

IMPLEMENTASI METODE LOCAL BINARY PATTERNS UNTUK PENGENALAN POLA HURUF HIRAGANA DAN KATAKANA PADA SMARTPHONE

Rabiuldien Amat¹⁾, Jayanti Yusmah Sari²⁾, Ika Purwanti Ningrum³⁾

^{1,2,3)}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo

Kampus Hijau Bumi Tridharma Anduonohu, JL. HEA. Mokodompit Kendari, 93231

e-mail: rabiuldien@yahoo.com¹⁾, jayanti.yusmah.sari@gmail.com²⁾, ika.purwanti.n@gmail.com³⁾

ABSTRAK

Pada saat ini Jepang merupakan salah satu negara yang menjadi pilihan bagi pelajar dunia untuk melanjutkan pendidikan. Pelajar yang ingin melanjutkan pendidikan ke Jepang harus mempelajari Bahasa Jepang. Huruf dasar yang harus dikuasai oleh pembelajar Bahasa Jepang pemula adalah Huruf Hiragana dan Katakana. Dalam mempelajari Huruf Hiragana dan Katakana pembelajar pemula mengalami kesulitan yaitu huruf tersebut memiliki pelafalan yang sama, namun penulisan dari huruf tersebut berbeda.

Penelitian ini bertujuan untuk menerapkan metode Local Binary Patterns (LBP) sebagai metode ekstraksi fitur untuk pengenalan pola Huruf Hiragana dan Katakana. Cara kerja operator LBP yaitu dengan mencari nilai tengah dari suatu kernel berukuran 3 x 3, dengan melakukan perbandingan nilai tengah piksel dengan nilai tetangga piksel terdekat pada citra grayscale.

Implementasi metode Local Binary Pattern (LBP) untuk pengenalan pola Huruf Hiragana dan Katakana pada Smartphone berbasis Android telah diujikan pada 460 data sampel. Hasil pengujian menunjukkan akurasi pengenalan yang cukup baik yaitu sebesar 81,1%.

Kata Kunci: Hiragana, Katakana, Local Binary Patterns.

ABSTRACT

Nowadays, Japan is one of the most famous study destination for students all around the world. Students who want to continue their study in Japan must learn Japanese. Basic letters such as Hiragana and Katakana must be mastered by those students. The main problem in mastering Hiragana and Katakana is that the letters have the same pronunciation but with different lettering.

This mini thesis aims to implement Local Binary Patterns (LBP) as method of feature extraction for Hiragana and Katakana pattern recognition. LBP operator works by finding the middle value of a kernel with a size of 3 x 3, by comparing the middle value of a pixel with the nearest neighboring pixels in the grayscale image.

The implementation of Local Binary Patterns (LBP) method for Hiragana and Katakana pattern recognition on Android-based Smartphone has been tested on 460 data samples. The test results show a good recognition accuracy of 81.1%.

Keywords: Hiragana, Katakana, Local Binary Patterns.

I. PENDAHULUAN

Pada saat ini Jepang merupakan salah satu negara yang menjadi pilihan bagi pelajar dunia untuk melanjutkan pendidikan, tanpa terkecuali pelajar Indonesia. Pada tahun 2004 tercatat sejumlah 1.451 pelajar Indonesia yang melanjutkan studinya ke Jepang, dan pada tahun 2015 jumlah tersebut meningkat menjadi 3.600 [1]. Data tersebut menunjukkan bahwa minat pelajar Indonesia untuk melanjutkan pendidikannya ke Jepang dari tahun ke tahun semakin meningkat.

Huruf yang digunakan dalam penulisan Bahasa Jepang ada 3, yaitu Hiragana, Katakana dan Kanji [2]. Huruf dasar yang harus dikuasai oleh pembelajar Bahasa Jepang pemula adalah Huruf Hiragana dan Katakana, sedangkan Huruf Kanji merupakan huruf untuk pembelajar tingkat lanjut.

Dalam proses mempelajari Huruf Hiragana dan Katakana dibutuhkan perangkat ajar untuk mempermudah proses belajar. Perangkat ajar yang digunakan sebaiknya bersifat *mobile* karena piranti *mobile* memiliki kemampuan untuk bergerak dan mampu memenuhi kebutuhan pengguna tanpa harus terikat pada satu tempat atau sumber daya [3].

Saat ini, telah banyak aplikasi pembelajaran Bahasa Jepang khususnya Huruf Hiragana dan Katakana yang bisa diunduh dengan mudah dan memiliki banyak variasi dalam cara pembelajaran penulisan hurufnya. Akan tetapi, aplikasi tersebut belum memiliki fungsi untuk mengenali pola huruf yang diinputkan *user*. Oleh karena itu, pada penelitian ini akan dibuat sebuah aplikasi yang dapat mengenali pola Huruf Hiragana dan Katakana.

Beberapa metode telah digunakan dalam penelitian untuk mengenali pola huruf seperti Direction Feature Extraction untuk mengenali pola Huruf Hiragana dan Katakana [4] dan Pola Busur Terlokalisasi untuk mengenali

pola Huruf Kanji [5]. Namun kedua metode tersebut belum menunjukkan hasil pengenalan yang optimal untuk pengenalan pola huruf. Hal ini ditunjukkan dengan tingkat akurasi pengenalan kedua metode tersebut yang hanya berkisar antara 60% sampai dengan 75%.

