

## Aplikasi Identifikasi Kata Berbasis Optical Character Recognition dan Augmented Reality

Devy Normalasari<sup>1</sup>, Irawan Afrianto<sup>2</sup>

<sup>1,2</sup>Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipatiukur 112-116 Bandung

E-mail : [devy.normalasari@yahoo.com](mailto:devy.normalasari@yahoo.com)<sup>1</sup>, [irawan.afrianto@email.unikom.ac.id](mailto:irawan.afrianto@email.unikom.ac.id)<sup>2</sup>

### ABSTRAK

Teks tertulis merupakan salah satu metode yang umum untuk menyampaikan informasi. Namun, ketika berwisata ke luar negeri informasi tersebut ditemui dalam bahasa asing. Bagi beberapa individu perbedaan bahasa membuat informasi tersebut tidak dapat tersampaikan dengan baik. Penggunaan kamus digital menjadi pilihan wisatawan untuk menerjemahkan bahasa dengan pencarian kata yang cepat dan mudah. Namun kamus digital masih memiliki kekurangan, yaitu wisatawan harus mengetikkan teks yang ingin diterjemahkan, maka dibutuhkan sebuah aplikasi alternatif yang dapat mendeteksi karakter teks dan menerjemahkannya langsung. Salah satu teknologi yang dapat dimanfaatkan adalah teknologi *Augmented Reality* dengan dukungan perangkat *mobile* bersistem operasi Android. Teknologi Optical Character Recognition juga akan diterapkan untuk pengenalan kata pada citra yang dideteksi. Pada sistem yang akan dibangun, *Augmented Reality* akan memanfaatkan kamera yang ada pada perangkat Android untuk mendeteksi teks tanpa harus menyimpan gambar terlebih dahulu. Kemudian objek atau target teks yang terdeteksi akan dikonversi kedalam teks yang dapat diedit oleh *Optical Character Recognition* sehingga teks tersebut dapat diterjemahkan. Penerjemahan dilakukan secara *offline* maupun *online* dengan Bing Microsoft Translator. Hasil terjemahan teks tersebut kemudian akan ditampilkan dengan *Augmented Reality* pada layar perangkat Android. Terjemahan kata akan tampil di luar wilayah pendeteksian atau ROI secara berurutan untuk setiap kata dari atas ke bawah. Implementasi AR dengan OCR pada aplikasi Word Translator diharapkan dapat membantu wisatawan dalam menerjemahkan kata-kata berbahasa asing secara *realtime* dan akurat tanpa harus mengetik.

**Kata kunci** : Augmented Reality, Optical Character Recognition, Pengenalan Kata, Penerjemah, Vuforia

### 1. PENDAHULUAN

Teks tertulis merupakan salah satu metode yang paling umum untuk menyampaikan informasi.

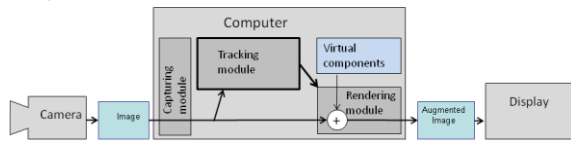
Namun, ketika melakukan traveling atau berwisata ke luar negeri informasi tersebut ditemui dalam bahasa asing. Bagi beberapa individu perbedaan bahasa membuat informasi tersebut tidak dapat tersampaikan dengan baik.

Para wisatawan yang tidak menguasai bahasa asing dapat memanfaatkan kamus saku untuk menerjemahkan informasi berupa teks yang ada di negara tersebut. Akan tetapi, penggunaan kamus saku dirasa kurang praktis untuk berwisata. Penggunaan kamus digital juga menjadi pilihan wisatawan untuk menerjemahkan bahasa dengan pencarian kata yang cepat dan mudah. Namun kamus digital tersebut masih memiliki kekurangan, yaitu wisatawan harus mengetikkan teks yang ingin diterjemahkan. Dari masalah-masalah tersebut, maka dibutuhkan sebuah aplikasi alternatif yang dapat mendeteksi karakter teks dan menerjemahkannya secara langsung dengan tingkat mobilitas yang cukup tinggi. Salah satu teknologi yang dapat dimanfaatkan adalah teknologi *Augmented Reality* dan perangkat *mobile* bersistem operasi Android. Berdasarkan data dari *International Data Corporation*, sistem operasi Android menguasai 84,7% pangsa pasar pada kuartal kedua tahun 2014[3]. Oleh karena itu, *platform* Android menjadi pilihan untuk menerapkan sistem *Augmented Reality* ini.

