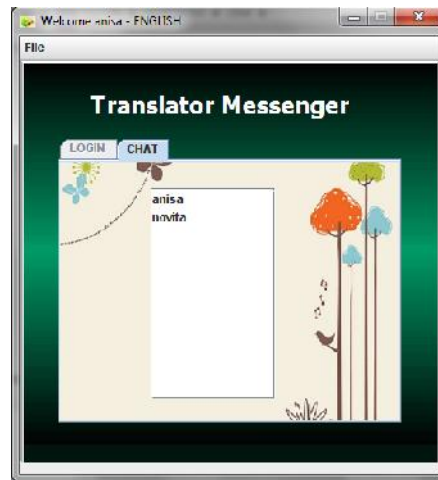
Tampilan pertama dari Aplikasi Translator Messenegr adalah form Login. Form ini berisi inputan username dan password. Form ini bertujuan untuk memvalidasi input username dan password user yang akan masuk ke dalam aplikasi. Jika validasi gagal, sistem akan menampilkan pesan error kepada user. Namun jika validasi berhasil, sistem akan langsung menampilkan form utama Aplikasi Translator Messenger.



Gambar 18. Halaman *Login*

c. Menu Utama

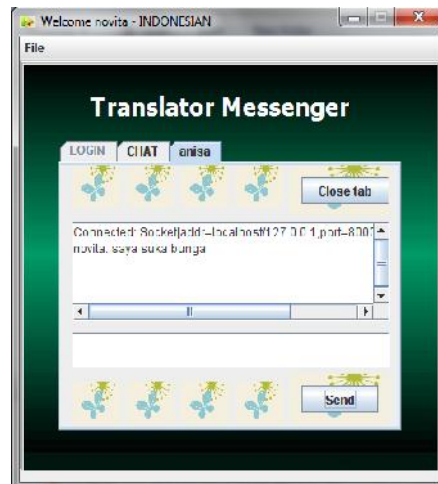
Menu utama berisi list user yang berfungsi untuk menampung online user. Jika ingin melakukan chatting, user dapat memilih user lawan dengan cara meng-klik nama user yang diinginkan.



Gambar 19. Halaman menu utama

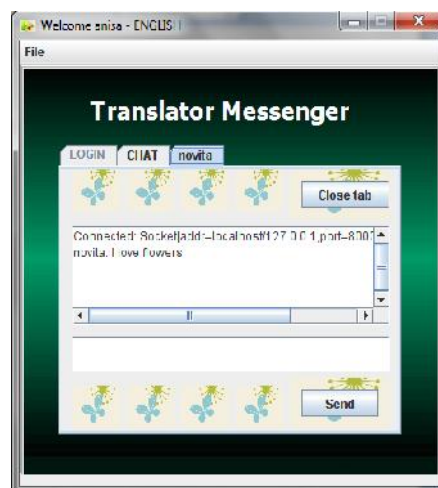
d. Form Chatting

Menu chatting berupa form percakapan antara user yang sedang melakukan chatting. Form ini akan muncul secara otomatis ketika user telah memilih user lawan yang diinginkan. Setelah user mengirim pesan kepada user lawan, secara otomatis pula form chatting akan muncul pada halaman user lawan. Pada Aplikasi Translator Messenger, pesan yang dikirimkan akan langsung diterjemahkan sesuai dengan bahasa yang dikuasai oleh user lawan. Sehingga output yang ditampilkan pada halaman user berbeda dengan output yang ditampilkan pada halaman user lawan.



Gambar 20. Halaman chatting

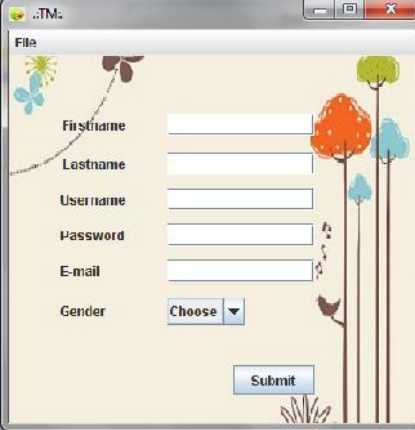
Dari gambar di bawah ini, dapat dilihat hasil terjemahan pesan yang dikirim dari bahasa Indonesia ke dalam bahasa Inggris.



Gambar 21. Halaman Chatting user lawan

e. Form Sign Up

Pada form sign up, user dapat mendaftarkan diri sebagai user baru. Sistem akan langsung menampilkan pesan error jika terdapat input yang belum diisi. Data yang dimasukkan oleh user akan disimpan ke dalam database jika validasi sukses.

A screenshot of a web browser window displaying a sign-up form. The window title is ".TM:". The form is titled "File" and contains the following fields: "Firstname", "Lastname", "Username", "Password", "E-mail", and "Gender" (with a "Choose" dropdown menu). A "Submit" button is located at the bottom right of the form. The background of the form is decorated with a light beige color and features a whimsical illustration of trees and butterflies.

Gambar 22. Halaman Sign Up

4. Pengujian

Pengujian Aplikasi Translator Messenger terdiri dari tiga tahapan yaitu:

a. Pengujian *Black-box*

Pengujian black box dilakukan oleh peneliti untuk mengetahui kinerja produk sebelum diujikan oleh penguji Ahli(*expert*).

b. Pengujian *correctness*

Pengujian correctness dilakukan oleh peneliti sendiri menggunakan tools yang sudah banyak digunakan yaitu LOC counter dan FindBugs. Loc counter untuk menghitung jumlah baris kode dan FindBugs untuk mengetahui apakah ada cacat pada perangkat lunak yang dikembangkan.

c. Pengujian *reliability*

Pengujian reliability juga dilakukan oleh peneliti sendiri dengan menghitung nilai MTBF perangkat lunak untuk mengetahui kehandalan perangkat lunak tersebut.

d. Pengujian *Alpha*

Pengujian alpha dilakukan oleh dua orang dosen Pendidikan Teknik Informatika Universitas Negeri Yogyakarta yaitu, Drs. Totok Sukardiyono, M.T. dan Adi Dewanto, M.Kom selaku penguji ahli(*expert*).

e. Pengujian *Beta*

Pengujian beta dilakukan oleh mahasiswa Pendidikan Teknik Informatika Universitas Negeri Yogyakarta sebanyak 16 orang.

B. Deskripsi Program

Aplikasi Translator Messenger merupakan aplikasi berbasis desktop yang bisa digunakan untuk berkomunikasi bahasa yang berbeda melalui pesan teks atau chatting. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan Google Translate API sebagai penerjemah bahasa yang disediakan dalam aplikasi tersebut. Perangkat lunak Aplikasi Translator Messenger memiliki beberapa bagian, yakni:

1. Login

Fitur login digunakan untuk mengatur keamanan dari Aplikasi Translator Messenger. Pengguna yang dapat mengakses aplikasi hanya user yang sudah mendaftarkan diri sebagai user. Pada fitur ini berisi inputan username, inputan password dan pilihan bahasa. Inputan username dan password yang sudah diisi oleh user akan divalidasi dengan cara melakukan pengecekan ke dalam database. Jika inputan username dan password sesuai maka sistem secara otomatis akan menampilkan menu utama aplikasi. Fitur ini juga dilengkapi

dengan menu sign up. Menu sign up diperuntukkan bagi pengguna yang belum mendaftarkan diri sebagai user.

2. Sign Up

Fitur ini dapat diakses pada halaman login. Pengguna dapat mengakses melalui menu file dan memilih sign up atau langsung mengklik sign up di bawah tombol log in.

Pada tampilan sign up berisi firstname, lastname, username, password, email, dan pilihan gender. Semua inputan harus diisi tanpa kecuali. Jika terdapat inputan yang belum terisi, secara otomatis sistem akan mengirimkan peringatan error.

3. Chatting

Fitur ini adalah antarmuka yang digunakan untuk proses komunikasi berbasis teks atau chatting. Fitur ini berisi area output pesan dan area input pesan yang akan dikirim ke user lawan.

4. Sign Out

Fitur ini digunakan untuk keluar dari Aplikasi Translator Messenger. User dapat mengakses melalui menu file dan memilih menu sign out. Jika memilih menu sign out secara otomatis aplikasi akan menutup dan koneksi ke database terputus.

C. Pembahasan

1. Pengujian Black-box

Pengujian black-box adalah pengujian yang dilakukan oleh peneliti sendiri untuk memastikan bahwa Aplikasi Translator Messenger sudah siap untuk digunakan. Pengujian ini dilakukan sebelum peneliti melakukan pengujian alpha.

Tabel 14. Pengujian black-box

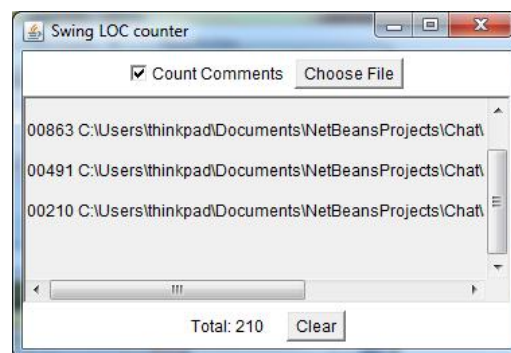
No	Use Case	Hasil Pengujian
Login		
1	Menampilkan pesan error	Ya
Sign Up		
2	Menampilkan pesan error	Ya
3	Menampilkan konfirmasi penyimpanan data sukses	Ya
Chatting		
4	Mengirim pesan sesuai dengan tujuan user	Ya
5	Menampilkan output pesan	Ya
6	Menutup tab chat	Ya
Sign Out		
7	Menampilkan pesan sign out	Ya
8	Menutup aplikasi secara sempurna	Ya
OS		
9	Pengujian pada OS Windows	Ya
10	Pengujian pada OS Mac	Ya

Dari pengujian *black-box*, dihasilkan bahwa fungsionalitas tiap fungsi pada Aplikasi Translator Messenger dapat berjalan sesuai dengan tujuan dari

fungsi tersebut. Sehingga, dapat dikatakan Aplikasi Translator Messenger sudah siap untuk dilakukan pengujian *alpha*.

2. Pengujian Correctness

Menurut Pressman(2010), pengujian correctness dapat diukur dengan analisis defect per KLOC (error pada setiap KLOC/Kilo Line of Code). Untuk mencari KLOC bisa menggunakan tool yang banyak disediakan. Pada kasus ini, peneliti menggunakan LOC counter.



Gambar 23. Penjumlahan baris kode

Class program yang telah dikembangkan kemudian dihitung LOCnya menggunakan LOC counter. Hasil dari perhitungan LOC dapat dilihat pada tabel berikut:

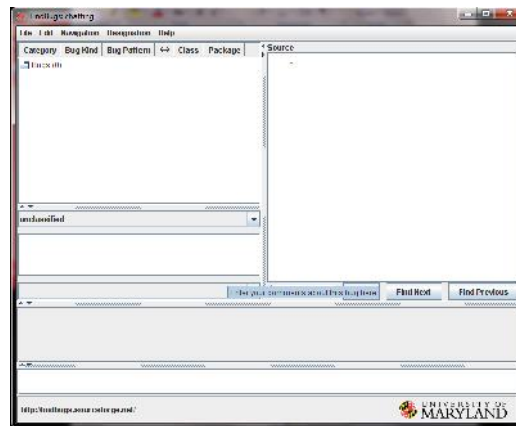
Tabel 15. Daftar kelas pada program

No	Keterangan	LOC
1	ChatServer.java	123
2	ChatServerThread.java	45
3	Login.java	863
4	Message.java	491
5	Sign.java	210
6	ThreadC.java	32

	Jumlah	1770
--	--------	------

Dari tabel diatas dapat dilihat total LOC dari perangkat lunak yang dibuat sebanyak 1770. Jadi LOC = 1,77 KLOC.

