

## PENERAPAN OPTICAL CHARACTER RECOGNITION PADA PENTERJEMAH INGGRIS ↔ INDONESIA MOBILE BERBASIS AUGMENTED REALITY

Ossy D.E.W<sup>1</sup>, T.M. Zaini<sup>2</sup>, Frizka<sup>3</sup>

<sup>1</sup>Jurusan Sistem Informasi, Universitas Lampung

<sup>2,3</sup>Jurusan Sistem Informasi - Fakultas Ilmu Komputer  
Informatics & Business Institute Darmajaya

Jl. Z.A Pagar Alam No 93, Bandar Lampung - Indonesia 35142

Telp. (0721) 787214 Fax. (0721)700261

e-mail : [ossy\\_dew@yahoo.co.id](mailto:ossy_dew@yahoo.co.id), [tmzaini@ymail.com](mailto:tmzaini@ymail.com), [frizka@yahoo.com](mailto:frizka@yahoo.com)

### ABSTRACT

*In general a translator use keypad or keyboard as input device. Sometimes user getting trouble when he or she must type long word or sentence. A new way to solve that problem is using Optical Character Recognition (OCR). In this research we developed English - Indonesian, Indonesian - English translator using OCR and Augmented Reality. This application running well in Android flatform, when support of good lighting and particular camera distance and point of view. This application have two output, namely text and sound which realized by Augmented Reality technology .*

**Keyword** : *Translator, Optical Character Recognition, Augmented reality.*

### ABSTRAK

Pada umumnya sebuah translator menggunakan keypad or keyboard sebagai alat input. Terkadang pengguna mengalami kesulitan saat harus mengetikkan kata atau kalimat yang panjang. Sebuah cara baru untuk mengatasi masalah tersebut yaitu penggunaan Optical Character Recognition (OCR). Pada penelitian ini, kami mengembangkan penterjemah bahasa Inggris ke bahasa Indonesia, bahasa Indonesia ke bahasa Inggris menggunakan OCR dan teknologi Augmented Reality. Aplikasi ini dapat dieksekusi dengan baik pada flatform Android, jika didukung dengan pencahayaan yang baik dan jarak serta sudut pandang yang tepat. Aplikasi ini memiliki dua keluaran, yaitu teks dan suara yang direalisasikan oleh teknologi augmented reality.

Kata kunci : *Translator, Optical Character Recognition, Augmented reality.*

### I. PENDAHULUAN

Kemajuan teknologi telah menembus berbagai bidang kehidupan, termasuk bidang komunikasi. Kebutuhan

akan perangkat penterjemah untuk mengatasi perbedaan bahasa antar negara memicu berkembangnya aplikasi translator elektronik yang praktis, mudah

digunakan, serta interaktif. Umumnya kamus atau penterjemah elektronik akan menterjemahkan kata atau kalimat yang diketik oleh user melalui keyboard atau keypad. Jika kata atau kalimat terlalu panjang, proses input terkadang mengalami masalah, misal salah ketik atau waktu input yang relatif panjang. Untuk memudahkan dan mempersingkat proses input tersebut, terdapat beberapa upaya yang dapat dilakukan. pada penelitian ini akan diterapkan salah satu metoda pada cabang ilmu *computer vision* yaitu *Optical Character Recognition* (OCR). Dengan kemampuan mengenal character pada huruf latin yang dicapture secara *real time* dari keyboard, user tidak perlu lagi membaca selanjutnya mengetikan kata atau kalimat yang ingin diterjemahkan. Proses tersebut cukup dilakukan dengan mengarahkan kamera smartphone ke kata yang ingin diterjemahkan, kemudian aplikasi akan melakukan *tracking* hasil *capture* kamera, selanjutnya jika kata tersebut sudah dikenal maka akan dimunculkan hasil terjemahan kata tersebut berupa teks disertai dengan suara secara *real time*, hal tersebut direalisasikan dengan menerapkan teknologi Augmented Reality (AR).

