

# PERANCANGAN DAN IMPLEMENTASI *DIRECTIONAL FEATURE EXTRACTION* DAN *SUPPORT VECTOR MACHINES* UNTUK MENERJEMAHKAN KATA DENGAN PENGENALAN HURUF HIRAGANA DALAM BAHASA JEPANG KE BAHASA INDONESIA BERBASIS ANDROID

Fardilla Zardi Putri<sup>1</sup>, Budhi Irawan<sup>2</sup>, Umar Ali Ahmad<sup>3</sup>

<sup>1,2,3</sup>Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>fardillaputri@gmail.com, <sup>2</sup>budhiirawan@telkomuniversity.ac.id, <sup>3</sup>umar@telkomuniversity.ac.id

---

## Abstrak

Pada era global ini menguasai bahasa selain bahasa Indonesia merupakan salah satu kebutuhan penting yang harus dimiliki setiap orang. Banyak orang berkunjung ke negara lain untuk melakukan banyak kegiatan seperti bekerja, belajar, bahkan berlibur. 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 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 Machine*. Pengujian yang dilakukan memberikan hasil terbaik pada nilai akurasi yang dicapai dengan menggunakan metode *Directional Feature Extraction* dan *Support Vector Machine* 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 baik.

**Kata Kunci:** *Optical Character Recognition, Directional Feature Extraction, Support Vector Machine, Android*

---

## Abstract

In this global era mastering language besides *bahasa Indonesia* 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 most 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 called *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 Machine*. Tests was done to give the best results on the value of accuracy is achieved by using *Directional Feature Extraction* and *Support Vector Machine* is 85.71%. This study is using 104 training data. Beta Testing result on four points, namely the display application, system response time, accuracy of translation, and the benefits of the application shows that the application can be classified as good.

**Keyword:** *Optical Character Recognition, Directional Feature Extraction, Support Vector Machine, Android*

---

## 1. Pendahuluan

Pada era global ini menguasai bahasa selain Bahasa Indonesia 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 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.

## 2. 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*.

## 3. 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 (misalnya m dan a menjadi ma) atau konsonan ditambah semi-vokal y (misalnya m, y, dan a menjadi mya).

あ	か	さ	た	な	は	ま	や	ら	わ		が	ざ	だ	ば	ぱ
a	ka	sa	ta	na	ha	ma	ya	ra	wa		ga	za	da	ba	pa
い	き	し	ち	に	ひ	み		り			ぎ	じ	ぢ	び	ぴ
i	ki	shi	chi	ni	hi	mi		ri			gi	ji	ji	bi	pi
う	く	す	つ	ぬ	ふ	む	ゆ	る			ぐ	ず	づ	ぶ	ぷ
u	ku	su	tsu	nu	fu	mu	yu	ru			gu	zu	zu	bu	pu
え	け	せ	て	ね	へ	め		れ			げ	ぜ	で	べ	ぺ
e	ke	se	te	ne	he	me		re			ge	ze	de	be	pe
お	こ	そ	と	の	ほ	も	よ	ろ	を	ん	ご	ぞ	ど	ぼ	ぽ
o	ko	so	to	no	ho	mo	yo	ro	o	n	go	zo	do	bo	po
きゃ	しゃ	ちゃ	にゃ	ひゃ	みゃ	りゃ	ぎゃ	じゃ	ぢゃ	びゃ	ぴゃ				
kyā	shā	chā	nyā	hyā	myā	ryā	gyā	jā	chā	byā	pyā				
きゅ	しゅ	ちゅ	にゅ	ひゅ	みゅ	りゅ	ぎゅ	じゅ	ぢゅ	びゅ	ぴゅ				
kyū	shū	chū	nyū	hyū	myū	ryū	gyū	jū	chū	byū	pyū				
きょ	しよ	ちよ	によ	ひよ	みよ	りよ	ぎよ	じよ	ぢよ	びよ	ぴよ				
kyō	shō	chō	nyō	hyō	myō	ryō	gyō	jō	chō	byō	pyō				

Gambar 1. Huruf Hiragana

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2. Matrik Citra Digital

## 4. Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi dua 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 atau hasil *CT Scan*. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer.

