

PERANCANGAN DAN IMPLEMENTASI *HISTOGRAM OF ORIENTED GRADIENTS* DAN *K-NEAREST NEIGHBOUR* UNTUK DETEKSI HURUF HIRAGANA JEPANG PADA APLIKASI *MOBILE PENERJEMAH KATA* DALAM BAHASA JEPANG KE BAHASA INDONESIA BERBASIS ANDROID

Nana Febriana¹, Budhi Irawan,S.Si.,M.T.²,Umar Ali Ahmad,S.T.,M.T.³

^{1,2,3}Sistem Komputer, Fakultas Teknik Elektro – Univesitas Telkom Jln.
Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia
febrian.nana@gmail.com¹, bir@telkomuniversity.ac.id², uaa@telkomuniversity.ac.id³

ABSTRAK

Pada era global ini menguasai bahasa lain merupakan salah satu kebutuhan penting yang harus dimiliki setiap orang. Banyak orang berkunjung ke negara lain untuk melakukan banyak kegiatan seperti bekerja, belajar bahkan untuk liburan. Salah satu negara yang banyak dikunjungi adalah negara Jepang. Negara Jepang memiliki bentuk huruf yang berbeda dengan huruf latin pada umumnya. Untuk mempelajari Bahasa Jepang tersebut dibutuhkan pemahaman dengan huruf-hurufnya. Seiring dengan berkembangnya teknologi, pengenalan karakter atau *Optical Character Recognition* (OCR) merupakan salah satu teknologi pada bidang pengenalan karakter atau pola dan kecerdasan buatan sebagai mesin pembaca. Pada penelitian tugas akhir ini dirancang sebuah aplikasi penerjemah kata dalam Bahasa Jepang ke Bahasa Indonesia pada *smartphone* berbasis Android dengan prinsip dasar *Optical Character Recognition* (OCR) untuk mengenali kata Bahasa Jepang yang tertangkap oleh kamera *smartphone*. Pada penelitian ini digunakan metode *Histogram of Oriented Gradients* (HOG) sebagai metode untuk mengekstrak setiap ciri masing-masing huruf jepang yang menjadi masukan. Kemudian ciri tersebut akan dicocokkan dengan data latih huruf hiragana. Dan akhirnya, aplikasi menampilkan terjemahan dari kata masukan kepada pengguna. Pengujian yang dilakukan memberikan hasil parameter terbaik untuk aplikasi yaitu 82.85 % dengan resolusi Resolusi 3264x2448 atau 8 MegaPiksel untuk 2 silabel sedangkan untuk 3 silabel adalah 76.67 %. Pengujian juga menunjukkan bahwa kecepatan *clock* prosesor *device* berpengaruh terhadap waktu respons sistem. Hasil Pengujian Beta atas empat poin, yaitu tampilan aplikasi, waktu respons sistem, ketepatan penerjemahan, dan manfaat aplikasi menunjukkan aplikasi dapat diklasifikasikan sebagai baik dan bermanfaat.

Kata Kunci: *Optical Character Recognition, Histogram of Oriented Gradients, K-Nearest Neighbor, Android*

ABSTRACT

In this global era mastering another language is one of the important needs that must be owned by everyone. Many people visit other countries to do a lot of activities such as work, study and even for a holiday. One of the countries visited is Japan. Japanese state has a different shape with the Latin alphabet in general. To learn the Japanese language requires an understanding with the letters. Along with the development of technology, character recognition or often *Optical Character Recognition* (OCR) is one of the technologies in the field of character or pattern recognition and artificial intelligence as reading machines. In this research designed a word translator application in Japanese to Indonesian on the Android-based *smartphone* with the basic principle of *Optical Character Recognition* (OCR) to recognize the Japanese word captured by a *smartphone* camera. In this research used methods *Histogram of Oriented Gradients* (HOG) as a method to extract each of the characteristics of each letter Japanese become inputs. Then these traits will be matched with training data hiragana. And finally, the application displays the translation of the word feedback to the user. Tests conducted deliver results best parameter for applications is 82.85 % with a resolution of 3264x2448 or 8 megapixel to 2 syllables while for 3 syllables is 76.67 %. Tests also showed that the clock speed of the processor device linear effect on the system response time. Beta Testing Results on four points, namely the display application, system response time, accuracy of translation, and the benefits of the application shows the application can be classified as either

