

Aplikasi Penerjemah Gambar Teks Berbahasa Inggris Menggunakan Teknologi Realitas Tertambah pada Perangkat Berbasis Android

Antonius Kevin Wiguna, Nanik Suciati, dan Wijayanti Nurul Khotimah
Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
e-mail: kevin.wiguna@gmail.com, nanik@if.its.ac.id, wijayanti@if.its.ac.id

Abstrak— Di dalam lingkungan dunia nyata, banyak informasi berupa teks yang disajikan menggunakan bahasa yang tidak kita mengerti. Informasi tersebut bisa kita temui di tempat-tempat umum. Biasanya, diperlukan alat bantu untuk memahami tulisan dalam bahasa asing tersebut. Kemudahan dan kecepatan dalam penggunaan alat penerjemah menjadi penting terutama bila kita sedang berada di tempat umum. Untuk itu, diajukan pengembangan sebuah aplikasi yang mampu menerjemahkan teks yang mudah digunakan. Pengguna memilih gambar teks yang ingin diterjemahkan dari tampilan kamera aplikasi dengan menyentuh gambar teks yang diinginkan. Area gambar teks diambil menggunakan algoritma berbasis deteksi tepi pada gambar. Teks dikenali menggunakan Optical Character Recognition (OCR) dengan bantuan pustaka Tesseract dan diterjemahkan menggunakan bantuan Yandex.Translate API. Hasil penerjemahan akan ditampilkan menggunakan konsep realitas tertambah, di mana hasil akan ditampilkan langsung pada tampilan kamera di aplikasi. Aplikasi ini dibuat untuk berjalan pada perangkat yang menggunakan sistem operasi Android. Penerjemahan yang dapat dilakukan adalah penerjemahan dari bahasa Inggris ke bahasa Indonesia. Pengujian fungsional aplikasi menunjukkan hasil yang cukup baik untuk pengambilan area ROI dan proses mendapatkan hasil terjemahan. Hasil pengujian pada pengguna menunjukkan bahwa pengguna puas dengan kenyamanan, performa, dan kemudahan penggunaan aplikasi.

Kata Kunci—Android, Bahasa Inggris, Bahasa Indonesia, Gambar Teks, Penerjemah, Realitas Tertambah.

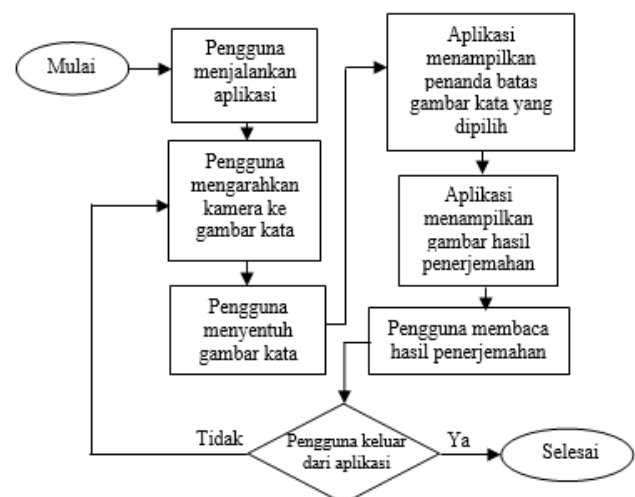
I. PENDAHULUAN

BERADA di tempat dengan bahasa yang tidak kita kuasai seringkali menimbulkan masalah. Kita tidak dapat menggunakan pengetahuan bahasa yang kita miliki untuk menyerap informasi dari lingkungan sekitar. Salah satunya adalah informasi yang berupa tulisan. Informasi berupa tulisan banyak sekali terdapat baik dalam lingkungan nyata seperti jalan raya dan fasilitas umum, maupun lingkungan maya seperti di Internet. Kesulitan akan muncul ketika informasi yang kita butuhkan adalah informasi yang penting, seperti informasi cara menggunakan transportasi umum atau informasi posisi dan arah untuk menuju suatu tempat yang biasanya terdapat pada rambu-rambu lalu lintas di jalan.

Salah satu solusi untuk mengatasi masalah tersebut adalah dengan menggunakan alat penerjemah bahasa. Alat penerjemah bahasa sudah tersedia dalam berbagai macam bentuk dan jenis. Dilihat dari segi metode masukannya, alat penerjemah bahasa menggunakan berbagai jenis metode, seperti mengetikkan masukan berupa teks [1], mengucapkan kata yang ingin diterjemahkan [2], ataupun dengan menggunakan deteksi tulisan tangan, khususnya untuk

bahasa yang menggunakan huruf non-Latin [3]. Metode-metode ini memiliki kekurangan dimana metode-metode tersebut tidak terlalu optimal jika kita melihat suatu teks dalam bahasa asing dan ingin menerjemahkannya ke dalam bahasa yang kita ketahui. Teks perlu dimasukkan ulang secara manual ke dalam alat penerjemah sehingga agak memakan waktu jika kita melihat sebuah teks dan ingin hasil terjemahan yang lebih cepat, misalnya ketika sedang berada dalam tempat umum, akan lebih baik tentunya jika kita tidak terlalu berlama-lama diam dalam satu tempat hanya untuk mengoperasikan alat penerjemah.