#### 1.1 Augmented Reality (AR)

Augmented Reality (AR) adalah campuran dari gambar dunia nyata dengan benda-benda buatan atau informasi yang dihasilkan oleh komputer [1]. AR juga didefinisikan sebagai perpanjangan lingkungan pengguna dengan konten buatan. Benda-benda nyata menampilkan informasi yang tidak dapat diterima oleh pengguna dalam inderanya sendiri. Hal ini membuat realitas bertambah sesuai sebagai alat untuk membantu persepsi dan interaksi pengguna dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata. AR dapat diaplikasikan untuk semua indera, termasuk pendengaran, dan sentuhan. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industry manufaktur, AR juga dapat digunakan untuk penerjemahan teks dalam berbagai

bahasa dengan tambahan OCR yang dimiliki pada AR.



Gambar 1. Proses Augmented Reality

Augmented reality memiliki 2 metode pengenalan target. Metode pengenalan target yang pertama adalah *marker based tracking* dan yang kedua adalah *markerless*. Augmented reality pada *marker based tracking* merupakan metode yang dikembangkan sejak tahun 80an. Marker merupakan ilustrasi dari hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi marker dan menciptakan dunia virtual 3D yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z [4]. Berbeda dengan *Marker Based Tracking*, metode *Markerless Augmented Reality* tidak lagi memerlukan penggunaan sebuah marker untuk menampilkan objek-objek digital. Metode ini memiliki beberapa teknik-teknik khusus yaitu *Face Tracking*, *3D Object Tracking*, *Motion Tracking*, *GPS Based Tracking*, dan *Text Recognition*.

1.2 Optical Character Recognition (OCR)

Optical character recognition (OCR) adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (*image character*) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar *output scanner* dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan[2]. OCR dibagi menjadi beberapa tahap yaitu, *data capture*, *preprocessing*, *segmentasi*, *normalisasi*, *pengenalan*, dan *postprocessing*.

a. Data Capture

Data capture merupakan proses konversi suatu dokumen (hardcopy) menjadi suatu file gambar digital.

b. Preprocessing

Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya.

c. Segmentasi

Segmentasi adalah proses memisahkan area pengamatan (region) pada tiap karakter yang akan dideteksi.

d. Normalisasi

Normalisasi adalah proses merubah dimensi region tiap karakter. Dalam OCR algoritma yang digunakan pada proses ini adalah algoritma scaling.

e. Pengenalan

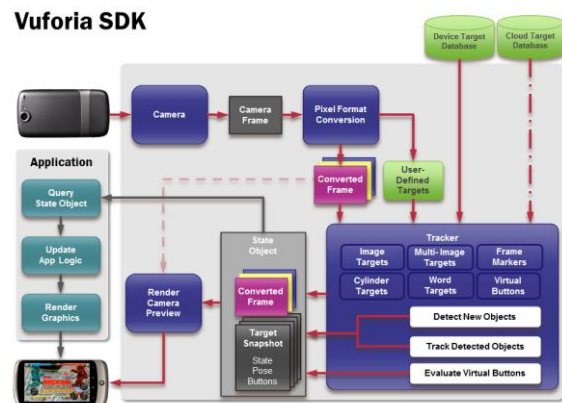
Pengenalan merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada database.

f. Postprocessing

Pada umumnya proses yang dilakukan pada tahap ini adalah proses koreksi ejaan sesuai dengan bahasa yang digunakan.

1.3 Vuforia SDK

Vuforia adalah Augmented Reality SDK (*Software Development Kit*) untuk pembangunan AR pada perangkat *mobile* yang disediakan oleh Qualcomm[12]. Target pada vuforia merupakan objek pada dunia nyata yang dapat dideteksi oleh kamera, untuk menampilkan objek virtual. Vuforia menggunakan teknologi *computer vision* untuk mengenali dan membuat pelacakan objek yang ditangkap oleh kamera video secara *realtime*.



Gambar 2. Arsitektur Vuforia SDK

Komponen *tracker* berisi algoritma *computer vision* yang dapat mendeteksi dan melacak benda-benda dunia nyata yang ada pada frame *kamera video*. Subtipe *Tracker* pada gambar 2.16 yang didefinisikan dalam Vuforia adalah sebagai berikut :

- a. *ImageTarget*, objek dari *Trackable class* yang terdiri dari gambar.
- b. *CylinderTarget*, objek dari *Trackable class* yang terdiri dari gambar yang diterapkan pada silinder atau permukaan kerucut.
- c. *MultiTarget*, objek dari *Trackable class* yang terdiri dari beberapa gambar yang dikombinasikan dalam konfigurasi spasial yang diberikan.
- d. *Marker*, objek dari *Trackable class* yang terdiri dari bingkai marker.
- e. *Word*, objek dari *Trackable class* yang mewakili elemen teks.