Selanjutnya, untuk menghitung cacat, peneliti menggunakan FindBugs. FindBugs digunakan untuk mendeteksi error yang terjadi pada kode-kode program yang telah dibuat. Pada gambar dibawah ini dapat dilihat bahwa tidak ada cacat pada perangkat lunak yang dibuat atau dapat dikatakan cacat = 0.



Gambar 24. Tampilan FindBugs

Selanjutnya, penghitungan dilakukan sesuai dengan teori Pressman (2010), yaitu cacat/KLOC = 0/1.77

$$= 0$$

Dari hasil tersebut dapat dikatakan bahwa tingkat correctness memenuhi standar. Hal ini dikarenakan error yang terjadi sebanyak 0. Sebagai pembanding standar jumlah error yang seharusnya, peneliti menggunakan rumus Watson – Felix Model seperti yang dijelaskan oleh Pressman (2010).

$$E = 5.2 \times (\text{KLOC})^{0.91}$$

$$\begin{aligned}
 &= 5.2 \times (1,77)^{0,91} \\
 &= 5.2 \times 1.68 \\
 &= 8.736
 \end{aligned}$$

Dari hasil perhitungan didapat maksimal jumlah error adalah 8.736. Sehingga dapat dilihat hasil perhitungan sebenarnya lebih kecil dari hasil perhitungan standar ($0 < 8.736$). Dari hasil perbandingan dapat disimpulkan bahwa perangkat lunak yang dikembangkan sudah memenuhi standar indikator *correctness*.

3. Pengujian Reliability

Menurut Scoot Speaks (2004), pengujian *reliability* dapat dihitung menggunakan pendekatan MTBF (Mean time between failure). *Mean time between failure* merupakan hasil dari banyaknya kesalahan yang terjadi selama pengoperasian perangkat lunak dalam jangka tertentu. Waktu yang digunakan dalam pengujian *reliability* merupakan prediksi singkat dari pengujian perangkat lunak yang sebenarnya. Maksudnya, waktu yang digunakan bukan merupakan lamanya perangkat lunak dapat digunakan hingga tidak dapat digunakan kembali. Di bawah ini merupakan hasil pengujian *reliability* perangkat lunak:

$$\begin{aligned}
 \text{MTBF} &= T/R \\
 &= 5/1 \\
 &= 5 \text{ hour/failure}
 \end{aligned}$$

Dari hasil perhitungan dapat disimpulkan terjadi 1 kesalahan selama pengoperasian perangkat lunak yang berdurasi 5 jam.

Sebagai perbandingan kehandalan produk perangkat lunak, peneliti membandingkan nilai MTBF perangkat lunak yang dikembangkan dengan nilai MTBF Microsoft Word. Menurut QuickSilver Controls (2005), nilai MTBF Microsoft Word pada pengujian selama 17 jam terjadi kesalahan (*failure*) sebanyak 1. Dari perbandingan tersebut dapat disimpulkan bahwa perangkat lunak yang dibangun memiliki kehandalan dibawah Microsoft Word.

4. Pengujian Alpha

Pengujian alpha adalah pengujian yang dilakukan setelah peneliti melakukan pengujian black-box. Pengujian ini bertujuan untuk mengetahui unjuk kerja dari perangkat lunak yang sudah dibangun.

Pengujian ini dilakukan oleh ahli aplikasi yang dikembangkan oleh peneliti. Dalam hal ini adalah Aplikasi Translator Messenger. Indikator pengujian alpha meliputi, indikator correctness, reliability, functionality, dan usability. Ahli akan menilai sejauh mana aplikasi sesuai dengan indikator kelayakan suatu perangkat lunak. Hasil dari penilaian ahli akan menjadi revisi awal untuk memperbaiki kekurangan Aplikasi Translator Messenger.

Secara keseluruhan, peneliti menilai Aplikasi Translator Messenger sudah sesuai dengan yang diharapkan. Ahli menilai unjuk kerja aplikasi tersebut sudah baik. Namun terdapat beberapa masukan untuk menyempurnakan aplikasi tersebut, yakni:

1. Penambahan form profil yang bertujuan agar user dapat melihat profile user lain.

2. Penambahan pengaturan font dalam form chatting. Hal ini bertujuan untuk mengubah level besar kecilnya tulisan pesan yang akan dikirim.
3. Membuat database menjadi tersentral ke server. Hal ini lebih memudahkan dalam pengelolaan aplikasi.
4. Pengaturan form login yang secara otomatis langsung menampilkan menu utamajika validasi username dan password berhasil. Hal ini bertujuan agar interface lebih memudahkan user.
5. Pengaturan scroll bar yang otomatis turun ke bawah mengikuti pesan baru yang ditampilkan pada area output pesan.

Berdasarkan penilai dari ahli, maka diperoleh data sebagai berikut:

- a. Pengujian alpha berdasarkan indikator *functionality*

Tabel 16. Hasil pengujian *functionality*

Ahli	Kuesioner functionality											
	1	2	3	4	5	6	7	8	9	10	11	12
1	5	5	5	5	5	4	4	4	4	5	5	5
2	4	5	4	4	4	4	4	4	4	5	5	5
Jumlah	9	10	9	9	9	8	8	8	8	10	10	10
n	108											
N	120											
DP(%)	100%											

Hasil dari penilaian ahli dari segi *functionality* dapat dilihat sebagai berikut:

$$DP = \frac{108}{120} \times 100 \%$$

$$= 90\%$$

Untuk skor ideal:

$$\text{Untuk jawaban sangat setuju: } 5 \times 12 \times 2 = 120$$

$$DP = \frac{120}{120} \times 100 \% = 100\%$$

Untuk jawaban tidak setuju : $1 \times 12 \times 2 = 24$

$$DP = \frac{24}{120} \times 100 \% = 20\%$$

Dengan interval 5 skala maka didapat sebagai berikut:

Tabel 17. Skala kelayakan

No	Presentase	Keterangan
1	20% - 35%	Sangat Tidak Layak
2	36% - 51%	Kurang Layak
3	52% - 67%	Cukup Layak
4	68% - 83%	Layak
5	84% - 100%	Sangat Layak

Sehingga dapat disimpulkan bahwa perangkat lunak menghasilkan presentase sebesar 90% dari rentang 84% - 100% dan dapat dikategorikan sangat layak.

b. Pengujian alpha berdasarkan indikator *usability*

Tabel 18. Hasil pengujian *usability*

Ahli	Kuesioner Usability			
	13	14	15	16
1	4	5	5	4
2	4	4	4	4
Jumlah	8	9	9	8
n	34			
N	40			
DP(%)	85%			

Hasil dari penilaian ahli dari segi *usability* dapat dilihat sebagai berikut:

$$DP = \frac{34}{40} \times 100 \%$$

$$= 85\%$$

Untuk skor ideal:

Untuk jawaban sangat setuju: $5 \times 4 \times 2 = 40$

$$DP = \frac{40}{40} \times 100 \% = 100\%$$

Untuk jawaban tidak setuju : $1 \times 4 \times 2 = 8$

$$DP = \frac{8}{40} \times 100 \% = 20\%$$

Sehingga dapat disimpulkan bahwa perangkat lunak menghasilkan presentase sebesar 85% dari rentang 84% - 100% dan dapat dikategorikan sangat layak.

5. Pengujian Beta

Pengujian beta merupakan pengujian terakhir dari suatu produk perangkat lunak. Pengujian ini dilakukan setelah pengujian alpha selesai. Pengujian ini dilakukan oleh responden sebanyak 16 orang mahasiswa Pendidikan Teknik Informatika Universitas Negeri Yogyakarta. Berikut ini hasil pengujian beta yang didapat:

Tabel 19. Hasil pengujian *beta*

User	Functionality	Usability
1	33	15
2	37	16
3	33	16
4	34	16
5	39	18
6	37	18

7	37	17
8	36	17
9	38	17
10	35	16
11	33	16
12	40	18
13	34	17
14	32	17
15	39	18
16	37	15
n	574	267
N	640	320
DP(%)	89.69%	83.45%

$$DP = \frac{n}{N} \times 100 \%$$

Keterangan :

DP = Deskriptif Persentase (%)

n = Skor empirik (Skor yang diperoleh)

N = Skor ideal (Ditentukan berdasarkan spesifikasi sistem)

Dari hasil penilaian di atas, dapat diperoleh presentase sebesar 89.69% dari segi *functionality* dan dikategorikan sangat layak serta 83,45% dari segi *usability* dan dikategorikan layak.

6. Revisi akhir

Perangkat lunak Aplikasi Translator Messenger telah selesai melakukan tahapan pengujian. Pengujian dimulai dari pengujian *black-box*, pengujian *alpha* dan terakhir pengujian *beta*. Pengujian *black-box* dilakukan oleh peneliti sendiri. Pengujian *black-box* digunakan untuk memastikan perangkat lunak telah siap untuk dilakukan pengujian lebih lanjut. Pengujian alpha

dilakukan oleh 2 orang ahli(*expert*) rekayasa perangkat lunak. Sedangkan pengujian beta dilakukan oleh 16 orang mahasiswa.

Perangkat lunak Aplikasi Translator Messenger yang dibangun memiliki beberapa kelebihan dan kelemahan. Beberapa kelebihan dari perangkat lunak tersebut adalah:

- a. Perangkat lunak Aplikasi Translator Messenger dapat langsung menerjemahkan pesan ke dalam bahasa yang dikuasai oleh user.
- b. Bahasa yang disediakan cukup banyak sehingga pilihan bahasa beragam.
- c. Perangkat lunak ini bisa diakses atau digunakan oleh siapa saja tanpa tergantung instansi atau kepemilikan email tertentu.

Sedangkan kelemahan dari perangkat lunak Aplikasi Translator Messenger sebagai berikut:

- a. Perangkat lunak Aplikasi Translator Messenger dibangun hanya berbasis text.
- b. Perangkat lunak Aplikasi Translator Messenger hanya bisa dijalankan pada local area saja. Hal ini dikarenakan server yang dibuat hanya sebagai server buatan.

BAB V

KESIMPULAN DAN SARAN

A. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan terhadap perangkat lunak Aplikasi Translator Messenger yang dibangun, maka dapat disimpulkan sebagai berikut:

1. Perangkat lunak Aplikasi Translator Messenger dibuat menggunakan metode penelitian dan pengembangan. Melalui tahapan analisis, desain, pengkodean dan pengujian. Perangkat lunak ini dikembangkan menggunakan bahasa pemrograman Java dan diintegrasikan dengan Google Translate API sebagai penerjemah bahasa.
2. Hasil pengujian *black-box* menyatakan bahwa unjuk kerja fungsionalitas semua fungsi pada Aplikasi Translator Messenger dapat dikatakan berfungsi sesuai dengan tujuan dari aplikasi tersebut.
3. Hasil pengujian kelayakan menyatakan sebagai berikut:
 - a. Hasil pengujian *correctness* menyatakan bahwa perangkat lunak Aplikasi Translator Messenger telah memenuhi standar indikator *correctness*.
 - b. Hasil pengujian *reliability* menyatakan bahwa setiap 5 jam pengoperasian perangkat lunak terjadi 1 kesalahan program.