### 4.1. Citra RGB

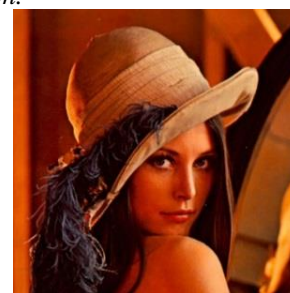
Citra RGB merupakan kombinasi dari elemen warna *Red* (merah), *Green* (hijau), dan *Blue* (biru). Format RGB ini direpresentasikan dengan menggunakan tiga *layer*. Setiap *pixel*nya merupakan kombinasi dari ketiga elemen warna tersebut, dengan tingkat warna 0 – 255. Pilihan skala 256 ini didasarkan pada cara mengungkap delapan digit bilangan biner yang digunakan oleh mesin komputer.

### 4.2. Citra Grayscale

Citra *grayscale* berbeda dengan citra "black and white". Pada konteks komputer, citra hitam putih hanya terdiri atas dua warna saja yaitu "hitam" dan "putih" saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna di antaranya sangat banyak. Terdapat tiga cara yang umum digunakan dalam mengubah citra RGB menjadi citra *grayscale*, yaitu Metode *Lightness*, Metode *Average*, dan Metode *Luminosity*.

## 5. Optical Character Recognition

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



Gambar 3. Citra RGB



Gambar 4. Citra Grayscale

## 6. Directional Feature Extraction

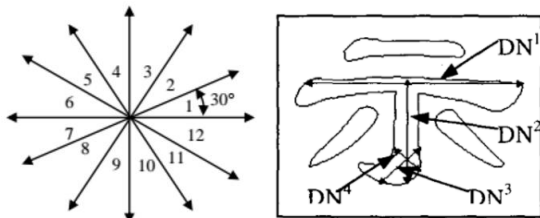
*Directional Feature Extraction* merupakan metode *Feature Extraction* dengan ciri arah dari setiap objek yang dideteksi. Pada metode ini, arah dibagi menjadi empat yaitu secara *horizontal* (—), *vertical* ( | ), *left up diagonal* ( / ) dan *right up diagonal* ( \ ). Untuk melakukan ekstraksi fitur berdasarkan arah dimulai dari piksel pertama yaitu piksel kiri paling atas hingga piksel paling kanan dan kemudian dilanjutkan lagi untuk nilai piksel di bawahnya. Untuk setiap piksel, dicek nilai tetangganya dari mulai piksel yang berada di atas piksel tersebut. Pengecekan dilakukan dengan searah jarum jam. Bila piksel tetangga bernilai 0 atau berwarna hitam, atur nilai sesuai ketentuan penomoran arah.

## 7. Support Vector Machine

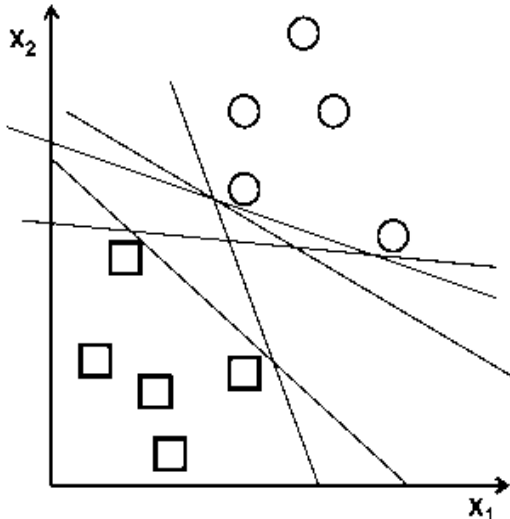
*Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun lalu, seperti *margin hyperplane*, kernel yang diperkenalkan oleh Aronszajn tahun 1950, dan juga konsep-konsep pendukung lain.



