

**PERANCANGAN DAN IMPLEMENTASI *DIRECTIONAL FEATURE EXTRACTION* DAN *SUPPORT VECTOR MACHINES* UNTUK DETEKSI HURUF HIRAGANA JEPANG PADA APLIKASI *MOBILE PENERJEMAH KATA* DALAM BAHASA JEPANG KE BAHASA INDONESIA BERBASIS ANDROID**

**Fardilla Zardi Putri<sup>1</sup>, Budhi Irawan,S.Si.,M.T.<sup>2</sup>,Umar Ali Ahmad,S.T.,M.T.<sup>3</sup>**

<sup>1,2,3</sup>Sistem Komputer, Fakultas Teknik Elektro – Univesitas Telkom Jln.

Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia

[fardillaputri@gmail.com](mailto:fardillaputri@gmail.com)<sup>1</sup>, [bir@telkomuniversity.ac.id](mailto:bir@telkomuniversity.ac.id)<sup>2</sup>, [uaa@telkomuniversity.ac.id](mailto:uaa@telkomuniversity.ac.id)<sup>3</sup>

**ABSTRACT**

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 sering *Optical Character Recognition* (OCR) merupakan salah satu aplikasi teknologi pada bidang pengenalan karakter atau pola dan kecerdasan buatan sebagai mesin pembaca. Pada penelitian tugas akhir ini, akan dirancang sebuah aplikasi penerjemah kata dalam Bahasa Jepang berbasis Android dengan memanfaatkan prinsip dasar OCR dengan menggunakan metode *Directional Feature Extraction* dan *Support Vector Mechine*. Pengujian yang dilakukan memberikan hasil terbaik pada nilai akurasi yang dicapai dengan menggunakan metode *Directional Feature Extraction* dan *Support Vector Mechine* adalah 85.71%. Pada penelitian ini menggunakan 104 data latih. Hasil Pengujian Beta atas empat poin, yaitu tampilan aplikasi, waktu respons sistem, ketepatan penerjemahan, dan manfaat aplikasi menunjukkan aplikasi dapat diklasifikasikan sebagai baik.

**Kata Kunci:** *Optical Character Recognition, Directional Feature Extraction, Support Vector Mechine, 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 application of technology in the field of character or pattern recognition and artificial intelligence as reading machines. In this research, we will design an application in Japanese language word translator based Android by utilizing basic principles of OCR using the *Directional Feature Extraction* and *Support Vector Mechine*. Tests were done to give the best results on the value of accuracy is achieved by using *Directional Feature Extraction* and *Support Vector Mechine* is 85.71%. In this study, using 104 training data. 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 good

**Keyword:** *Optical Character Recognition, Directional Feature Extraction, Support Vector Mechine, 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. Saat mengunjungi negara tersebut para pengunjung akan kesulitan membaca dan mengartikan setiap kata dalam Bahasa Jepang. Saat ini, android merupakan salah satu teknologi yang sangat berkembang dan banyak diminati berbagai kalangan. Android merupakan *Operating System* yang banyak digunakan oleh pengguna *mobile phone*. Sifatnya yang open source memungkinkan pengguna menambahkan aplikasi sesuai kebutuhan. *Mobile phone* yang berbasis android tersebut juga sering dibawa kemana-mana oleh penggunanya. Seiring dengan berkembangnya teknologi, pengenalan karakter dapat dibantu dengan sebuah alat seperti *scanner*. *Optical Character Recognition* (OCR) merupakan salah satu aplikasi teknologi pada bidang pengenalan karakter atau pola dan kecerdasan buatan sebagai mesin pembaca.

Pada penelitian tugas akhir ini akan dirancang sebuah aplikasi penerjemah Bahasa Jepang berbasis Android. Dengan menggunakan prinsip *Optical Character Recognition* (OCR) untuk mengenali huruf dalam Bahasa Jepang yang tertangkap oleh kamera *handphone*. Citra masukan diolah dan diekstraksi untuk mendapatkan ciri dari huruf dalam Bahasa Jepang menggunakan metode *Directional Feature Extraction*. Sedangkan metode *Support Vector Machine* akan digunakan pada tahap pengklasifikasian. Oleh karena itu, dengan adanya aplikasi ini diharapkan dapat membantu para pengunjung negara Jepang terutama pengunjung dari Indonesia dalam menerjemahkan setiap kata dalam Bahasa Jepang.