- f. *VirtualButton*, merupakan daerah persegi panjang dari *ImageTarget* dimana dapat memicu aktivitas jika disentuh atau terhalangi dalam tampilan kamera.

**1.4 Vuforia Word Target**

Word Targets merupakan elemen textual dimana Vuforia dapat mengenali dan melacak kata yang valid. Berikut adalah kriteria kata yang dapat dikenali oleh Vuforia :

- a. Sebuah kata yang memiliki panjang hingga 24 karakter
- b. Sebuah kata yang mungkin berisi *hyphens* (kata majemuk tertutup, misalnya : “Mother-in-law”)
- c. Sebuah kata yang mungkin berisi ruang (kata majemuk terbuka, misalnya : “Hot dog”)
- d. Sebuah kata yang tidak mengandung angka

Vuforia SDK dapat mengenali kata-kata dan melacaknya yang sama seperti jenis dari target lain, dengan dua mode pengakuan yang tersedia yaitu “*Words*” dan “*Characters*.” Apabila menggunakan “*Words*” mode, kata tersebut diakui (dan kemudian dilacak) jika termasuk dalam *wordlist* yang diberikan, dimana *wordlist* dapat disimpan pada perangkat dan dimuat oleh aplikasi pada saat *runtime*. *Wordlist* juga dapat diperpanjang dengan kata-kata tambahan, dan filter kata-kata tertentu yang dapat dimuat dan diterapkan melalui SDK API. Selain itu, ketika menggunakan “*Characters*” mode, setiap kata yang dihasilkan dari urutan karakter sembarang (termasuk nomor) juga dapat terdeteksi.

Pada sistem yang akan dibangun, *Augmented Reality* akan memanfaatkan kamera yang ada pada perangkat Android untuk mendeteksi teks tanpa harus menyimpan gambar terlebih dahulu. Kemudian objek atau target teks yang terdeteksi akan dikonversi kedalam teks yang dapat diedit oleh OCR sehingga teks tersebut dapat diterjemahkan. Hasil terjemahan teks tersebut kemudian akan ditampilkan kembali dengan *Augmented Reality* secara realtime pada layar perangkat Android. Dengan sistem seperti ini wisatawan dapat dengan mudah menerjemahkan teks tanpa mengetik terlebih dahulu ataupun mencari kata pada kamus saku.

Maksud dari penelitian ini adalah untuk mengimplementasikan *augmented reality* dengan *optical character recognition* pada aplikasi Word TranslatAR.

Sedangkan tujuan yang dicapai pada penelitian ini adalah:

- a. Memberikan alat bantu alternatif untuk membantu para wisatawan untuk menerjemahkan kata-kata berbahasa asing tanpa harus mengetik.
- b. Menguji ketepatan pendeteksian teks dengan berbagai macam kriteria huruf latin.
- c. Menguji keakuratan penerjemahan kata dari bahasa inggris ke bahasa indonesia.

**2. ISI PENELITIAN**

Aplikasi yang dibangun akan dibagi menjadi 3 proses besar, yaitu deteksi kata, penerjemahan teks dan menampilkan hasil terjemahan. Pendeteksian dilakukan terhadap objek berupa teks dengan maksimal 24 karakter yang diambil secara *realtime* menggunakan kamera dari perangkat android tanpa melakukan proses penyimpanan terlebih dahulu. Objek berupa teks yang berbentuk citra digital akan berfungsi sebagai input dari Itern, kemudian citra tersebut dikonversi menjadi karakter yang dapat diedit. Setelah semua karakter diperoleh lalu dilakukan proses penerjemahan pada karakter tersebut dan menampilkannya kembali dalam bentuk teks dengan *augmented reality*.

**2.1 Analisis Marker dan Target**

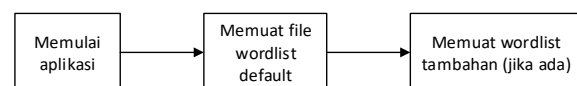
Pada Itern ini, target merupakan kata sehingga pendeteksian target menggunakan teks atau huruf sebagai *marker*-nya. Vuforia SDK dapat mengenali kata-kata dan melacaknya, dengan dua mode pelacakan yang tersedia yaitu “*Words*” dan “*Characters*.” Pada penelitian ini metode yang dipakai menggunakan mode “*Words*”, kata tersebut akan diakui (dan kemudian dilacak) jika termasuk dalam *wordlist* yang ada pada sistem.