## II. METODE PENELITIAN

Penelitian ini dilakukan melalui tahap-tahap proses seperti diuraikan berikut ini :

### C.1 Analisis kebutuhan

Tahap analisis kebutuhan merupakan tahap awal yang dilakukan dalam pembuatan aplikasi berdasarkan hirarki metode yang digunakan, metode waterfall. Analisis kebutuhan menganalisis kebutuhan-kebutuhan baik kebutuhan perangkat keras (hardware), kebutuhan perangkat lunak (software), dan kebutuhan pengguna (user) yang diperlukan untuk membangun aplikasi ini.

### C.2 Dekripsi Sistem

Aplikasi mobile translator ini merupakan aplikasi penerjemah yang menerapkan teknologi Augmented Reality untuk menampilkan output yang realtime pada smartphone, khususnya smartphone berbasis platform Android. Output yang dihasilkan dari aplikasi ini adalah suara. Augmented Reality merupakan teknik/teknologi untuk menambahkan objek virtual ke dalam lingkungan nyata dengan menggunakan penanda (marker) ataupun tidak (markerless), baik berupa objek 2D/3D maupun suara. Pada aplikasi mobile translator ini, objek virtual yang ditambahkan ke dalam lingkungan nyata

adalah suara. Di awal perancangan aplikasi ini, diinginkan aplikasi mobile translator yang melakukan proses translasi secara langsung tanpa perlu mengetikkan kata dan hanya mengarahkan kamera pada kata yang ingin diterjemahkan. Output suara yang dihasilkan oleh aplikasi menggunakan librari suara (speech engine) eksternal yang harus diinstal langsung pada smartphone. Aplikasi ini bekerja dengan meng-capture (mengambil) gambar dari teks yang akan diterjemahkan, kemudian gambar diubah menjadi teks dengan menggunakan OCR. Terjemahan kata yang dimasukan dicari pada database SQLite dan kemudian menjalankan librari suara untuk pelafalan arti dari kata yang diinputkan sebagai markernya.

### C.3 Kebutuhan Perangkat Keras

Spesifikasi minimal perangkat keras (hardware) untuk membangun aplikasi ini adalah sebagai berikut :

- Smartphone dengan platform Android versi min. 2.2
- CPU : 600 Mhz
- Ruang Penyimpanan : 164MB + microSD (2 GB)
- Memory : 256 MB
- Dimensi Layar : 240 x 320
- Masukkan : Touch screen kapasitif, volume controls, *accelerometer*.

- Konektivitas : Wi-Fi (802.11b/g/n), Bluetooth 2.1, USB, A-GPS, Quad band, HSDPA/HSUPA.

### C.4 Kebutuhan Perangkat Lunak

Perangkat lunak (software) yang diperlukan dalam penelitian ini adalah:

- Eclipse Helios SR 1 for Java Developers
- Android SDK Tools Revision 10
- SDK Platform Android 2.2 Froyo
- Android Development Tools (ADT) plugin 9.0.0 for Eclipse
- Virtual Box Linux for Windows
- Speech Engine
- Sistem Operasi Windows XP
- Sistem Operasi Linux Ubuntu 10.0

### C.5 Kebutuhan Pengguna

Kebutuhan pengguna (*user requirement*) didasarkan pada keinginan pengguna/calon pengguna terhadap aplikasi, yaitu user friendly. Karakter dari User friendly adalah easy to use, up to date, efisien dan attractive.

- *Easy to use*

Prosedur menjalankan atau eksekusi aplikasi harus mudah dipahami dan dimengerti oleh user. Sehingga tidak menyulitkan pengguna saat menggunakan aplikasi.

- *Up to date*

Aplikasi dikembangkan/dibangun mengikuti perkembangan teknologi terkini.

- Efisien

Penggunaan aplikasi tidak membutuhkan waktu lama.

- Attractive

Aplikasi memiliki daya tarik untuk menarik minat user, seperti dari kegunaan aplikasi.