Keyword: *Optical Character Recognition, Histogram of Oriented Gradients, K-Nearest Neighbor, Android*

I. PENDAHULUAN

1.1. Latar Belakang

Pada era global ini menguasai bahasa lain merupakan salah satu kebutuhan penting yang harus dimiliki setiap orang. Banyak orang berkunjung ke negara lain untuk melakukan banyak kegiatan seperti bekerja, belajar bahkan untuk liburan. Salah satu negara yang banyak dikunjungi adalah negara Jepang. Negara Jepang memiliki

bentuk huruf yang berbeda dengan huruf latin pada umumnya. Jenis huruf dalam Bahasa Jepang ada tiga kelompok yaitu Katakana, Hiragana dan Kanji. Saat ini, android merupakan salah satu teknologi yang sangat berkembang dan banyak diminati oleh berbagai kalangan. Penelitian ini diajukan untuk meningkatkan kemampuan dalam mengenali huruf Jepang yang merupakan standar huruf dokumen yang dicetak. Penelitian masih terus dilakukan untuk meningkatkan kemampuan sistem OCR yang masih terdapat beberapa kesulitan dalam sistem OCR. Seperti, keambiguan dalam teks dan *noise* dan kesalahan klasifikasi.

Pada penelitian tugas akhir ini dirancang sebuah aplikasi penerjemah Bahasa Jepang berbasis Android. Prinsip dasar yang digunakan pada aplikasi ini adalah *Optical Character Recognition* (OCR) untuk mengenali huruf dalam Bahasa Jepang yang tertangkap oleh kamera *Smartphone*. Citra masukan diolah dan diekstraksi untuk mendapatkan ciri dari huruf dalam Bahasa Jepang menggunakan metode *Histogram of Oriented Gradients*. Sedangkan metode *K-Nearest Neighbour* digunakan pada tahap pengklasifikasian. Diharapkan dengan menggunakan kedua metode tersebut yaitu *Histogram of Oriented Gradients* dan *K-Nearest Neighbour* akan memberikan nilai akurasi yang tinggi dalam menerjemahkan kata dalam Bahasa Jepang sesuai dengan penelitian sebelumnya.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas ada beberapa masalah yang akan diselesaikan dalam penelitian tugas akhir ini, yaitu.

1. Bagaimana merancang dan membuat aplikasi penerjemah kata dalam Bahasa Jepang menggunakan metode *Histogram of Oriented Gradient* dan *K-Nearest Neighbour* berbasis *smartphone* android?
2. Bagaimana implementasi aplikasi penerjemah kata dalam bahasa Jepang berbasis *smartphone* Android?
3. Bagaimana pengujian dan analisa performansi dari aplikasi pengenalan karakter huruf dalam Bahasa Jepang berbasis *smartphone* android?

1.3. Tujuan

Adapun tujuan dilaksanakannya tugas akhir ini, yaitu.

1. Merancang dan membuat aplikasi penerjemah kata dalam Bahasa Jepang dengan menggunakan metode *Histogram of Oriented Gradients* dan *K-Nearest Neighbour* berbasis *smartphone* android.
2. Mengimplementasikan aplikasi penerjemah kata dalam Bahasa Jepang berbasis *smartphone* android.
3. Melakukan pengujian dan analisa performansi aplikasi penerjemah kata dalam Bahasa Jepang berbasis *smartphone* Android.

1.4. Batasan Masalah

Agar tidak terjadi perluasan pembahasan dalam penelitian tugas akhir ini, penulis membatasi beberapa masalah, sebagai berikut.

1. Huruf Jepang yang dideteksi adalah huruf Jepang Hiragana.
2. Panjang maksimal kata yang diterjemahkan adalah lima huruf Bahasa Jepang.
3. Bahasa Jepang diterjemahkan dalam Bahasa Indonesia
4. Proses ekstraksi ciri menggunakan *Histogram of Oriented Gradients* (HOG).
5. Proses klasifikasi menggunakan *K-Nearest Neighbor* (KNN).
6. Menggunakan sistem operasi android.
7. Sistem berbasis *mobile offline*.
8. Teks Jepang yang dideteksi harus dalam posisi mendatar.
9. Citra yang diolah adalah karakter tulisan cetak dalam format JPEG.