Pada tahun 2014, Biglari dkk menggunakan metode Local Binary Patterns (LBP) untuk pengenalan huruf Persia/Arabic [6]. Metode LBP digunakan sebagai metode ekstraksi fitur/ciri. Cara kerja operator LBP yaitu dengan mencari nilai tengah dari suatu kernel berukuran 3 x 3, dengan melakukan perbandingan nilai tengah piksel dengan nilai tetangga piksel terdekat pada citra *grayscale*. Apabila nilai tengah bernilai sama atau lebih besar, diberi nilai 1 selain itu, diberi nilai 0. Kemudian, nilai LBP diperoleh dari penjumlahan dua pangkat nilai angka yang bernilai 1. Hasil pengujian menunjukkan persentase akurasi yang sangat baik yaitu sebesar 99.72%. Sehingga penelitian tersebut menyimpulkan bahwa kelebihan dari LBP yaitu mudah diimplementasikan dan merupakan metode ekstraksi fitur yang cukup cepat dengan proses komputasi yang rendah.

Berdasarkan hal tersebut maka penulis akan melakukan penelitian dengan judul “Implementasi Metode Local Binary Patterns (LBP) untuk Pengenalan Pola Huruf Hiragana dan Katakana pada *Smartphone* Berbasis Android”. Penelitian ini diharapkan dapat menghasilkan aplikasi yang dapat mengenali pola Huruf Hiragana dan Katakana sehingga dapat digunakan sebagai media pendamping pembelajar pemula dalam mempelajari Huruf Hiragana dan Katakana dasar.

II. TINJAUAN PUSTAKA

1. Metode Local Binary Patterns

LBP adalah metode analisis tekstur yang menggunakan model statistika dan struktur [7]. LBP pertama kali diperkenalkan oleh Timo Ojala. Operator LBP menggunakan perbandingan nilai keabuan dari piksel-piksel ketetanggaan. Operator dasar LBP berukuran 3 x 3 menggunakan 8 piksel ketetanggaan i_n dari sebuah piksel tengah i_c . Piksel ketetanggaan ke- n tersebut di-*threshold* menggunakan nilai keabuan dari piksel tengah seperti yang ditunjukkan pada persamaan (1) dan fungsi *thresholding* $s(x)$ seperti yang ditunjukkan pada persamaan (2). Kode *binary* hasil operator LBP piksel ketetanggaan akan digunakan untuk merepresentasikan fitur dari piksel tengah i_c [8].

$$LBP(x_c, y_c) = \sum_{n=0}^7 s(i_n - i_c)2^n \tag{1}$$

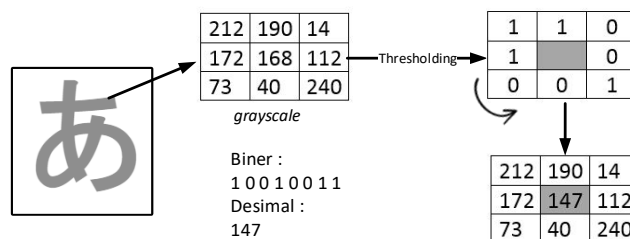
$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{2}$$

Gambar 1 merupakan ilustrasi dari proses LBP. Proses pertama adalah melakukan pengurangan piksel ketetanggaan dengan piksel tengah (1). Selanjutnya hasil pengurangan di-*threshold* menggunakan persamaan (2), jika hasilnya ≥ 0 maka diberi nilai 1 dan jika hasilnya < 0 maka diberi nilai 0. Setelah itu, nilai biner piksel ketetanggaan akan disusun berlawanan arah jarum jam dan 8 bit biner tersebut dikonversi ke dalam nilai desimal untuk menggantikan nilai piksel tengah i_c .

Menurut [6] kelebihan dari LBP adalah mudah diimplementasikan dan tingkat komputasinya lebih rendah sehingga tidak membutuhkan waktu yang lama dalam ekstraksi fitur.

2. Euclidean Distance

Euclidean Distance adalah metrik yang paling sering digunakan untuk menghitung kesamaan dari dua vektor dengan menghasilkan nilai yang berupa jarak dari kedua vektor tersebut. Nilai Euclidean Distance diperoleh dari akar kuadrat selisih 2 vektor yang akan dihitung jaraknya. Untuk menghitung nilai Euclidean Distance dari vektor fitur masukan dan vektor fitur pembanding digunakan persamaan (3) [9].



Gambar 1 Ilustrasi LBP

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \tag{3}$$

Keterangan:

- d_{ij} = nilai/besaran jarak
- n = panjang vektor
- x_{ik} = vektor fitur masukan
- x_{jk} = vektor fitur pembandingan

Semakin kecil nilai d_{ij} maka semakin mirip kedua vektor yang dicocokkan. Sebaliknya, semakin besar nilai d_{ij} maka semakin berbeda pula kedua vektor yang dicocokkan [9].

III. METODE PENELITIAN

Pada aplikasi ini terdapat 4 menu utama yaitu menu Hiragana, Katakana, latihan dan tentang. Pada menu latihan diimplementasikan metode Local Binary Patterns (LBP) untuk mengidentifikasi benar atau salahnya pola yang diinputkan oleh *user*. Adapun perancangan proses terbagi atas dua yaitu perancangan proses citra sampel dan perancangan proses citra uji yang ditunjukkan pada Gambar 2.

