

Implementasi Moment Invariant Untuk Pengenalan Label Buku Perpustakaan Berbasis Android

Hendy Yudhitya Susetya¹, Antonius Rachmat², Kristian Adi Nugraha³

^{1,2,3}Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

Jl. Dr. Wahidin Sudiro Husodo No.5 – 25, Yogyakarta

hendy.ys@ti.ukdw.ac.id

anton@ti.ukdw.ac.id

adinugraha@ti.ukdw.ac.id

Abstract— *Duta Wacana Christian University Library has many collection of book. The book are neatly organized on the shelves and labeled but there are still many people who put the book on the wrong shelves. This research aims to create an application that can identify whether a book is misplaced or not with help of Moment Invariant and K-Nearest Neighbor method. Moment invariant is an extraction method that produce 7 features and this features is used to identify an object. While K-Nearest Neighbor is a method to classify object based on training data or template. The system receives input in form of image that contain row of books. This image will go through several processes. The processes are Region of Interest process to obtain book label, Histogram Approach process to obtain character from each label, thinning process to obtain skeleton of each character, and then Moment invariant for extraction method to obtain features that will be classified using K-nearest neighbor based on template. The research gave out two result and both of them use $k=1$ and 100×100 for character size. The first one has a result of 85.39% accuracy by using 30 test images where templates taken from the test images. The second result has 81.54% accuracy by using 20 test image where no template taken from the test images.*

Intisari— Perpustakaan Universitas Kristen Duta Wacana(UKDW) memiliki koleksi buku yang cukup banyak dan beragam. Buku-buku tersebut sudah disusun rapi pada rak-rak buku yang ada dan diberi label. Akan tetapi masih banyak peminjam yang meletakkan buku pada rak yang salah. Penelitian ini bertujuan untuk membuat aplikasi yang dapat mengetahui apakah sebuah buku salah letak atau tidak dengan menggunakan metode Moment Invariant dan K-Nearest Neighbor. Moment invariant merupakan sebuah metode ekstraksi fitur yang menghasilkan 7 fitur yang digunakan untuk mengenali sebuah obyek. Sedangkan K-Nearest Neighbor merupakan metode klasifikasi yang digunakan untuk mengklasifikasikan obyek berdasarkan data latih yang ada. Citra input berupa citra yang berisi deret buku. Citra ini akan melalui beberapa proses yaitu proses Region of Interest untuk mendapatkan label buku, proses segmentasi Histogram Approach untuk mendapatkan citra karakter tiap labelnya, proses thinning yang digunakan untuk mendapatkan kerangka tiap karakter, kemudian proses ekstraksi fitur Moment Invariant yang akan menghasilkan fitur dari citra karakter dan fitur-fitur tersebut akan diklasifikasikan menggunakan K-Nearest Neighbor sesuai dengan template yang ada sehingga citra karakter dapat dikenali. Dari penelitian yang telah dilakukan dapat disimpulkan bahwa sistem berhasil mengimplementasikan metode Moment Invariant dan K-Nearest Neighbor sehingga berhasil dalam mengenali karakter pada label buku perpustakaan UKDW dengan

menggunakan $K=1$ dan ukuran citra karakter sebesar 100×100 . Terdapat 2 hasil persentase keakuratan yaitu sebesar 85.39% dengan menggunakan 30 citra uji yang didalamnya terdapat beberapa citra karakter yang digunakan sebagai template dan sebesar 81.54% dengan menggunakan 20 citra uji di mana tidak ada template yang berasal dari citra uji.

Kata Kunci— *OCR, moment invariant, K-NN, thinning, buku perpustakaan*

I. PENDAHULUAN

Buku merupakan sarana atau alat yang banyak digunakan masyarakat dalam mencari ilmu. Meskipun pada era globalisasi sekarang banyak terdapat buku-buku elektronik, buku teks tetap digunakan oleh masyarakat terutama mahasiswa untuk mencari ilmu. Perpustakaan banyak ditemukan buku dengan berbagai macam jenis. Buku-buku di perpustakaan biasanya memiliki sebuah penanda yang berupa label. Label ini bertujuan untuk menunjukkan di mana seharusnya buku tersebut diletakkan, sehingga buku-buku di perpustakaan dapat diklasifikasikan dengan rapi di rak ada.