- c. Hasil penilaian pengujian alpha yang dilakukan oleh ahli, menghasilkan dari segi *functionality* sebesar 90% dari rentang 84% - 100% dan dapat dikategorikan sangat layak serta dari segi *usability* 85% dari rentang 84% - 100% dan dapat dikategorikan sangat layak.
- d. Sedangkan hasil pengujian beta menghasilkan 89.69% dari segi *functionality* dan dikategorikan sangat layak serta 83.45% dari segi *usability* dan dikategorikan layak.

B. SARAN

Berdasarkan penelitian terhadap Aplikasi Translator Messenger yang telah dilakukan, maka dapat disimpulkan beberapa saran sebagai berikut:

1. Perlu adanya penambahan fitur-fitur yang lebih mendayagunakan aplikasi messenger, seperti video call, group discussion, mengirim dokumen atau file, mengirim gambar dsb.
2. Perlu adanya penambahan format aplikasi yang berbasis mobile dan web-based.

LAMPIRAN

Lampiran 1. SK Skripsi

**KEPUTUSAN DEKAN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 09/ELK/Q-1/I/2011
TENTANG
PENGANGKATAN PEMBIMBING TUGAS AKHIR SKRIPSI
BAGI MAHASISWA FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

**DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

- Menimbang** : 1. Bahwa sehubungan dengan telah dipenuhi syarat untuk penulisan Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta, perlu diangkat pembimbing.
2. Bahwa untuk keperluan dimaksud perlu ditetapkan dengan Keputusan Dekan.
- Mengingat** : 1. Undang-undang Nomor 20 tahun 2003.
2. Peraturan Pemerintah RI Nomor 60 tahun 1999.
3. Keputusan Presiden RI: a. Nomor 93 tahun 1999; b. 305/M tahun 1999.
4. Keputusan Menteri Pendidikan dan Kebudayaan RI: Nomor 274/O/1999.
5. Keputusan Mendiknas RI Nomor 003/O/2001.
6. Keputusan Rektor UNY Nomor : 1160/UN34/KP/2011

MEMUTUSKAN

Menetapkan

Pertama : Mengangkat Pembimbing Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta sebagai berikut :

Nama Pembimbing : Herman Dwi Surjono, Ph.D.
Bagi mahasiswa :
Nama/No.Mahasiswa : Novita Anindita / 08520241003
Jurusan/ Prodi : Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika

Kedua : Dosen pembimbing disertai tugas membimbing penulisan Tugas Akhir Skripsi sesuai dengan Pedoman Tugas Akhir Skripsi.

Ketiga : Keputusan ini berlaku sejak ditetapkan

Keempat : Segala sesuatu akan diubah dan dibetulkan sebagaimana mestinya apabila di kemudian hari ternyata terdapat kekeliruan dalam Keputusan ini.

Ditetapkan : di Yogyakarta
Pada tanggal : 10 Januari 2012



Mooh. Bruri Triyono
NIP. 19560216 198603 1 003

Tembusan Yth :

1. Pembantu Dekan I, II, III FT UNY
2. Ketua Jurusan Pendidikan Teknik Elektronika
3. Ka Bag Tata Usaha FT UNY
4. Yang bersangkutan

Lampiran 2. Source Code

Chatser

```

import java.net.*;
import java.io.*;
import java.util.logging.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.text.DefaultCaret;

public class ChatServer extends javax.swing.JFrame implements Runnable {
private ChatServerThread clients[] = new ChatServerThread[50];
private ServerSocket server = null;
private Thread thread = null;
private int clientCount = 0;

public Connection conn;
public Statement st;
public ResultSet rs;
public DefaultListModel list;
String usercocok, sql,sqllis;
//AmbilClient ambil = new AmbilClient();

    public ChatServer() {
        try {
            initComponents();
            textser.setEditable(false);
            setTitle(".:TM:");
            setIconImage(new javax.swing.ImageIcon(getClass().
getResource("sun.jpg")).getImage());
            server = new ServerSocket(8000);
            textser.setText("Server started: " + server);
            start();
        } catch (IOException ex) {
            Logger.getLogger(ChatServer.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

    public void run()
    {
        while (thread != null)
        {
            try
            {
                textser.append("\n"+"Waiting for a client ...");

```

```

        addThread(server.accept());
        DefaultCaret caret = (DefaultCaret)textser.getCaret();
        caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);
    }
    catch(IOException ioe)
    {
        System.out.println("Server accept error: " + ioe); stop(); }
    }
}

public void start()
{
    if (thread == null)
    {
        thread = new Thread(this);
        thread.start();
    }
}

public void stop()
{
    if (thread != null)
    {
        thread.stop();
        thread = null;
    }
}

private int findClient(int ID)
{
    for (int i = 0; i < clientCount; i++)
        if (clients[i].getID() == ID)
            return i;
    return -1;
}

public synchronized void handle(int ID, String input)
{
    String [] hikhik = input.split(" ");
    String usersend = hikhik[3];

    if (usersend.equals("saiaremove")){
        for (int i = 0; i < clientCount; i++) {
            clients[i].send(input);
        }
        remove(ID);
    }
}

```

```

        } else {
            for (int i = 0; i < clientCount; i++) {
                clients[i].send(input);
            }
        }
    }

    @Override
    public synchronized void remove(int ID)
    {
        int pos = findClient(ID);
        if (pos >= 0)
        {
            ChatServerThread toTerminate = clients[pos];
            textser.append("\n"+"Removing client thread " + ID);
            if (pos < clientCount-1)
                for (int i = pos+1; i < clientCount; i++)
                    clients[i-1] = clients[i];
            clientCount--;
        }
        try
        {
            toTerminate.close();
        }
        catch(IOException ioe)
        {
            System.out.println("Error closing thread: " + ioe);
        }
        toTerminate.stop();
    }

    private void addThread(Socket socket)
    {
        if (clientCount < clients.length)
        {
            textser.append("\n" + "Client accepted: " + socket);
            clients[clientCount] = new ChatServerThread(this, socket);
            try {
                clients[clientCount].open();
                clients[clientCount].start();
                clientCount++;
            } catch (IOException ioe) {
                System.out.println("Error opening thread: " + ioe);
            }
        }
        else
            System.out.println("Client refused: maximum " + clients.length + " reached.");
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```

private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    textser = new javax.swing.JTextArea();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    textser.setColumns(20);
    textser.setRows(5);
    jScrollPane1.setViewportView(textser);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 345,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(28, Short.MAX_VALUE))
        );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 250, Short.MAX_VALUE)
                .addGap(29, 29, 29))
            );

    pack();
} // </editor-fold>

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ChatServer().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JScrollPane jScrollPane1;

```

```
private javax.swing.JTextArea textser;  
// End of variables declaration  
}
```

ChatServerThread

```
import java.net.*;
import java.io.*;

public class ChatServerThread extends Thread
{ private ChatServer  server  = null;
  private Socket      socket  = null;
  private int         ID      = -1;
  private DataInputStream streamIn = null;
  private DataOutputStream streamOut = null;
  // AmbilClient ambil = new AmbilClient();

  public ChatServerThread(ChatServer server1, Socket socket1)
  {
    server = server1;
    socket = socket1;
    ID = socket.getPort();
  }

  public void send(String msg)
  { try
    { streamOut.writeUTF(msg);
      streamOut.flush();
    }
    catch(IOException ioe)
    { System.out.println(ID + " ERROR sending: " + ioe.getMessage());
      server.remove(ID);
      stop();
    }
  }

  public int getID()
  { return ID;
  }

  @Override
  public void run()
  {
    System.out.println("Server Thread " + ID + " running.");
    while (true)
    { try
      {
        server.handle(ID, streamIn.readUTF());
      }
      catch(IOException ioe)
      {
```

```
        System.out.println(ID + " ERROR reading: " + ioe.getMessage());
        server.remove(ID);
        stop();
    }
}

public void open() throws IOException
{
    streamIn = new DataInputStream(new
        BufferedInputStream(socket.getInputStream()));
    streamOut = new DataOutputStream(new
        BufferedOutputStream(socket.getOutputStream()));
}

public void close() throws IOException
{
    if (socket != null) socket.close();
    if (streamIn != null) streamIn.close();
    if (streamOut != null) streamOut.close();
}
}
```

Login

```

import com.google.api.GoogleAPI;
import com.google.api.GoogleAPIException;
import com.google.api.translate.Language;
import com.google.api.translate.Translate;
import java.awt.event.*;
import java.util.logging.*;
import javax.swing.*;
import java.sql.*;
import java.util.*;
import java.io.*;
import java.net.*;
import javax.swing.Timer;

public class Login extends javax.swing.JFrame implements Runnable{
    public Connection conn;
    public Statement st;
    public ResultSet rs;
    private Socket socket          = null;
    private Thread thread          = null;
    private DataInputStream console = null;
    private DataOutputStream streamOut = null;
    private ThreadC client        = null;
    DefaultListModel list        = new DefaultListModel();
    DefaultListModel check       = new DefaultListModel();
    String username, password, chatWith, pilihbahasa, useraccept,
        usersend, languagesend, languageaccept, hasiltranslate, passcocok;
    String [] hikhik;
    Timer timer;
    JPanel panel2;
    int x;

    Vector pms = new Vector();
    Login clientApplication;
    ArrayList ar = new ArrayList();

    public Login() {
        initComponents();
        koneksi("localhost", "chatting", "root", "");
        tabbedPane.setEnabledAt(1, false);
        combolanguage();

        GoogleAPI.setHttpReferrer("http://ajax.googleapis.com/ajax/services/language/translate");
        GoogleAPI.setKey("AIzaSyCGyGqo7zI-
rHLQbeWA_ixxBg3pmLEqv4M");

```

```

        menuSignout.setVisible(false);
        inputUsername.setFocusable(true);
        Login.this.
        setTitle(".:TM:");
        setIconImage(new javax.swing.ImageIcon(
        getClass().
        getResource("sun.jpg")).getImage());
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        panelbackk1 = new chat.panelbackk();
        jLabel1 = new javax.swing.JLabel();
        tabbedPane = new javax.swing.JTabbedPane();
        loginPanel = new javax.swing.JPanel();
        panelback1 = new chat.panelback();
        labelUserNName = new javax.swing.JLabel();
        inputUsername = new javax.swing.JTextField();
        labelPassword = new javax.swing.JLabel();
        inputPassword = new javax.swing.JPasswordField();
        bahasa = new javax.swing.JComboBox();
        connect = new javax.swing.JButton();
        SignUp = new javax.swing.JLabel();
        chatPanel = new javax.swing.JPanel();
        panelback2 = new chat.panelback();
        jScrollPane2 = new javax.swing.JScrollPane();
        listUser = new javax.swing.JList();
        jMenuItemBar1 = new javax.swing.JMenuBar();
        File = new javax.swing.JMenu();
        menuSignup = new javax.swing.JMenuItem();
        menuSignout = new javax.swing.JMenuItem();
        menuExit = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                formWindowClosing(evt);
            }
        });

        jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