1. Konversi Citra RGB ke Grayscale

Proses konversi citra RGB ke *grayscale* bertujuan untuk mengubah citra huruf dari citra RGB ke citra *grayscale*. Adapun tahap pertama dari proses konversi citra RGB ke *grayscale* adalah inisialisasi nilai x dan y . Tahap selanjutnya mengambil nilai warna merah, hijau dan biru. Ketiga nilai tersebut dijumlahkan dan dibagi 3 sehingga didapatkan nilai *grayscale*. Selanjutnya nilai *grayscale* tersebut digunakan untuk menggantikan nilai warna merah, hijau dan biru pada piksel (x, y) . Selanjutnya dilakukan penambahan nilai y untuk memproses piksel selanjutnya. Setelah nilai $y \leq \text{tinggi citra}$ maka dilakukan penambahan nilai x . Setelah semua piksel telah diproses maka terbentuklah citra *grayscale*.

2. Cropping

Proses *cropping* adalah proses seleksi sub piksel dari citra menjadi citra baru. Proses ini bertujuan untuk mengambil objek huruf pada citra dan membuang bagian yang tidak diperlukan sehingga proses ekstraksi fitur lebih minim.

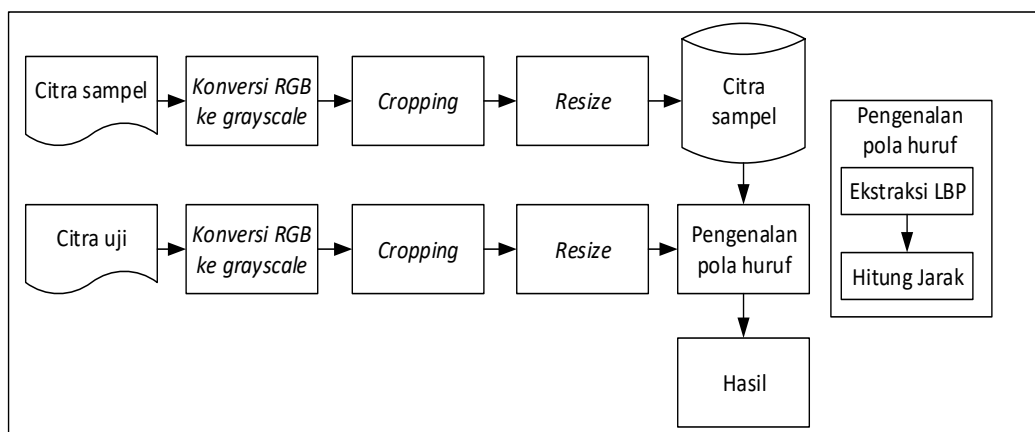
3. Resize

Proses *resize* bertujuan untuk menormalisasi ukuran citra hasil *cropping* ke ukuran 128 x 128 piksel. Pada penelitian ini digunakan fungsi penskalaan yang tersedia pada Android Studio.

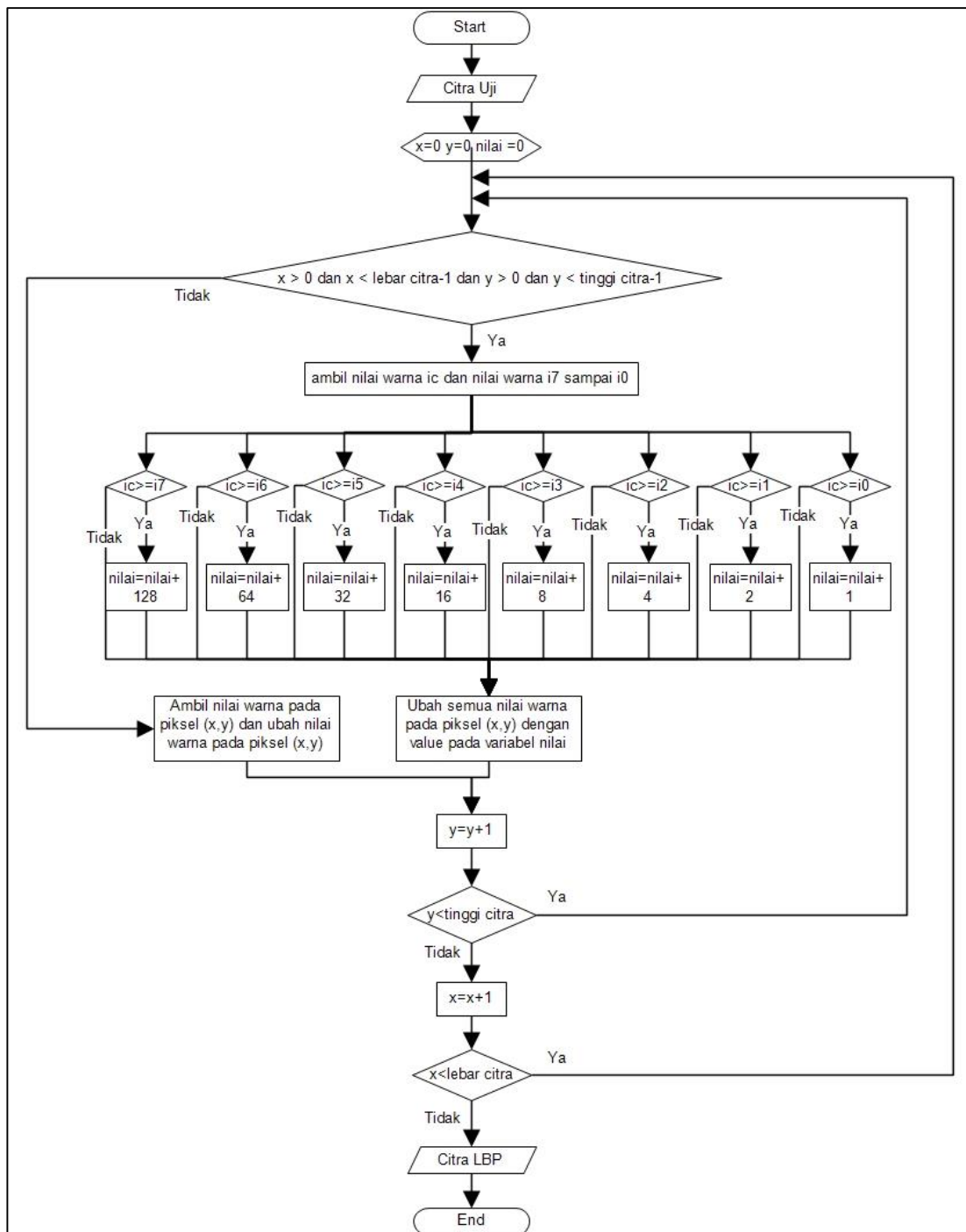
4. Local Binary Patterns (LBP)

