

# Aplikasi Pengenalan Kata Pada Huruf Braille dan Pelafalannya

Agung Mega Iswara, Surya Sumpeno, Mauridhi Hery Purnomo  
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111  
e-mail: surya@ee.its.ac.id, hery@ee.its.ac.id

**Abstrak**—Penelitian dan karya dalam bidang *optical character recognition* terutama pengenalan karakter Braille, hingga saat ini masih jarang dilakukan dan pada umumnya penelitian yang telah ada untuk pengenalan huruf Braille masih berorientasi pada citra statis yang menggunakan *scanner*. Untuk mengubah lembaran Braille menjadi citra untuk dikenali sebagai karakter *alphabet* harus menggunakan alat *scanner* yang mahal, tidak *portable*, dan tidak praktis. Untuk menyederhanakan proses pengenalan huruf Braille tersebut perlu digunakan sistem pengenalan huruf Braille secara dinamis dengan kamera *webcam* atau kamera telpon genggam yang lebih murah dan lebih praktis. Tujuan utama dari pembuatan karya ini adalah untuk menyederhanakan proses pengenalan huruf Braille tersebut. Pengenalan huruf Braille secara dinamis ini menggunakan metode *chain approximation* untuk mendeteksi titik-titik Braille dan metode *horizontal and vertical projection* untuk pengelompokan posisi-posisi Braille menjadi karakter sesuai dengan aturan pembacaan huruf Braille yang berlaku. Hasil yang didapat dari proyek ini adalah sebuah program yang dapat bekerja untuk mengenali huruf Braille secara dinamis dengan menggunakan kamera *webcam* dengan tingkat akurasi sebesar 84% benar.

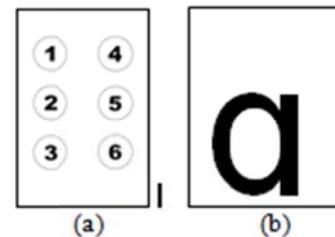
**Kata Kunci**—*image processing*, orang dengan cacat penglihatan, pengenalan Braille dinamis.

## I. PENDAHULUAN

PERTUKARAN informasi di era yang serba digital terutama dalam bentuk visual dan cetak sudah umum dilakukan oleh orang dengan penglihatan normal tetapi pertukaran informasi antara orang yang berpenglihatan kurang (*visually impaired*) atau bahkan yang tuna netra tentu saja akan menjadi sangat problematis. Sistem huruf Braille merupakan alat paling sering digunakan oleh penderita tuna netra untuk mengetahui informasi dari media cetak, namun apabila orang normal ingin mengetahui informasi yang tercetak dalam huruf Braille, orang normal tersebut harus terlebih dahulu mempelajari sistem huruf Braille, dengan aplikasi pengenalan huruf Braille pembacaan informasi dalam huruf Braille akan semakin mudah sehingga proses pertukaran informasi antara orang normal dengan orang tuna netra akan semakin baik.

Aplikasi pengenalan huruf Braille pada umumnya masih menggunakan *scanner* yang mahal dan kurang *portable*. Pada karya perangkat lunak ini dikembangkan aplikasi pengenalan huruf Braille yang dapat digunakan dengan kamera *webcam* sehingga dapat memudahkan dalam pengenalan huruf Braille secara dinamis dan lebih *portable* apabila dibandingkan dengan *scanner*.

Metode yang digunakan dalam pendeteksian titik-titik Braille secara dinamis adalah metode *chain approximation* [1],



Gambar 1. Huruf Braille  
(a) Aturan posisi titik-titik Braille  
(b) Contoh huruf Braille

untuk menentukan titik-titik Braille yang valid dengan menentukan luas area dari kontur dan untuk mendapatkan koordinat titik tengah tiap titik Braille digunakan metode *moments invariants* [2], dan untuk pengelompokan titik-titik Braille menjadi karakter digunakan metode *horizontal and vertical projection* [3].