### 1.2. Rumusan Masalah

Berdasarkan latar belakang diatas ada beberapa masalah yang akan diselesaikan dalam penelitian tugas akhir ini adalah:

- (1) Bagaimana merancang dan membuat aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang menggunakan metode *Directional Feature Extraction* dan *Support Vector Machines* pada *smartphone* Android?
- (2) Bagaimana implementasi aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang pada *smartphone* Android?
- (3) Bagaimana pengujian dan analisa performansi dari aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang pada *smartphone* Android?

### 1.3. Tujuan

Adapun tujuan dilaksanakannya tugas akhir ini yaitu:

- 1) Merancang dan membuat aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang menggunakan metode *Directional Feature Extraction* dan *Support Vector Machines* pada *smartphone* Android.
- 2) Mengimplementasikan aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang pada *smartphone* Android.
- 3) Melakukan pengujian dan analisa performansi aplikasi untuk menerjemah kata dengan pengenalan huruf *hiragana* dalam Bahasa Jepang pada *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 Hiragana
- (2) Bahasa Jepang diterjemahkan dalam Bahasa Indonesia
- (3) Proses ekstraksi ciri menggunakan *Directional Feature Extraction* (DFE)
- (4) Proses klasifikasi menggunakan *Support Vector Machines* (SVM)
- (5) Menggunakan *Operation System* Android
- (6) Sistem bersifat *mobile-based offline*
- (7) Citra yang diolah adalah karakter tulisan cetak atau diketik dalam format JPEG
- (8) Background citra hanya terdiri dari satu warna
- (9) Maksimal jumlah silabel dalam satu kata adalah 5 silabel
- (10) Deteksi berdasarkan per-kata

## II. TEORI DASAR

### 2.1. Android

Android adalah suatu sistem operasi yang didesain sebagai *platform open source* untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android menyediakan semua *tools* dan *framework* untuk mengembangkan aplikasi dengan mudah dan cepat. Dengan adanya Android SDK (*Software Development Kit*) pengembang aplikasi dapat memulai pembuatan aplikasi pada *platform* android menggunakan bahasa pemrograman java.

### 2.2. Hiragana Jepang

Dalam bahasa jepang terdapat tiga jenis aksara yaitu hiragana, katakana dan kanji. Huruf Hiragana dan katakana dibuat oleh orang 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).

**2.3. Citra Digital**

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil CT Scan dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer.

**2.3.1. Citra RGB**

Citra RGB merupakan kombinasi dari elemen warna Red (merah), Green (hijau), dan Blue (biru). Format RGB ini direpresentasikan dengan menggunakan 3 layer. Setiap pikselnya merupakan kombinasi dari ketiga elemen warna tsb, dengan tingkat warna 0-255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin computer.

**2.3.2. Citra Grayscale**

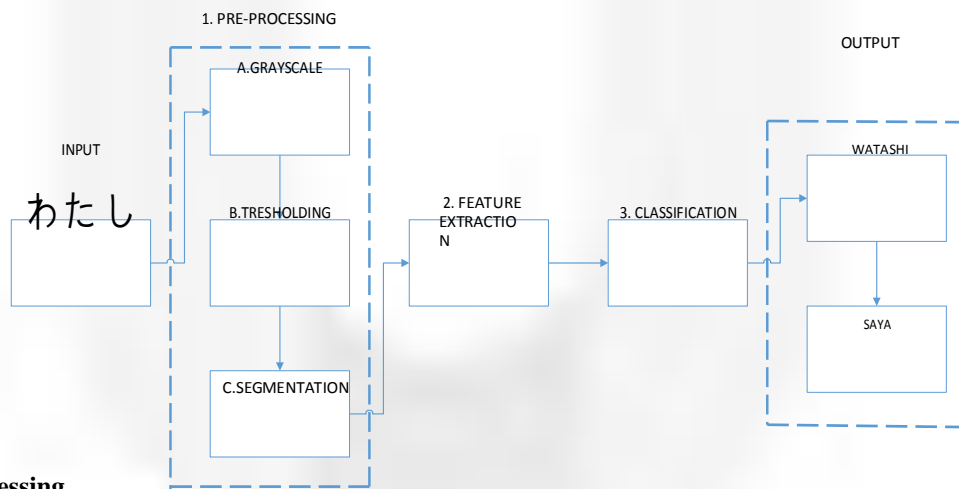
Citra grayscale berbeda dengan citra ”hitam-putih”, dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu ”hitam” dan ”putih” saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Terdapat tiga cara yang umum 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 dari format RGB menjadi format *grayscale* (memiliki derajat keabuan). Untuk mengubah citra RGB menjadi *grayscale* menggunakan rumus berikut, dimana nilai pada masing-masing R,G,B tersebut dari 0-255.