Pada penelitian ini metode ekstraksi fitur yang digunakan adalah Local Binary Patterns (LBP). Adapun tahap-tahap dari proses LBP dengan menggunakan persamaan (1) dan *thresholding* berdasarkan Persamaan (2) yang ditunjukkan Gambar 3. Adapun tahap-tahapnya yaitu:

1. Langkah pertama adalah melakukan inisialisasi terhadap (x, y) dan variabel *nilai* yang berguna sebagai penampung nilai yang nantinya menggantikan nilai piksel tengah.
2. Langkah selanjutnya adalah menggunakan kondisi $x > 0$ dan $x < \text{lebar citra dikurang 1}$ dan $y > 0$ dan $y < \text{tinggi citra dikurang 1}$.
3. Apabila kondisi tersebut terpenuhi maka langkah selanjutnya adalah mengambil nilai piksel tengah i_c dan piksel ketetanggaan dari i_7 sampai i_0 .



Gambar 2 Gambaran umum sistem



Gambar 3 Flowchart local binary pattern

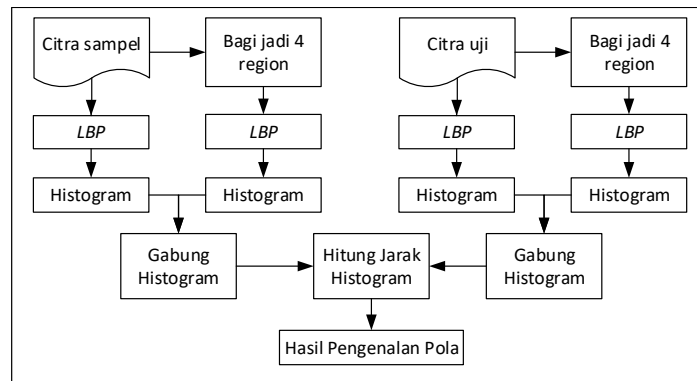
4. Langkah selanjutnya melakukan perbandingan nilai piksel tengah i_c dengan piksel ketetanggaannya, apabila nilai $i_c \geq$ piksel ketetanggaannya maka dilakukan penjumlahan pada variabel $nilai$ sesuai dengan bobot masing-masing piksel ketetanggaannya.
5. Langkah selanjutnya adalah mengubah semua nilai warna pada piksel (x, y) dengan $value$ pada variabel $nilai$.
6. Selanjutnya dilakukan penjumlahan nilai y dan memproses piksel selanjutnya. Setelah semua piksel diproses maka akan terbentuk citra hasil LBP.

5. Pengenalan Pola Huruf

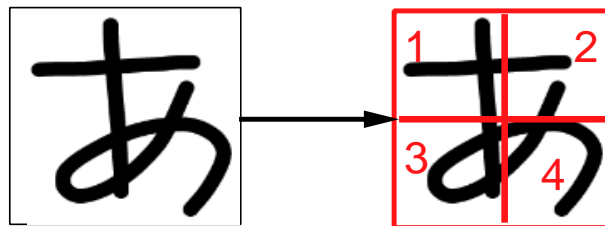
Berdasarkan Gambar 4 tahap-tahap proses pengenalan pola pada penelitian ini yaitu:

1. Membagi citra sampel dan citra uji menjadi 4 region seperti yang ditunjukkan pada Gambar 5, sehingga masing-masing region berukuran 64×64 .