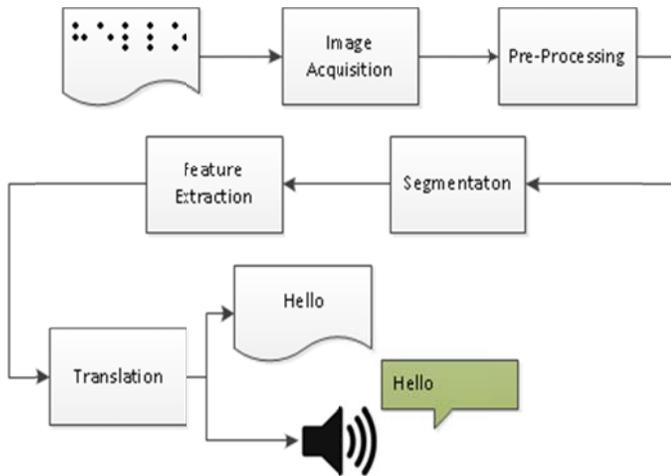
## II. SISTEM BRAILLE

Huruf Braille terdiri dari titik-titik yang disusun dalam bentuk persegi panjang dengan 3 baris dan 2 kolom seperti pada gambar 1. Huruf Braille didesain untuk dibaca dengan menggerakkan ujung jari dari kiri ke kanan untuk setiap huruf Braille. Kedua tangan biasanya terlibat dalam proses pembacaan dan pada umumnya yang digunakan adalah jari telunjuk.

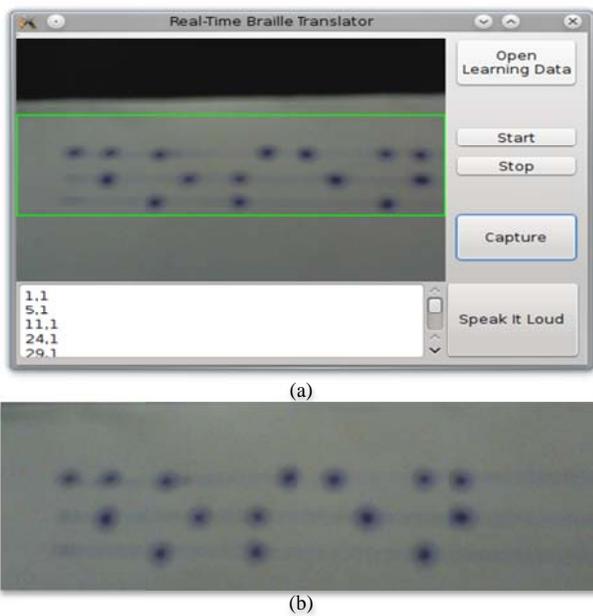
Gambar 1 menunjukkan posisi tiap titik pada huruf Braille. Titik-titik tersebut selanjutnya digunakan untuk diterjemahkan ke karakter alfabet yang dapat digunakan pada bahasa Indonesia, bahasa Inggris, dan lain-lain.

## III. SISTEM PENGENALAN HURUF BRAILLE

Pada bagian ini akan dijelaskan tentang sistem yang digunakan pada pengenalan huruf Braille secara dinamis. Keseluruhan prosesnya dapat digambarkan dalam *flow chart* yang ditunjukkan pada gambar 2. Pada gambar 2 dapat dilihat alur dari proses yang digunakan pada sistem pengenalan huruf Braille secara dinamis. Sistem pengenalan huruf Braille dimulai dengan pengambilan gambar dengan kamera *webcam*, setelah didapatkan citra dari kamera, citra diproses dengan perubahan citra ke format warna HSV (*Hue-Saturation-Value*) yang kemudian *noise*-nya di-filter dengan tujuan mereduksi gangguan pada citra. Proses tersebut dilanjutkan dengan segmentasi, fitur ekstraksi, klasifikasi, dan



Gambar 2 Flow Chart Sistem Pengenalan Braille



Gambar 3 Proses pengambilan gambar citra huruf Braille

Keterangan Gambar 3:

- (a) Interface dari program, proses perolehan gambar didapatkan setelah user menekan tombol "Capture".
- (b) Hasil citra yang didapatkan setelah tombol "Capture" ditekan.

pada akhirnya didapatkan hasil pengenalan huruf Braille dalam bentuk teks dan suara.