**2.2 Deteksi dan Pengenalan Kata**

Proses pendeteksian teks merupakan proses pertama dari 3 proses utama dalam Itern ini. Proses pendeteksian kata dilakukan untuk mendapatkan nilai dari teks yang dideteksi untuk kemudian digunakan pada proses penerjemahan. Proses pendeteksian teks dilakukan dengan teknologi *augmented reality*. Teknologi pendeteksian teks ini telah disediakan oleh Vuforia sebagai salah satu bentuk *tracking* untuk *augmented reality*.

**2.2.1 Memuat Wordlist**

Inisialisasi pertama, sistem memuat *word list* untuk mempercepat pemilihan kata dalam pendeteksian. Sebelum kamera di inisialisasi dan pendeteksian target dilakukan, Itern akan memuat *wordlist* ke dalam proses. Hal ini bertujuan untuk mengambil nilai dari semua daftar kata untuk pencocokan elemen teks yang muncul dari kamera. *Wordlist* memiliki format biner khusus dari Vuforia yang dirancang untuk menyertakan satu set besar kata. Gambar 3 menunjukkan diagram blok memuat wordlist.



**Gambar 3.** Diagram Blok Memuat Wordlist

**2.2.2 Menentukan ROI**

Inisialisasi kedua, menentukan *Region Of Interest* (ROI) untuk memberikan daerah deteksi pada kamera dalam mendeteksi objek. Langkah pertama dalam mendeteksi sebuah teks adalah menentukan *Region of Interest* (ROI). ROI ditentukan untuk membatasi wilayah pelacakan dan deteksi dimana terdapat suatu teks. Gambar 4 adalah contoh dari orientasi *Region of Interest*.



**Gambar 4** Orientasi *Region Of Interest*

**2.2.3 Menangkap Target**

Kamera menangkap citra secara *realtime*. Dalam hal ini, kamera akan menangkap citra dengan mengambil *frame* tanpa melakukan *pause* seperti layaknya menangkap *frame* dari video. Pada proses penangkapan target dari kamera, Itern mengambil citra dari *frame* yang ditangkap kamera.

Setelah melakukan instansiasi untuk modus kamera yang akan digunakan, kemudian tahapan selanjutnya adalah melakukan *render* pada objek yang tertangkap kamera. Adapun beberapa tahapan sebelum melakukan *render* menjadi citra yang dapat dilihat pada Gambar 5.

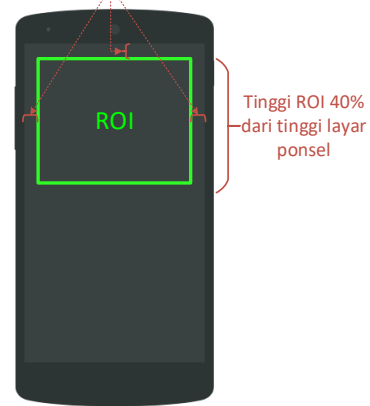


**Gambar 5.** Flow Proses Menangkap Target Citra dari Kamera

Tinggi dan lebar layar ditentukan dengan mengambil nilai dari layout Itern. Pertama instansiasi objek untuk mengambil nilai dari ukuran layar. Jika perangkat yang digunakan wisatawan adalah tablet, maka lebar margin adalah 20% dari lebar layar sedangkan jika perangkat yang digunakan adalah ponsel maka lebar margin adalah 5% dari lebar layar.

Jika perangkat yang digunakan wisatawan adalah tablet, maka tinggi ROI adalah 30% dari tinggi layar sedangkan jika perangkat yang digunakan adalah ponsel maka tinggi ROI adalah 40% dari tinggi layar.

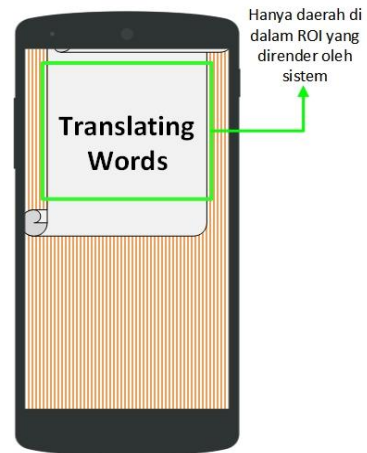
Lebar margin 5% dari lebar layar ponsel



**Gambar 6.** Tampilan Akhir ROI yang Ditampilkan

Setelah ROI terbentuk lalu lakukan konversi koordinat layar dari ROI yang telah dikalkulasikan ke koordinat kamera, Setelah mendapatkan koordinat kamera, kemudian tentukan sebuah persegi sebagai objek dari ROI yang telah dikalkulasi.