## C.6. Perancangan

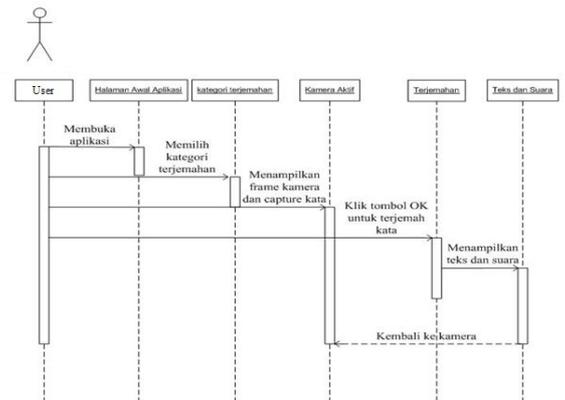
Konsep rancangan dituangkan dalam bentuk diagram pada Unified Modelling Language (UML), yang meliputi : use case, activity diagram, sequence diagram dan user interface.

### C.6.1. Use Case Diagram

*Use case diagram* digunakan untuk mendeskripsikan atau memberikan gambaran mengenai layanan system terhadap user, dalam hal ini pengguna dapat melakukan tracking teks, melihat hasil terjemahan dalam bentuk teks dan mendengarkan pelafalan atau cara pengucapan kata.

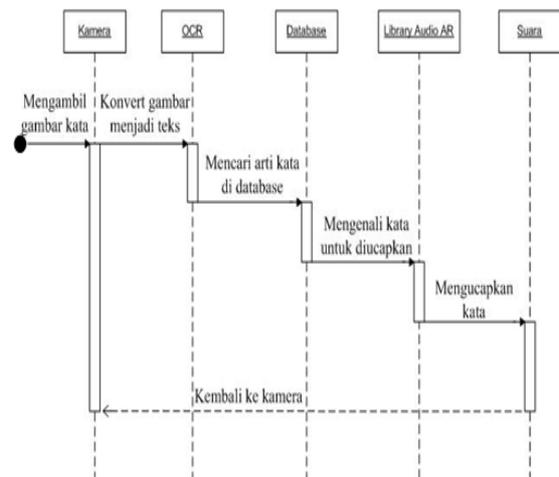
*Augmented Reality* pada aplikasi ini digunakan untuk menghasilkan output yang berupa suara yang melafalkan arti dari kata yang diinputkan. Aplikasi ini juga menerapkan teknik OCR dan SQLite.

### C.6.2. Sequence Diagram



Gambar 1. Sequence diagram

### Mobile Translator



Gambar 2. Sequence Diagram Proses tracking dan pencarian kata

Proses Mobile Translator dimulai ketika pengguna mengaktifkan kamera smartphone dan menginputkan gambar dari tulisan yang dikehendaki. Gambar tersebut akan tersimpan langsung pada smartphone dan digunakan sebagai penanda (marker) oleh AR yang nantinya akan digunakan untuk me-load suara

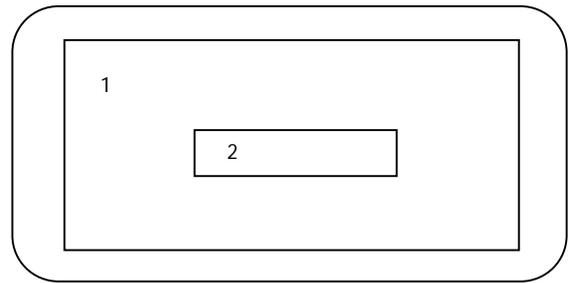
sesuai dengan teks yang diinputkan. OCR akan memproses gambar tersebut secara langsung dengan mengubah gambar menjadi teks. Setelah gambar diubah menjadi teks, sistem akan membuka database SQLite untuk mencari arti dari teks tersebut. Apabila arti dari teks ditemukan dalam database, hasilnya ditampilkan pada layar smartphone berupa teks dan terjemahan kata tersebut akan dilafalkan. Sedangkan bila artinya tidak ditemukan, akan ada pemberitahuan

### C.6.3 Desain Antarmuka Sistem

Desain antarmuka sistem menunjukkan gambaran dari desain awal aplikasi yang dibuat, terdiri dari desain tampilan penangkapan objek, desain tampilan output, dan desain tampilan yang menunjukkan terdapat kata yang tidak dapat diterjemahkan.