```

```

panelbackk1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24));
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("Translator Messenger");

labelUserName.setText("Username");

inputUsername.setCursor(new
java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
inputUsername.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        inputUsernameKeyReleased(evt);
    }
});

labelPassword.setText("Password");

bahasa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bahasaActionPerformed(evt);
    }
});

connect.setText("Log In");
connect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        connectActionPerformed(evt);
    }
});

SignUp.setText("Sign Up");
SignUp.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        SignUpMouseClicked(evt);
    }
});

javax.swing.GroupLayout panelback1Layout = new
javax.swing.GroupLayout(panelback1);
panelback1.setLayout(panelback1Layout);
panelback1Layout.setHorizontalGroup(

panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(panelback1Layout.createSequentialGroup())

    .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelback1Layout.createSequentialGroup())
        .addGap(52, 52, 52)

    .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(panelback1Layout.createSequentialGroup())
        .addComponent(labelUserName)
        .addGap(18, 18, 18)
        .addComponent(inputUsername))
        .addGroup(panelback1Layout.createSequentialGroup())
        .addComponent(labelPassword)
        .addGap(18, 18, 18)

    .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(inputPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(connect))))
        .addGroup(panelback1Layout.createSequentialGroup())
        .addGap(90, 90, 90)
        .addComponent(bahasa,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(panelback1Layout.createSequentialGroup())
        .addGap(135, 135, 135)
        .addComponent(SignUp)))
        .addContainerGap(116, Short.MAX_VALUE)
    );
    panelback1Layout.setVerticalGroup(

panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
panelback1Layout.createSequentialGroup())
        .addContainerGap(69, Short.MAX_VALUE)

    .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(labelUserName,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

104,

14,

```

javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(inputUsername,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
        .addComponent(labelPassword)
        .addComponent(inputPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(bahasa,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(connect,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(SignUp)
        .addGap(32, 32, 32)
    );

    javax.swing.GroupLayout loginPanelLayout = new
javax.swing.GroupLayout(loginPanel);
    loginPanel.setLayout(loginPanelLayout);
    loginPanelLayout.setHorizontalGroup(

loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(loginPanelLayout.createSequentialGroup()
            .addComponent(panelback1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    loginPanelLayout.setVerticalGroup(

```

```

loginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(panelback1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

tabbedPane.addTab("LOGIN", loginPanel);

listUser.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        listUserMouseClicked(evt);
    }
});
jScrollPane2.setViewportView(listUser);

javax.swing.GroupLayout panelback2Layout = new
javax.swing.GroupLayout(panelback2);
panelback2.setLayout(panelback2Layout);
panelback2Layout.setHorizontalGroup(

panelback2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelback2Layout.createSequentialGroup()
        .addGap(90, 90, 90)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 124,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(122, Short.MAX_VALUE))
    );
panelback2Layout.setVerticalGroup(

panelback2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
panelback2Layout.createSequentialGroup()
        .addContainerGap(26, Short.MAX_VALUE)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(21, 21, 21))
    );

javax.swing.GroupLayout chatPanelLayout = new
javax.swing.GroupLayout(chatPanel);
chatPanel.setLayout(chatPanelLayout);

```

```

chatPanelLayout.setHorizontalGroup(

chatPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(panelback2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
chatPanelLayout.setVerticalGroup(

chatPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(panelback2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

tabbedPane.addTab("CHAT", chatPanel);

javax.swing.GroupLayout panelbackk1Layout = new
javax.swing.GroupLayout(panelbackk1);
panelbackk1.setLayout(panelbackk1Layout);
panelbackk1Layout.setHorizontalGroup(

panelbackk1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelbackk1Layout.createSequentialGroup())

.addGroup(panelbackk1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelbackk1Layout.createSequentialGroup()
        .addGap(67, 67, 67)
        .addComponent(jLabel1))
        .addGroup(panelbackk1Layout.createSequentialGroup()
            .addGap(34, 34, 34)
            .addComponent(tabbedPane,
javax.swing.GroupLayout.PREFERRED_SIZE, 341,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(36, Short.MAX_VALUE))
    );
panelbackk1Layout.setVerticalGroup(

panelbackk1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelbackk1Layout.createSequentialGroup()
        .addGap(26, 26, 26)
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(tabbedPane,
javax.swing.GroupLayout.PREFERRED_SIZE,          284,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(46, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout    jPanel2Layout    =    new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(panelbackk1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(panelbackk1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    File.setText("File");
    File.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            FileActionPerformed(evt);
        }
    });

    menuSignup.setText("Sign Up");
    menuSignup.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            menuSignupActionPerformed(evt);
        }
    });
    File.add(menuSignup);

```

```

menuSignout.setText("Sign Out");
menuSignout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        menuSignoutActionPerformed(evt);
    }
});
File.add(menuSignout);

menuExit.setText("Exit");
menuExit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        menuExitActionPerformed(evt);
    }
});
File.add(menuExit);

jMenuBar1.add(File);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 400,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

pack();
} // </editor-fold>

public void run()

```

```

{ while (thread != null)
  { try
    { // streamOut.writeUTF(console.readLine());
      // streamOut.writeUTF(hasil);
      streamOut.flush();
    }
    catch(IOException ioe)
    { // System.out.println("Sending error: " + ioe.getMessage());
      stop();
    } } }

public void handle(String msg)
{
  try {
    hikhik = msg.split(" ");
    usersend = hikhik[0];
    useraccept = hikhik[2];
    int pecah1 = msg.indexOf(" ");
    int pecah2 = msg.indexOf(" ", pecah1 + 1);
    int pecah3 = msg.indexOf(" ", pecah2 + 1);
    int pecah4 = msg.indexOf(" ", pecah3 + 1);
    String ky = msg.substring(pecah4);

    String sqllanguagesend = "select * from mes where userMes LIKE '" +
usersend + "'";
    st = conn.createStatement();
    rs = st.executeQuery(sqllanguagesend);
    if (rs.next()) {
      languagesend = rs.getString(3);
    }

    String sqllanguageaccept = "select * from mes where userMes LIKE '" +
useraccept + "'";
    st = conn.createStatement();
    rs = st.executeQuery(sqllanguageaccept);
    if (rs.next()) {
      languageaccept = rs.getString(3); }

    if (ky.equals(" saiaremove")) {
      for(int i = 0; i < pms.size(); i++){
        Message tmp = (Message)pms.get(i);
        if(tmp.chattingWith.equals(usersend)){
          (tmp.textmes).append("\n" + usersend + " sign out" + "\n");
          (tmp.kirim).setEnabled(false);
        }
      }
    }
  }
}

```

```

    }
    } else {
    if (ky.equals(" saiaecit")) {
    for(int i = 0; i < pms.size(); i++){
    Message tmp = (Message)pms.get(i);
    if(username.equals(usersend)) {
    if(tmp.chattingWith.equals(useraccept)){
        tabbedPane.removeTabAt(i+2);
        pms.removeElementAt(i); }
    // else if (tmp.chattingWith.equals(usersend)){
    //     (tmp.textmes).append("\n" + usersend + " close chat" + "\n");
    // }
    } }
    } else {
    // INI FIX LHO JANGAN DIHAPUS
    try {
        hasiltranslate = Translate.DEFAULT.execute(ky,
Language.valueOf(languagesend), Language.valueOf(languageaccept));
        } catch (GoogleAPIException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);
        }

        // if (username.equals(usersend)) {
        // } else {
        if (username.equals(useraccept)) {
            if (!(alreadyChatting(usersend))) {
                createNewPM(usersend);
            } else {
            }
        }
        // }
        for (int i = 0; i < pms.size(); i++) {
            Message tmp = (Message) pms.get(i);
            if ((username).equals(useraccept)) {
                if ((tmp.chattingWith).equals(usersend)) {
                    (tmp.textmes).append(usersend + ": " + hasiltranslate + "\n");
                } else if ((username).equals(usersend)) {
                    if ((tmp.chattingWith).equals(useraccept)) {
                        (tmp.textmes).append(usersend + ":" + ky + "\n");
                    }
                }
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }

```

```

    }
}

public void start() throws IOException
{ //console = new DataInputStream(System.in);
  // console = new DataInputStream(inputText.getText());
  console = new DataInputStream(socket.getInputStream());
  streamOut = new DataOutputStream(socket.getOutputStream());
  if (thread == null)
  { client = new ThreadC(this, socket);
    thread = new Thread(this);
    thread.start();
  } }

public void stop()
{ if (thread != null)
  { thread.stop();
    thread = null;
    client.close();
    client.stop();
  } }

public void connect(){
  tabbedPane.setEnabledAt(0, false);
  tabbedPane.setEnabledAt(1, true);
  tabbedPane.setSelectedIndex(1);
  connect.setEnabled(false);
  inputUsername.setText("");
  inputPassword.setText("");
  setTitle("Welcome " + username + " - " + pilihbahasa);
  menuSignout.setVisible(true);
  menuSignup.setVisible(false);
  menuExit.setVisible(false);
}

public void combolanguage(){
  bahasa.addItem("Language");
  bahasa.addItem("AFRIKAANS");
  bahasa.addItem("ALBANIAN");
  bahasa.addItem("AMHARIC");
  bahasa.addItem("ARABIC");
  bahasa.addItem("ARMENIAN");
  bahasa.addItem("AZERBAIJANI");
  bahasa.addItem("BASQUE");
  bahasa.addItem("BELARUSIAN");
}

```

```
bahasa.addItem("BENGALI");
bahasa.addItem("BIHARI");
bahasa.addItem("BULGARIAN");
bahasa.addItem("BURMESE");
bahasa.addItem("CATALAN");
bahasa.addItem("CHEROKEE");
bahasa.addItem("CHINESE");
bahasa.addItem("CHINESE_SIMPLIFIED");
bahasa.addItem("CHINESE_TRADITIONAL");
bahasa.addItem("CROATIAN");
bahasa.addItem("CZECH");
bahasa.addItem("DANISH");
bahasa.addItem("DHIVEHI");
bahasa.addItem("DUTCH");
bahasa.addItem("ENGLISH");
bahasa.addItem("ESPERANTO");
bahasa.addItem("ESTONIAN");
bahasa.addItem("FILIPINO");
bahasa.addItem("FINNISH");
bahasa.addItem("FRENCH");
bahasa.addItem("GALICIAN");
bahasa.addItem("GEORGIAN");
bahasa.addItem("GERMAN");
bahasa.addItem("GREEK");
bahasa.addItem("GUARANI");
bahasa.addItem("GUJARATI");
bahasa.addItem("HEBREW");
bahasa.addItem("HINDI");
bahasa.addItem("HUNGARIAN");
bahasa.addItem("ICELANDIC");
bahasa.addItem("INDONESIAN");
bahasa.addItem("INUKTITUT");
bahasa.addItem("IRISH");
bahasa.addItem("ITALIAN");
bahasa.addItem("JAPANESE");
bahasa.addItem("KANNADA");
bahasa.addItem("KAZAKH");
bahasa.addItem("KHMER");
bahasa.addItem("KOREAN");
bahasa.addItem("KURDISH");
bahasa.addItem("KYRGYZ");
bahasa.addItem("LAOTHIAN");
bahasa.addItem("LATVIAN");
bahasa.addItem("LITHUANIAN");
bahasa.addItem("MACEDONIAN");
bahasa.addItem("MALAY");
```