Dapatkan data dan ukuran dari objek persegi untuk memberikan nilai pada *viewport* dimana *viewport* sendiri merupakan area dimana objek yang menarik akan di-*render*. *Viewport* dinyatakan dalam koordinat *render-device-specific*. Sistem akan me-*render* objek yang ditangkap oleh kamera menjadi citra.



**Gambar 7.** Daerah yang di-*render* oleh Sistem

Setelah Itern mendapatkan citra yang ditangkap oleh kamera yang mana citra tersebut merupakan *frame* dari video yang ditangkap kamera, kemudian Itern akan me-*render frame* tersebut untuk melakukan *binary masking*.

Setelah sistem melakukan render pada target untuk mendapatkan citra dari setiap *frame* video, kemudian *Vuforia* akan melakukan pelacakan objek dimana objek tersebut merupakan teks (kata). Ada beberapa langkah yang perlu dilakukan untuk melakukan pelacakan teks dengan *Vuforia*. Gambar 8 adalah langkah-langkah untuk melakukan



pelacakan dan pengenalan kata dengan memanfaatkan API dari *Vuforia*.

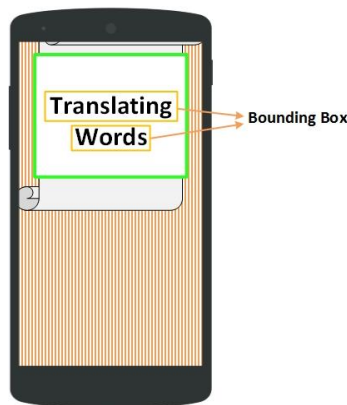


**Gambar 8.** Alur Proses dari Pelacakan dan Pengenalan Kata

Sebelum melakukan pelacakan dan pengenalan kata, perlu dilakukan suatu pencarian terhadap target yang dapat dilacak pada *frame*. Untuk mendapatkan objek dari target yang sedang dilacak, pertama lakukan instansiasi objek dari kelas *TrackableResult* yang telah disediakan oleh *vuforia* untuk menampung hasil dari pelacakan. Kemudian periksa apakah jenis objek *result* dari *TrackableResult* sama dengan jenis dari kelas *WordResult* dengan membandingkan operasi *result.getType()* dengan operasi *WordResult.getClassType()*. Jika kondisinya benar maka objek yang terlacak memiliki teks. Itu berarti dapat dilakukan pelacakan kata dari *frame* yang di deteksi tersebut. Kemudian langkah selanjutnya adalah mendapatkan kata.

Setelah memeriksa jenis objek *result* dari *TrackableResult* dan asilnya adalah sama dengan jenis kelas *WordResult*, kemudian dilakukan pelacakan kata guna mendapatkan kata yang ada pada *frame*

Apabila terdapat kata yang terdeteksi oleh sistem, maka sistem akan melakukan *render* pada objek *wordBox2D* yang mana merupakan kotak pembatas kata. *Render* dilakukan untuk menampilkan kotak pembatas kata ke layar. Hal ini bertujuan untuk memberi tanda pada kata yang terdeteksi atau dikenali oleh sistem.



**Gambar 9.** Kotak Pembatas Kata

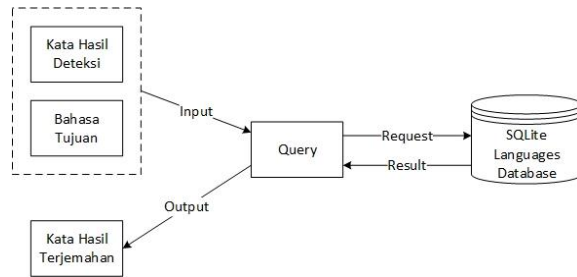
Langkah berikutnya dalam mendapatkan kata adalah mengambil nilai dari kata itu sendiri. *Vuforia* telah menyediakan fungsi untuk mengambil nilai dari kata yang terdeteksi. Fungsi tersebut merupakan operasi *getString()* yang ada pada kelas *Word*.

Operasi ini mengembalikan nilai *string* dari karakter *Unicode* untuk kata yang terdeteksi.