Perpustakaan UKDW memiliki koleksi buku yang cukup beragam. Buku-buku tersebut juga ditandai dengan sebuah label. Akan tetapi masih banyak mahasiswa yang kurang teliti dan mengembalikan buku tidak pada tempatnya. Hal ini tentu membuat mahasiswa lain kesulitan dalam mencari buku yang diinginkan, serta membuat rak buku terlihat tidak rapi, maka dibutuhkan sebuah aplikasi yang memanfaatkan teknologi pengolahan citra yang dapat digunakan untuk mencari kesalahan penempatan letak buku di perpustakaan UKDW.

Optical Character Recognition (OCR) merupakan salah satu dari teknologi pengolahan citra yang dapat digunakan untuk mengenali karakter di dalam label. Karakter yang sudah diidentifikasi dapat digunakan untuk mengetahui letak dari buku tersebut benar atau salah. Banyak metode dapat digunakan dalam OCR. Metode yang digunakan oleh Pujiyanto [1] pada penelitian yang sebelumnya adalah *Template Matching*. Sebelum dilakukan proses pengenalan label buku, terlebih dahulu dilakukan proses *Region of Interest* yang memperkecil ruang lingkup citra yang akan diproses. Kemudian dilakukan segmentasi menggunakan

Histogram Approach, yang dilanjutkan dengan proses mengenali label buku dengan *Template Matching*.

Dari hasil penelitian Pujiyanto [1], metode ROI, *Histogram Approach* dan *Template Matching* dapat mendeteksi kesalahan letak buku pada citra. Persentase keberhasilan ROI sebesar 98.61%, keberhasilan *Histogram Approach* 81.27% dan keberhasilan *Template Matching* 83.78%. Penulis dalam penelitiannya akan menggunakan metode yang berbeda yaitu ekstraksi fitur *Moment Invariant* dan metode klasifikasi K-NN. *Moment Invariant* merupakan sebuah metode ekstraksi fitur yang dapat menangani translasi, penyekalaan dan rotasi pada gambar. Metode ini akan menghasilkan 7 ciri yang nantinya dapat digunakan dalam pengenalan karakter [2].

Rizanti, Suprpto dan Regasari [3] melakukan penelitian untuk mengenali citra alphabet. Di dalam penelitian tersebut digunakan metode ekstraksi fitur *Moment Invariant* yang menghasilkan 7 fitur. Kemudian fitur-fitur tersebut diklasifikasikan menggunakan metode klasifikasi *K-Nearest Neighbor* (K-NN). K-NN adalah sebuah metode untuk melakukan pengklasifikasian berdasarkan data yang jaraknya paling dekat dengan objek tersebut. Rumus yang digunakan dalam menghitung jarak adalah *Euclidian Distance*. Dari penelitian ini didapatkan hasil dengan akurasi terbaik yaitu 84.74% dengan menggunakan 45 data set. Oleh karena itu diharapkan dengan menggunakan kedua metode tersebut dapat menambah keakuratan sistem dari penelitian sebelumnya.

Pada penelitian ini, penulis akan membuat sebuah sistem untuk mengetahui kesalahan letak buku di perpustakaan UKDW. Berdasarkan hasil penelitian di atas, penulis memutuskan untuk menggunakan metode *Moment Invariant* sebagai metode ekstraksi fitur dan metode K-NN yang digunakan untuk mengklasifikasikan citra. Penulis berharap dengan menggunakan metode *Moment Invariant* dan K-NN dalam sistem ini dapat menghasilkan hasil persentase akurasi yang lebih baik dari penelitian yang sebelumnya.

II. LANDASAN TEORI

A. Region of Interest

Region of Interest (ROI) adalah sebuah algoritma yang membatasi perhitungan hanya pada sebuah area tertentu dari sebuah citra utuh sesuai keinginan user. Dengan hanya memilih area ini secara khusus, maka akan meningkatkan kecepatan perhitungan dari komputer. Hal ini dikarenakan perhitungan hanya terjadi pada area yang dipilih dan tidak pada area yang lain. [4]

Tahapan-tahapan yang terjadi pada proses ROI adalah sebagai berikut.