```

bahasa.addItem("MALAYALAM");
bahasa.addItem("MALTESE");
bahasa.addItem("MARATHI");
bahasa.addItem("MONGOLIAN");
bahasa.addItem("NEPALI");
bahasa.addItem("NORWEGIAN");
bahasa.addItem("ORIYA");
bahasa.addItem("PASHTO");
bahasa.addItem("PERSIAN");
bahasa.addItem("POLISH");
bahasa.addItem("PORTUGUESE");
bahasa.addItem("PUNJABI");
bahasa.addItem("ROMANIAN");
bahasa.addItem("RUSSIAN");
bahasa.addItem("SANSKRIT");
bahasa.addItem("SERBIAN");
bahasa.addItem("SINDHI");
bahasa.addItem("SINHALESE");
bahasa.addItem("SLOVAK");
bahasa.addItem("SLOVENIAN");
bahasa.addItem("SPANISH");
bahasa.addItem("SWAHILI");
bahasa.addItem("SWEDISH");
bahasa.addItem("TAGALOG");
bahasa.addItem("TAJIK");
bahasa.addItem("TAMIL");
bahasa.addItem("TELUGU");
bahasa.addItem("THAI");
bahasa.addItem("TIBETAN");
bahasa.addItem("TURKISH");
bahasa.addItem("UIGHUR");
bahasa.addItem("UKRANIAN");
bahasa.addItem("URDU");
bahasa.addItem("UZBEK");
bahasa.addItem("VIETNAMESE");
bahasa.addItem("WELSH");
bahasa.addItem("YIDDISH");
}

private void menuSignoutActionPerformed(java.awt.event.ActionEvent evt) {
    String metyyu = (username + " to " + chatWith + " : " + "saiaremove");
    try {
        streamOut.writeUTF(metyyu);
    } catch (IOException ex) {
        Logger.getLogger(Message.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

```

```

    }

    String query = "delete from mes where userMes LIKE '" + username + "'";
    try {
        st.executeUpdate(query);
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }

    stop();
    timer.stop();

    try {
        conn.close();
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        st.close();
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        rs.close();
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }

    this.dispose();
}

private void FileActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void menuSignupActionPerformed(java.awt.event.ActionEvent evt) {
    new Sign().setVisible(true);
}

private void menuExitActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void listUserMouseClicked(java.awt.event.MouseEvent evt) {
    if (evt.getClickCount() == 1 || evt.getClickCount() == 2 ||
    evt.getClickCount() == 3) {

```



```

        st.execute(sqlid);
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE,
null, ex);
    }
    ActionListener actionListener = new ActionListener() {
        public void actionPerformed(ActionEvent actionEvent) {
            try {
                list.clear();
                listUser.setModel(list);
                String sqluseronline = "select * from mes order by
userMes";

                st = conn.createStatement();
                rs = st.executeQuery(sqluseronline);
                while (rs.next()) {
                    list.addElement(rs.getObject(1));
                }
            } catch (SQLException ex) {

Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    };
    timer = new Timer(1000, actionListener);
    timer.start();
    listUser.setModel(list);
} else {
    JOptionPane.showMessageDialog(this, "Username or Password
invalid");
}
} else if (password.equals(password)){
    JOptionPane.showMessageDialog(this, "Username or Password
invalid");
}
}
} catch (UnknownHostException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
} catch (IOException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

private void bahasaActionPerformed(java.awt.event.ActionEvent evt) {
    if(bahasa.getSelectedItem().equals("AFRIKAANS")){

```

```
pilihbahasa = "AFRIKAANS";
} else if(bahasa.getSelectedItemAt().equals("ALBANIAN")){
    pilihbahasa = "ALBANIAN";
} else if(bahasa.getSelectedItemAt().equals("AMHARIC")){
    pilihbahasa = "AMHARIC";
} else if(bahasa.getSelectedItemAt().equals("ARABIC")){
    pilihbahasa = "ARABIC";
} else if(bahasa.getSelectedItemAt().equals("ARMENIAN")){
    pilihbahasa = "ARMENIAN";
} else if(bahasa.getSelectedItemAt().equals("AZERBAIJANI")){
    pilihbahasa = "AZERBAIJANI";
} else if(bahasa.getSelectedItemAt().equals("BASQUE")){
    pilihbahasa = "BASQUE";
} else if(bahasa.getSelectedItemAt().equals("BELARUSIAN")){
    pilihbahasa = "BELARUSIAN";
} else if(bahasa.getSelectedItemAt().equals("BENGALI")){
    pilihbahasa = "BENGALI";
} else if(bahasa.getSelectedItemAt().equals("BIHARI")){
    pilihbahasa = "BIHARI";
} else if(bahasa.getSelectedItemAt().equals("BULGARIAN")){
    pilihbahasa = "BULGARIAN";
} else if(bahasa.getSelectedItemAt().equals("BURMESE")){
    pilihbahasa = "BURMESE";
} else if(bahasa.getSelectedItemAt().equals("CATALAN")){
    pilihbahasa = "CATALAN";
} else if(bahasa.getSelectedItemAt().equals("CHEROKEE")){
    pilihbahasa = "CHEROKEE";
} else if(bahasa.getSelectedItemAt().equals("CHINESE")){
    pilihbahasa = "CHINESE";
} else if(bahasa.getSelectedItemAt().equals("CHINESE_SIMPLIFIED")){
    pilihbahasa = "CHINESE_SIMPLIFIED";
} else if(bahasa.getSelectedItemAt().equals("CHINESE_TRADITIONAL")){
    pilihbahasa = "CHINESE_TRADITIONAL";
} else if(bahasa.getSelectedItemAt().equals("CROATIAN")){
    pilihbahasa = "CROATIAN";
} else if(bahasa.getSelectedItemAt().equals("CZECH")){
    pilihbahasa = "CZECH";
} else if(bahasa.getSelectedItemAt().equals("DANISH")){
    pilihbahasa = "DANISH";
} else if(bahasa.getSelectedItemAt().equals("DHIVEHI")){
    pilihbahasa = "DHIVEHI";
} else if(bahasa.getSelectedItemAt().equals("DUTCH")){
    pilihbahasa = "DUTCH";
} else if(bahasa.getSelectedItemAt().equals("ENGLISH")){
    pilihbahasa = "ENGLISH";
} else if(bahasa.getSelectedItemAt().equals("ESPERANTO")){
```

```
pilihbahasa = "ESPERANTO";
} else if(bahasa.getSelectedItemAt().equals("ESTONIAN")){
    pilihbahasa = "ESTONIAN";
} else if(bahasa.getSelectedItemAt().equals("FILIPINO")){
    pilihbahasa = "FILIPINO";
} else if(bahasa.getSelectedItemAt().equals("FINNISH")){
    pilihbahasa = "FINNISH";
} else if(bahasa.getSelectedItemAt().equals("FRENCH")){
    pilihbahasa = "FRENCH";
} else if(bahasa.getSelectedItemAt().equals("GALICIAN")){
    pilihbahasa = "GALICIAN";
} else if(bahasa.getSelectedItemAt().equals("GEORGIAN")){
    pilihbahasa = "GEORGIAN";
} else if(bahasa.getSelectedItemAt().equals("GERMAN")){
    pilihbahasa = "GERMAN";
} else if(bahasa.getSelectedItemAt().equals("GREEK")){
    pilihbahasa = "GREEK";
} else if(bahasa.getSelectedItemAt().equals("GUARANI")){
    pilihbahasa = "GUARANI";
} else if(bahasa.getSelectedItemAt().equals("GUJARATI")){
    pilihbahasa = "GUJARATI";
} else if(bahasa.getSelectedItemAt().equals("HEBREW")){
    pilihbahasa = "HEBREW";
} else if(bahasa.getSelectedItemAt().equals("HINDI")){
    pilihbahasa = "HINDI";
} else if(bahasa.getSelectedItemAt().equals("HUNGARIAN")){
    pilihbahasa = "HUNGARIAN";
} else if(bahasa.getSelectedItemAt().equals("ICELANDIC")){
    pilihbahasa = "ICELANDIC";
} else if(bahasa.getSelectedItemAt().equals("INDONESIAN")){
    pilihbahasa = "INDONESIAN";
} else if(bahasa.getSelectedItemAt().equals("INUKTITUT")){
    pilihbahasa = "INUKTITUT";
} else if(bahasa.getSelectedItemAt().equals("IRISH")){
    pilihbahasa = "IRISH";
} else if(bahasa.getSelectedItemAt().equals("ITALIAN")){
    pilihbahasa = "ITALIAN";
} else if(bahasa.getSelectedItemAt().equals("JAPANESE")){
    pilihbahasa = "JAPANESE";
} else if(bahasa.getSelectedItemAt().equals("KANNADA")){
    pilihbahasa = "KANNADA";
} else if(bahasa.getSelectedItemAt().equals("KAZAKH")){
    pilihbahasa = "KAZAKH";
} else if(bahasa.getSelectedItemAt().equals("KHMER")){
    pilihbahasa = "KHMER";
} else if(bahasa.getSelectedItemAt().equals("KOREAN")){
```

```

    pilihbahasa = "KOREAN";
} else if(bahasa.getSelectedItemAt().equals("KURDISH")){
    pilihbahasa = "KURDISH";
} else if(bahasa.getSelectedItemAt().equals("KYRGYZ")){
    pilihbahasa = "KYRGYZ";
} else if(bahasa.getSelectedItemAt().equals("LAOTHIAN")){
    pilihbahasa = "LAOTHIAN";
} else if(bahasa.getSelectedItemAt().equals("LATVIAN")){
    pilihbahasa = "LATVIAN";
} else if(bahasa.getSelectedItemAt().equals("LITHUANIAN")){
    pilihbahasa = "LITHUANIAN";
} else if(bahasa.getSelectedItemAt().equals("MACEDONIAN")){
    pilihbahasa = "MACEDONIAN";
} else if(bahasa.getSelectedItemAt().equals("MALAY")){
    pilihbahasa = "MALAY";
} else if(bahasa.getSelectedItemAt().equals("MALAYALAM")){
    pilihbahasa = "MALAYALAM";
} else if(bahasa.getSelectedItemAt().equals("MALTESE")){
    pilihbahasa = "MALTESE";
} else if(bahasa.getSelectedItemAt().equals("MARATHI")){
    pilihbahasa = "MARATHI";
} else if(bahasa.getSelectedItemAt().equals("MONGOLIAN")){
    pilihbahasa = "MONGOLIAN";
} else if(bahasa.getSelectedItemAt().equals("NEPALI")){
    pilihbahasa = "NEPALI";
} else if(bahasa.getSelectedItemAt().equals("NORWEGIAN")){
    pilihbahasa = "NORWEGIAN";
} else if(bahasa.getSelectedItemAt().equals("ORIYA")){
    pilihbahasa = "ORIYA";
} else if(bahasa.getSelectedItemAt().equals("PASHTO")){
    pilihbahasa = "PASHTO";
} else if(bahasa.getSelectedItemAt().equals("PERSIAN")){
    pilihbahasa = "PERSIAN";
} else if(bahasa.getSelectedItemAt().equals("POLISH")){
    pilihbahasa = "POLISH";
} else if(bahasa.getSelectedItemAt().equals("PORTUGUESE")){
    pilihbahasa = "PORTUGUESE";
} else if(bahasa.getSelectedItemAt().equals("PUNJABI")){
    pilihbahasa = "PUNJABI";
} else if(bahasa.getSelectedItemAt().equals("ROMANIAN")){
    pilihbahasa = "ROMANIAN";
} else if(bahasa.getSelectedItemAt().equals("RUSSIAN")){
    pilihbahasa = "RUSSIAN";
} else if(bahasa.getSelectedItemAt().equals("SANSKRIT")){
    pilihbahasa = "SANSKRIT";
} else if(bahasa.getSelectedItemAt().equals("SERBIAN")){

```