**2.3 Penerjemahan Kata**

Setelah setelah sistem mendapatkan nilai hasil pendeteksian kata, kemudian sistem akan menerjemahkannya sesuai dengan bahasa tujuan yang dipilih oleh wisatawan. Ada dua cara yang dilakukan untuk menerjemahkan kata berdasarkan ketersediaan *resource* dari bahasa yang ada, yaitu secara *Offline* dan *Online*.

Penerjemahan kata secara *offline* dilakukan dengan memanfaatkan database untuk menyimpan *resource* dari kumpulan kata dalam berbagai bahasa. Walaupun bahasa yang disediakan tidak sebanyak yang disediakan oleh penyedia layanan terjemahan online, penerjemahan kata secara *offline* dapat menerjemahkan kata tanpa harus terhubung ke jaringan internet sehingga tidak tergantung dengan kecepatan dan ketersediaan jaringan internet di tempat wisatawan berada.



**Gambar 10.** Alur Proses Terjemahan Offline

Pada penelitian ini, Microsoft Bing Translator dimanfaatkan untuk menerjemahkan kata secara *online* apabila bahasa tujuan yang dipilih wisatawan tidak terdaftar sebagai bahasa yang memiliki *resource offline*. Microsoft Bing Translator API merupakan API yang digunakan untuk fitur penerjemah yang terdapat pada sistem. Secara default API ini akan menerjemahkan semua pesan yang diterima ke dalam bahasa yang digunakan oleh *user*.

Hasil dari deteksi kata akan ditampilkan dalam *Internat* untuk kemudian dikirimkan ke server Microsoft Translator oleh sistem untuk diterjemahkan. Apabila server Microsoft Translator memberikan balasan berupa pesan kesalahan atau kata yang dikirim tidak memiliki terjemahan pada bahasa tujuan, maka sistem akan menampilkan kata awal (kata hasil deteksi) ke layar ponsel sebagai hasil terjemahan.

**2.4 Menampilkan Hasil Terjemahan**

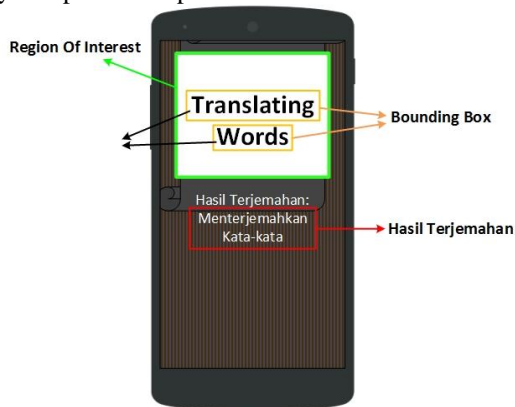
Setelah hasil pengenalan kata diterjemahkan ke bahasa tujuan, kemudian hasil terjemahan kata tersebut ditampilkan ke layar. Untuk menampilkannya kelayar, terdapat dua tahapan yaitu

menentukan posisi objek kemudian melakukan *render* terhadap *frame*.

Untuk menentukan posisi objek, pertama periksa jika kedua kata berada dalam garis yang sama. Kedua kata dikatakan berada dalam garis yang sama jika titik tengah (pada sumbu Y) dari titik pertama adalah diantara nilai dari titik kedua. Apabila kedua kata berada pada garis yang berbeda maka hasil terjemahan yang ditampilkan pun demikian. Jika yang terdeteksi lebih dari satu kata maka kata kedua letaknya akan berada di bawah kata pertama. Posisi letak kata hasil terjemahan akan ditampilkan dibawah persegi panjang ROI dan menggunakan *center alignment* atau konten berada di tengah layar.

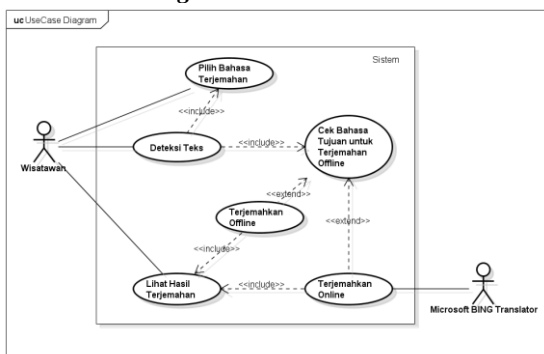
Setelah menentukan posisi untuk menampilkan hasil dari kata yang telah diterjemahkan, kemudian dilakukan *render frame* untuk menampilkan objek tersebut ke layar.