II. TEORI DASAR

2.1. Bahasa Jepang

Dalam bahasa Jepang terdapat tiga jenis aksara yaitu hiragana, katakana dan kanji. Huruf Hiragana dan katakana asli dari Jepang sendiri sedangkan huruf Kanji berasal dari Cina. Huruf hiragana digunakan untuk menuliskan kata asli dari bahasa Jepang (bukan kata serapan). Huruf hiragana didasarkan pada lima bunyi vokal, yaitu a, i, u, e, dan o. Vokal ini dapat digunakan sendiri ataupun digunakan bersama sebuah konsonan (misal m+a = ma) atau konsonan ditambah semi-vokal "y" (misal m+y+a = mya).

2.2. Pemrograman Android^[1]

Android adalah sebuah sistem operasi *open source* yang dapat dikembangkan oleh siapapun secara bebas. Android menyediakan *tools* dan *framework* untuk mengembangkan sebuah aplikasi bagi penggunanya secara mudah dan cepat. Selain itu, android memiliki SDK (*software development kit*) yaitu sebuah *tools* yang digunakan untuk mengembangkan aplikasi *platform* android dengan Bahasa pemrograman Java. Kode pada java dikompilasi bersama dengan data file *resource* yang dibutuhkan oleh aplikasi, dimana dalam prosesnya file *resource* tersebut

dipaket (*package*) oleh tools yang dinamakan “apt tools” ke dalam paket android sehingga menghasilkan file dengan ekstensi *.apk. File *.apk tersebut yang disebut aplikasi yang dapat di pasang (*install*) di perangkat *mobile*.

2.3. Citra Digital

Citra digital merupakan gambar dua dimensi yang terbentuk dari piksel-piksel. Secara matematis direpresentasikan dengan $f(x, y)$, dimana x dan y menyatakan koordinat posisi piksel dan nilai $f(x, y)$ menunjukkan intensitas (derajat keabuan) piksel atau *picture element* pada koordinat tersebut. Saat nilai-nilai dari x, y dan $f(x, y)$ berada pada range tertentu maka citra tersebut dinamakan dengan citra digital.

2.3.1. Citra RGB

RGB merupakan model warna yang dapat menghasilkan nilai RGB yang berbeda, karena elemen warna tersebut bervariasi. Setiap warna dasarnya memiliki rentang nilai. Misalnya, ketika dimasukkan ke monitor RGB, tiga warna tersebut dapat memiliki kedalaman 24 bit, dimana setiap warnanya memiliki 8 bit. Jumlah bit yang digunakan untuk mempresentasikan setiap piksel dalam ruang RGB disebut kedalaman piksel. Setiap pikselnya merupakan kombinasi dari ketiga elemen warna tersebut dengan tingkat warna dengan rentang nilai antara 0-255.

2.3.2. Citra Grayscale

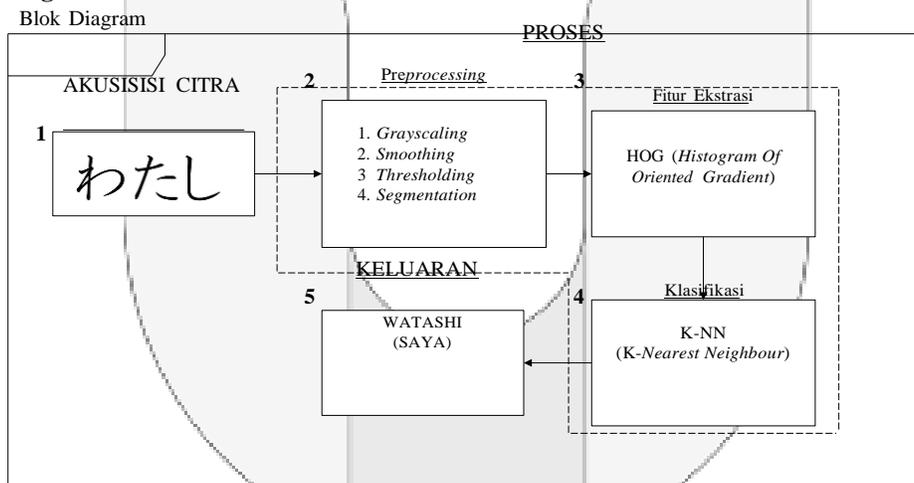
Citra grayscale merupakan suatu citra yang nilai pikselnya merepresentasikan derajat keabuan atau intensitas warna putih. Terdapat tiga cara yang umum digunakan dalam mengubah citra RGB menjadi citra *grayscale*, yaitu Metode *Lightness*, Metode *Average*, dan Metode *Luminosity*.