Untuk mengatasi masalah tersebut, dikembangkan sebuah aplikasi penerjemah teks dengan metode masukan yang mudah digunakan, yaitu dengan menggunakan kamera pada perangkat bergerak dengan sistem operasi Android, dimana teks yang ingin diterjemahkan dimasukkan melalui kamera dan hasil translasi akan ditampilkan menggunakan teknologi realitas tertambah (*augmented reality*) langsung pada gambar teks yang didapatkan melalui kamera. Pengguna memilih tulisan yang ingin diterjemahkan dengan cara menyentuh gambar teks yang ingin diterjemahkan pada layar perangkat bergerak. Gambar tulisan akan diambil dari gambar kamera dengan membentuk batas-batas dari gambar teks yang dipilih berdasarkan masukan pengguna. Nantinya akan dilakukan ekstraksi data teks dari gambar tersebut menggunakan metode Optical Character Recognition (OCR). Setelah itu teks akan diterjemahkan dan hasil terjemahan tersebut akan diubah menjadi sebuah gambar yang diletakan di layar dalam bentuk realitas tertambah. Penerjemahan yang dapat dilakukan oleh aplikasi ini adalah penerjemahan dari bahasa Inggris ke bahasa Indonesia.



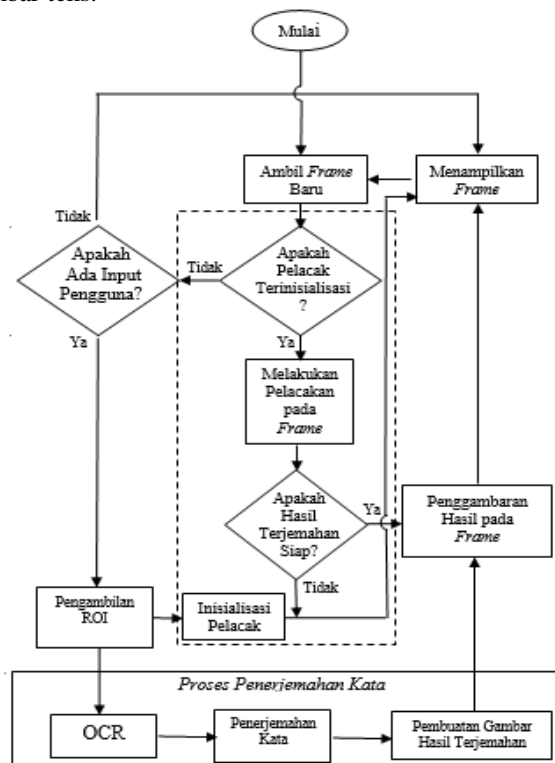
Gambar 1. Diagram alur interaksi pengguna dengan aplikasi.

Diharapkan aplikasi teks berbasis realitas tertambah yang dibangun ini menjadi sebuah aplikasi penerjemah yang mudah digunakan dan dapat memberikan hasil terjemahan yang cepat serta memiliki keakuratan translasi yang baik, sehingga nantinya dapat digunakan untuk masalah-masalah dalam kehidupan nyata.

II. PENELITIAN YANG SUDAH PERNAH DILAKUKAN

Realitas tertambah sudah banyak diaplikasikan ke dalam kehidupan sehari-hari. Pada [4], teknologi realitas tertambah digunakan untuk memberikan sebuah alternatif pengalaman bermain yang baru, dengan menggunakan teknologi realitas tertambah berbasis lokasi yang menjadikan posisi dan lingkungan pada dunia nyata sebagai bagian dari lingkungan permainan. Pada [5], teknologi realitas tertambah digunakan sebagai media untuk mengobati fobia pada serangga.

Penelitian mengenai penggunaan teknologi realitas tertambah pada aplikasi penerjemah gambar teks sudah pernah dilakukan pada [6]. Aplikasi menerima masukan dari pengguna berupa sentuhan pada gambar kata yang ingin diterjemahkan pada tampilan kamera. Hasil terjemahan ditampilkan pada posisi gambar yang dipilih dengan warna dibuat menyesuaikan warna dari gambar teks yang dipilih beserta latar belakangnya. Aplikasi mengambil gambar teks berdasarkan hasil pengenalan tepi menggunakan deteksi gradien gambar. Aplikasi menggunakan algoritma ESM sebagai algoritma pelacakan *region of interest* (ROI) yang dipilih untuk menentukan posisi penggambaran hasil terjemahan. Aplikasi tersebut berjalan pada perangkat bergerak Nokia N900 dengan sistem operasi Maemo. Pustaka yang digunakan adalah OpenCV, libCVD, dan GStreamer. Resolusi *frame* yang ditangkap dari kamera perangkat sebesar 320x240 piksel. Pengujian dilakukan dengan menggunakan dataset ICDAR 2003 dan video klip yang mengandung gambar teks.



Gambar 2. Diagram alur arsitektur aplikasi.

Penelitian yang dibahas pada artikel ini mengadopsi aplikasi yang dibahas pada [6], dengan melakukan beberapa perubahan. Aplikasi pada penelitian ini berjalan pada sistem operasi Android dengan menggunakan resolusi *frame* masukan yang lebih besar, yaitu 1920x1080 piksel. Pada proses pengambilan ROI, akan ditambahkan dua proses, yaitu pencarian jarak antar huruf pada gambar kata yang dipilih serta melakukan penghitungan ulang batas-batas vertikal gambar kata setelah penghitungan batas-batas horizontal. Kedua hal tersebut dilakukan untuk menangani jarak antar huruf yang dapat berbeda-beda pada setiap gambar teks, dan untuk memastikan agar setiap huruf pada kata yang dipilih masuk ke dalam wilayah ROI. Pada proses pelacakan, akan digunakan algoritma yang berbeda, yaitu menggunakan sebuah metode pelacakan berbasis pencocokan histogram HSV. Algoritma ini dipilih karena sifatnya yang sederhana untuk menyesuaikan dengan lingkungan perangkat aplikasi, agar aplikasi tidak menjadi terlalu lambat untuk digunakan. Selain itu, aplikasi ini dirancang untuk digunakan dalam keadaan perangkat relatif diam, sehingga tidak dibutuhkan algoritma yang terlalu kompleks. Pada [7], histogram HSV digunakan sebagai fitur warna dalam sistem klasifikasi motif batik dan memberikan hasil yang baik. Penggambaran hasil penerjemahan tidak berada tepat di atas gambar kata yang dipilih, namun berada diatas gambar kata tersebut, disertai persegi yang menandai gambar kata yang dipilih. Hal tersebut dilakukan agar pengguna dapat melihat gambar kata yang diterjemahkan beserta hasilnya pada waktu yang bersamaan, mempertimbangkan aspek ingatan jangka pendek (*short-term memory*) dari pengguna, sehingga pengguna tidak perlu mengingat-ingat kata yang dipilih.