Gambar 5. Tahap-Tahap OCR



Gambar 6. Directional Feature Extraction



Gambar 7. Support Vector Machine

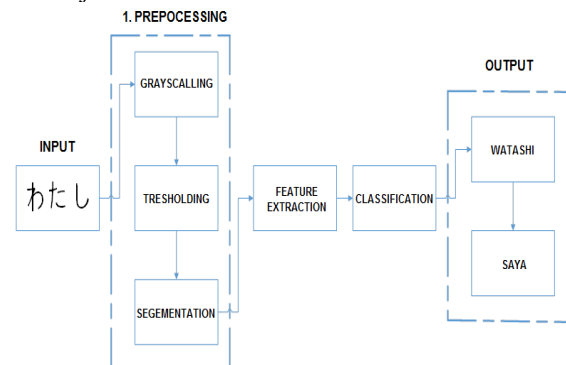
## 8. Perancangan Sistem

Sistem dirancang dan dikembangkan pada *smartphone* Android. Sistem yang dirancang bertujuan untuk mengenali setiap kata pada bahasa Jepang yang menggunakan huruf Hiragana dalam aplikasi penerjemah. Proses pengenalan kata bahasa Jepang diawali dengan pengambilan gambar kata Jepang. Proses pengambilan gambar dapat langsung dari *capture* kamera atau berupa gambar yang sudah ada pada *gallery handphone*. Gambar tersebut akan dideteksi menggunakan prinsip dasar *Optical Character Recognition* (OCR). Dengan menggunakan OCR, kata bahasa Jepang yang awalnya dalam bentuk *image* diubah menjadi *text*.

### 8.1. Pre-processing

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.

Setelah *grayscale*, dilanjutkan dengan proses *thresholding*. *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 8. Rancangan Sistem



Gambar 9.(a) Sebelum *Grayscale*,  
(b) Sesudah *Grayscale*



Gambar 10. Hasil *Thresholding*



Gambar 11. Hasil Segmentasi

Tahap selanjutnya segmentasi yang merupakan proses mempartisi citra menjadi beberapa daerah atau objek. Pada penelitian ini tahap segmentasi, dengan melakukan *cropping* setiap silabel dari sebuah kata.

Metode yang digunakan adalah *line segmentation*, baik secara *horizontal* maupun *vertical*, dan *Projection Block*. Silabel yang didapatkan dari segmentasi akan menjadi masukan proses ekstraksi ciri.

## 8.2. 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 [9].

## 8.3. 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  $1 \times 2500$ , yang 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 vektor ciri. Terdapat 104 kelas yang merupakan jumlah huruf Hiragana. Secara garis besar cara kerja dari *Support Vector Machine* adalah sebagai berikut:

- a. Vektor ciri dari setiap silabel Hiragana didapat dari ekstraksi ciri *Directional Feature Extraction* yaitu matrik yang berukuran  $1 \times 2500$  yang dijadikan data *input* awal. Data yang digunakan terbagi dua, yaitu data *training* dan data *testing*.
  - 1) Data *training* terdiri dari 104 silabel yang berasal dari huruf Hiragana dalam bahasa Jepang yang berukuran  $50 \times 50$  yang telah melewati proses *pre-processing* dan ekstraksi ciri. Setiap silabel terdiri dari tiga bentuk yaitu ada *bold*, *italic* dan normal silabel, sehingga ada  $3 \times 104$  silabel (312).
  - 2) Data *testing* adalah citra kata bahasa Jepang yang terdiri dari huruf Hiragana Jepang. Maksimal silabel pada kata tersebut ada 5 silabel. Citra tersebut akan di *pre-processing* dan diekstraksi cirinya. Dan disetiap kata tersebut akan diklasifikasi oleh *Support Vector Machine*.
- b. Data *input* awal 312 yang telah didapatkan ekstraksi cirinya yaitu berupa matrik  $312 \times 2500$ .

Setiap baris pada matriks mewakili setiap ciri pada silabel huruf Hiragana Jepang.

- c. Data *input* awal 312 yang telah didapatkan ekstraksi cirinya yaitu berupa matrik  $312 \times 2500$ . Setiap baris pada matriks mewakili setiap ciri pada silabel huruf Hiragana Jepang.

## 9. 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 dua silabel dan tiga silabel yang sama. Masing-masing kata diambil dengan melakukan kontrol terhadap jarak pengambilan dan pencahayaan.

## 10. Pengujian Waktu Respon Sistem

Tujuan pengujian ini adalah mengetahui pengaruh kecepatan *clock* prosesor terhadap kecepatan waktu respon 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 respon yang diperlukan aplikasi untuk menerjemahkan citra masukan.