$$Average\ Grayscale = \frac{R + G + B}{3}$$



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

Setelah *grayscale* dilanjutkan dengan proses *thresholiding*. *Thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas.



Gambar 3.2 Hasil Tresholding

Tahap selanjutnya segmentasi merupakan proses mempartisi citra menjadi beberapa daerah atau objek. Pada penelitian ini tahap segmentasi malakukan *cropping* setiap silabel dari sebuah kata. Metode yang digunakan adalah *line segmentation*, baik secara horizontal dan vertical dan *Projection Block*. Silabel yang didapatkan dari segmentasi akan menjadi masukan proses ekstraksi ciri.



Gambar 3.3 Hasil Segmentasi

### 3.3. Ekstraksi Ciri

Proses ekstraksi ciri merupakan pengambilan informasi – informasi penting yang terdapat pada sebuah citra. Tujuan proses ekstraksi ciri adalah mengambil beberapa informasi penting dari citra. Metode yang digunakan pada tahap ekstraksi ciri ini adalah *Directional Feature Extraction*. Perhitungan pada penelitian ini mengacu pada sebuah *paper*<sup>[9]</sup>.

### 3.4. Klasifikasi

Tujuan dari proses pembelajaran SVM adalah untuk mendapatkan hipotesis berupa bidang pemisah terbaik (*hyperplane*) yang tidak hanya mengurangi *empirical risk*. Secara garis besar cara kerja dari *Support Vector Machine* adalah vektor ciri dari setiap silabel hiragana didapat dari ekstraksi ciri DFE yang berukuran 1x2500, matrix 1x2500 merupakan data *input* awal. Data *training* terdiri dari 104 kata yang berasal dari huruf Hiragana dalam Bahasa Jepang. Setiap kata dibatasi panjangnya sebanyak maksimal 5 silabel. Data awal yang telah diolah kemudian disatukan dan dihasilkan matrik ciri, gabungan dari seluruh vector ciri. Terdapat 104 kelas yang merupakan jumlah huruf hiragana.

Secara garis besar cara kerja dari *Support Vector Machine* adalah sebagai berikut:

1. Vektor ciri dari setiap silabel hiragana didapat dari ekstraksi ciri *Directional Feature Extraction* yaitu matrik yang berukuran 1x2500 dijadikan data *input* awal. Data yang digunakan terbagi 2 yaitu data *training* dan data *testing*.
  - Data *training* terdiri dari 104 silabel yang berasal dari huruf *Hiragana* dalam Bahasa Jepang yang berukuran 50x50 yang telah melewati proses *pre-processing* dan ekstraksi ciri. Setiap silabel terdiri dari 3 bentuk yaitu ada **bold**, *italic* dan normal silabel, sehingga ada 3x104 silabel yaitu 312.
  - Data *testing* adalah citra kata bahasa jepang yang terdiri dari huruf *hiragana* Jepang. Maksimal silabel pada kata tersebut adal 5 silabel. Citra tersebut akan dipre-processing dan diekstraksi cirinya. Dan disetiap kata tersebut akan diklasifikasi oleh *Support Vector Machine*.
2. Data *input* awal yang 312 yang telah didapatkan ekstraksi cirinya yaitu berupa matrik 312x2500. Dimana setiap baris pada matrik mewakili setiap ciri pada silabel huruf *hiragana* Jepang.