III. ANALISIS DAN PERANCANGAN

A. Analisis Sistem

Aplikasi penerjemah gambar teks yang dibuat menggunakan teknologi realitas tertambah. Teknologi realitas tertambah dipilih agar masukan gambar yang mengandung kata yang ingin diterjemahkan dapat diambil secara langsung (*real time*) melalui kamera perangkat bergerak pengguna. Pengambilan gambar masukan melalui kamera bertujuan untuk mengurangi waktu pengguna dalam memasukan kata yang ingin diterjemahkan dibandingkan dengan cara masukan lain seperti mengetikan kata.

Cara pengguna untuk berinteraksi dengan aplikasi adalah dengan memilih kata yang diinginkan pada gambar dari kamera perangkat yang ditampilkan di layar perangkat pengguna. Pemilihan kata dilakukan dengan cara melakukan sentuhan pada gambar kata yang ingin diterjemahkan oleh pengguna.

Hasil terjemahan yang digambarkan akan berupa persegi panjang yang menandai wilayah dari gambar kata yang dipilih. Hasil penerjemahan kata akan berupa gambar persegi panjang berisikan kata hasil penerjemahan dalam bahasa Indonesia. Gambar tersebut diletakan di atas persegi penanda wilayah gambar teks yang dipilih. Gambar 1 menunjukkan alur interaksi pengguna saat menggunakan aplikasi.

B. Perancangan Arsitektur Aplikasi

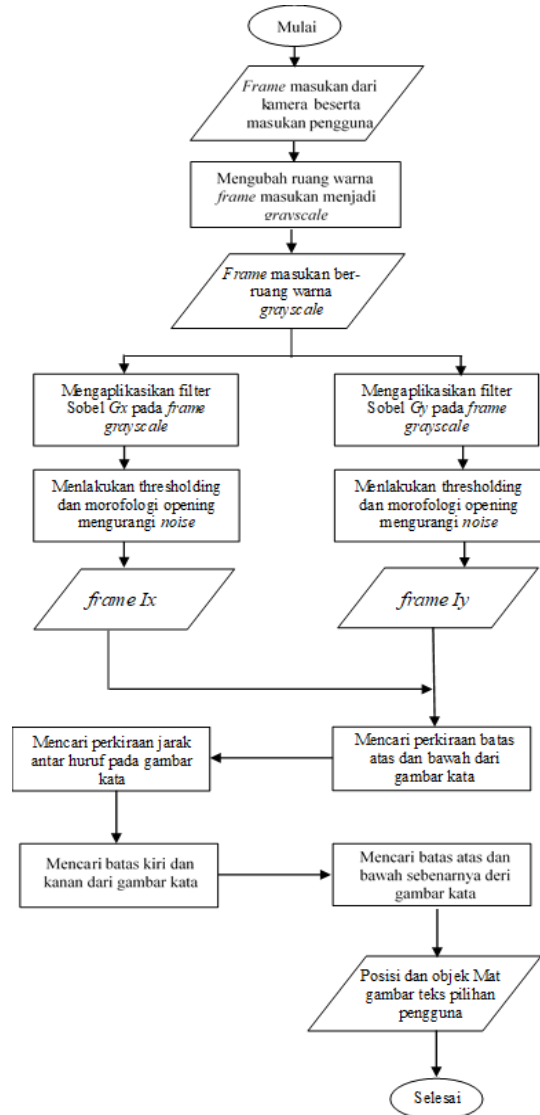
Gambar 2 menunjukkan diagram alur dari arsitektur aplikasi secara keseluruhan. Arsitektur aplikasi dapat dibagi menjadi tiga bagian utama, yaitu pengambilan gambar kata yang akan diterjemahkan sebagai *region of interest* (ROI),

penerjemahan kata, dan penambahan tampilan kamera perangkat dengan hasil terjemahan.