Setelah melakukan *render* maka sistem akan menampilkan terjemahan ke layar ponsel. Hasil terjemahan ini akan ditampilkan secara *realtime* saat sistem dapat mengenali kata yang ada pada objek. Untuk lebih lengkapnya, ilustrasi hasil tampilan di layar dapat dilihat pada Gambar 11.



Gambar 11. Menampilkan Objek Hasil Terjemahan

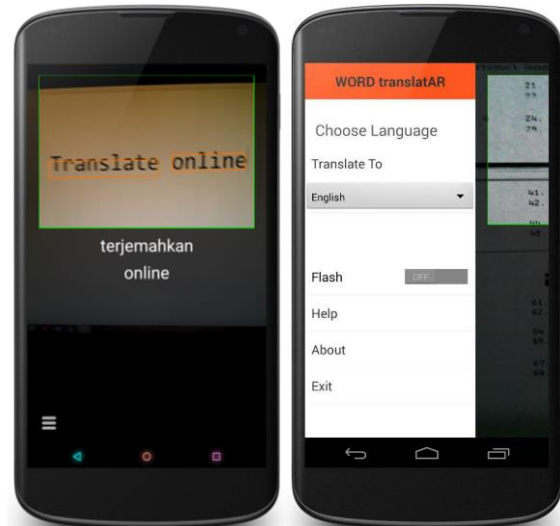
2.5 Use Case Diagram



Gambar 12. Use Case Diagram Word TranslatAR

2.6 Implementasi

Antarmuka dari aplikasi dibuat pada platform Android. Terdapat dua bagian antarmuka yaitu halaman utama dan halaman konfigurasi.



Gambar 13. Implementasi Antarmuka Word TranslatAR

2.7 Hasil Pengujian

Pengujian yang dilakukan adalah melakukan pendeteksian terhadap berbagai jenis teks. Pengujian dilakukan dengan 6 tahap, rincian dari rencana pengujian yang akan dilakukan dapat dilihat pada tabel 1.

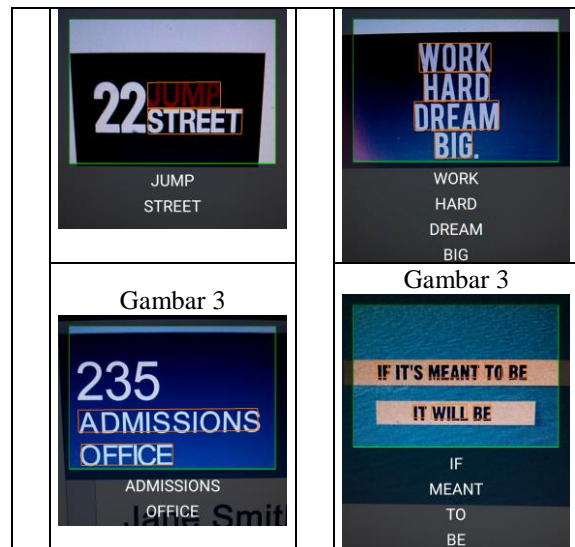
Tabel 1. Rencana Pengujian Deteksi Teks

No	Pengujian	Kegiatan
1	Pengujian 1	Mendeteksi teks tulisan tangan
2	Pengujian 2	Mendeteksi teks dengan font face yang berbeda
3	Pengujian 3	Mendeteksi teks dengan warna yang berbeda
4	Pengujian 4	Mendeteksi teks dengan style tertentu
5	Pengujian 5	Mendeteksi teks yang mengandung angka dan nomor
6	Pengujian 6	Mendeteksi lebih dari 1 kata

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian deteksi teks terhadap sistem ini. Pengujian ini bertujuan untuk mengetahui kemampuan deteksi terhadap teks dengan berbagai macam kriteria huruf latin. Berikut adalah rincian dari kasus dan hasil pengujian deteksi teks.

Tabel 2. Hasil Pengamatan dari Setiap Pengujian

	Hasil Pengamatan Gambar 1	Hasil Pengamatan Gambar 1	
Pengujian 1		Pengujian 2	



Pada pengujian keakuratan deteksi teks ini, pengujian akan dilakukan sebanyak 10 kali terhadap 3 buah gambar untuk setiap pengujian, dengan waktu tunggu pendeteksian sebanyak 5 detik. Jika dalam waktu tersebut tidak terdeteksi, maka sistem dianggap tidak mampu untuk mendeteksi teks tersebut.