**A. Image Acquisition**

Mekanisme pengambilan gambar pada proyek ini menggunakan kamera webcam standar. Proses pengambilan gambar dilakukan dengan menekan tombol ketika sedang menjalankan program yang dibuat dengan Qt Framework [4]. Hasil dari proses tersebut dapat dilihat pada gambar 3 (a) dan (b).

**B. Pre-processing**

**1) Konversi Warna RGB ke HSV**

Pengubahan citra dari format RGB (Red-Green-Blue) ke dalam format HSV (Hue-Saturation-Value) [5] mempunyai tujuan untuk mengurangi komputasi pada citra dan untuk mengurangi noise yang disebabkan oleh iluminasi cahaya.

**2) Contrast-Stretching**

Terkadang terdapat daerah-daerah gelap dan terang pada saat pengambilan citra, sehingga perlu dilakukan perbaikan kontras [6] pada citra sehingga nantinya titik Braille dapat terdeteksi lebih jelas.

**3) Median Filter**

Pada saat pengambilan gambar, hampir pasti terdapat noise yang dapat mengganggu kualitas citra dan mengurangi tingkat kualitas pengenalan terhadap huruf Braille. Median filter [7] juga dapat membantu mengurangi noise tersebut.

**C. Segmentation**

**1) Adaptive Thresholding**

Thresholding adalah suatu metode yang digunakan untuk membagi sebuah citra menjadi nilai foreground dan background dengan mengatur pixel dengan nilai intensitas yang berada diatas nilai threshold menjadi nilai foreground atau 1 (satu) dan pixel sisanya menjadi nilai background atau 0 (nol) [8]. Adaptive thresholding dapat mengubah nilai batasan/threshold tersebut secara dinamis sehingga dapat menyesuaikan perubahan kondisi cahaya sehingga metode ini sangat berguna apabila karya aplikasi pengenalan huruf Braille ini digunakan pada kondisi pencahayaan yang berubah-ubah.

**2) Morfolgi Erode**

Morfologi Erode dapat didefinisikan sebagai:

$$A \ominus B = \{x | (B)_x \subseteq A\} \tag{1}$$

A adalah citra dan B adalah structuring element atau elemen pembentuknya [9].

**3) Morfolgi Dilate**

Dilate dapat didefinisikan sebagai:

$$A \oplus B = \{x | (B')_x \cap A \neq \emptyset\} \tag{2}$$

Seperti halnya pada morfologi erode, B adalah struktur elemen dan A adalah citranya [9].

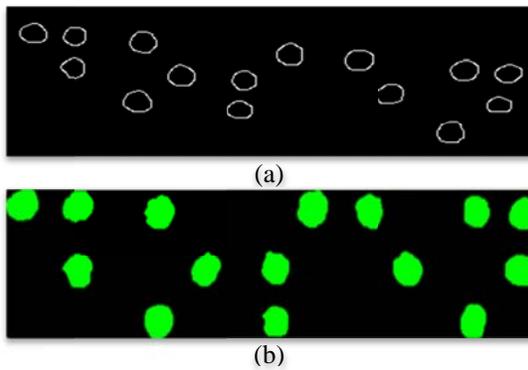
**4) Deteksi Tepi Canny**

Deteksi tepi Canny ini digunakan untuk mendapatkan garis-garis tepi pada titik Braille sehingga nantinya dapat mempermudah dalam pencarian koordinat garis tepi pada titik Braille tersebut [10].