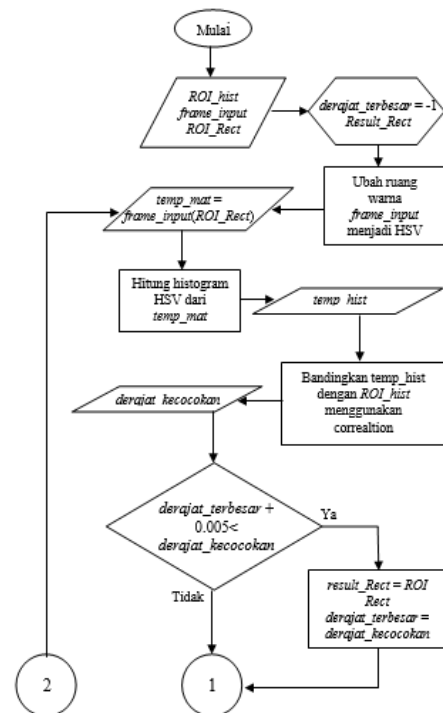
Pengambilan ROI merupakan tahap untuk mencari batas-batas dari gambar kata yang dipilih. Gambar 3 menunjukkan alur keseluruhan dari proses pengambilan ROI. Masukan dari tahap ini adalah hasil deteksi tepian *frame* masukan dari kamera perangkat menggunakan filter Sobel G_x dan G_y [8], sehingga didapatkan tepian vertikal dan horizontal. Selain itu juga dilakukan *thresholding* dan morfologi *opening* untuk mengurangi *noise*. Proses deteksi wilayah ROI terbagi menjadi tiga bagian, yaitu pencarian batas vertikal (batas atas dan bawah), pencarian jarak antar huruf pada gambar kata yang dipilih, dan pencarian batas horizontal (batas kiri dan kanan). Pencarian batas-batas vertikal dilakukan dengan menggunakan hasil deteksi tepian vertikal. Pada posisi sentuhan pengguna, dibuat sebuah garis imajiner dengan panjang 200 piksel. Pada posisi setiap piksel dari garis tersebut, dicari apakah terdapat tepian vertikal. Apabila ya, garis akan digeser sebanyak 2 piksel ke arah sumbu Y negatif untuk batas atas dan sumbu Y positif untuk batas bawah. Panjang garis imajiner dan besar pergeseran ditentukan berdasarkan eksperimen. Pergeseran dihentikan ketika sudah tidak terdeteksi tepian pada setiap piksel garis imajiner tersebut. Batas atas dan bawah diambil dari posisi koordinat sumbu Y dari garis tersebut. Kemudian dilakukan pencarian jarak antar huruf pada gambar teks yang dipilih. Pencarian batas-batas horizontal dilakukan dengan menggunakan prinsip yang sama, tetapi menggunakan hasil deteksi tepian vertikal sebagai masukannya dan garis imajiner yang digunakan berupa garis vertikal sepanjang tinggi gambar kata yang didapatkan dari pencarian batas-batas vertikal. Garis digeser secara horizontal ke arah sumbu X positif dan negatif sebanyak 1 piksel. Besar pergeseran didapatkan dari hasil eksperimen. Untuk menangani ruang kosong antar huruf yang ditemui saat pencarian batas-batas horizontal, sebelumnya dilakukan pencarian jarak antar huruf. Pencarian jarak antar huruf dilakukan dengan menggunakan prinsip yang sama dengan pencarian batas-batas horizontal, namun keluarannya adalah lebar ruang kosong antara tepian horizontal dari dua huruf pada gambar kata yang dipilih. Saat pencarian batas-batas horizontal, garis imajiner akan digeser sesuai dengan jarak antar huruf ketika menemui ruang kosong. Jika ruang kosong melebihi jarak antar huruf, maka disimpulkan ruang kosong tersebut sudah berada diluar gambar kata dan pencarian dihentikan. Setelah batas-batas horizontal didapatkan, dilakukan pencarian ulang tinggi kata. Pencarian ulang dilakukan dengan cara yang sama dengan pencarian tinggi kata pada proses sebelumnya, tetapi panjang garis imajiner yang digunakan sesuai dengan lebar gambar kata. Dari proses ini didapatkan tinggi kata yang sebenarnya. Hasil dari proses pencarian ROI adalah koordinat-koordinat batas-batas ROI dan gambar ROI itu sendiri.

Proses penerjemahan kata terbagi menjadi tiga bagian, yaitu pengenalan kata, penerjemahan, dan penggambaran hasil terjemahan. Pengenalan kata pada ROI dilakukan dengan menggunakan pustaka OCR Tesseract versi Android bernama tess-two [9]. Kata bahasa Inggris hasil pengenalan akan dikirimkan ke penerjemah. Penerjemah yang digunakan adalah

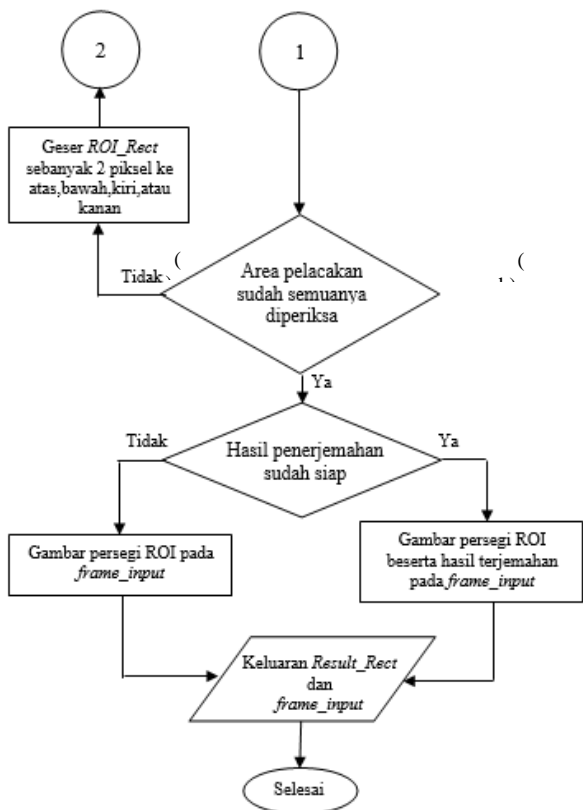
Yandex. Translate API [10]. Kemudian aplikasi akan membuat gambar yang berisikan hasil penerjemahan dari bagian penerjemah. Gambar hasil penerjemahan yang dibuat berupa objek Mat OpenCV.



Gambar 3. Diagram alur pengambilan gambar kata sebagai ROI.



Gambar 4. Diagram alur proses pelacakan ROI (1).



Gambar 5. Diagram alur proses pelacakan ROI (2).