**Tabel 3.** Hasil Pengujian

	Gambar	Jumlah Teks terdeteksi / jumlah uji	Presentase	Rata - rata
Pengujian 1	Gambar			
	Gambar 1	0/10	0%	43%
	Gambar 2	6/10	60%	
	Gambar 3	7/10	70%	
Pengujian 2	Gambar			
	Gambar 1	0/10	0%	46%
	Gambar 2	8/10	80%	
	Gambar 3	6/10	60%	
Pengujian 3	Gambar			
	Gambar 1	10/10	100%	100%
	Gambar 2	10/10	100%	
	Gambar 3	10/10	100%	
Pengujian 4	Gambar			
	Gambar 1	10/10	100%	100%
	Gambar 2	10/10	100%	
	Gambar 3	10/10	100%	
Pengujian 5	Gambar			
	Gambar 1	0/10	0%	0%
	Gambar 2	0/10	0%	
	Gambar 3	0/10	0%	
Pe	Gambar			
		Jumlah Teks terdeteksi /	Presentase	Rata - rata

	jumlah uji		
Gambar 1	8/10	80%	73%
Gambar 2	8/10	80%	
Gambar 3	6/10	60%	

Berdasarkan hasil dari pengujian deteksi teks dengan beberapa kriteria huruf latin, dapat ditarik kesimpulan bahwa aplikasi Word TranslatAR dapat melakukan deteksi dengan baik terhadap teks berwarna, teks dengan *style* dan teks lebih dari satu kata, namun aplikasi Word TranslatAR tidak dapat melakukan pendeteksian dengan baik terhadap tulisan tangan, teks dengan *font face*, dan teks yang memiliki angka. Sehingga dianjurkan untuk melakukan pendeteksian terhadap teks dengan huruf latin tanpa *font face*, angka dan bukan tulisan tangan.

### 3. PENUTUP

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

- a. Word TranslatAR dapat menjadi alat bantu alternatif untuk penerjemahan kata bagi wisatawan.
- b. Implementasi *Augmented Reality* dengan *Optical Character Recognition* pada aplikasi AR Word Translator membantu wisatawan untuk menerjemahkan teks berbahasa asing yang terdapat di papan petunjuk tanpa harus menetik.
- c. Keakuratan kata yang dideteksi pada aplikasi Word TranslatAR terhadap font tanpa style sudah sangat baik sehingga layak digunakan untuk penerjemahan teks-teks pendek.

Untuk pengembangan aplikasi dalam jangka waktu ke depan, ada beberapa saran yang dapat dilakukan, antara lain :

- a. Memungkinkan deteksi teks untuk beberapa bahasa dengan huruf-huruf non latin seperti huruf kanji dan huruf arab.
- b. Melengkapi semua bahasa untuk proses terjemahan secara offline sehingga aplikasi dapat digunakan tanpa harus terkoneksi dengan internet.

### DAFTAR PUSTAKA

- [1] S. Siltanen, *Theory and applications of marker-based augmented reality*. VTT Technical Research Centre of Finland, 2012.
- [2] C. Patel, *et al*, "Optical Character Recognition by Open Source OCR Tool Tesseract", *J. Computer Applications.*, Vol 55, No.10, 2012.
- [3] International Data Corporation..( 2014, September). *Smartphone OS Market Share, Q3 2014* [online]. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Second ed. New Jersey: Prentice-Hall, 2002.
- [5] G. X. Ritter and J. N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*: CRC Press, 1996.
- [6] R.T. Azuma, "A Survey of Augmented Reality", *Teleoperators and Virtual Environments* 6, 1997.
- [7] M. Paul, Kishino, Fumio., *Augmented Reality: A class of displays on the reality-virtuality continuum*, Japan: ATR Communication Systems Research Laboratories, 1994.
- [8] M. Cheriet, *et al.*, *Character Recognition Systems*. New Jersey: John Wiley & Sons, 2007.
- [9] Novhard. (2014, September). *Pattern Recognition atau Pengenalan Pola* [online]. <http://novhard.wordpress.com/2007/09/07/pattern-recognitionatau-pengenalan-pola/>
- [10] Developers.(2014,September).*About* [online]. <http://developer.android.com/about/index.html>
- [11] Developers.(2014,September).*SDK*[online]. <http://developer.android.com/sdk/index.html>
- [12] Vuforia.(2014, September). *Qualcomm Vuforia Developer Portal*[online]. <https://developer.vuforia.com/>
- [13] (WCECS), vol.1, October 2012.
- [14] T. Khan, *et al.*, "Augmented Reality Based Word Translator," *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, vol.2, March 2014.