#### a. Desain Tampilan Capture

Desain ini memberikan gambaran proses peng-capture-an objek teks yang akan diterjemahkan. Pada kata tersorot yang akan diterjemahkan, terdapat kotak atau field dengan warna yang lebih terang dibanding sekitarnya. Desain tampilan penangkapan objek dapat dilihat pada gambar 3.



**Gambar 3. Desain Tampilan Capture**

### C.7 Coding

Realisasi rancangan aplikasi dilakukan dengan menulis kode program atau tahap coding program, terdapat beberapa bagian program yaitu :

#### C.7.1 Capture

Coding merupakan penulisan sistem kedalam bahasa pemrograman. Setelah merancang tampilan interface dari aplikasi yang dibuat, selanjutnya mengimplementasikannya kedalam bahasa pemrograman. Pada aplikasi ini digunakan bahasa pemrograman Java. Tampilan awal ketika aplikasi mobile translator dieksekusi (pada emulator) :

Memulai menjalankan aplikasi dengan meng-klik tombol START. Saat icon kamera diklik, coding program berikut dieksekusi. Image dicapture oleh kamera



**Gambar 4. Screenshot emulator**

Proses mengambil gambar (capture) ini sangat dipengaruhi oleh pencahayaan. Intensitas cahaya yang sangat besar mempermudah proses pembacaan gambar sebagai marker untuk kemudian dilakukan proses load suara dan teks hasil. Sebaliknya, intensitas cahaya yang sedikit menyebabkan kesalahan dalam pembacaan string dari teks inputan sehingga hasilnya pun akan salah. Selain itu, hal ini juga mengakibatkan teks awal tidak terdeteksi.

### C.7.2 Speak

Proses pelafalan kata terjemahan pada aplikasi ini ditunjukkan oleh potongan program dibawah ini:

```
String kata_kata =
    HasilPencarian.toString();
if (kata_kata!=null
    &&kata_kata.length(>0) {
    hpMembaca.speak(kata_kata,
        TextToSpeech.QUEUE_ADD, null);
    } }));
```

Code program tersebut digunakan untuk menghasilkan output suara yang melafalkan arti dari kata yang diterjemahkan.

*HasilPencarian* merupakan variabel global dimana variable tersebut berisi arti kata dari kata yang diinputkan untuk diterjemahkan. *kata\_kata* adalah variable yang digunakan untuk proses pengucapan kata. *kata\_kata* = *HasilPencarian*, artinya nilai dari variable *kata\_kata* sama dengan nilai pada variable *HasilPencarian* yaitu arti kata dari kata yang diinputkan untuk diterjemahkan. Dengan kata lain, bila nilai *kata\_kata* sama dengan *HasilPencarian*, maka library suara TextToSpeech Engine akan dipanggil dan menyuarakan atau mengucapkan kata sesuai dengan *HasilPencarian*. Output suara dihasilkan dari library suara yang bersifat eksternal dan telah terinstal pada smartphone. Library suaranya adalah TextToSpeech Engine. Oleh karena bersifat eksternal, library tersebut harus dipanggil kedalam project translator menggunakan fungsi *Intent*. Berikut potongan coding program untuk memanggil library suara :

```
Intent dengar_tts_di_hp = new Intent();

dengar_tts_di_hp.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
```

```
startActivityResult(dengar_tts_di
_hp,
KODE_UNTUK_PRIKSA_KATA);
```