Proses penambahan hasil penerjemahan pada tampilan kamera aplikasi dilakukan dengan terlebih dahulu mendapatkan posisi ROI pada setiap *frame* yang didapatkan dari kamera perangkat. Pencarian posisi ROI dilakukan dengan menggunakan pelacak. Gambar 4 dan 5 menunjukkan alur dari pelacak yang digunakan. Pelacak yang dibangun menggunakan histogram HSV dari gambar ROI sebagai masukan. Pada *frame* masukan, diambil potongan gambar seluas wilayah ROI pada posisi ROI. Potongan gambar ini merupakan kandidat hasil pelacakan ROI. Kemudian akan dibuat histogram HSV dari potongan gambar tersebut. Histogram HSV ROI dan potongan gambar yang dibuat memiliki jumlah bin 8,3,3 untuk channel H, S, dan V. Histogram HSV dari potongan gambar tersebut akan dicocokkan dengan histogram HSV ROI menggunakan rumus Histogram Correlation dari OpenCV [11], yang akan menghasilkan nilai derajat kecocokan. Rumus Histogram Correlation memiliki pendefinisian sebagai berikut:

$$d(H_1, H_2) = \frac{\sum_i (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_i (H_1(I) - \bar{H}_1)^2 \sum_i (H_2(I) - \bar{H}_2)^2}} \quad (1)$$

$$\bar{H}_k = \frac{1}{N} \sum_j H_k(J) \quad (2)$$



Gambar 6. Contoh gambar kata sebelum (a) dan sesudah (b). Dalam gambar di atas, kata yang dipilih adalah “BACKUP”.

Pada rumus (1) dan (2), H_1 dan H_2 masing-masing adalah histogram pertama dan kedua, k adalah nomor urut histogram, I dan J adalah nomor urut bin histogram, dan N adalah jumlah bin histogram. Keluaran dari rumus ini berupa nilai desimal dengan nilai 1 sebagai nilai maksimal (histogram sama). Nilai tersebut akan disimpan dan dibandingkan dengan hasil derajat kecocokan histogram HSV ROI dengan histogram kandidat pelacakan lainnya. Kandidat pelacakan lainnya merupakan potongan-potongan *frame* dalam jangkauan 50 piksel dari posisi ROI ke arah atas, bawah, kiri, dan kanan. Setiap kandidat berjarak 2 piksel antara satu sama lain. Kandidat pelacakan ROI dengan derajat kecocokan tertinggi akan diambil sebagai hasil pelacakan dan akan digunakan pada proses pelacakan selanjutnya. Derajat kecocokan yang dianggap paling tinggi adalah derajat dengan nilai yang lebih tinggi dengan derajat tertinggi sebelumnya dengan selisih 0.005. Pada posisi hasil pelacakan, hasil penerjemahan akan digambar apabila telah siap. Jumlah bin histogram, jangkauan pelacakan, jarak antar kandidat, serta nilai selisih antara derajat kecocokan tertinggi ditentukan berdasarkan eksperimen dengan mempertimbangkan performa aplikasi dan kualitas pelacakan. Gambar 6 menunjukkan contoh penggambaran hasil penerjemahan.

IV. UJI COBA DAN EVALUASI

Uji coba yang dilakukan terhadap aplikasi akan terbagi menjadi dua, yaitu uji coba kotak hitam untuk mengetahui apakah setiap komponen aplikasi telah berjalan dengan baik atau tidak, dan uji coba terhadap pengguna untuk mengetahui tingkat kepuasan pengguna terhadap aplikasi berdasarkan aspek-aspek pengujian yang ditetapkan.

Perangkat uji yang juga digunakan dalam pengembangan aplikasi adalah perangkat *smartphone* Asus Zenfone 2 Laser ZE500KL.

A. Uji Coba Kotak Hitam

Komponen-komponen aplikasi yang diujicoba dalam uji coba kotak hitam adalah pengambilan ROI, pengenalan kata, dengan OCR, penerjemahan kata, penggambaran hasil, dan pelacakan posisi ROI.

Data uji akan terbagi menjadi dua jenis, yaitu data uji buatan dan data uji objek nyata. Pembagian data uji dilakukan untuk menguji kemampuan aplikasi dalam menangani keadaan uji dan keadaan dunia nyata sehari-hari.

Data uji buatan akan dicetak pada kertas HVS ukuran A4. Faktor-faktor pengujian yang dipilih digunakan dalam data uji adalah jumlah kata, bentuk karakter, dan warna. Faktor-faktor tersebut dipilih untuk mengetahui efeknya pada setiap komponen aplikasi. Untuk jumlah kata, data uji buatan dibagi menjadi dua, yaitu satuan dan banyak. Untuk kata satuan, kata yang digunakan adalah “Copyrights”. Untuk kata banyak, digunakan kata-kata “One”, “Two”, “Three”, “Four”, “Five”, “Six”, “Seven”, “Eight”, dan “Nine”. Data uji kata banyak disusun menjadi tiga kelompok, yaitu “One Two Three”, “Four Five Six”, dan “Seven Eight Nine”. Kelompok-kelompok tersebut dilatakan berurutan dari atas ke bawah.

Pada data uji kata satuan dan kata banyak, warna yang digunakan adalah hitam di atas latar belakang putih, merah di atas latar belakang putih, dan warna kuning di atas latar belakang biru. Untuk data uji kata satuan, jenis *font* yang digunakan adalah New Courier dan Consolas untuk jenis *font* dengan luas wilayah karakter yang sama (*fixed-pitch*) serta

Arial Black dan Times New Roman untuk jenis *font* dengan luas wilayah karakter yang tidak sama (*non fixed-pitch*). Pada data uji kata banyak, jenis *font* yang digunakan hanya Arial Black. Ukuran huruf yang digunakan adalah 28 *point* dan 48 *point*.