2.4. Optical Character Recognition

OCR adalah pengenalan dari karakter teks yang tercetak atau tertulis oleh computer. Tahap-tahap dalam proses OCR mencakup *image acquisition, image preprocessing, feature extraction, dan classification*.

III. PERANCANGAN SISTEM

3.1. Blok Diagram Sistem



3.2. Preprocessing

Tahap awal dari proses *pre-processing* adalah *grayscale*. Tahap ini mengubah citra RGB menjadi citra *grayscale*. Untuk mengubah citra RGB menjadi *grayscale* menggunakan rumus berikut, dimana nilai pada masing-masing R, G, B tersebut dari 0-255.

$$\text{Luminosity} = 0.21 R + 0.72 G + 0.07 B$$



Gambar 3.1 (a) Sebelum *Grayscale* (b) Sesudah *Grayscale*

Tahap berikutnya adalah *smoothing*. *Smoothing* bertujuan untuk menghilangkan gangguan (*noise*) yang terdapat pada citra masukan. Operasi pada proses *smoothing* menggunakan teknik konvolusi matriks.



Gambar 3.2 Hasil *Smoothing*

Tahap berikutnya adalah *thresholding*. Tujuan dari proses *thresholding* yaitu mengubah citra masukan menjadi citra hitam atau putih dengan menentukan batas ambang (*thresholding*).



Gambar 3.3 Hasil *Thresholding*

Tahap berikutnya segmentasi merupakan proses mempartisi (membagi) citra menjadi beberapa daerah atau objek dengan melakukan *cropping* terhadap citra tersebut. Pada penelitian ini tahap segmentasi menggunakan teknik *projection* yaitu mendapatkan informasi pada citra dengan menjumlahkan setiap piksel baris atau kolom secara berulang. Silabel yang didapatkan dari segmentasi akan menjadi masukan proses ekstraksi ciri.



Gambar 3.4 Hasil Segmentasi

3.3. Ekstraksi Ciri

Pada tahap ekstraksi ciri bertujuan untuk mengambil informasi penting pada setiap citra masukan. Pada proses ini terjadinya pengurangan terhadap jumlah piksel tetapi tetap mendapatkan atau mengambil informasi penting tersebut. Metode yang digunakan pada tahap ekstraksi ciri ini adalah *Histogram of Oriented*.

3.4. Klasifikasi

Algoritma KNN bertujuan untuk mengklasifikasikan objek baru berdasarkan atribut *training sample*. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori^[9]. Ada 2 tahap pada proses klasifikasi yaitu data *training* dan data *testing*.

- Data *training* terdiri dari silabel yang berasal dari citra huruf *Hiragana* dalam Bahasa Jepang yang telah melewati proses *pre-processing* dan ekstraksi ciri. Setiap ekstraksi ciri/vektor ciri dari silabel huruf *Hiragana* pada data *training* disimpan di *database* yaitu setiap ciri berukuran 105x36 piksel.
- Data *testing* adalah citra kata bahasa Jepang yang terdiri dari huruf *hiragana* Jepang. Maksimal silabel pada kata tersebut adalah 5 silabel. Citra *testing* atau data uji sebelumnya akan melalui tahap *pre-processing* dan mendapatkan setiap ekstraksi cirinya pada fitur ekstraksi. Dari setiap ciri tersebut dicocokkan atau membandingkan data dari hasil *training* dengan data *testing* dan dicari nilai terdekat berdasarkan nilai *Euclidean distance* yang diklasifikasi oleh *K-Nearest Neighbor*.

Hasil pencocokan data uji dengan data *training* akan menghasilkan keluaran yaitu terjemahan kata dalam Bahasa Indonesia.