## 11. Pengujian Ketepatan Segmentasi

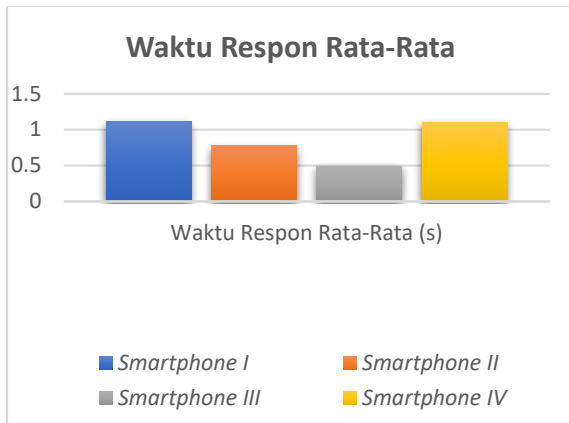
Pengujian ini bertujuan untuk mengetahui ketepatan dalam proses segmentasi pada setiap kata yang diujikan. Pengujian akan dilakukan pada setiap kata berdasarkan jumlah silabel. Dari 130 data yang diuji terdapat 100 data yang hasil segmentasinya benar dan 30 data mengalami kesalahan dalam segmentasi. Pada pengujian ini dapat dilihat bahwa akurasi segmentasi tertinggi adalah 94% untuk dua silabel dan 90% untuk tiga silabel.

## 12. Pengujian Deteksi Kata

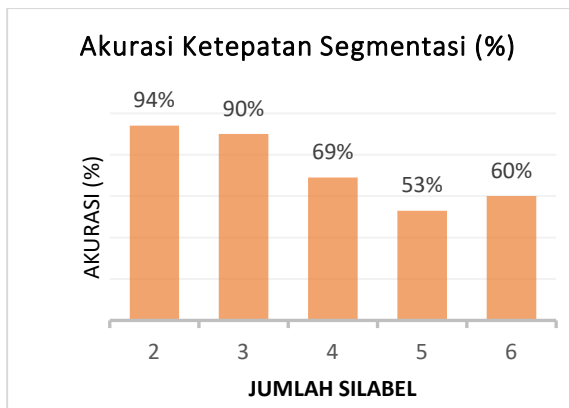
Tujuan dari pengujian ini adalah mengetahui tingkat keakuratan sistem dalam melakukan pendeteksian setiap citra masukan. Dari 130 data yang telah diuji, ada 107 data yang dideteksi benar dan 23 dideteksi salah. Pengujian ini dilakukan

**Tabel 1. Hasil Pengujian Ukuran Resolusi Citra**

Resolusi	Silabel	Akurasi (%)
2048×1152	Silabel 2	71.43%
	Silabel 3	60%
3264×1836	Silabel 2	80%
	Silabel 3	66.67%
4128×3096	Silabel 2	85.71%
	Silabel 3	83.33%
5312×2988	Silabel 2	91.42%
	Silabel 3	90%



Gambar 12. Hasil Pengujian Waktu Respon Rata-Rata



Gambar 13. Hasil Pengujian Ketepatan Segmentasi

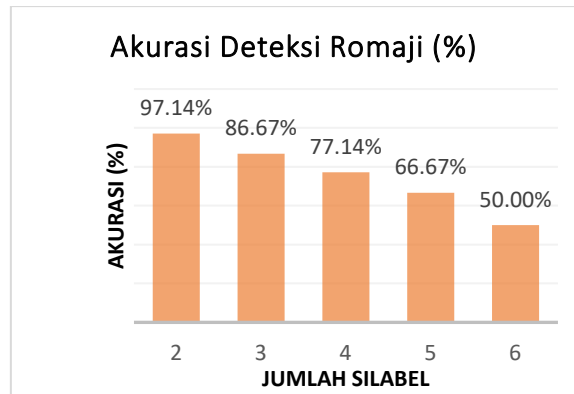
berdasarkan banyak silabel yang terdapat pada setiap kata yang akan dideteksi. Silabel terdiri dari 2, 3, 4 dan 5. Grafik pada Gambar 14 menjelaskan nilai akurasi tertinggi pada ketepatan deteksi ini pada kata yang terdiri dari dua silabel adalah 97.14%. Dari 35 jumlah kata terdapat satu kata yang terdeteksi salah.