## IV. PENGUJIAN DAN ANALISIS

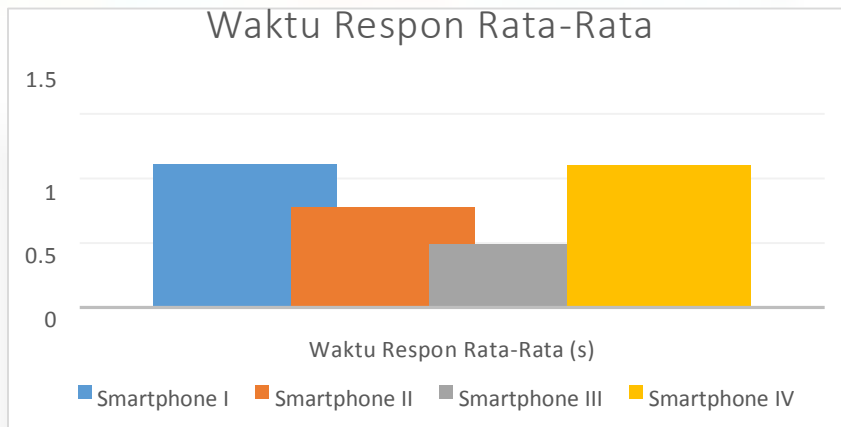
### 4.1. Pengujian Ukuran Resolusi Citra

Pada pengujian ini, menggunakan empat ukuran resolusi citra yang umum didapatkan pada mobile Android, yaitu 2.4MP, 6MP, 13.1MP, dan 16MP. Tiap resolusi diuji dengan citra masukan yang terdiri atas 30 kata 2 silabel dan 3 silabel yang sama. Masing-masing kata diambil dengan melakukan kontrol terhadap jarak pengambilan dan pencahayaan.

| Resolusi           | Silabel   | Akurasi (%) |
|--------------------|-----------|-------------|
| Resolusi 2048x1152 | Silabel 2 | 71.43%      |
|                    | Silabel 3 | 60%         |
| Resolusi 3264x1836 | Silabel 2 | 80%         |
|                    | Silabel 3 | 66.67%      |
| Resolusi 4128x3096 | Silabel 2 | 85.71%      |
|                    | Silabel 3 | 83.33%      |
| Resolusi 5312x2988 | Silabel 2 | 91.42%      |
|                    | Silabel 3 | 90%         |

4.2. Pengujian Waktu Respon Sistem

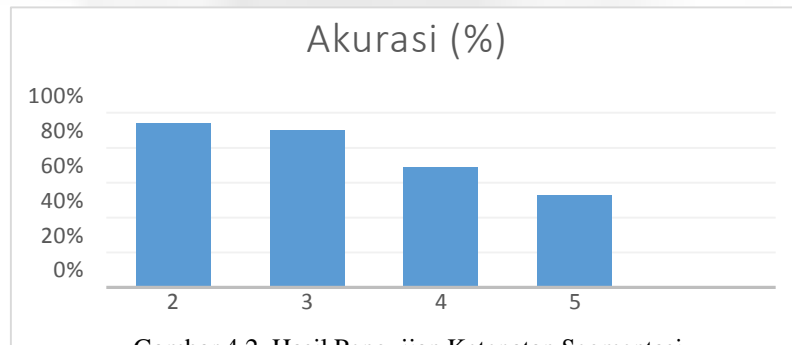
Tujuan pengujian ini adalah mengetahui pengaruh kecepatan *clock* prosesor terhadap kecepatan waktu respons aplikasi. Aplikasi dijalankan pada beberapa jenis *smartphone* dengan kapasitas prosesor yang berbeda. *Device* dengan *clock* lebih besar memiliki kecepatan eksekusi data yang lebih cepat. Oleh karena itu, 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