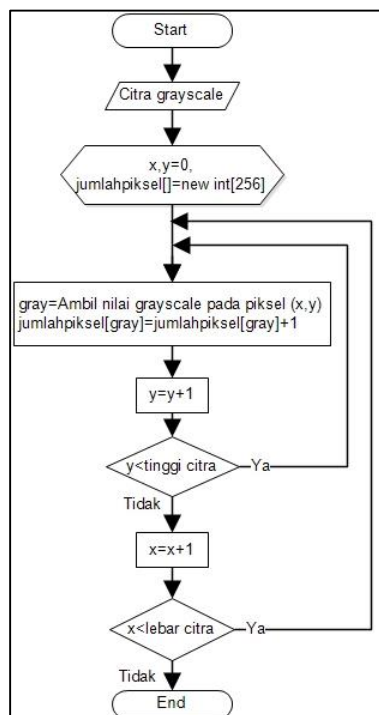
2. Mengekstraksi fitur dari citra sampel region, citra uji region, citra sampel dan citra uji.
3. Setelah proses ekstraksi selesai, tahap selanjutnya yaitu memproses histogram masing-masing citra. Proses histogram ditunjukkan Gambar 6.
4. Menggabungkan histogram dari citra asli dengan citra yang telah dibagi menjadi region.
5. Langkah selanjutnya menghitung jarak dari histogram gabungan citra uji dengan histogram gabungan citra sampel.
6. Setelah hasil perhitungan jarak dari 5 citra sampel didapatkan, tahap selanjutnya mencari nilai persen kemiripan.
7. Apabila persen kemiripan \geq nilai ambang maka hasil pengenalan berupa “benar”, dan apabila persen kemiripan $<$ nilai ambang maka hasil pengenalan “salah”.



Gambar 4 Gambaran proses pengenalan pola huruf



Gambar 5 Ilustrasi pembagian citra



Gambar 6 flowchart histogram

IV. HASIL UJI COBA DAN PEMBAHASAN

Pengujian Aplikasi pada penelitian ini dilakukan sebanyak dua kali. Pengujian I bertujuan untuk mendapatkan nilai *threshold* kemiripan setiap Huruf Hiragana dan Katakana. Nilai *threshold* diperoleh dengan menghitung rata-rata. Nilai *threshold* akan digunakan sebagai penentu hasil pengenalan pola huruf pada pengujian II. Hasil dari pengujian I ditunjukkan pada Tabel 1.

TABEL 1 HASIL PENGUJIAN I UNTUK Mencari Nilai THRESHOLD

Huruf	Hasil Percobaan ke-					Rata – rata (%)
	1(%)	2(%)	3(%)	4(%)	5(%)	
Hiragana A	85,3	74,7	82,6	82,3	83	81,58
Hiragana I	87,6	89	85	89,2	92,5	88,66
Hiragana U	92,4	90,2	92	94,5	98,3	93,48
Hiragana E	85,8	68,6	89,7	66,3	85,6	79,2
Hiragana O	76,7	60,7	63,5	71,9	71,4	68,84
Hiragana Ka	74,6	77,6	79,9	86,6	76,1	78,96
Hiragana Ki	72,7	68,5	94,6	91,3	78,9	81,2
Hiragana Ku	95,1	98,8	95	99,3	93,3	96,3
Hiragana Ke	74	78,9	88,2	80,4	87,4	81,78
Hiragana Ko	79,7	86,3	92,8	79,6	91,5	85,98
Hiragana Sa	81,6	82,7	93,9	79,5	82,6	84,06
Hiragana Shi	80,4	79	95,7	91	84,5	86,12
Hiragana Su	79,1	82,8	86,4	80,9	82,5	82,34
Hiragana Se	78,6	75,3	76,2	77,9	72,7	76,14
Hiragana So	87,2	78,8	71,6	77,3	84,3	79,84
Hiragana Ta	73,7	83	70,2	76,7	82	77,12
Hiragana Chi	62,7	61,2	77,6	81,2	61,2	68,78
Hiragana Tsu	88,3	79,3	90,4	86,3	78,4	84,54
Hiragana Te	84,1	72,5	86,5	85,4	89,6	83,62
Hiragana To	82,2	81,8	85,3	87,5	87,7	84,9
Hiragana Na	88,1	90,9	90,7	87,3	81,1	87,62
Hiragana Ni	45,7	55,8	47,4	62,6	49,9	52,28
Hiragana Nu	85,1	79	87,5	81,2	77,1	81,98
Hiragana Ne	76,5	86,6	85,7	78,1	84,6	82,3
Hiragana No	63,9	67,9	71	65,2	72,8	68,16
Hiragana Ha	79,6	84,3	91,1	81	78,1	82,82
Hiragana Hi	62,1	78,2	64,5	87,5	82,3	74,92
Hiragana Fu	60,7	75,4	78,6	73,9	65,1	70,74
Hiragana He	89,6	81,1	90,4	78,7	89,2	85,8
Hiragana Ho	87,7	76,7	84	82,3	74,9	81,12
Hiragana Ma	78,8	81,3	82,6	76,5	72,8	78,4
Hiragana Mi	51,7	45,6	62,3	63,8	48,6	54,4
Hiragana Mu	78,9	90	87,1	86,8	86,3	85,82
Hiragana Me	61,8	72,4	69,1	65,1	73,1	68,3
Hiragana Mo	88,7	77,7	75,4	92	86,2	84
Hiragana Ra	73,9	81,6	81,3	78,2	83,1	79,62






