1) Binarization dan Grayscale

Pada penelitian ini proses *Binarization* dan *Grayscale* digabung menjadi satu proses. Proses *grayscale* merupakan salah satu tahap awal pada proses ROI. Fungsi utama dari proses ini adalah mengubah citra menjadi citra *Grayscale*. Untuk mengubah citra menjadi citra *Grayscale* dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B [5]. Rumus (1) merupakan rumus *Grayscale*.

$$s = (R + G + B)/3 \quad (1)$$

Sedangkan *Binarization* sendiri merupakan metode untuk mengubah citra *grayscale* menjadi citra biner(hitam-putih).

2) Image Cropping

Image Cropping merupakan tahapan di mana memotong bagian yang tidak diperlukan dan menyisakan bagian yang diperlukan. Dalam penelitian ini proses *cropping* bertujuan untuk mencari baris label dan kemudian baris label tersebut dipotong menjadi potongan label. Gambar 1 merupakan *pseudocode* dari proses *Image Cropping*.

```

Start
  Get image cb
  width = Get cb size
  height = Get cb size
  For y=0 to height
    batasAtas = Get y
    batasBawah = Get y
  End for
  Crop cb use batasAtas and batasBawah
  n = 0
  For x=0 to width
    xAwal[n] = Get x
    xAkhir[n] = Get x
    n++
  End for
  For i=0 to n
    Create new image label[i]
    use xAwal[i] and xAkhir[i]
  End for
Finish
    
```

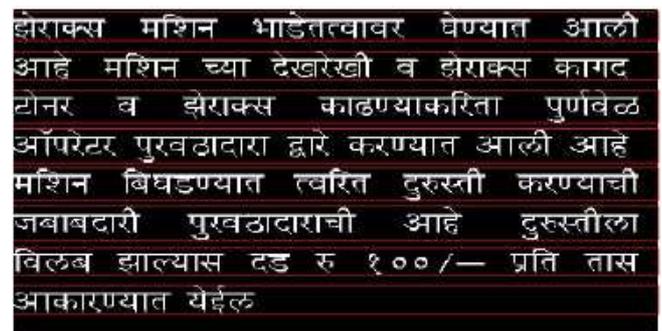
Gambar. 1. Pseudocode proses *Image Cropping*

B. Histogram Approach

Histogram Approach merupakan salah satu metode segmentasi dengan menggunakan grafik. Grafik yang dihasilkan dapat digunakan untuk menentukan jeda antar baris maupun jeda antar karakter, sehingga mempermudah proses segmentasi. Berdasarkan Dongre [6] *Histogram Approach* memiliki 3 proses yaitu *line segmentation*, *word segmentation*, dan *character segmentation*.

1) Line segmentation

Line Segmentation digunakan untuk mengidentifikasi baris karakter dan memisahkan 1 baris dengan baris lainnya. Gambar 2 merupakan contoh dari *Line Segmentation*.



a) Sebelum *Line Segmentation*



b) Setelah Line Segmentation

Gambar. 2. Line Segmentation ([6])

2) Word segmentation

Word segmentation digunakan untuk mengidentifikasi antar kata dalam suatu baris dan memisahkan 1 kata dengan kata lainnya. Gambar 3 merupakan contoh dari Word Segmentation.



a) Sebelum Word Segmentation



b) Sesudah Word Segmentation

Gambar. 3. Word Segmentation ([6])

3) Character segmentation

Character segmentation digunakan untuk mengidentifikasi karakter pada sebuah kata dan memisahkan antar karakter. Gambar 4 merupakan contoh dari Character Segmentation.



a) Sebelum Character Segmentation



b) Sesudah Character Segmentation

Gambar. 4. Character Segmentation ([6])

C. Thinning

Thinning merupakan proses di mana citra biner akan direduksi hingga tersisa garis tengah atau kerangka dari objek. Tujuan dari thinning adalah mencari informasi dasar dari citra sehingga dapat mempermudah proses perhitungan, pengolahan serta analisis pada proses selanjutnya. [7]

Menurut Parker [8] untuk memperoleh kualitas hasil thinning yang baik dapat menggunakan Stentiford's preprocessing, algoritma thinning Zhang-Suen algorithm dan Holt's staircase removal sebagai post-processing. Pada penelitian ini penulis akan menambahkan algoritma dilasi sebelum dilakukannya Stentiford's preprocessing.