4.3. Pengujian Ketepatan Segmentasi

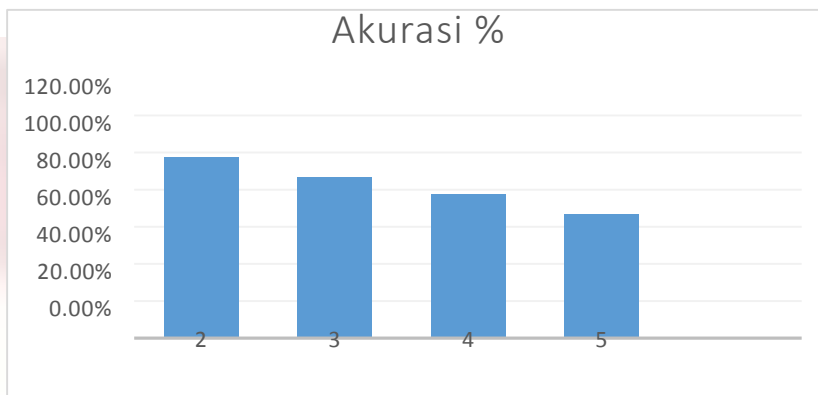
Pengujian ini bertujuan untuk mengetahui ketepatan dalam proses segmentasi pada setiap kata yang diujikan. Pengujian akan dilakukan pada setiap kata berdasarkan jumlah silabel. Dari 130 data yang diuji terdapat 100 data yang hasil segmentasinya benar dan 30 data mengalami kesalahan dalam segmentasi. Pada pengujian ini dapat dilihat bahwa akurasi segmentasi tertinggi adalah 94% untuk 2 silabel dan 90% untuk 3 silabel. Banyaknya jumlah silabel memiliki hubungan *non-linier* dengan ketepatan segmentasi yang dilakukan aplikasi.



Gambar 4.2. Hasil Pengujian Ketepatan Segmentasi

#### 4.4. Pengujian Deteksi Kata

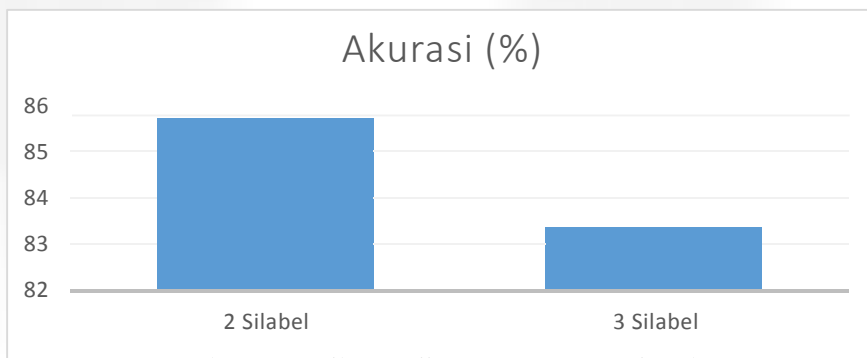
Tujuan dari pengujian ini adalah mengetahui tingkat keakuratan sistem dalam melakukan pendeteksian setiap citra masukan. Dari 130 data yang telah diuji, ada 107 data yang dideteksi benar dan 23 dideteksi salah. Pengujian ini dilakukan berdasarkan banyak silabel yang terdapat pada setiap kata yang akan dideteksi. Silabel terdiri dari 2,3,4 dan 5. Grafik dibawah menjelaskan nilai akurasi tertinggi pada ketepatan deteksi ini pada kata yang terdiri dari 2 silabel adalah 97.14%, dimana dari 35 jumlah kata terdapat 1 kata yang salah deteksi.



Gambar 4.3. Hasil Pengujian Deteksi Kata

#### 4.5. Pengujian Ketepatan Penerjemah

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 30 kata. Hasil pengujian menunjukkan nilai akurasi yang cukup tinggi, yaitu 85.71%. Hal ini menunjukkan bahwa ciri yang diambil dari masing-masing silabel cukup mewakili keunikan silabel tersebut sehingga dapat dikenali oleh sistem.