IV. PENGUJIAN DAN ANALISIS

4.1. Pengujian Ukuran Resolusi Citra

Tujuan pengujian ini adalah untuk memastikan bahwa citra uji yang mempunyai ukuran yang berbeda dapat menampilkan hasil deteksi sesuai yang diharapkan. Pada pengujian ini, menggunakan empat ukuran resolusi citra yang umum didapatkan pada *smartphone* Android, yaitu 2MP, 4MP, 6MP, dan 8MP. Tiap resolusi diuji dengan citra masukan yang terdiri atas 35 kata 2 silabel dan 30 kata 3 silabel.

Tabel 4.1. Hasil Pengujian Ukuran Resolusi Citra

| Resolusi | Silabel | Akurasi (%) |
|-----------------------------|-----------|-------------|
| Resolusi 2048x1152 (2MP) | Silabel 2 | 54.28 |
| | Silabel 3 | 46.67 |
| Resolusi 2592x1456 (4MP) | Silabel 2 | 54.28 |
| | Silabel 3 | 53.33 |
| Resolusi 3264x1836 (6MP) | Silabel 2 | 71.42 |
| | Silabel 3 | 66.67 |
| Resolusi 3264x2448 (8MP) | Silabel 2 | 82.85 |
| | Silabel 3 | 76.67 |

4.2. Pengujian Pengaruh Jarak Akuisisi Terhadap Objek

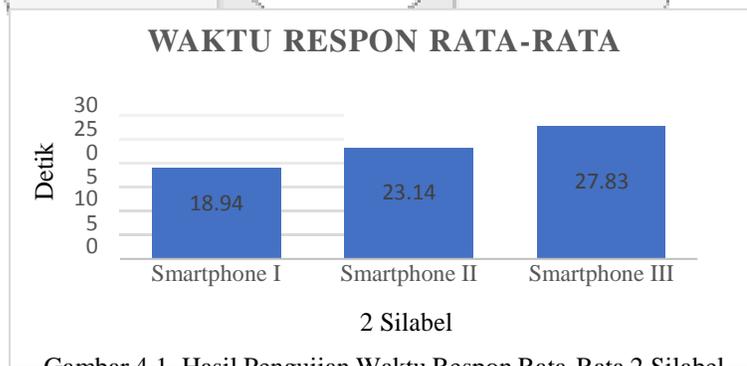
Pengujian ini bertujuan untuk mengetahui akurasi sistem dengan akuisisi citra dengan berbagai jarak yang ditentukan. Pengujian ini dilihat tidak hanya dari jarak tapi ukuran *font* serta kesempurnaan pengambilan gambar mempengaruhi hasil pengujian. Pengujian dilakukan pengambilan terhadap gambar dengan beberapa jarak yang telah ditentukan yakni menggunakan jarak 12cm, 17cm dan 20 cm menggunakan kamera asus zenfone 5 dengan resolusi 8 Megapiksel.

Tabel 4.2. Hasil Pengujian Pengaruh Jarak Terhadap Objek

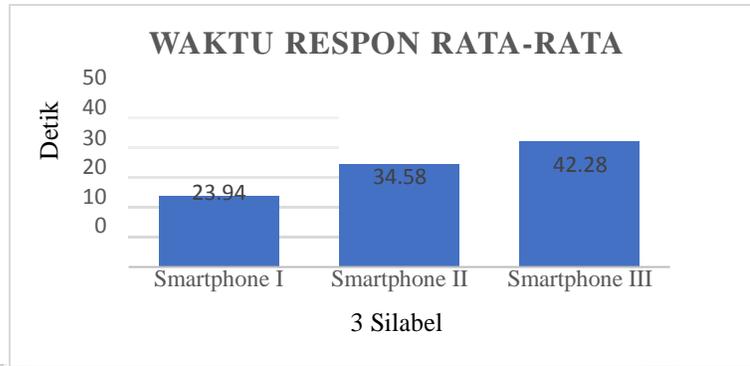
| Jarak (cm) | Silabel | Akurasi % |
|------------|-----------|-----------|
| 12 | Silabel 2 | 77.14% |
| | Silabel 3 | 73.33% |
| 17 | Silabel 2 | 74.29% |
| | Silabel 3 | 73.33% |
| 20 | Silabel 2 | 71.43% |
| | Silabel 3 | 70.00% |

4.3. Pengujian Waktu Respon Sistem

Tujuan pengujian ini adalah mengetahui pengaruh kecepatan *clock* prosesor terhadap kecepatan waktu respons aplikasi. Aplikasi dijalankan pada beberapa jenis *smartphone* dengan kapasitas prosesor yang berbeda. *Device* dengan *clock* lebih besar memiliki kecepatan komputasi data yang lebih cepat. Dengan demikian semakin besar *clock prosesor device*, semakin cepat pula waktu respons yang diperlukan aplikasi untuk menerjemahkan citra masukan.