Hiragana Ri	81,7	65,1	70,2	77,6	70,4	73
Hiragana Ru	50,7	69,9	74	55,6	70,4	64,12
Hiragana Re	80,2	89,5	88,4	84,4	87,9	86,08
Hiragana Ro	89,5	58,8	85	68,9	86,5	77,74
Hiragana Ya	66,2	67,4	77,3	77,5	76,9	73,06
Hiragana Yu	79,9	71,8	74,9	86,4	78,1	78,22
Hiragana Yo	77,6	73,2	84,9	68,1	83,1	77,38
Hiragana Wa	82,8	74,9	83,1	91,5	87,4	83,94
Hiragana Wo	79,9	83,6	75,6	77,1	78,1	78,86
Hiragana N	77,2	64,9	67,6	73,6	82,4	73,14
Katakana A	84,3	76,3	88,3	77,8	87,5	82,84
Katakana I	79,4	79,3	76,1	77	73,2	77
Katakana U	85,9	79,3	81,7	78,1	86,3	82,26
Katakana E	86,4	76,3	77,7	72,1	73,4	77,18
Katakana O	83,4	79	84	85,9	74,7	81,4
Katakana Ka	75,5	50,3	63,5	55,3	66,7	62,26
Katakana Ki	87,3	62,6	94	70,8	87,9	80,52
Katakana Ku	81	67,9	89,7	83,8	84,2	81,32
Katakana Ke	92,9	91,4	91,6	97	89,2	92,42
Katakana Ko	83,2	83,5	84	76,6	80,1	81,48
Katakana Sa	92,4	93,6	94,3	93,6	89	92,58
Katakana Shi	92,1	90	91,7	91,1	86,8	90,34
Katakana Su	73,9	67,4	70,5	71	76,7	71,9
Katakana Se	86,2	76,1	67	77,9	79,8	77,4
Katakana So	88,8	84,1	81	88,7	88,5	86,22
Katakana Ta	73,9	67,1	75,5	68,1	80,5	73,02
Katakana Chi	74	78,7	85,2	77,9	86,8	80,52
Katakana Tsu	84,5	76,8	73,6	86,8	83,5	81,04
Katakana Te	80,6	81,9	83,1	81,4	79,5	81,3
Katakana To	79,1	85,6	86,6	84,1	75,5	82,18
Katakana Na	92	89,4	87	84,3	88,5	88,24
Katakana Ni	91,3	89,7	82,8	76,9	82	84,54
Katakana Nu	62	61,6	72,8	78,7	79,9	71
Katakana Ne	80,1	77,6	76,5	74,8	88	79,4
Katakana No	93,2	96,6	95,9	93,9	87,9	93,5
Katakana Ha	96,4	97,6	98,2	94,2	90,6	95,4
Katakana Hi	87,2	84,8	80,4	73,6	83,3	81,86
Katakana Fu	83	84,6	74,9	78,6	85,4	81,3
Katakana He	96,3	95	96,1	95,5	93,7	95,32
Katakana Ho	87,5	88,2	83,2	87,9	85,9	86,54
Katakana Ma	65,1	73,9	88,2	84,9	77,3	77,88
Katakana Mi	81,9	82,4	91,2	84,6	72,8	82,58
Katakana Mu	88,6	89,9	76,1	94	92,2	88,16
Katakana Me	81	75,3	82	83,5	81,7	80,7

Katakana Mo	74,4	76,4	78,9	85	78,7	78,68
Katakana Ra	84,8	75,3	75,6	82,8	76,6	79,02
Katakana Ri	74	65,6	63,3	72,2	83,6	71,74
Katakana Ru	89,9	95,7	85,5	86,4	85,8	88,66
Katakana Re	92,6	90,2	93,4	89,9	85,2	90,26
Katakana Ro	90,6	73,3	75,1	90,4	75,9	81,06
Katakana Ya	81,2	79,5	76,6	91,1	76,5	80,98
Katakana Yu	87,3	82	75	89,6	88,6	84,5
Katakana Yo	76	79,4	80,1	81	73	77,9
Katakana Wa	74,6	71,9	71,8	84	80,2	76,5
Katakana Wo	80	82,4	83,9	83,9	86,5	83,34
Katakana N	75,3	73,8	71,7	76,1	86,3	76,64
						52,28
						96,3
						80,4

Tabel 1 merupakan hasil pengujian Huruf Hiragana dan Huruf Katakana. Dari tabel tersebut dapat diketahui bahwa rata-rata kemiripan untuk semua huruf adalah 80,4%. Nilai tersebut tidak dapat dijadikan nilai *threshold* pada aplikasi, karena terdapat beberapa huruf yang memiliki nilai kemiripan di bawah 80,4% seperti Huruf Hiragana Mi, Hiragana Me, Hiragana Ni, Hiragana Hi dan Hiragana Ya. Oleh karena itu, pada aplikasi ini digunakan nilai rata-rata kemiripan dari setiap huruf sebagai nilai *threshold* masing-masing huruf.