Tabel 1.
Hasil Pengujian pada Data Uji Buatan Kata Satuan

Faktor Pengujian	Nilai Rata-Rata Pengujian					
	(1)	(2)	(3)	(4)	(5)	
Jenis <i>Font</i>	Consolas New	0.83	0.83	0.83	0.83	0.79
	Courier	1	1	1	1	0.62
	Arial Black	0.83	0.83	0.83	1	0.83
	Times New Roman	1	1	1	1	0.41
	28 <i>point</i>	1	1	1	0.92	0.56
Ukuran	48 <i>point</i>	0.83	0.83	0.83	1	0.79
	Hitam di atas putih	1	1	1	0.87	0.57
Warna	Merah di atas putih	0.87	0.87	0.87	1	0.75
	Kuning di atas biru	0.87	0.87	0.87	1	0.72

Keterangan: (1): pengambilan ROI, (2): pengenalan kata dengan OCR, (3): penggambaran hasil penerjemahan (4): pelacakan ROI (keadaan diam), (5): pelacakan ROI (keadaan bergerak)

Tabel 2.
Hasil Pengujian pada Data Uji Buatan Kata Banyak

Faktor Pengujian	Nilai Rata-Rata Pengujian					
	(1)	(2)	(3)	(4)	(5)	
Kata	One	1	1	1	0.83	0.75
	Two	0.33	0.33	0.33	0.67	0.5
	Three	0.5	0.5	0.5	1	0.7
	Four	0.83	0.83	0.83	0.67	0.5
	Five	0.66	0.66	0.66	0.83	0.42
	Six	1	1	1	0.83	0.62
	Seven	0.66	0.66	0.66	0.83	0.58
	Eight	1	1	1	0.83	0.62
	Nine	1	0.67	0.67	0.83	0.62
Ukuran	28 <i>point</i>	0.63	0.59	0.59	0.74	0.44
	48 <i>point</i>	0.93	0.89	0.89	0.89	0.75
Warna	Hitam di atas putih	0.72	0.72	0.72	0.61	0.43
	Merah di atas putih	0.89	0.89	0.89	0.94	0.63
	Kuning di atas biru	0.72	0.61	0.61	0.89	0.71

Keterangan: (1): pengambilan ROI, (2): pengenalan kata dengan OCR, (3): penggambaran hasil penerjemahan (4): pelacakan ROI (keadaan diam), (5): pelacakan ROI (keadaan bergerak)

Tabel 3.
Hasil Pengujian pada Data Uji Objek Nyata

Pengujian	Nilai Rata-Rata
Pengambilan area ROI	0.87
Pengenalan kata dengan OCR	0.87
Penggambaran hasil penerjemahan	0.87
Pelacakan ROI (keadaan diam)	0.5
Pelacakan ROI (keadaan bergerak)	0.31

Untuk data uji yang menggunakan objek nyata, digunakan sisi depan dari kardus kemasan produk “Seagate Backup Plus Slim 2 TB”. Data uji yang digunakan adalah kata-kata dalam Bahasa Inggris diluar nama merek dan singkatan. Kata-kata tersebut adalah: “BACKUP”, “PLUS”, “Slim”, “Portable”, “Storage”, “Backup”, “made”, dan “easy”.

Pada pengujian pengambilan ROI, pengenalan kata dengan OCR, penerjemahan kata, penggambaran hasil, dan pelacakan ROI pada keadaan diam, perangkat uji diletakan pada tripod dan diletakan sejauh 25 sentimeter pada ruangan berpencahayaan baik. Sentuhan dilakukan pada huruf di bagian tengah masing-masing gambar kata. Hasil pengujian diberi nilai 1 jika berhasil dan nilai 0 jika gagal.

Pada pengujian pengambilan ROI, dilihat apakah persegi ROI melingkupi kata yang dipilih dengan baik. Dari hasil pengujian pada Tabel 1, 2,dan 3, terlihat bahwa hasil pengujian cukup baik, namun terdapat beberapa kata dalam data uji buatan kata banyak dengan nilai yang kecil, yaitu “Two” dan “Three”. Berdasarkan hasil analisa data deteksi tepi, terdapat huruf-huruf yang tepiannya saling berimpitan

atau saling tumpang tindih antara yang satu dengan yang lain. Hal ini menyebabkan jarak antar huruf menjadi sulit untuk ditemukan. Untuk mengatasi hal ini, diperlukan penanganan yang lebih baik terhadap data deteksi tepi, salah satunya dengan melakukan *preprocessing* yang lebih baik terhadap data deteksi tepi atau *frame* masukan.

Pada pengujian pengenalan kata dengan OCR, dilihat apakah OCR mampu memberikan hasil pengenalan kata yang baik. Pengujian OCR dilakukan dalam satu rangkaian dengan pengujian pengambilan area ROI. Apabila pada saat pengujian ROI area ROI tidak bisa didapatkan, maka hasil OCR dianggap tidak ada. Dari hasil pengujian pada Tabel 1, 2,dan 3, terlihat bahwa hasil pengujian cukup baik. Untuk setiap pengambilan ROI yang sukses, pengenalan kata mampu berjalan dengan baik. Namun terdapat kegagalan pada data uji kata banyak pada kata “Nine”. OCR tidak mampu memberikan hasil pengenalan sama sekali untuk kata tersebut.

Pada pengujian penerjemahan kata, dilihat apakah penerjemah mampu menerjemahkan kata bahasa Inggris ke bahasa Indonesia dengan baik. Masukan dari pengujian ini adalah hasil pengenalan OCR dari setiap kata pada data uji yang dapat dengan sukses dikenali dan sesuai dengan gambar kata. Dari hasil pengujian pada Tabel 4, terlihat bahwa kata-kata pada data uji buatan mampu diterjemahkan ke dalam bahasa Indonesia dengan baik. Pada data uji objek nyata, terdapat kata-kata yang tidak mampu diterjemahkan oleh penerjemah.

Pada pengujian penggambaran hasil penerjemahan, dilihat apakah penggambaran hasil penerjemahan berjalan dengan baik atau tidak. Pengujian penggambaran hasil penerjemahan dilakukan dalam satu rangkaian dengan pengujian pengambilan area ROI dan pengenalan kata dengan OCR.

