

# APLIKASI TRANSLATOR MESSENGER BERBASIS JAVA DAN GOOGLE TRANSLATE API

**Novita Anindita (08520241003)**

Program Studi Pendidikan Teknik Informatika, Universitas Negeri Yogyakarta  
novita.anin@gmail.com

## ABSTRAK

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. Adapun teknik analisis yang digunakan adalah teknik analisis deskriptif kuantitatif.

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.

## I. PENDAHULUAN

Teknologi informasi telah mengubah cara berkomunikasi zaman dahulu secara konvensional menjadi komunikasi digital pada masa sekarang. Di dunia komunikasi, perkembangan teknologi yang semakin maju dapat dilihat dengan semakin berkembangnya berbagai fitur-fitur untuk lebih memudahkan pengguna dalam berkomunikasi. Salah satu aplikasi komunikasi yang banyak digunakan adalah aplikasi *messenger*.

Aplikasi-aplikasi *messenger* sudah menjadi tren dalam masyarakat, baik kalangan remaja, mahasiswa maupun pebisnis. Di era modern, para pebisnis tidak lagi harus bertatap muka untuk menjalin negosiasi ataupun kontrak kerja. Terkadang penggunaan aplikasi messenger dirasa sudah cukup mewakili untuk menjalin komunikasi secara mudah dan efisien. Terkadang tidak sedikit, para pebisnis melakukan negosiasi 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. 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.

## II. KAJIAN PUSTAKA

### A. Messenger

*Messenger* adalah suatu *feature*/program dalam Internet untuk berkomunikasi langsung sesama pemakai Internet yang sedang. Seseorang mengirim pesan dengan teks kepada orang

lain yang sedang *online*, kemudian orang yang dituju membalas pesan dengan teks, demikian seterusnya.

#### **B. Java**

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Kelebihan-kelebihan java menurut Wahana (2010) antara lain: *multiplatform*, OOP (Object Oriented Programming), *library* Lengkap, pengumpulan *spam* otomatis.

#### **C. 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.

#### **D. 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 terbagi menjadi beberapa jenis diagram yang masing-masing memiliki aturan-aturan tertentu dalam penyusunannya.

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

Pada penelitian ini, peneliti menggunakan empat faktor kualitas *software* menurut McCall dan ISO 9126 model. Keempat faktor tersebut adalah *correctness*, *reliability*, *usability* (McCall) dan *functionality* (ISO 9126 model).

### **III. METODE PENELITIAN**

Metode yang digunakan dalam penelitian ini menggunakan metode Penelitian dan Pengembangan (*Research and Development*). Menurut Sujadi (2002), 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). Langkah-langkah metode penelitian tersebut adalah analisis (*Analysis*), desain (*design*), pengkodean (*Code*), pengujian (*Testing*)

### **IV. PEMBAHASAN**

#### **A. Analisis**

Analisis yang dilakukan peneliti meliputi analisis kebutuhan dan analisis desain. 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*. Sedangkan untuk menerjemahkan diperlukan *translator* berupa Google Translate API. Analisis desain dilakukan untuk menyesuaikan kebutuhan agar perangkat lunak Aplikasi Translator Messenger yang dibangun sesuai dengan harapan.

## B. Desain

Desain sistem dalam penelitian ini meliputi:

### 1. Perancangan *Database*

Pada pembuatan Aplikasi Translator Messenger, terlebih dahulu peneliti merancang *database* yang akan digunakan untuk menjalankan sistem aplikasi tersebut. *Database* Aplikasi Translator Messenger meliputi tabel identitas dan tabel mes

### 2. Pemodelan Visual

Pada penelitian ini, pemodelan visual yang digunakan adalah notasi UML (*Unified Modeling Language*). Adapun beberapa UML yang digunakan oleh peneliti adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*

### 3. 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.

## C. 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.

## D. Pengujian

### 1. Pengujian Black-box

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

### 2. Pengujian Correctness

Menurut Pressman (2010), pengujian correctness dapat diukur dengan analisis defect per KLOC (error pada setiap KLOC/Kilo Line of Code). Pada kasus ini, peneliti menggunakan LOC counter. Hasil dari perhitungan LOC dari perangkat lunak yang dibuat sebanyak 1770. Jadi  $LOC = 1,77$  KLOC.

Selanjutnya, untuk menghitung cacat, peneliti menggunakan FindBugs. Pada aplikasi ini, tidak ada cacat pada perangkat lunak yang dibuat atau dapat dikatakan cacat (0). Selanjutnya, penghitungan dilakukan sesuai dengan teori Pressman (2010), yaitu  $cacat/KLOC = 0/1,77 = 0$

Sebagai perbandingan standar jumlah error yang seharusnya, peneliti menggunakan rumus Watson – Felix Model seperti yang dijelaskan oleh Pressman (2010).

$$E = 5.2 \times (KLOC)^{0.91} = 5.2 \times (1,77)^{0.91} = 8.736$$

Dari hasil perhitungan 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. Di bawah ini merupakan hasil pengujian *reliability* perangkat lunak:

$$MTBF = T/R = 5/1 = 5 \text{ hour/failure}$$

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*

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

$$DP = (108/120) \times 100\% = 90\%$$

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

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

$$DP = (34/40) \times 100 \% = 85\%$$

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

#### 5. Pengujian beta

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

### 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 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.
3. Hasil pengujian kelayakan menyatakan hasil pengujian *correctness* menyatakan bahwa Aplikasi Translator Messenger telah memenuhi standar indikator *correctness*. Hasil pengujian *reliability* menyatakan bahwa setiap 5 jam pengoperasian perangkat lunak terjadi 1 kesalahan program. Hasil penilaian pengujian alpha yang dilakukan oleh ahli, menghasilkan dari segi *functionality* sebesar 90% dan dapat dikategorikan sangat layak serta dari segi *usability* 85% dan dapat dikategorikan sangat layak. 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.

### DAFTAR PUSTAKA

Pressman, R.(2010). Software Engineering: A Practitioner's Approach Seventh Edition. New York: McGraw Hill.

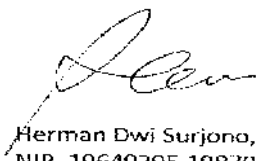
Pressman, Roger S.(2001).Software Engineering A Practitioner's Approach Fifth Edition. McGraw-Hill Companies, Inc, New York.

QuickSilver Controls.(2005).*Technical Document:QCI-TD009*.Inc.

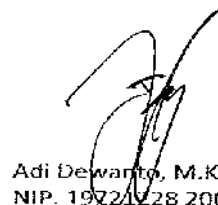
Speaks Scoot.(2004).Reliability and MTBF Overview VICOR Reliability Engineering.

Sujadi.(2002).Metodologi Penelitian Pendidikan. Jakarta: Rineka Cipta.

Wahana.(2010).The 40 Best Java Applications. Jakarta: Elex Media Komputindo.



Herman Dwi Surjono, Ph.D.  
NIP. 19640205 198701 1 001



Adi Dewanto, M.Kom.  
NIP. 19721128 200501 1 001