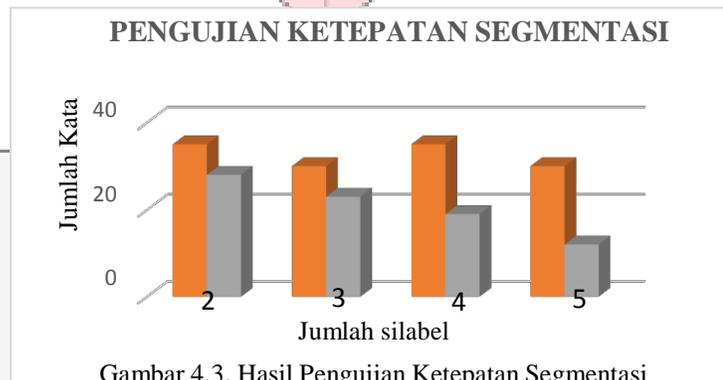
Gambar 4.1. Hasil Pengujian Waktu Respon Rata-Rata 2 Silabel



Gambar 4.2. Hasil Pengujian Waktu Respon Rata-Rata 3 Silabel

4.4. Pengujian Ketepatan Segmentasi

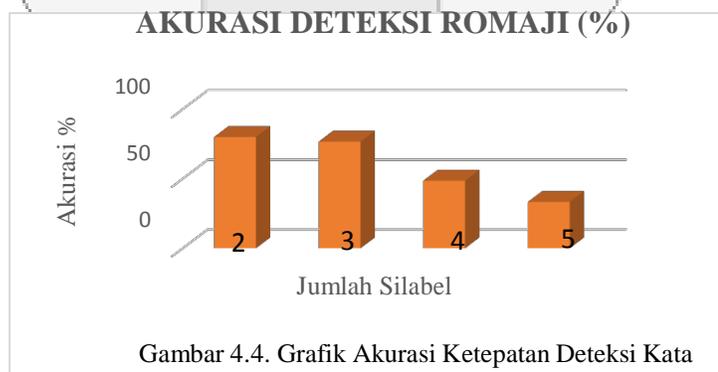
Pengujian ini bertujuan untuk mengetahui ketepatan dalam proses segmentasi pada setiap kata yang diujikan. Pengujian akan dilakukan pada setiap kata berdasarkan jumlah silabel. Hasil segmentasi dari beberapa kata yang disegmentasi diketahui bahwa setiap kata yang mengandung silabel terpisah seperti huruf hiragana “i” akan mendapatkan kesalahan pada proses segmentasinya.



Gambar 4.3. Hasil Pengujian Ketepatan Segmentasi

4.5. Pengujian Ketepatan Deteksi

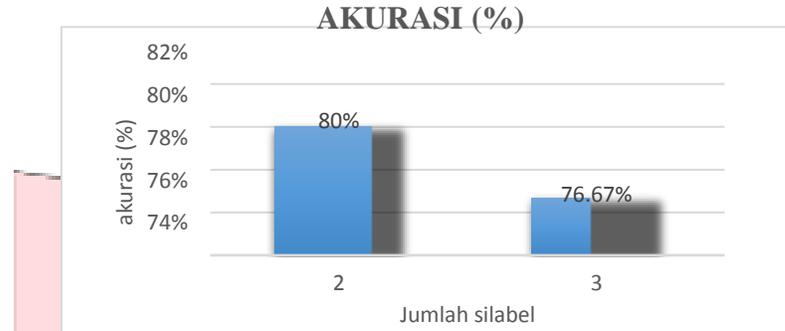
Tujuan dari pengujian ini adalah mengetahui tingkat keakuratan sistem dalam melakukan pendeteksian setiap citra masukan. Hasil yang diharapkan pada pengujian ini yaitu romaji jepang (tulisan latin jepang). Pengujian dilakukan pada setiap kata berdasarkan jumlah silabel yaitu masing masing 2, 3, 4 dan 5 silabel dengan masing masing kata yaitu 35, 30, 35 dan 30.