Apabila pada suatu kasus pengujian ROI area ROI tidak bisa didapatkan atau OCR gagal mengenali gambar kata, maka kasus pengujian tersebut dianggap gagal. Dari hasil pengujian pada Tabel 1, 2,dan 3, terlihat bahwa untuk pengambilan ROI dan pengenalan kata dengan OCR yang sukses, penggambaran hasil penerjemahan mampu berjalan dengan baik.

Pada pengujian pelacakan ROI pada keadaan diam, dilihat apakah pelacak mampu mendapatkan posisi ROI dengan baik dalam keadaan diam. Sentuhan dilakukan pada bagian tengah masing-masing gambar kata. Apabila setelah melakukan sentuhan ROI agal terambil, sentuhan diulangi hingga ROI terambil dengan baik dan sesuai dengan area gambar kata yang diuji. Pada setiap kata yang diuji, hasil uji dikatakan sukses apabila lebih dari 50 persen gambar kata berada pada persegi ROI selama 2 detik setelah hasil penerjemahan tergambar pada layar. Dari hasil pengujian pada Tabel 1, 2, dan 3, terlihat bahwa hasil uji pada data uji kata buatan cukup baik. Pada data uji objek nyata, nilai rata-rata hasil pengujian rendah. Kesalahan terjadi pada kata berwarna hitam dengan latar belakang putih. Dari hasil analisa histogram HSV ROI dengan histogram HSV potongan hasil pelacakan yang salah, terlihat bahwa nilai derajat kecocokan yang didapatkan tinggi. Hal ini menunjukkan bahwa histogram tidak mencerminkan perbedaan yang signifikan antara daerah gambar dengan piksel berwarna gelap kehitaman (gambar kata) dengan daerah dengan piksel yang lebih terang (gambar latar belakang). Untuk mengatasi hal ini, diperlukan bin histogram yang lebih banyak pada *channel* Value, yang mengatur tingkat kecerahan warna, sehingga aplikasi dapat

dengan lebih baik membedakan piksel berwarna gelap menuju hitam, berwarna terang menuju putih, dan diantaranya.

Pada pengujian pelacakan ROI pada keadaan bergerak, dilihat apakah pelacak mampu mendapatkan posisi ROI dengan baik dalam keadaan bergerak. Prosedur pengambilan masukan sama dengan pengujian pelacakan dalam keadaan diam, namun pada pengujian ini, perangkat dipegang di tangan dengan jarak sejauh 25 sentimeter dari data uji. Perangkat digerakan ke atas, bawah, kiri, dan kanan sejauh 1.5 sentimeter. Pada setiap kata yang diuji, hasil uji dikatakan sukses apabila lebih dari 50 persen gambar kata berada padapersegi ROI. Dari hasil pengujian pada Tabel 1, 2, dan 3, terlihat bahwa pada hasil pengujian menggunakan data uji buatan, terlihat bahwa ukuran kata 28 *point* memiliki nilai pengujian rata-rata yang lebih rendah dibandingkan dengan nilai rata-rata dari ukuran kata 48 *point*. Hal ini dapat disebabkan oleh jangkauan pelacakan yang ditetapkan sebanyak 50 piksel ke atas, bawah, kiri, dan kanan. Terdapat perbedaan antara pergeseran sebanyak 50 piksel pada ROI yang lebih besar dengan ROI yang lebih kecil. Jangkauan pelacakan relatif terhadap ROI lebih besar pada ukuran teks yang lebih besar dibandingkan dengan ukuran teks yang lebih kecil. Selain itu, masalah derajat kecocokan histogram HSV yang ditemui pada pengujian pelacakan pada kondisi diam juga ditemui disini, namun kesalahan juga terjadi pada warna selain teks hitam di atas latar belakang putih. Gambar kata yang berbeda dapat memiliki derajat kecocokan histogram yang tinggi.

Tabel 4.
Hasil Pengujian Penerjemahan Kata

Data Uji	Kata	Hasil Penerjemahan	Nilai
Buatan	Copyrights	Hak Cipta	1
	One	Satu	1
	Two	Dua	1
	Three	Tiga	1
	Four	Empat	1
	Five	Lima	1
	Six	Enam	1
	Seven	Tujuh	1
	Eight	Delapan	1
	Nine	Sembilan	1
Objek Nyata	BACKUP	CADANGAN	1
	PLUS	PLUS	0
	Slim	-	0
	Portable	Portable	0
	Storage	Penyimpanan	1
	Backup	Cadangan	1
	made	dibuat	1
	easy	mudah	1

A. Uji Coba pada Pengguna

Uji coba pada pengguna dilakukan menggunakan perangkat uji pada ruangan berpencahayaan baik. Uji coba dilakukan pada lima orang pengguna. Pengguna diminta untuk mengoperasikan aplikasi untuk menerjemahkan kata-kata pada data uji. Untuk uji coba ini, data uji yang digunakan adalah benda sehari-hari, yaitu papan penanda, kemasan produk, dan surat kabar. Setelah mencoba aplikasi, pengguna diminta untuk mengisi kuesioner untuk memberikan penilaian terhadap indikator dari masing-masing aspek

pengujian. Nilai yang diberikan berada pada rentang nilai 1 untuk sangat tidak puas dan 6 untuk sangat puas. Aspek pengujian yang dinilai ada tiga macam, yaitu kenyamanan aplikasi, performa aplikasi dan kemudahan penggunaan aplikasi.

Tabel 5 menunjukkan nilai rata-rata dari setiap indikator dan aspek pengujian. Hasil pengujian menunjukkan bahwa pengguna merasa puas untuk setiap aspek pengujian aplikasi.