```

        pilihbahasa = "SERBIAN";
    } else if(bahasa.getSelectedItemAt().equals("SINDHI")){
        pilihbahasa = "SINDHI";
    } else if(bahasa.getSelectedItemAt().equals("SINHALESE")){
        pilihbahasa = "SINHALESE";
    } else if(bahasa.getSelectedItemAt().equals("SLOVAK")){
        pilihbahasa = "SLOVAK";
    } else if(bahasa.getSelectedItemAt().equals("SLOVENIAN")){
        pilihbahasa = "SLOVENIAN";
    } else if(bahasa.getSelectedItemAt().equals("SPANISH")){
        pilihbahasa = "SPANISH";
    } else if(bahasa.getSelectedItemAt().equals("SWAHILI")){
        pilihbahasa = "SWAHILI";
    } else if(bahasa.getSelectedItemAt().equals("SWEDISH")){
        pilihbahasa = "SWEDISH";
    } else if(bahasa.getSelectedItemAt().equals("TAGALOG")){
        pilihbahasa = "TAGALOG";
    } else if(bahasa.getSelectedItemAt().equals("TAJIK")){
        pilihbahasa = "TAJIK";
    } else if(bahasa.getSelectedItemAt().equals("TAMIL")){
        pilihbahasa = "TAMIL";
    } else if(bahasa.getSelectedItemAt().equals("TELUGU")){
        pilihbahasa = "TELUGU";
    } else if(bahasa.getSelectedItemAt().equals("THAI")){
        pilihbahasa = "THAI";
    } else if(bahasa.getSelectedItemAt().equals("TIBETAN")){
        pilihbahasa = "TIBETAN";
    } else if(bahasa.getSelectedItemAt().equals("TURKISH")){
        pilihbahasa = "TURKISH";
    } else if(bahasa.getSelectedItemAt().equals("UIGHUR")){
        pilihbahasa = "UIGHUR";
    } else if(bahasa.getSelectedItemAt().equals("UKRANIAN")){
        pilihbahasa = "UKRANIAN";
    } else if(bahasa.getSelectedItemAt().equals("URDU")){
        pilihbahasa = "URDU";
    } else if(bahasa.getSelectedItemAt().equals("UZBEK")){
        pilihbahasa = "UZBEK";
    } else if(bahasa.getSelectedItemAt().equals("VIETNAMESE")){
        pilihbahasa = "VIETNAMESE";
    } else if(bahasa.getSelectedItemAt().equals("WELSH")){
        pilihbahasa = "WELSH";
    } else if(bahasa.getSelectedItemAt().equals("YIDDISH")){
        pilihbahasa = "YIDDISH";
    }
}
}

```

```

private void inputUsernameKeyReleased(java.awt.event.KeyEvent evt) {
    username = new String(inputUsername.getText());
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    if (tabbedPane.getTitleAt(0).equals("LOGIN")){
        this.dispose();
    } else {
        String metyyu = (username + " to " + chatWith + " : " + "saiaremove");
        try {
            streamOut.writeUTF(metyyu);
        } catch (IOException ex) {
            Logger.getLogger(Message.class.getName()).log(Level.SEVERE, null,
ex);
        }

        String query = "delete from mes where userMes LIKE '" + username + "'";
        try {
            st.executeUpdate(query);
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }

        stop();
        timer.stop();

        try {
            conn.close();
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }
        try {
            st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }
        try {
            rs.close();
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }

        this.dispose(); }
}

public void privateMessageManager(String chatWith) {

```

```

        if(chatWith.equalsIgnoreCase(username))
            return;
        if(!(alreadyChatting(chatWith)))
            createNewPM(chatWith);
    }

    public boolean alreadyChatting(String chatWith) {
        for(int i=0;i<pms.size();i++)
        {
            Message tmp = (Message)pms.get(i);
            if((tmp.chattingWith).equals(chatWith))
            {
                tabbedPane.setSelectedIndex(i+2);
                return true;
            }
        }
        return false;
    }

    public void createNewPM(String chatWith) {
        panel2 = new Message(clientApplication, chatWith, username, socket,
streamOut);
        tabbedPane.addTab(chatWith, panel2);
        tabbedPane.setEnabledAt(tabbedPane.getTabCount()-1, true);
        tabbedPane.setSelectedIndex(tabbedPane.getTabCount()-1);
        pms.addElement(panel2);
    }

    public void koneksi(String server, String db, String username, String
password){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            // conn = null;
            conn = DriverManager.getConnection("jdbc:mysql://" + server + ":3306/"
+ db, username, password);
            if (conn == null) {
                JOptionPane.showMessageDialog(this, "gagal");
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "tidak bisa koneksi ke database");
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JMenu File;
private javax.swing.JLabel SignUp;
private javax.swing.JComboBox bahasa;
private javax.swing.JPanel chatPanel;
public javax.swing.JButton connect;
private javax.swing.JPasswordField inputPassword;
public javax.swing.JTextField inputUsername;
private javax.swing.JLabel jLabel1;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JLabel labelPassword;
private javax.swing.JLabel labelUserName;
private javax.swing.JList listUser;
private javax.swing.JPanel loginPanel;
private javax.swing.JMenuItem menuItemExit;
private javax.swing.JMenuItem menuItemSignout;
private javax.swing.JMenuItem menuItemSignup;
private chat.panelback panelback1;
private chat.panelback panelback2;
private chat.panelbackk panelbackk1;
private javax.swing.JTabbedPane tabbedPane;
// End of variables declaration
}
```

Message

```

import java.sql.*;
import java.io.*;
import java.net.*;
import java.util.logging.*;
import javax.swing.text.DefaultCaret;

public class Message extends javax.swing.JPanel {
    String chattingWith=null;
    private Login parent;
    private Socket socket = null;
    private DataOutputStream streamOut = null;
    String text, sql, idcocok, tambahtext;
    String username = null;

    public Connection conn;
    public Statement st;
    public ResultSet rs;

    public Message(Login parent, String chattingWith, String username, Socket
socket,DataOutputStream streamOut) {
        initComponents();
        this.parent = parent;
        this.username = username;
        this.chattingWith = chattingWith;
        this.socket = socket;
        this.streamOut = streamOut;
        textmes.append("Connected: " + socket + "\n");
        textmes.setEditable(false);
        DefaultCaret caret = (DefaultCaret)textmes.getCaret();
        caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        textmes = new javax.swing.JTextArea();
        textInput = new javax.swing.JTextField();
        panelbackmes1 = new chat.panelbackmes();
        panelbackmes2 = new chat.panelbackmes();
        panelbackmes4 = new chat.panelbackmes();
        panelbackmes5 = new chat.panelbackmes();
        panelbackmes6 = new chat.panelbackmes();
        exit = new javax.swing.JButton();
    }
}

```

```

panelbackmes7 = new chat.panelbackmes();
panelbackmes8 = new chat.panelbackmes();
panelbackmes9 = new chat.panelbackmes();
panelbackmes10 = new chat.panelbackmes();
panelbackmes11 = new chat.panelbackmes();
kirim = new javax.swing.JButton();

textmes.setColumns(20);
textmes.setRows(5);
jScrollPane1.setViewportView(textmes);

textInput.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        textInputActionPerformed(evt);
    }
});

javax.swing.GroupLayout panelbackmes1Layout = new
javax.swing.GroupLayout(panelbackmes1);
panelbackmes1.setLayout(panelbackmes1Layout);
panelbackmes1Layout.setHorizontalGroup(

panelbackmes1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGroup(0, 48, Short.MAX_VALUE)
);
panelbackmes1Layout.setVerticalGroup(

panelbackmes1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGroup(0, 42, Short.MAX_VALUE)
);

javax.swing.GroupLayout panelbackmes2Layout = new
javax.swing.GroupLayout(panelbackmes2);
panelbackmes2.setLayout(panelbackmes2Layout);
panelbackmes2Layout.setHorizontalGroup(

panelbackmes2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGroup(0, 48, Short.MAX_VALUE)
);
panelbackmes2Layout.setVerticalGroup(

panelbackmes2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

```

```

        .addGap(0, 42, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout panelbackmes4Layout = new
javax.swing.GroupLayout(panelbackmes4);
    panelbackmes4.setLayout(panelbackmes4Layout);
    panelbackmes4Layout.setHorizontalGroup(

panelbackmes4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGap(0, 48, Short.MAX_VALUE)
    );
    panelbackmes4Layout.setVerticalGroup(

panelbackmes4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGap(0, 42, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout panelbackmes5Layout = new
javax.swing.GroupLayout(panelbackmes5);
    panelbackmes5.setLayout(panelbackmes5Layout);
    panelbackmes5Layout.setHorizontalGroup(

panelbackmes5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGap(0, 48, Short.MAX_VALUE)
    );
    panelbackmes5Layout.setVerticalGroup(

panelbackmes5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGap(0, 42, Short.MAX_VALUE)
    );

    exit.setText("Close tab");
    exit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            exitActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout panelbackmes6Layout = new
javax.swing.GroupLayout(panelbackmes6);
    panelbackmes6.setLayout(panelbackmes6Layout);
    panelbackmes6Layout.setHorizontalGroup(

```

```

panelbackmes6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGroup(panelbackmes6Layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(exit)
    .addContainerGap(9, Short.MAX_VALUE))
);
panelbackmes6Layout.setVerticalGroup(

```

```

panelbackmes6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGroup(panelbackmes6Layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(exit)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

```

```

    javax.swing.GroupLayout    panelbackmes7Layout    =    new
javax.swing.GroupLayout(panelbackmes7);
panelbackmes7.setLayout(panelbackmes7Layout);
panelbackmes7Layout.setHorizontalGroup(

```

```

panelbackmes7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 48, Short.MAX_VALUE)
);
panelbackmes7Layout.setVerticalGroup(

```

```

panelbackmes7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 42, Short.MAX_VALUE)
);

```

```

    javax.swing.GroupLayout    panelbackmes8Layout    =    new
javax.swing.GroupLayout(panelbackmes8);
panelbackmes8.setLayout(panelbackmes8Layout);
panelbackmes8Layout.setHorizontalGroup(

```

```

panelbackmes8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 48, Short.MAX_VALUE)
);
panelbackmes8Layout.setVerticalGroup(

```