Sintaks program di atas dieksekusi saat proses OCR selesai dan output akan ditampilkan dan diucapkan. Library AR untuk suara dipanggil dengan fungsi Intent. Library AR suara akan memeriksa kata apa yang dihasilkan dari proses

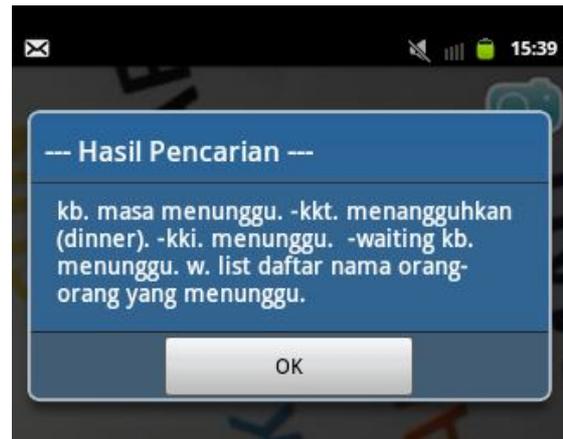
### C.8. Coding Install Speech Engine

Sebelum digunakan, library suara, TextToSpeech Engine harus diinstall terlebih dahulu. Berikut perintah untuk proses install speech Engine:

```
Intent di_pasang_lagi = new Intent();
di_pasang_lagi.setAction(TextToSpeec
h.Engine.ACTION_INSTALL_TTS_D
ATA);
startActivity(di_pasang_lagi); }
```

## III. HASIL DAN PEMBAHASAN

Proses penerjemahan dengan aplikasi mobile translator bila berhasil akan menampilkan kata terjemahan sesuai kata yang di capture oleh kamera ponsel. Tampilan output terjemahannya dapat dilihat pada Gambar 5.

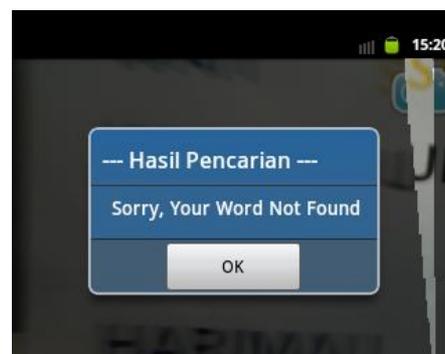


**Gambar 5. Interface hasil terjemahan yang berhasil**

Teks terjemahan berupa bahasa Indonesia atau bahasa Inggris dimuat pada kotak dialog pesan yang ditunjukkan dengan sintak `myAlertDialog.setMessage(HasilPencarian)`; .

### D.2 Hasil Tidak Ditemukan

Bila terjemahan kata tidak ditemukan, pesan pada kotak dialog memberitahukan bahwa kata yang dicari tidak ditemukan.



**Gambar 6. Screenshot aplikasi untuk kata yang tidak ditemukan hasil terjemahannya**

## E. Testing

Testing (pengujian sistem) dilakukan untuk menguji sistem secara keseluruhan, mulai dari desain hingga coding program. Hal ini bertujuan untuk memastikan bahwa aplikasi ini dapat berjalan dengan baik dan tidak/memiliki sedikit *bug*. Pengujian aplikasi Mobile Translator ini menggunakan metode Blackbox Testing yang focus pada persyaratan fungsional perangkat lunak. Blackbox Testing digunakan untuk mencari kesalahan interface, kesalahan pada fungsi yang salah atau hilang, kesalahan pada struktur data atau akses database, kesalahan performance, kesalahan inisialisasi dan tujuan akhir.

### E.1 Tahap Pengujian

Pengujian aplikasi dilakukan pada beberapa kondisi, seperti terlihat pada Tabel 1.