**D. Feature Extraction**

**1) Pendeteksian Titik Braille**

Metode chain approximation [1] ini akan mendeteksi kontur pada citra hasil pendeteksian garis tepi Canny, kemudian setelah semua kontur berhasil terdeteksi digunakan moments invariant [2] untuk mendapatkan luas area tiap kontur yang terdeteksi dengan menggunakan rumus:



Gambar 4. Deteksi dan Koreksi Kemiringan Citra Braille  
 (a) Citra Braille dengan sudut  $-7^\circ$ ;  
 (b) hasil citra setelah kemiringannya dikoreksi

$$A = m_{0,0} = \iint dx dy f(x, y) \quad (3)$$

Untuk mendapatkan koordinat titik tengah dari setiap titik pembentuk huruf Braille yang valid digunakan bantuan metode *moments invariants* orde pertama dengan rumus sebagai berikut:

$$m_{1,0} = \iint dx dy x f(x, y) \quad (4)$$

$$m_{0,1} = \iint dx dy y f(x, y) \quad (5)$$

Dengan menggunakan rumus *moments invariants* orde pertama (4) dan (5), perhitungan koordinat titik tengah titik Braille didapatkan dengan rumus:

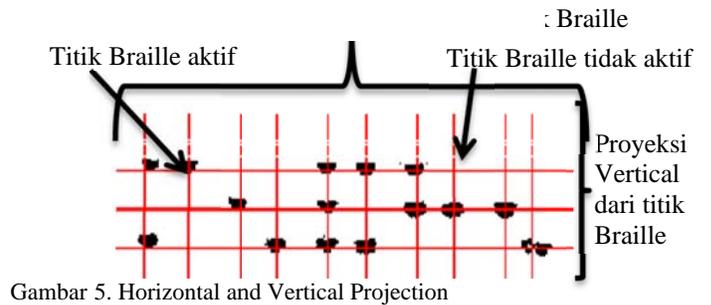
$$C(x, y) = \left\{ \frac{m_{1,0}}{m_{0,0}}, \frac{m_{0,1}}{m_{0,0}} \right\} \quad (6)$$

2) *Deteksi dan Koreksi Kemiringan*

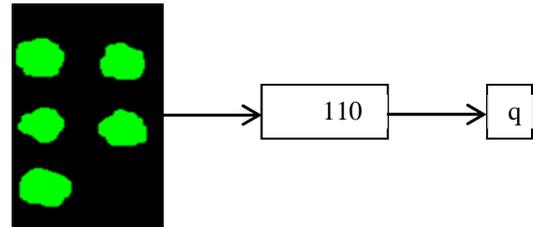
Deteksi sudut pada citra menggunakan koordinat dari titik-titik yang berada di posisi ekstrim paling kiri atas, paling kanan atas, paling kiri bawah, serta paling kanan bawah dari citra. Setelah didapatkan koordinat-koordinat tersebut, sudutnya dapat dihitung dengan metode yang diusulkan pada [11]. Gambar hasil koreksi kemiringan pada pendeteksian titik Braille diperlihatkan pada gambar 4.

3) *Horizontal and Vertical Projection*

Setelah didapatkan semua posisi titik Braille yang valid dari citra, posisi dari semua titik Braille valid disimpan ke array yang terpisah untuk masing-masing posisi horizontal dan vertical dari Braille. Identifikasi titik Braille valid di dalam *cell* Braille dapat dilakukan dengan cara memeriksa nilai proyeksi horizontal dan vertical yang tersimpan dalam *array* dengan nilai koordinat dari posisi titik yang aktif [3]. Kalau nilai pada proyeksi horizontal dan vertical cocok dengan koordinat titik Braille maka titik tersebut valid, jika sebaliknya maka nilai tersebut adalah sebagai *background*. Ilustrasi proyeksi horizontal dan vertical dapat dilihat pada gambar 5.



Gambar 5. Horizontal and Vertical Projection



Gambar 6. Ilustrasi penerjemahan karakter Braille ke huruf Alfabet

4) *Penerjemahan Titik-titik Braille*

Hasil posisi titik Braille ini dapat diterjemahkan ke huruf Alfabet. Penerjemahan posisi titik-titik Braille menjadi alfabet menggunakan klasifikasi *naïve-bayes* [12] dengan rumus seperti berikut:

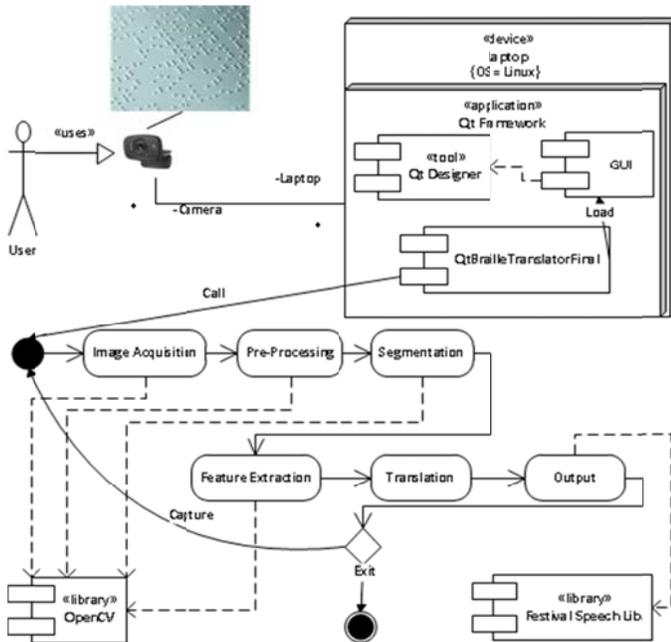
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (7)$$

Proses penerjemahan dapat diilustrasikan pada gambar 6.

IV. ILUSTRASI SISTEM

Bagian ini menjelaskan ilustrasi sistem yang digunakan untuk mengenali huruf Braille. Dari gambar 7 terlihat bahwa pengguna atau *user* menggunakan kamera untuk *capturing* citra dokumen Braille. Proses penangkapan citra menggunakan aplikasi yang sepenuhnya dikembangkan dengan bahasa pemrograman C++ menggunakan *tool* Qt Creator untuk mempermudah proses pembuatan aplikasi dan Qt Designer untuk merancang *user interface* aplikasi (Qt Creator dan Qt Designer tergabung dalam satu susunan bernama Qt Framework [4]). Proses *image acquisition*, *pre-processing*, *segmentation*, *feature extraction*, dan *classifier* pada karya ini menggunakan *OpenCV library* [13]. *OpenCV library* digunakan sebagai *library* yang dapat membantu dalam pengembangan aplikasi *image processing*.

Hasil luaran yang didapatkan setelah berhasil menerjemahkan huruf Braille adalah huruf alfabet yang ditampilkan pada *textbox* aplikasi dan disuarakan melalui *speaker*. Proses menyuarakan teks tersebut menggunakan *festival speech library* [14] yang berbahasa inggris dikarenakan pada saat pengembangan aplikasi hanya bahasa inggris yang didukung oleh Qt Framework. *Festival speech library* ini adalah *library text-to-speech* yang berguna untuk mengubah teks menjadi suara.



Gambar 7. Ilustrasi sistem pengenalan huruf Braille



Gambar 8. Proses pengambilan gambar dengan kamera webcam

Proses pengambilan citra dari dokumen Braille dengan menggunakan kamera dapat digambarkan pada gambar 8. Dokumen yang digunakan untuk pengenalan huruf Braille dapat berupa dokumen standar huruf Braille dengan cetakan timbul ataupun dengan cetakan tinta.

### V. HASIL PENGUJIAN

Sistem diujikan pada laptop dengan memakai kamera webcam Logitech C525. Dokumen yang digunakan pada saat pengujian adalah dokumen standar Braille dengan cetakan timbul dan tulisan huruf Braille dengan cetakan tinta. Hasil yang didapatkan untuk pengenalan per huruf adalah sebanyak 22 dari 26 huruf berhasil diterjemahkan dengan benar, kesalahan dalam proses penerjemahan dikarenakan metode *horizontal and vertical projection* belum mampu membaca



(a)



(b)

Gambar 9. Pengujian huruf pada cetakan tinta  
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille



(a)