```

panelbackmes8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 42, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout    panelbackmes9Layout    =    new
javax.swing.GroupLayout(panelbackmes9);
    panelbackmes9.setLayout(panelbackmes9Layout);
    panelbackmes9Layout.setHorizontalGroup(

panelbackmes9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 48, Short.MAX_VALUE)
    );
    panelbackmes9Layout.setVerticalGroup(

panelbackmes9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
    .addGap(0, 42, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout    panelbackmes10Layout    =    new
javax.swing.GroupLayout(panelbackmes10);
    panelbackmes10.setLayout(panelbackmes10Layout);
    panelbackmes10Layout.setHorizontalGroup(

panelbackmes10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGap(0, 48, Short.MAX_VALUE)
    );
    panelbackmes10Layout.setVerticalGroup(

panelbackmes10Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGap(0, 42, Short.MAX_VALUE)
    );

    kirim.setText("Send");
    kirim.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            kirimActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout    panelbackmes11Layout    =    new
javax.swing.GroupLayout(panelbackmes11);

```

```

panelbackmes11.setLayout(panelbackmes11Layout);
panelbackmes11Layout.setHorizontalGroup(

panelbackmes11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(panelbackmes11Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(kirim, javax.swing.GroupLayout.DEFAULT_SIZE, 76,
Short.MAX_VALUE)
        .addContainerGap())
    );
panelbackmes11Layout.setVerticalGroup(

panelbackmes11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
panelbackmes11Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(kirim)
        .addContainerGap())
    );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
        .addGroup(layout.createSequentialGroup()
            .addComponent(panelbackmes7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(panelbackmes8,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(panelbackmes9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes11,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(textInput)
        .addGroup(layout.createSequentialGroup())
        .addComponent(panelbackmes4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(panelbackmes6, 0, 96, Short.MAX_VALUE))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 312, Short.MAX_VALUE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

```

```

);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
    .addComponent(panelbackmes2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(panelbackmes5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
ILING, false)
    .addComponent(panelbackmes4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(panelbackmes1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addComponent(panelbackmes6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
96,

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(textInput,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
34,

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
ILING)

```

```

        .addComponent(panelbackmes7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(panelbackmes8,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(panelbackmes9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(panelbackmes10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(panelbackmes11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
    );
} // </editor-fold>

private void kirimActionPerformed(java.awt.event.ActionEvent evt) {
    text = textInput.getText();
    tambahtext = (username + " to " + chattingWith + " : " + text);
    try {
        streamOut.writeUTF(tambahtext);
    } catch (IOException ex) {
        Logger.getLogger(Message.class.getName()).log(Level.SEVERE, null,
ex);
    }
    textInput.setText("");
    DefaultCaret caret = (DefaultCaret)textmes.getCaret();
    caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);
}

private void exitActionPerformed(java.awt.event.ActionEvent evt) {
    String metu = (username + " to " + chattingWith + " : " + "saiacit");
    try {
        streamOut.writeUTF(metu);
    } catch (IOException ex) {
        Logger.getLogger(Message.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

```

```

}

private void textInputActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultCaret caret = (DefaultCaret)textmes.getCaret();
    caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);
    textInput.setFocusable(true);
    text = textInput.getText();
    tambahtext = (username + " to " + chattingWith + " : " + text);
    try {
        streamOut.writeUTF(tambahtext);
    } catch (IOException ex) {
        Logger.getLogger(Message.class.getName()).log(Level.SEVERE, null,
ex);
    }
    textInput.setText("");
}

// Variables declaration - do not modify
public javax.swing.JButton exit;
private javax.swing.JScrollPane jScrollPane1;
public javax.swing.JButton kirim;
private chat.panelbackmes panelbackmes1;
private chat.panelbackmes panelbackmes10;
private chat.panelbackmes panelbackmes11;
private chat.panelbackmes panelbackmes2;
private chat.panelbackmes panelbackmes4;
private chat.panelbackmes panelbackmes5;
private chat.panelbackmes panelbackmes6;
private chat.panelbackmes panelbackmes7;
private chat.panelbackmes panelbackmes8;
private chat.panelbackmes panelbackmes9;
private javax.swing.JTextField textInput;
public javax.swing.JTextArea textmes;
// End of variables declaration
}

```

Sign

```

import java.sql.*;
import java.util.logging.*;
import javax.swing.*;

public class Sign extends javax.swing.JFrame {
    public Connection conn;
    public Statement st;
    String firstname, lastname, signuser, signpass, signemail, gender;

    public Sign() {
        initComponents();
        setTitle(".:TM:");
        setIconImage(new javax.swing.ImageIcon(getClass().
getResource("sun.jpg")).getImage());
        koneksi("localhost","chatting","root","");
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        panelback1 = new chat.panelback();
        Submit = new javax.swing.JButton();
        SignEmail = new javax.swing.JTextField();
        ComboGender = new javax.swing.JComboBox();
        SignUser = new javax.swing.JTextField();
        SignPass = new javax.swing.JPasswordField();
        SignFirstname = new javax.swing.JTextField();
        SignLastname = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenuItem1 = new javax.swing.JMenuItem();
        menuItem = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent evt) {
                formWindowClosing(evt);
            }
        });
    }
}

```



```

.addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jLabel2)

.addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel4)
    .addComponent(jLabel3)
    .addComponent(jLabel6)
    .addComponent(jLabel5))
    .addComponent(jLabel1))
.addGap(37, 37, 37)

.addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(SignFirstname,
javax.swing.GroupLayout.PREFERRED_SIZE, 134,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(SignEmail,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)
    .addComponent(ComboGender,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(SignPass,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)
    .addComponent(SignUser,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)
    .addComponent(SignLastname,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE))))))
.addGap(92, 92, 92))
);
panelback1Layout.setVerticalGroup(

panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelback1Layout.createSequentialGroup()
        .addGap(51, 51, 51)

.addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)

```

```

        .addComponent(SignFirstname,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(15, 15, 15)

    .addGroup(panelback1Layout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)
        .addComponent(SignLastname,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))

    .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(panelback1Layout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.LEADING)

    .addGroup(panelback1Layout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)
        .addComponent(SignUser,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))
        .addGroup(panelback1Layout.createSequentialGroup())
        .addGap(32, 32, 32)

    .addGroup(panelback1Layout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)
        .addComponent(SignPass,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4))

    .addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(panelback1Layout.createParallelGroup(javafx.swing.GroupLayout.Align
ment.BASELINE)
        .addComponent(SignEmail,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel5))

```

```

        .addGap(14, 14, 14)

        .addGroup(panelback1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(ComboGender,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel6)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 36,
            Short.MAX_VALUE)
            .addComponent(Submit)
            .addGap(25, 25, 25)
        );

        jMenu1.setText("File");

        menuExit.setText("Exit");
        menuExit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                menuExitActionPerformed(evt);
            }
        });
        jMenu1.add(menuExit);

        jMenuBar1.add(jMenu1);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(panelback1,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
        );
        layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(panelback1, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    
```

```

);

    pack();
} // </editor-fold>
public void koneksi(String server, String db, String username, String password){
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://" + server + ":3306/"
+ db, username, password);
        if (conn == null) {
            JOptionPane.showMessageDialog(this, "gagal");
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "tidak bisa koneksi ke database");
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void ComboGenderActionPerformed(java.awt.event.ActionEvent evt) {
if(ComboGender.getSelectedItem().equals("L")){
    gender = "L";
}
else if(ComboGender.getSelectedItem().equals("P")){
    gender = "P";
}
}

private void SubmitActionPerformed(java.awt.event.ActionEvent evt) {
    if (SignFirstname.getText().equals("") || SignLastname.getText().equals("") ||
        SignUser.getText().equals("") || SignPass.getText().equals("") ||
        SignEmail.getText().equals("") ||
ComboGender.getSelectedItem().equals("Choose")){
        JOptionPane.showMessageDialog(this, "You must fill out of the blank");
    } else {
        try {
            firstname = SignFirstname.getText();
            lastname = SignLastname.getText();
            signuser = SignUser.getText();
            signpass = SignPass.getText();
            signemail = SignEmail.getText();

            String sql = "insert into identitas(username,firstname,lastname,password,
email,gender)" +
                " values ('" + signuser + "', '" + firstname + "'" + "','" + lastname + "','"

```

```

+
        "" + signpass + "," + signemail + "," + gender + "");
    st = conn.createStatement();
    st.execute(sql);
} catch (SQLException ex) {
    Logger.getLogger(Sign.class.getName()).log(Level.SEVERE, null, ex);
}
    SignFirstname.setText("");
    SignLastname.setText("");
    SignUser.setText("");
    SignPass.setText("");
    SignEmail.setText("");
    JOptionPane.showMessageDialog(this, "Your data is saved"); }
}

private void menuExitActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        this.dispose();
        conn.close();
        st.close();
    } catch (SQLException ex) {
        Logger.getLogger(Sign.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    this.dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Sign().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JComboBox ComboGender;
private javax.swing.JTextField SignEmail;
private javax.swing.JTextField SignFirstname;
private javax.swing.JTextField SignLastname;
private javax.swing.JPasswordField SignPass;
private javax.swing.JTextField SignUser;
private javax.swing.JButton Submit;
private javax.swing.JLabel jLabel1;

```

```
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JMenu jMenu1;  
private javax.swing.JMenuBar jMenuBar1;  
private javax.swing.JMenuItem menuItemExit;  
private chat.panelback panelback1;  
// End of variables declaration
```

```
}
```

ThreadC

```
import java.net.*;
import java.io.*;
public class ThreadC extends Thread
{ private Socket socket = null;
  private Login client = null;
  private DataInputStream streamIn = null;

  public ThreadC(Login parent, Socket _socket)
  { client = parent;
    socket = _socket;
    open();
    start();
  }

  public void open()
  { try
    { streamIn = new DataInputStream(socket.getInputStream());
    }
    catch(IOException ioe)
    { System.out.println("Error getting input stream: " + ioe);
      client.stop();
    }
  }

  public void close()
  { try
    { if (streamIn != null) streamIn.close();
    }
    catch(IOException ioe)
    { System.out.println("Error closing input stream: " + ioe);
    }
  }