### 13. Pengujian Ketepatan Penerjemah

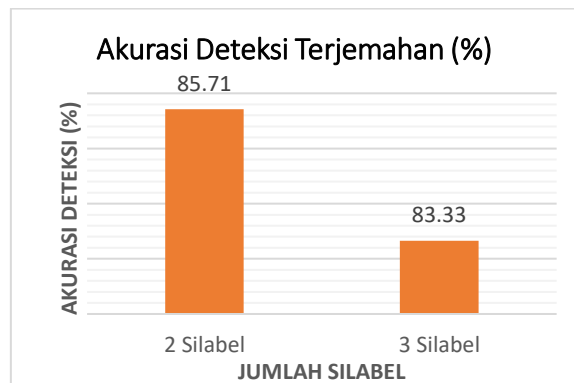
Tujuan dari pengujian ini adalah mengetahui tingkat keakuratan sistem dalam melakukan penerjemahan citra masukan. Berdasarkan pengujian segmentasi, maka dipilih kata-kata bahasa Jepang yang terdiri atas dua dan tiga silabel, masing-masing 30 kata. Hasil pengujian menunjukkan nilai akurasi yang cukup 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.

### 14. Pengujian Beta

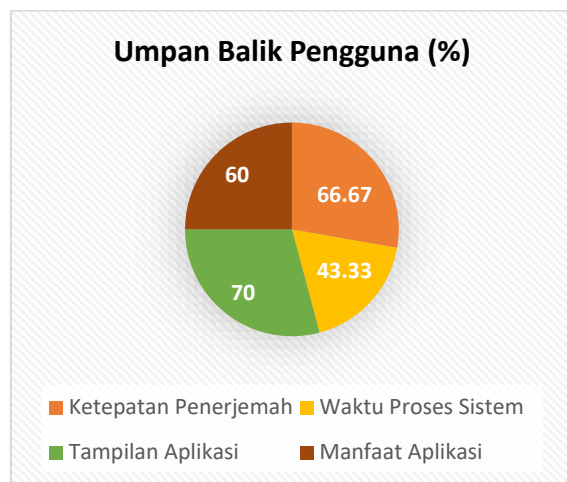
Pengujian ini dilakukan pada sisi pengguna secara objektif. Pengujian *Beta* merupakan pengujian berdasarkan umpan-balik dari pengguna aplikasi. Terdapat empat poin yang diujikan kepada responden, yaitu tampilan aplikasi, ketepatan



Gambar 14. Hasil Pengujian Deteksi Kata



Gambar 15. Hasil Pengujian Ketepatan Penerjemahan



Gambar 16. Umpan Balik Pengguna

penerjemahan, lama waktu respon, serta tingkat kebermanfaatan aplikasi. Pengujian *Beta* dilakukan pada 30 responden yang memenuhi kriteria yang ditentukan yaitu diantaranya para pengguna *smartphone* android.

### 15. Kesimpulan

Penelitian ini menghasilkan kesimpulan sebagai berikut.

- Deteksi kata pada Huruf Hiragana dalam bahasa Jepang dapat berhasil dengan menggunakan

- metode *Directional Feature Extraction* (DFE) pada proses ekstraksi ciri dan hasil ekstraksi ciri diklasifikasikan menggunakan metode *Support Vector Machine* (SVM) dengan nilai akurasi sebesar 85,71% dengan waktu respon sistem pada *smartphone* android dengan prosesor 1,5 GHz *Quad Core* adalah 0,778 detik.
- b. Pada pengujian berdasarkan ukuran resolusi kamera akurasi tertinggi mencapai 91,42% dengan resolusi 5312×2988 piksel untuk dua silabel sedangkan untuk tiga silabel adalah 90%. Dari pengujian dapat disimpulkan bahwa ukuran resolusi sebuah kamera, posisi objek yang akan dideteksi dan pencahayaan mempengaruhi ketepatan dalam menerjemahkan kata.
  - c. Perancangan dengan menggunakan ekstraksi ciri *Directional Feature Extraction* (DFE) serta klasifikasi dengan menggunakan metode *Support Vector Machine* (SVM) dapat diterapkan pada perangkat *smartphone* android dalam bentuk aplikasi. Berdasarkan pengujian *Beta* dinyatakan bahwa sebanyak 60% responden memberikan penilaian bahwa aplikasi bermanfaat untuk pengguna dalam menerjemahkan kata huruf Hiragana dalam bahasa Jepang dan 70% responden menilai tampilan aplikasi ini menarik.

#### 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] Murtiwyati 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.