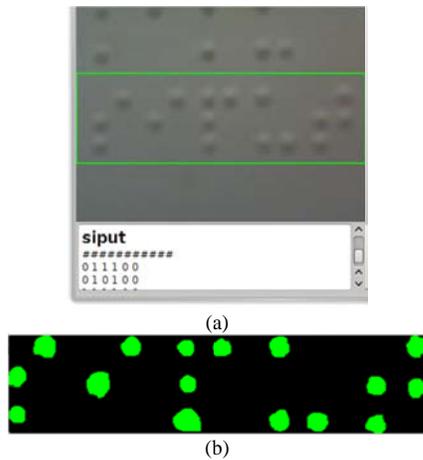


(b)

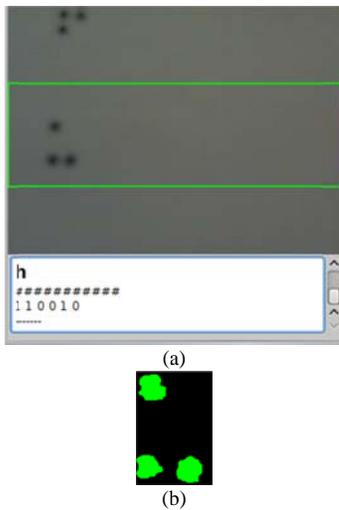
Gambar 10. Pengujian kata pada cetakan tinta  
Keterangan Gambar 10  
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille

Braille yang mempunyai *gap/kekosongan* pada baris dan atau kolomnya. Untuk pengenalan per kata, apabila pada kata tersebut terdapat karakter Braille yang hanya mempunyai 1 kolom maka hasil terjemahan dari pengenalan huruf Braille akan memberikan hasil yang salah. Gambar 9 memperlihatkan proses pengenalan huruf Braille dengan terjemahan benar pada cetakan tinta. Gambar 10 dan 11 memperlihatkan proses pengenalan kata dengan terjemahan benar pada masing-masing cetakan tinta dan dokumen timbul. Sedangkan pada gambar 12 dan 13 memperlihatkan proses pengenalan yang memberikan hasil terjemahan yang salah.

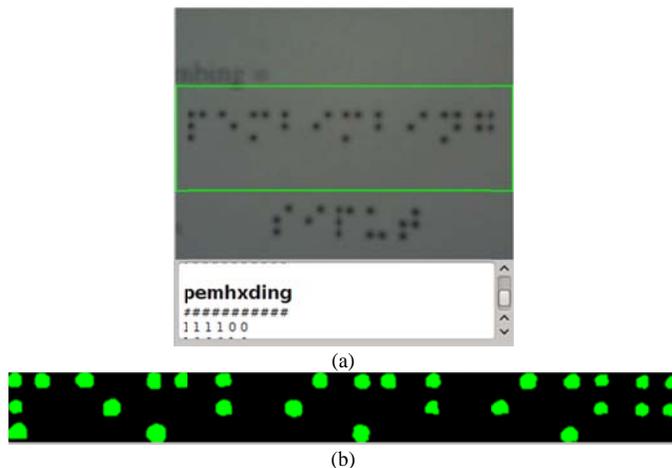
Pengujian pengenalan huruf Braille juga dilakukan dengan menambahkan sudut kemiringan pada proses penangkapan citra. Besarnya sudut pada pengujian dibatasi antara  $-10^\circ$  hingga  $+10^\circ$ . Pembatasan pada kemiringan citra dilakukan karena citra dengan sudut kemiringan yang besar akan memberikan hasil terjemahan yang salah. Gambar 13 memperlihatkan pengujian pada citra Braille dengan kemiringan  $-7^\circ$ .