Gambar 4.4. Grafik Akurasi Ketepatan Deteksi Kata

4.6. Pengujian Ketepatan Terjemahan

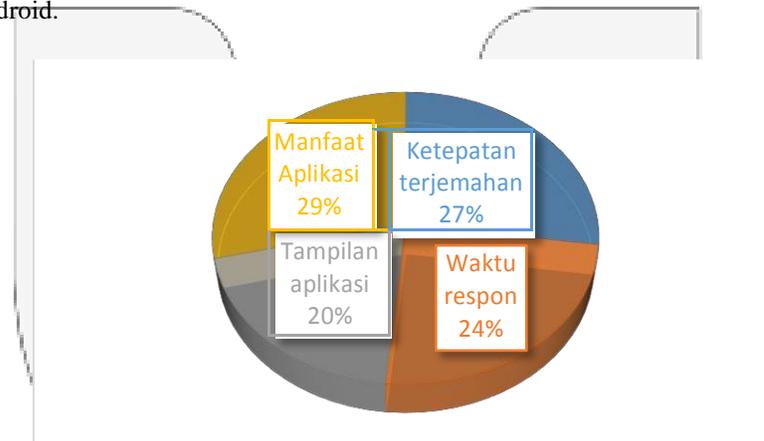
Tujuan dari pengujian ini adalah mengetahui tingkat keakuratan sistem dalam melakukan penerjemahan citra masukan. Berdasarkan pengujian segmentasi, maka dipilih kata-kata Bahasa Jepang yang terdiri atas dua dan tiga silabel, masing-masing 35 dan 30 kata. Hasil pengujian menunjukkan nilai akurasi yang cukup tinggi, yaitu 80% untuk 2 silabel. Hal ini menunjukkan bahwa ciri yang diambil dari masing-masing silabel cukup mewakili keunikan silabel tersebut sehingga dapat dikenali oleh sistem.



Gambar 4.5 Hasil Pengujian Ketepatan Penerjemahan

4.7. Pengujian Beta

Pengujian ini dilakukan pada sisi pengguna secara objektif. Pengujian Beta merupakan pengujian berdasarkan umpan-balik dari pengguna aplikasi. Terdapat empat poin yang diujikan kepada responden, yaitu tampilan aplikasi, ketepatan penerjemahan, lama waktu respons, serta manfaat aplikasi. Pengujian beta dilakukan pada 25 responden yang memenuhi kriteria yang ditentukan yaitu diantaranya para pengguna *smartphone* Android.



Gambar 4.6. Hasil Pengujian Beta

V. PENUTUP

Berdasarkan tujuan pembuatan sistem, dapat diambil beberapa kesimpulan.

- 1 Deteksi kata pada Huruf Hiragana dalam Bahasa Jepang dapat berhasil dengan menggunakan metode *Histogram of Oriented Gradients* (HOG) pada proses ekstraksi ciri, dan hasil ekstraksi ciri diklasifikasikan menggunakan metode *K-Nearest Neighbor* (KNN) dengan nilai akurasi sebesar 80.00% dengan waktu respon sistem pada *smartphone* Android dengan prosesor 2.0 GHz Dual-core dengan waktu 18.94 detik.
- 2 Pada pengujian berdasarkan ukuran resolusi piksel citra akurasi tertinggi mencapai 82.85 % dengan resolusi Resolusi 3264x2448 atau 8 MegaPiksel untuk 2 silabel sedangkan untuk 3 silabel adalah 76.67 %.
- 3 Perancangan dengan menggunakan ekstraksi ciri *Histogram of Oriented Gradients* (HOG) serta klasifikasi dengan menggunakan metode *K-Nearest Neighbor* (KNN) dapat diterapkan pada perangkat *smartphone* Android dalam bentuk aplikasi. Berdasarkan pengujian Beta dinyatakan

bahwa 76% responden memberikan penilaian manfaat untuk aplikasi dalam menterjemahkan kata huruf Hiragana ke dalam Bahasa Indonesia dan 52% responden menilai bahwa tampilan aplikasi menarik.

Berdasarkan penelitian yang telah dilakukan dan hasil yang telah diperoleh, penulis memberikan beberapa saran untuk pengembangan riset selanjutnya.