Tabel 5.
Hasil Pengujian Aplikasi pada Pengguna

Aspek	Indikator	Rata-Rata Nilai
Kenyamanan Aplikasi	Kenyamanan tampilan antar muka	4.80
	Kenyamanan menggunakan sentuhan sebagai metode masukan	5.20
	Kenyamanan melihat hasil terjemahan	4.60
Rata-Rata Pembulatan		4.87 5
Performa Aplikasi	Performa Aplikasi	4.80
	Kecepatan Mendapatkan Hasil Terjemahan	5.20
Rata-Rata Pembulatan		5.00 5
Kemudahan Penggunaan Aplikasi	Kemudahan penggunaan saat uji coba	4.80
	Kemudahan penggunaan untuk kebutuhan sehari-hari	4.80
Rata-Rata Pembulatan		4.80 5

V. KESIMPULAN DAN SARAN

Pada artikel ini, dibahas perancangan dan pengujian dari aplikasi aplikasi penerjemah gambar teks berbahasa Inggris menggunakan teknologi realitas tertambah pada perangkat berbasis Android.

Proses pengambilan ROI dilakukan dengan mendeteksi wilayah kata yang dipilih menggunakan hasil deteksi tepi dengan filter Sobel. Hasil pengujian menunjukkan nilai yang baik, dengan adanya masalah dalam menangani ruang antar huruf dan diperlukannya *preprocessing* untuk mengolah data tepian yang tidak baik.

Pengenalan teks dilakukan dengan bantuan pustaka Tesseract OCR, sementara penerjemahan teks dari bahasa Inggris ke bahasa Indonesia dilakukan dengan kakas hitam Yandex. Translate API. Hasil uji coba kotak hitam menunjukkan nilai yang baik untuk pengenalan kata dengan OCR. Untuk perjemahan teks, masih terdapat bahasa dalam bahasa Inggris yang tidak mampu diterjemahkan ke bahasa Indonesia. Untuk pengembangan ke depannya, dapat dicari kemungkinan penggunaan penerjemah dengan kosakata bahasa Inggris dan Indonesia yang lebih lengkap

Penambahan grafis pada *frame* masukan dari kamera dengan hasil terjemahan dilakukan dengan membuat sebuah gambar menggunakan objek Mat OpenCV yang berisi hasil terjemahan. Hasil terjemahan digambar pada *frame* masukan di atas gambar teks yang diterjemahkan berdasarkan posisi gambar teks yang didapatkan dari proses pelacakan. Proses pelacakan ROI dilakukan dengan membandingkan histogram HSV dari ROI dengan histogram HSV dari daerah pelacakan. Pelacakan mencari tingkat kemiripan histogram yang paling tinggi dengan menggunakan rumus Histogram Correlation.

Hasil uji coba kotak hitam menunjukkan nilai yang rendah saat pelacak digunakan dalam kondisi bergerak dikarenakan faktor jangkauan pelacakan dan histogram HSV. Pengembangan selanjutnya dapat mencoba algoritma pelacak lainnya yang dapat berjalan dengan baik pada lingkungan perangkat bergerak Android serta cocok untuk pelacakan posisi gambar kata.

Berdasarkan pengujian oleh pengguna, penguji merasa puas dengan kenyamanan, performa, dan kemudahan penggunaan aplikasi. Penguji memberikan saran untuk memperbaiki tingkat *frame* per detiknya. Hal ini menjadi salah satu kesulitan utama dalam pengembangan aplikasi ini. Untuk pengembangan selanjutnya, dapat digunakan perangkat Android dengan spesifikasi yang lebih baik dan memperkecil resolusi *frame* masukan sehingga jumlah data yang harus diolah menjadi lebih kecil.

DAFTAR PUSTAKA

- [1] "Google Terjemahan." [Online]. Available: <https://translate.google.co.id/?hl=id>. [Accessed: 24-Jul-2018].
- [2] "Menerjemahkan dengan ucapan - Komputer - Bantuan Google Translate." [Online]. Available: https://support.google.com/translate/answer/6142468?hl=id&ref_topic=7011659. [Accessed: 24-Jul-2018].
- [3] "Menerjemahkan dengan tulisan tangan atau keyboard virtual - Komputer - Bantuan Google Translate." [Online]. Available: https://support.google.com/translate/answer/6142469?hl=id&ref_topic=7011659. [Accessed: 24-Jul-2018].
- [4] R. R. H. D. Herumurti, I. Kuswardayan, A. Yuniarti, W. N. Khotimah, and N. B. Fauzan, "Location Based Augmented Reality Game Using Kudan SDK," *Inf. Commun. Technol. Syst. (ICTS), 2017 11th Int. Conf.*, pp. 307–310, 2017.
- [5] F. Fatharany, R. R. Hariadi, D. Herumurti, and A. Yuniarti, "Augmented reality application for cockroach phobia therapy using everyday objects as marker substitute," in *Proceedings of 2016 International Conference on Information and Communication Technology and Systems, ICTS 2016*, 2017.
- [6] V. Frago, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk, "TranslatAR: A mobile augmented reality translator," in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 497–502.
- [7] N. Suciati, A. Kridanto, M. F. Naufal, M. Machmud, and A. Y. Wicaksono, "Fast discrete curvelet transform and HSV color features for batik image classification," in *2015 International Conference on Information & Communication Technology and Systems (ICTS)*, 2015, pp. 99–104.
- [8] "Feature Detectors - Sobel Edge Detector." [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>. [Accessed: 25-Jul-2018].
- [9] "GitHub - rmttheis/tess-two: Fork of Tesseract Tools for Android." [Online]. Available: <https://github.com/rmttheis/tess-two>. [Accessed: 25-Jul-2018].
- [10] "Translate API — Yandex Technologies." [Online]. Available: <https://tech.yandex.com/translate/>. [Accessed: 25-Jul-2018].
- [11] "Histogram Comparison — OpenCV 2.4.13.4 documentation." [Online]. Available: https://docs.opencv.org/2.4.13.4/doc/tutorials/imgproc/histograms/histogram_comparison/histogram_comparison.html. [Accessed: 25-Jul-2018].