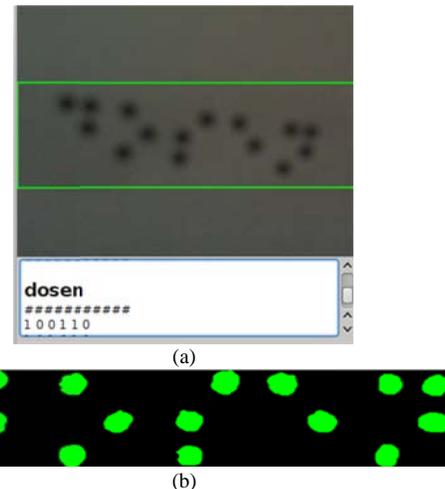
Gambar 11. Pengujian kata pada dokumen timbul  
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille



Gambar 12. Hasil terjemahan salah pada pengujian huruf  
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille



Gambar 13. Hasil terjemahan salah pada pengujian kata  
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille



Gambar 14. Proses pengenalan citra huruf Braille dengan kemiringan  $-7^{\circ}$   
(a) Proses penangkapan citra huruf Braille  
(b) Hasil pendeteksian titik-titik Braille

## VI. KESIMPULAN/RINGKASAN

Dari hasil pengembangan perangkat lunak sistem pengenalan huruf Braille secara dinamis didapatkan kesimpulan bahwa pendeteksian titik-titik Braille pada citra dengan metode *chain approximation* menunjukkan hasil yang baik, hal ini terbukti dari semua titik Braille terdeteksi dengan baik. Untuk membedakan titik-titik Braille yang valid dari *noise* pada citra berdasarkan luas area menggunakan metode *moment invariants* orde ke-nol menunjukkan hasil yang baik walaupun terkadang ada beberapa *noise* yang terdeteksi sebagai Braille yang valid. Hasil pengelompokan titik-titik Braille menjadi karakter dengan metode *horizontal and vertical projection* memiliki tingkat akurasi sebesar 84,6% dan belum cocok apabila digunakan dalam pengenalan huruf Braille secara dinamis.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada bapak Mauridhi Hery Purnomo dan bapak Surya Sumpeno atas bimbingan teknis selama mengerjakan sistem pengenalan huruf Braille secara dinamis.

## DAFTAR PUSTAKA

- [1] Teh, C.-H. and Chin, R.T..On the detection of dominant points on digital curves. Pattern Analysis and Machine Intelligence. 1989
- [2] Kilian, Johannes.Simple Image Analysis By Moments.2001 <URL: <http://public.cranfield.ac.uk/c5354/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf>, diakses tanggal 31 Mei 2013>
- [3] Shumet Tadesse, "Feature Extraction and Classification Schemes for Enhancing Amharic Braille Recognition System", M.Sc. Thesis, Faculty of Informatics, Addis Ababa University, Addis Ababa, June 2010
- [4] Qt Framework <URL: <http://qt-project.org/>, diakses 21 Juni 2013)
- [5] RGB & HSV Colorspaces, Vanderbilt University. <URL: <http://structbio.vanderbilt.edu/chazin/wisdom/xv-3.10a/rgb-hsv-colorspace.html>, diakses tanggal 20 Juni 2013>
- [6] Contrast Stretching <URL: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm> , diakses tanggal 2 Mei 2013>

- [7] Median Filter <URL:  
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>, diakses tanggal 2 Mei 2013>
- [8] Adaptive Thresholding <URL:  
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm>, diakses tanggal 1 Mei 2013>
- [9] Morphological Operations <URL:  
<http://bme.med.upatras.gr/improc/Morphological%20operators.htm>,  
diakses tanggal 31 Mei 2013)
- [10] Canny, John."A Computational Approach to Edge Detection". IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. PAMI-8, No. 6. November 1986
- [11] Safabakhsh, R. "Document skew detection using minimum-area bounding rectangle". Proceedings. International Conference on Information Technology: Coding and Computing. 2000.
- [12] Naïve-Bayes Classifier  
<URL:[http://digilib.ittelkom.ac.id/index.php?option=com\\_content&view=article&id=486:naive-bayes-classifier&catid=11:sistem-komunikasi&Itemid=14](http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=486:naive-bayes-classifier&catid=11:sistem-komunikasi&Itemid=14), diakses tanggal 20 Juni 2013>
- [13] OpenCV <URL: <http://opencv.org/>, diakses 21 Juni 2013)
- [14] The Festival Speech Synthesis System. University of Edinburgh.<URL:  
<http://www.cstr.ed.ac.uk/projects/festival/>, diakses tanggal 20 Juni 2013)