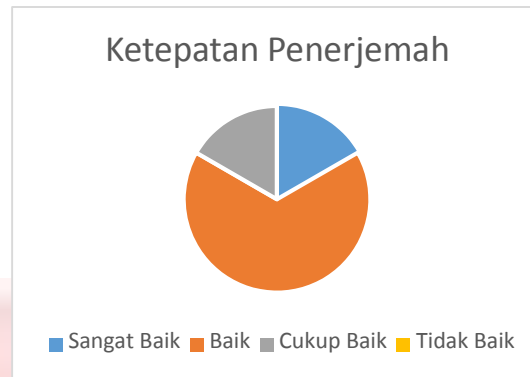


Gambar 4.4 Hasil Pengujian Ketepatan Penerjemahan

#### 4.6. Pengujian Beta

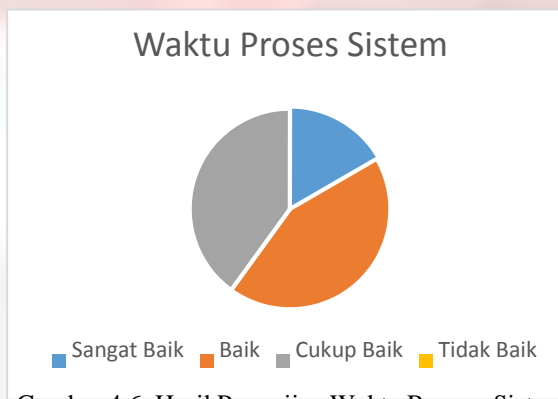
Pengujian ini dilakukan pada sisi pengguna secara objektif. Pengujian Beta merupakan pengujian berdasarkan umpan-balik dari pengguna aplikasi. Terdapat empat poin yang diujikan kepada responden, yaitu tampilan aplikasi, ketepatan penerjemahan, lama waktu respons, serta tingkat kebermanfaatan aplikasi. Pengujian beta dilakukan pada 30 responden yang memenuhi kriteria yang ditentukan yaitu diantaranya para pengguna *smartphone* Android.

Pada pengujian ketepatan penerjemahan, responden difasilitasi dengan daftar kata yang diujikan beserta artinya untuk pengecekan ketepatan. Dari pengujian ini, 66.67% responden menjawab penerjemahan tepat, 16.67% menjawab sangat tepat dan 16.67% menjawab cukup tepat.



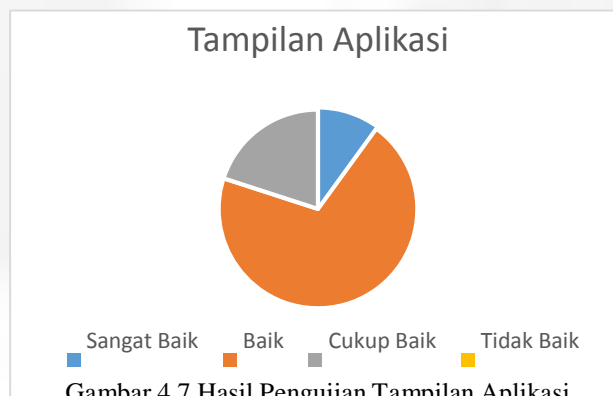
Gambar 4.5. Hasil Pengujian Ketepatan Penerjemahan

Pengujian waktu respons sistem menghasilkan persentase 43.33% untuk cepat, 16.67 untuk sangat cepat dan 40% cukup cepat.



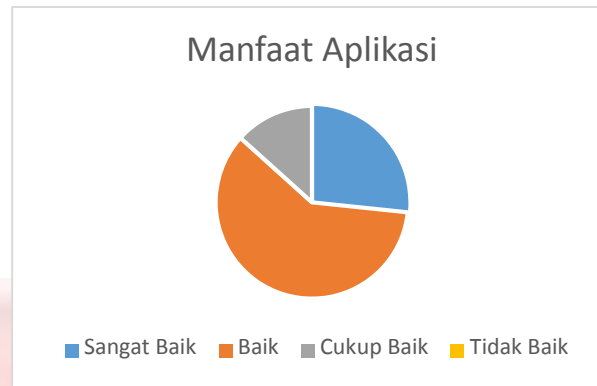
Gambar 4.6. Hasil Pengujian Waktu Respon Sistem

Pada pengujian ini, 70% responden menjawab bahwa tampilan baik, 10% menjawab sangat baik dan 20% menjawab cukup baik



Gambar 4.7 Hasil Pengujian Tampilan Aplikasi

Poin terakhir pada Pengujian Beta adalah mengenai tingkat manfaat aplikasi yang telah dibuat. 60% responden memberikan jawaban baik, 13.33 menjawab cukup baik dan 26.67% menjawab sangat baik.



Gambar 4.8 Hasil Pengujian Manfaat Penggunaan Aplikasi

## V. PENUTUP

Berdasarkan tujuan pembuatan sistem, dapat diambil beberapa kesimpulan.