Pengujian II bertujuan untuk mendapatkan akurasi pengenalan pola dari aplikasi. Pada pengujian II dilakukan percobaan sebanyak tiga kali yang dilakukan oleh tiga *user*. Pada pengujian ini setiap *user* melakukan pengujian terhadap 10 Huruf Hiragana dan 10 Katakana yang dilakukan secara acak. Jadi jumlah percobaan untuk setiap *user* adalah 60 kali percobaan. Hasil akurasi dari setiap percobaan dievaluasi menggunakan penilaian benar atau salah oleh pakar (A) dan penilaian benar atau salah berdasarkan nilai *threshold* yang diperoleh dari pengujian I (B). Hasil evaluasi (C) bernilai benar apabila penilaian dari pakar dan dari nilai *threshold* bernilai sama. Persentase akurasi untuk setiap *user* diperoleh dari perbandingan antara hasil evaluasi percobaan yang benar dengan jumlah percobaan yang dilakukan.

TABEL 2 HASIL PENGUJIAN II UNTUK MEMPEROLEH AKURASI

Huruf	Percobaan											
	I			II			III					
	Citra	A	B	C	Citra	A	B	C	Citra	A	B	C
<i>User ke-1</i>												
Hiragana E		0	0	1		1	1	1		1	0	0
Hiragana MA		0	0	1		1	1	1		1	1	1
Hiragana MO		1	0	0		1	1	1		1	1	1
Hiragana O		0	0	1		1	1	1		1	1	1
Hiragana RI		1	1	1		1	1	1		1	1	1
Hiragana SA		0	0	1		0	0	1		1	1	1
Hiragana SHI		1	1	1		1	1	1		1	1	1




Hiragana SO	そ	0	0	1	そ	1	1	0	そ	1	0	0
Hiragana TA	た	0	0	1	た	1	1	1	た	1	1	1
Hiragana TE	て	0	0	1	て	1	1	1	て	1	1	1
Katakana FU	フ	0	0	1	フ	1	1	1	フ	1	1	1
Katakana HA	ハ	0	0	1	ハ	1	0	0	ハ	1	1	1
Katakana KO	コ	0	0	1	コ	0	0	1	コ	1	1	1
Katakana MA	マ	0	0	1	マ	1	1	1	マ	1	1	1
Katakana NA	ナ	0	0	1	ナ	1	0	0	ナ	1	1	1
Katakana SA	サ	0	0	1	サ	1	0	0	サ	1	0	0
Katakana SHI	シ	0	0	1	シ	1	1	1	シ	1	0	0
Katakana SU	ス	0	0	1	ス	1	1	1	ス	1	1	1
Katakana U	ウ	0	0	1	ウ	1	1	1	ウ	1	1	1
Katakana YO	ヨ	0	0	1	ヨ	1	0	0	ヨ	1	0	0

Akurasi Pengenalan dari User ke-1 **81,7%**

User ke-2

Hiragana HA	は	1	1	1	は	1	1	1	は	1	1	1
Hiragana HI	ひ	1	1	1	ひ	1	1	1	ひ	1	1	1
Hiragana KA	か	1	1	1	か	1	1	1	か	1	1	1
Hiragana KO	こ	1	1	1	こ	1	1	1	こ	1	1	1
Hiragana NU	ぬ	1	1	1	ぬ	1	1	1	ぬ	1	1	1
Hiragana RI	り	1	1	1	り	1	1	1	り	1	1	1
Hiragana SA	さ	1	1	1	さ	1	0	0	さ	1	0	0
Hiragana SO	そ	1	1	1	そ	1	1	1	そ	1	0	0
Hiragana TO	と	1	1	1	と	1	1	1	と	1	1	1
Hiragana WO	を	0	0	1	を	1	1	1	を	1	1	1
Katakana RE	レ	0	0	1	レ	1	1	1	レ	1	1	1
Katakana RU	ロ	0	0	1	ロ	1	1	1	ロ	1	1	1
Katakana TO	ト	1	1	1	ト	1	0	0	ト	1	1	1

Katakana WO	ヲ	1	1	1	ヲ	1	1	1	ヲ	1	1	1
Katakana HO	ホ	0	0	1	ホ	1	0	0	ホ	1	1	1
Katakana MA	マ	1	1	1	マ	1	1	1	マ	1	1	1
Katakana NA	ナ	1	1	1	ナ	1	1	1	ナ	1	1	1
Katakana NE	ネ	0	1	0	ネ	1	1	1	ネ	1	1	1
Katakana SE	セ	1	1	1	セ	1	1	1	セ	1	1	1
Katakana TSU	ツ	0	0	1	ツ	1	1	1	ツ	1	1	1
Akurasi Pengenalan dari User ke-2						90%						
Hiragana HI	ひ	1	1	1	ひ	1	1	1	ひ	1	1	1
Hiragana HO	ほ	1	1	1	ほ	1	0	0	ほ	1	0	0
Hiragana KO	こ	0	0	1	こ	1	1	1	こ	1	0	0
Hiragana N	ん	1	0	0	ん	1	1	1	ん	1	1	1
Hiragana O	お	1	1	1	お	1	1	1	お	1	1	1
Hiragana RU	る	1	1	1	る	1	1	1	る	1	1	1
Hiragana TA	た	1	0	0	た	1	0	0	た	1	1	1
Hiragana WA	わ	1	1	1	わ	1	1	1	わ	1	1	1
Hiragana A	あ	0	0	1	あ	1	0	0	あ	1	0	0
Hiragana ME	め	0	1	0	め	1	1	1	め	1	1	1
Katakana FU	フ	1	1	1	フ	1	1	1	フ	1	1	1
Katakana KO	コ	1	0	0	コ	1	0	0	コ	1	1	1
Katakana NA	ナ	0	0	1	ナ	1	1	1	ナ	1	1	1
Katakana RA	ラ	1	0	0	ラ	1	0	0	ラ	1	0	0
Katakana TO	ト	1	0	1	ト	1	1	1	ト	1	1	1
Katakana YA	ヤ	0	0	1	ヤ	1	1	1	ヤ	1	0	0
Katakana MA	マ	0	0	1	マ	1	1	1	マ	1	1	1
Katakana TE	テ	0	0	1	テ	1	0	0	テ	1	1	1
Katakana TSU	ツ	1	1	1	ツ	1	1	1	ツ	1	1	1