**Tabel 1. Hasil Pengujian aplikasi**

Test Case	Hasil Harapan	Hasil Keluaran
Kata yang diinput tidak ada	Muncul pesan peringatan dan diucapkan	sesuai
Capture kata dengan sedikit cahaya	Tidak muncul kotak dialog	sesuai
Teks dengan warna	Muncul kata terjemahan dan pelafalan	sesuai

### E.2. Analisis Hasil Pengujian

Berdasarkan beberapa tahap pengujian pada aplikasi mobile translator diatas, didapatkan hasil analisis sebagai berikut:

1. Pengujian dengan memasukkan (capture) kata yang terjemahannya tidak ada pada database, maka ditampilkan kotak dialog berisi pesan pemberitahuan bahwa kata yang dicari tidak ada, serta suara yang mengucapkan pesan pemberitahuan tersebut. Pada proses menerjemahkan kata dari bahasa Indonesia ke dalam bahasa Inggris, suara dan pelafalan kata yang dihasilkan cukup jelas. Sedangkan untuk proses menerjemahkan kata dari bahasa Inggris ke dalam bahasa Indonesia, pelafalan kata yang dihasilkan kurang jelas. Hal ini dikarenakan library suara yang digunakan berdialek asli Inggris. Selain itu juga belum ada library suara dengan dialek bahasa Indonesia.
2. Pengujian memasukkan (capture) kata dengan intensitas cahaya yang kecil menyebabkan kata tersebut tidak terdeteksi, sehingga proses OCR tidak dapat berjalan dan AR

juga tidak dapat memanggil library suara untuk pengucapannya.

3. Pengujian dengan teks berwarna dan ukuran font di atas 12 tidak mengalami kendala apapun (terlepas dari faktor cahaya). Mobile translator tetap memberikan output yang sesuai dengan masukan.

#### E.4. Maintenance

*Maintenance* atau pemeliharaan merupakan tahap terakhir dalam metode Waterfall. Tahap maintenance bertujuan untuk meningkatkan performa aplikasi. Pemeliharaan termasuk memeriksa dan memperbaiki kesalahan yang terdapat pada fungsi program. Selain itu juga terdapat perbaikan pada database kata.

#### IV. SIMPULAN

Telah berhasil dikembangkan sebuah aplikasi mobile yang berfungsi sebagai penterjemah Inggris – Indonesia, Indonesia - Inggris yang menggunakan OCR yang mengenali input kata menggunakan kamera smartphone dan menampilkan hasil terjemahan berupa teks sekaligus suara yang akan bermanfaat sebagai petunjuk pelafalan kata yang benar untuk pelafalan kata-kata pada bahasa Inggris, namun sebaliknya untuk pelafalan kata berbahasa Indonesia, sebagian kata pelafalannya belum tepat

karena keterbatasan belum tersedianya library pelafalan berbahasa Indonesia.

#### DAFTAR PUSTAKA

- [1] Rashid S.F, Shafait F, Breuel T, “Scanning Neural Network for Text Line Recognition”, in The DFKI website, 2010.
- [2] Charles P.K, Harish P, M. Swathi CH, “A Review of the Various Techniques used for Optical Character Recognition” in Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 1, Jan-Feb 2012, pp.659-662
- [3] Lopresti D, “Optical Character Recognition Errors and Their Effects on Natural Language Processing”, in International Journal on Document Analysis and Recognition manuscript, December 2008.
- [4] Mollah A.F, Majumder M, Basu S, Nasipuri M, “Design of an Optical Character Recognition System for Camera-based Handheld Devices”, in JCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.
- [5] Patel C, “Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study” in International Journal of Computer

- Applications (0975 –8887)Volume 55–No.10, October 2012.
- [6] Hahn Jim, “Optical Character Recognition Software in Library Mobile Apps” in Int Conference IFLA, May 2012.
- [7] Anonym, “Word Lens application for Android”, in <http://www.theverge.com/2012/7/9/3146525/>, posting July 2012.
- [8] Anonym, “TranslatAR : A mobile Augmented Reality Translator” in <http://ilab.cs.ucsb.edu/index.php/component/content/article/10/14>, posting 2011.