- 1) Perancangan sistem aplikasi terdiri atas proses akuisisi citra, preprocessing, ekstraksi ciri, klasifikasi, dan pengecekan terjemahan. Ekstraksi ciri menggunakan *Directional Feature Extraction*, sedangkan proses klasifikasi menggunakan *Support Vector Machine*.
- 2) Aplikasi yang telah dirancang diimplementasikan pada Sony Xperia Z dengan Operating System Android 5.1.1 (Lollipop).
- 3) Dari pengujian dan analisis yang dilakukan, dapat disimpulkan beberapa hal.
  - a) Tingkat akurasi tertinggi dicapai ukuran resolusi 4605x3456 piksel atau 16 *Megapixel* yaitu 91.42% dengan menggunakan 2 silabel dan 90% dengan menggunakan 3 silabel.
  - b) Kecepatan clock prosesor berpengaruh linier terhadap kecepatan waktu respons sistem. Semakin besar clock, semakin cepat eksekusi data yang dilakukan sehingga waktu respons juga lebih cepat. Prosesor dengan menggunakan *device* berprosesor 1.4 GHz *Quad-core* dan 1.8 GHz *Dual-core* memiliki waktu respon sistem 0.489 detik.
  - c) Ketepatan segmentasi dipengaruhi oleh beberapa hal yaitu metode segmentasi yang digunakan dan morfologi silabel yang dideteksi. Akurasi tertinggi didapatkan yang menggunakan 2 silabel yaitu 94% dan 3 silabel 90%.
  - d) Metode *Directional Feature Extraction* sesuai untuk diterapkan bagi ekstraksi ciri kata-kata Bahasa Jepang. Pengujian terhadap akurasi penerjemahan aplikasi menunjukkan nilai keberhasilan 85.71%.
  - e) Berdasarkan Pengujian Beta, secara umum aplikasi dikatakan baik, dilihat dari segi tampilan dengan persentase 70%, untuk ketepatan penerjemahan 66.67%, dan manfaat aplikasi 60%. Sedangkan untuk persentase waktu respon sistem masih perlu diperbaiki lagi karena persentasenya cukup rendah 43.33%.

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 dan Windows Phone.
- 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 dan mengandung *teng-teng* dan *moru*.
- 3) Mengembangkan penelitian metode ekstraksi ciri DFE jenis lain seperti, *Thinning Directional Decomposition*, *New Stroke based Directional Decomposition*, dan *Contour Directional Decomposition*, 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.



## VI. DAFTAR PUSTAKA

- [1] Cahyan, Pramuda Akariusta, “**Segmentasi Citra Digital Dengan Menggunakan Algoritma Watershed dan Lowpass Filter Sebagai Proses Awal**”, Teknik Elektro Universitas Brawijaya. November 2013.
- [2] Xue Gao, Lian-Wen Ji, “**A New Stroke-Based Directional Feature Extraction Approach for Handwritten Chinese Character Recognition**”, Department of Electronics and Communication Engineering South China University of Technology, Guangzhou. 2001.
- [3] Murtiwiayati dan Glenn Lauren, “**Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar Berbasis Android**”, Fakultas Ilmu Komputer, Universitas Gunadarma. Desember 2013.
- [4] Satriyo Nugroho, Anto dan Budi Witarto, Arif, “**Application Of Support Vector Machine in Bioinformatics**”. Proceeding of Indonesian Scientific Meeting in Central Japan. Desember 2003.
- [5] B. Hallale, Sumedha dan D. Salunke, Geeta, “**Twelve Directional Feature Extraction for Handwritten English Character Recognition**”. International Journal of Recent Technology and Engineering. May 2013.
- [6] A.A Tayade dan R.V Mante, “**Text Recognition and Translation Application for Smartphone**”, International Journal of Advanced Research in Computer and Communication Engineering Vol 2, Issue VI. November 2013.
- [7] R.Ramanathan dan S.Ponmathavan, “**Optical Character Recognition for English and Tamil Using Support Vector Machine**”. Department of Electronics and Communication Engineering, India. International Conference on Advances in Computing, Control and Telecommunication Technologies. 2009.
- [8] Munir, Renaldi. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Penerbit Informatika.
- [9] Singh, Dayashankar, “**Handwritten Character Recognition Using Twelve Directional Feature Input and Neural Network**”. International Journal of Computer Application. India. 2010.
- [10] V. N. Vapnik, “**The Nature of Support Vector Machine**”. Berlin Heidelberg. 1999.
- [11] N. Cristianini, “**An Introduction to Support Vector Machines and Other KernelBased Learning Methods**”, Cambridge Press University, 2000.
- [12] Chen, Jiun-Lin, “**Chinese Handwritten Character Segmentation in Form Documents**”, National Chiao Tung University, Taiwan. 1999.