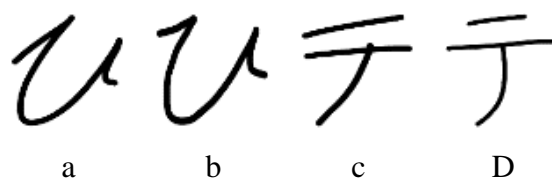
Katakana ME		0	0	1		1	0	0		1	1	1
Akurasi Pengenalan dari User ke-3										71,6%		
Rata-rata Akurasi Pengenalan dari Ketiga User										81,1%		

Tabel 2 menunjukkan bahwa hasil akurasi pengenalan *user* ke-1 adalah 81,7%, *user* ke-2 adalah 90% dan *user* ke-3 adalah 71,6%. Sehingga diperoleh rata-rata akurasi pengenalan sebesar 81,1%.

Ada pun contoh kesalahan pengenalan huruf yang disebabkan oleh kesalahan *user* dalam menginputkan pola huruf ditunjukkan pada Gambar 7. Kesalahan pengenalan tersebut menunjukkan bahwa metode Local Binary Patterns yang digunakan pada aplikasi ini kurang mampu mengenali perbedaan pola huruf yang kurang signifikan.

V. KESIMPULAN DAN SARAN

Berdasarkan analisis hasil pengujian maka dapat disimpulkan bahwa tingkat keakuratan aplikasi pengenalan pola Huruf Hiragana dan Katakana dengan mengimplementasikan metode Local Binary Patterns adalah sebesar 81,1% yang disebabkan oleh metode Local Binary Patterns yang kurang mampu mengenali perbedaan pola huruf yang kurang signifikan. Akurasi tersebut juga dipengaruhi oleh adanya perbedaan karakteristik huruf yang diinput-



Gambar 7 Contoh perbedaan hasil pengenalan pola huruf

kan oleh *user* berupa ukuran dan rotasi. Hal ini menunjukkan bahwa aplikasi ini kurang mampu mengatasi adanya perbedaan ukuran dan rotasi huruf.

Adapun saran untuk penelitian selanjutnya mengenai pengenalan pola Huruf Hiragana dan Katakana pada *smartphone* berbasis Android yaitu perlu digunakan metode ekstraksi fitur yang dapat mengatasi masalah variasi ukuran dan rotasi huruf.

DAFTAR PUSTAKA

- [1] Japan Student Services Organization, 2016, The Summary of Result on an Annual Survey of International Students in Japan, http://www.jasso.go.jp/en/about/statistics/intl_student_e/index.html [diakses 9 April 2016].
- [2] Nugroho Kusumo, O. K. Y., Dharmawan, D., Kurniawan, H., & Halim, A. (2013). Pengembangan Aplikasi Android Realtime Translation Untuk Menerjemahkan Bahasa Jepang Ke Bahasa Indonesia (Doctoral dissertation, BINUS).
- [3] Roßnagel, H., & Muntermann, J. (2009, October). Introducing sim-based security tokens as enabling technology for mobile real-time services. In *Nordic Conference on Secure IT Systems* (pp. 163-178). Springer Berlin Heidelberg.
- [4] Wirayuda, T. A. B., Wardhani, M.L.D.K., & Adiwijaya. (2008). Pengenalan Pola Huruf Jepang (Kana) Menggunakan Direction Feature Extraction dan Learning Vektor Quantization, *Jurnal Penelitian dan Pengembangan Telekomunikasi*, 13(2), ISSN: 1410-7066.
- [5] Madjid, D.H.A. (2016). Aplikasi Pengenalan Pola Karakter Kanji Dasar Menggunakan Pola Busur Terlokalisasi dan Jaringan Saraf Tiruan Learning Vector Quantization (LVQ) pada Smartphone Berbasis Android (Skripsi, Universitas Halu Oleo).
- [6] Biglari, M., Mirzaei, F., & Neycharan, J. G. (2014). Persian/Arabic handwritten digit recognition using local binary patterns. *International Journal of Digital Information and Wireless Communications (IJDWC)*, 4(4), 486-492.
- [7] Kurniawardhani, A., Suciati, N., & Ariesianti, I. (2014). Klasifikasi Citra Batik Menggunakan Metode Ekstraksi Ciri Yang Invariant Terhadap Rotasi. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 12(2), 48-60.
- [8] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), 971-987.
- [9] Putra, D. (2010). *Pengolahan citra digital*. Penerbit Andi.