- 1 Mengembangkan penelitian serupa dengan menggunakan bahasa pemrograman lain dan menerapkan pada *platform* lain seperti iOS, *Windows Phone* ataupun *blackbery*.
- 2 Menggunakan metode lain pada tahap segmentasi, sehingga dapat mendeteksi semua jenis huruf dengan sempurna dan mendapatkan hasil segmentasi yang akurat, termasuk huruf yang terpisah.
- 3 Mengembangkan penelitian metode ekstraksi ciri HOG seperti menggunakan CHOG untuk mendapatkan ciri dari setiap *cell* piksel pada citra agar didapatkan ciri-ciri setiap huruf jepang dengan jelas dan tepat.
- 4 Untuk tahap klasifikasi dapat dikembangkan dengan menggunakan metode yang lebih *advance*, seperti metode pengembangan Jaringan Syaraf Tiruan yang lainnya.

VI. DAFTAR PUSTAKA

- [1] Sfaat, Nazruddin, 2011. "Pemrograman aplikasi *mobile smartphone* dan tablet pc berbasis android". Bandung: Penerbit Informatika.
- [2] Fitria, Whielda, 2010. "Desain dan Implementasi Sistem Penerjemah Huruf Jepang Kana (Hiragana-Katakana) ke Huruf Latin Berbasis Pengolahan Citra Digital dan Jaringan Syaraf Tiruan *Self-Organizing Map* (SOM)". Bandung: Institut Teknologi Telkom.
- [3] Munir, Rinaldi, 2011. "Pengolahan Citra Digital". Bandung: Penerbit Informatika.
- [4] Riyadi, Widyanto, 2011. "Pengenalan Huruf Jepang (KANJI) Menggunakan *Independent Component Analysis* (ICA) dan *Learning Vector Quantization* (LVQ)". Bandung: Institut Teknologi Telkom.
- [5] Basith, Linazizah, Aidatunisadina, 2013. "Aplikasi Penerjemah Tulisan Bahasa Jepang ke Bahasa Indonesia menggunakan *Holistic Feature Extraction* pada *Handset* Berbasis Android". Bandung: Institut Teknologi Telkom.
- [6] Robi, Firmanda, 2014. "Rancang Bangun Aplikasi Deteksi Motif Batik Berbasis Pengolahan Citra Digital Pada *Platform* Android". Bandung: Universitas Telkom
- [7] Mustofa, Aril, Yasin, 2014. "Penentuan *Threshold* Menggunakan Algoritma Fuzzy C-Means Untuk Segmentasi *Region* Pada Plat Nomor Kendaraan". Volume 10 Nomor 1: Universitas Dian Nuswantoro.
- [8] Munir, Rinaldi, 2006. "Aplikasi *Image Thresholding* Untuk Segmentasi Objek" Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
- [9] Anwar, Khaerul, 2011. "Desain dan Implementasi Sistem Pembacaan dan Penerjemahan Kata Dalam Bahasa Arab Ke Bahasa Indonesia Berbasis Pengolahan Citra Digital". Bandung: Insitut Teknologi Telkom.
- [10] Vaulin, Syylvia, 2009. "Implementasi dan Analisis Pengenalan Huruf Menggunakan Algoritma Berbasis *Chain Code* dan *K-Nearest Neighbor*". Bandung: Institut Teknologi Telkom
- [11] Gonzalez, C. Rafael dan Woods E. Richard, 2002. "*Digital Image Processing*". New Jersey: Prentice-Hall, Inc.
- [12] Reza Ebrahimzadeh dan Mahdi Jampour, 2014. "*Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM*". International Journal of Computer Applications (0975 – 8887): Volume 104 – No.9, October 2014
- [13] Ilmi, Reza, 2015. "Perancangan dan Implementasi *Histogram of Oriented Gradients* dan *Support Vector Machine* (HOG+SVM) untuk deteksi obyek pejalan kaki pada aplikasi *Mobile* Berbasis Android". Bandung: Universitas Telkom
- [14] H. El Bahi, Z. Mahani, A. Zatni and S. Saoud, 2015. "*A robust system for printed and handwritten character recognition of images obtained by camera phone*". E-ISSN: 2224-3488 Volume 11, 2015
- [15] Wilhelm Burger, Mark J. Burge. "*Digital Image Processing an algorithmic Introduction using java 1st*".