  public void run()
  { while (true)
    { try
      { client.handle(streamIn.readUTF());
      }
      catch(IOException ioe)
      { System.out.println("Listening error: " + ioe.getMessage());
        client.stop();
      }
    }
  }
}}
```

Lampiran 3. Validasi Instrumen

SURAT KETERANGAN VALIDASI INSTRUMEN

Yang bertanda tangan dibawah ini:

Nama : Umi Rochayati, M.T.

NIP : 19630528 198710 2 001

Setelah memeriksa isi dari instrumen penelitian yang berjudul "*Aplikasi Translator Messenger Berbasis Java dan Google Translate API*", oleh peneliti:

Nama : Novita Anindita

NIM : 08520241003

Prodi : Pendidikan Teknik Informatika

Maka dengan ini menyatakan bahwa instrumen penelitian tersebut *):

- a.) Layak digunakan untuk penelitian dengan revisi sesuai saran sebagai berikut:

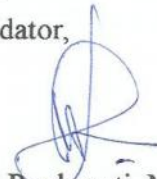
- Instrumen disesuaikan dengan kisi?.....
 - Nomor instrumen disesuaikan dengan.....
 Quality factor.....

- b. Layak digunakan untuk penelitian tanpa revisi.
 c. Tidak layak digunakan untuk penelitian.

Demikian surat keterangan ini dibuat.

Yogyakarta November 2012

Validator,



Umi Rochayati, M.T.

NIP. 19630528 198710 2 001

*) Lingkari pada huruf yang sesuai dengan pendapat Bapak/Ibu

SURAT KETERANGAN VALIDASI INSTRUMEN

Yang bertanda tangan dibawah ini:

Nama : Mohammad Munir, M.Pd.

NIP : 19630512 198901 1 001

Setelah memeriksa isi dari instrumen penelitian yang berjudul "*Aplikasi Translator Messenger Berbasis Java dan Google Translate API*", oleh peneliti:

Nama : Novita Anindita

NIM : 08520241003

Prodi : Pendidikan Teknik Informatika

Maka dengan ini menyatakan bahwa instrumen penelitian tersebut *):

- a. Layak digunakan untuk penelitian dengan revisi sesuai saran sebagai berikut:

1. Costlah bsing supaya di cekna miring -
 2. Functionality apa tidak kurang
 Kemungkinan ? -

- b. Layak digunakan untuk penelitian tanpa revisi.
 c. Tidak layak digunakan untuk penelitian.

Demikian surat keterangan ini dibuat.

Yogyakarta, 20 November 2012

Validator,

Mohammad Munir, M.Pd.

NIP. 19630512 198901 1 001

*) Lingkari pada huruf yang sesuai dengan pendapat Bapak/Ibu

SURAT KETERANGAN VALIDASI INSTRUMEN

Yang bertanda tangan dibawah ini:

Nama : Masduki Zakaria, M.T.

NIP : 19640917 198901 1 001

Setelah memeriksa isi dari instrumen penelitian yang berjudul "*Aplikasi Translator Messenger Berbasis Java dan Google Translate API*", oleh peneliti:

Nama : Novita Anindita

NIM : 08520241003

Prodi : Pendidikan Teknik Informatika

Maka dengan ini menyatakan bahwa instrumen penelitian tersebut *):

- a. Layak digunakan untuk penelitian dengan revisi sesuai saran sebagai berikut:

*Aplikasi ini bersyarat Prototipe → shg perlu penjelasan lebih lanjut tgl "Keterbatasan" ?
jika mungkin ada.*

*Saran: jika tidak "merupakan" judul di tambahkan
Prototipe jika tidak perlu ada*

- b. Layak digunakan untuk penelitian tanpa revisi.

- c. Tidak layak digunakan untuk penelitian.

penjelasan dalam laporan penelitian

Demikian surat keterangan ini dibuat.

Yogyakarta 22 November 2012

Validator,

Masduki Zakaria, M.T.

NIP. 19640917 198901 1 001

*) Lingkari pada huruf yang sesuai dengan pendapat Bapak/Ibu

Lampiran 4. Validasi Aplikasi

VALIDASI APLIKASI

“Aplikasi Translator Messenger Berbasis Java dan Google Translate API”

Nama Ahli : Drs. Totok Sukardiyono, M.T.

Pekerjaan : Dosen Pend. Teknik Informatika UNY

A. PETUNJUK

1. Mohon dengan hormat bantuan dan ketersediaan Ibu untuk menjawab seluruh pertanyaan yang ada.
2. Berilah tanda *check* () pada kolom sesuai dengan pendapat Ibu sesuai dengan keadaan yang sebenarnya.
3. Jawaban diberikan pada kolom yang sudah disediakan, dengan penilaian :
 - SS = Sangat Setuju
 - S = Setuju
 - RG = Ragu-ragu
 - TS = Tidak Setuju
 - STS = Sangat Tidak Setuju
4. Untuk menjalankan aplikasi, diperlukan:
 - a. JAVA(JRE)
 - b. MYSQL

Sebelum menjalankan aplikasi, terlebih dahulu silahkan restore database yang bernama **dbchat**.

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat list untuk menampung user yang online	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan password	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada user tertentu yang telah ditentukan	✓				
6.	Melalui <i>menu Sign up</i> , user terbantu dalam mendaftar sebagai user baru		✓			
7.	Melalui <i>menu Sign out</i> , user terbantu keluar dari aplikasi secara sempurna		✓			
8.	Melalui <i>tombol close tab pada tab chat</i> , user terbantu menutup tab chat yang sudah tidak digunakan		✓			
9.	Tampilan sistem ini sangat memudahkan		✓			
10.	Tab chat memuat nama sesuai dengan nama user yang dipilih pada list user	✓				
11.	Terdapat pesan user lawan yang telah sign out jika tab chat masih dalam kondisi aktif	✓				
12.	Translator Messenger dapat dihubungkan dengan Google Translate API sebagai penerjemah bahasa	✓				

13.	Cara penggunaan sistem ini sangat simpel		✓			
14.	Sistem ini sangat mudah dipelajari	✓				
15.	Translator Messenger memudahkan komunikasi bahasa yang berbeda	✓				
16.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

B. SARAN

Maka dengan ini, aplikasi tersebut dinyatakan *):

- a. Layak digunakan untuk penelitian dengan revisi sesuai saran sebagai

berikut:

- Buat database tercentral di server
- menu login belum langsung muncul sebaiknya seperti login tema menu
- scroll bar tidak langsung up otomatis, masih hrs manual menu sebaiknya dibuat otomatis

- b. Layak digunakan untuk penelitian tanpa revisi.

- c. Tidak layak digunakan untuk penelitian.

Yogyakarta, 18-12-2012

Validator,



Drs. Totok Sukardiyono, M.T.

NIP. 19670930 199303 1 005

*) Lingkari pada huruf yang sesuai dengan pendapat Bapak

VALIDASI APLIKASI

“Aplikasi Translator Messenger Berbasis Java dan Google Translate API”

Nama Ahli : Adi Dewanto, M.Kom.
Pekerjaan : Dosen Pend. Teknik Informatika UNY

B. PETUNJUK

1. Mohon dengan hormat bantuan dan ketersediaan Bapak untuk menjawab seluruh pertanyaan yang ada.
2. Berilah tanda *check* () pada kolom sesuai dengan pendapat Bapak sesuai dengan keadaan yang sebenarnya.
3. Jawaban diberikan pada kolom yang sudah disediakan, dengan penilaian :
 - SS = Sangat Setuju
 - S = Setuju
 - RG = Ragu-ragu
 - TS = Tidak Setuju
 - STS = Sangat Tidak Setuju
4. Untuk menjalankan aplikasi, diperlukan:
 - c. JAVA(JRE)
 - d. MYSQLSebelum menjalankan aplikasi, terlebih dahulu silahkan restore database yang bernama **dbchat**.

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat list untuk menampung user yang online		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan password		✓			
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada user tertentu yang telah ditentukan		✓			
6.	Melalui <i>menu Sign up</i> , user terbantu dalam mendaftar sebagai user baru		✓			
7.	Melalui <i>menu Sign out</i> , user terbantu keluar dari aplikasi secara sempurna		✓			
8.	Melalui <i>tombol close tab pada tab chat</i> , user terbantu menutup tab chat yang sudah tidak digunakan		✓			
9.	Tampilan sistem ini sangat memudahkan		✓			
10.	Tab chat memuat nama sesuai dengan nama user yang dipilih pada list user	✓				
11.	Terdapat pesan user lawan yang telah sign out jika tab chat masih dalam kondisi aktif		✓			
12.	Translator Messenger dapat dihubungkan dengan Google Translate API sebagai penerjemah bahasa		✓			

13.	Cara penggunaan sistem ini sangat simpel		✓		
14.	Sistem ini sangat mudah dipelajari		✓		
15.	Translator Messenger memudahkan komunikasi bahasa yang berbeda		✓		
16.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓		

B. SARAN

Maka dengan ini, aplikasi tersebut dinyatakan *):

- a. Layak digunakan untuk penelitian dengan revisi sesuai saran sebagai

berikut:

1) Seorang user dpt melihat profil user lain.

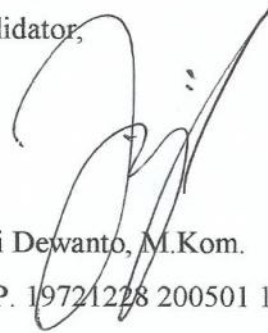
2) Pengaturan Font pesan.

- b. Layak digunakan untuk penelitian tanpa revisi.

- c. Tidak layak digunakan untuk penelitian.

Yogyakarta, Desember 2012

Validator,



Adi Dewanto, M.Kom.

NIP. 19721228 200501 1 001

*) Lingkari pada huruf yang sesuai dengan pendapat Bapak

Lampiran 5. Hasil User

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>		✓			
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		✓			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	✓				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda	✓				
7.	Tampilan sistem ini sangat memudahkan	✓				
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>	✓				
10.	Sistem ini sangat mudah dipelajari	✓				
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>		✓			
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda	✓				
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	✓				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	✓				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	✓				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda	✓				
7.	Tampilan sistem ini sangat memudahkan	✓				
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>	✓				
10.	Sistem ini sangat mudah dipelajari	✓				
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		✓			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		✓			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>		✓			
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		✓			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	✓				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		✓			
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	✓				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini			✓		

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		✓			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>		✓			
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		✓			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		✓			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>			✓		
10.	Sistem ini sangat mudah dipelajari		✓			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	✓				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	✓				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	✓				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	✓				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	✓				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		✓			
7.	Tampilan sistem ini sangat memudahkan		✓			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		✓			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>	✓				
10.	Sistem ini sangat mudah dipelajari	✓				
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		✓			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		✓			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	√				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	√				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	√				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	√				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		√			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda		√			
7.	Tampilan sistem ini sangat memudahkan		√			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>	√				
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		√			
10.	Sistem ini sangat mudah dipelajari		√			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda			√		
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		√			

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>	v				
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user	v				
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	v				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih	v				
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan	v				
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda	v				
7.	Tampilan sistem ini sangat memudahkan	v				
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		v			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>		v			
10.	Sistem ini sangat mudah dipelajari		v			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda	v				
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini	v				

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1.	Terdapat <i>list</i> untuk menampung <i>user</i> yang <i>online</i>		√			
2.	Terdapat pilihan bahasa untuk memilih bahasa yang dikuasai user		√			
3.	Terdapat tempat input pada tab login untuk memasukkan username dan <i>password</i>	√				
4.	Pesan yang diterjemahkan sesuai dengan bahasa yang telah dipilih		√			
5.	Pesan yang dikirim hanya kepada <i>user</i> tertentu yang telah ditentukan		√			
6.	Menu-menu dalam aplikasi messenger memudahkan bagi anda			√		
7.	Tampilan sistem ini sangat memudahkan		√			
8.	<i>Tab chat</i> memuat nama sesuai dengan nama <i>user</i> yang dipilih pada <i>list user</i>		√			
9.	Cara penggunaan sistem ini sangat <i>simpel</i>	√				
10.	Sistem ini sangat mudah dipelajari		√			
11.	<i>Translator Messenger</i> memudahkan komunikasi bahasa yang berbeda		√			
12.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini		√			