1) Dilation

Dilasi merupakan sebuah operasi morfologi yang digunakan untuk melebarkan gambar. Proses dilasi akan melebarkan foreground dengan cara mengubah background menjadi foreground sesuai dengan structural element yang digunakan. [9]

Structural Element yang dipakai dalam penelitian ini dapat dilihat pada gambar 5.

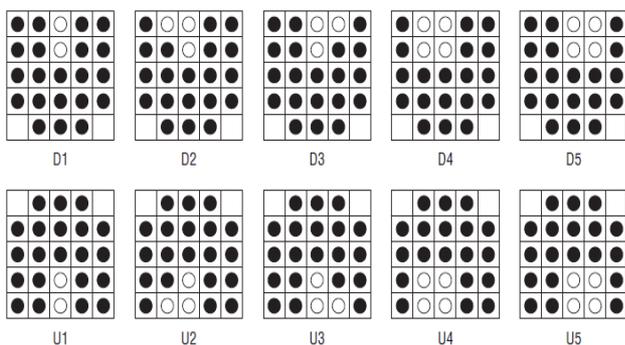
1	1	1
1	1	1
1	1	1

Gambar. 5. Structural element pada dilasi

2) Stentiford's preprocessing

Stentiford's preprocessing terdiri dari 2 tahap yaitu smoothing dan acute angle emphasis. Kedua tahap ini bertujuan untuk mengurangi hal-hal yang dapat mengganggu hasil proses thinning. Tahap smoothing dilakukan dengan menghapus semua pixel yang memiliki kurang dari sama dengan 2 pixel hitam sebagai tetangga dan memiliki nilai konektivitas kurang dari 2. Tahap kedua yaitu acute angle emphasis. Tahap ini menggunakan template yang dapat dilihat pada gambar 6. Jika ada template yang cocok, pixel

tengah pada template akan ditandai untuk dihapus dan menyebabkan terjadinya iterasi ke 2 yang hanya menggunakan 3 *template* dari tiap tipe. Jika ada *pixel* yang dihapus dari iterasi ini, maka akan dilakukan iterasi terakhir yang menggunakan template pertama dari tiap tipe.



Gambar. 6. Template Untuk proses *Acute Angle Emphasis* ([8])

3) Algoritma *thinning Zhang-Suen*

Algoritma *thinning Zhang-Suen* merupakan sebuah algoritma *thinning* yang mudah untuk diimplementasikan dan sudah dipakai bertahun-tahun. Algoritma ini dibagi menjadi 2 subiterasi. Pada iterasi pertama sebuah *pixel* akan ditandai dan dihapus jika memenuhi kondisi berikut :

- Memiliki nilai konektivitas 1
- Memiliki setidaknya 2 *pixel* tetangga yang hitam dan tidak lebih dari 6
- Setidaknya salah satu dari $I(i,j+1), I(i-1,j)$ dan $I(i,j-1)$ adalah putih
- Setidaknya salah satu dari $I(i-1,j), I(i+1,j)$ dan $I(i,j-1)$ adalah putih

Untuk sub iterasi kedua memiliki kondisi yang sama dengan sub iterasi pertama kecuali kondisi ke 3 dan 4 menjadi :

- Setidaknya salah satu dari $I(i-1,j), I(i,j+1)$ dan $I(i+1,j)$ adalah putih
- Setidaknya salah satu dari $I(i,j+1), I(i+1,j)$ dan $I(i,j-1)$ adalah putih
- Pada akhir tiap iterasi *pixel* yang memenuhi syarat akan ditandai dan dihapus. Jika pada salah satu iterasi tidak ada *pixel* yang dihapus maka algoritma berhenti.

4) *Holt's staircase removal*

Holt's staircase removal bertujuan untuk menghapus *pixel* yang masih dapat dihapus ada akhir proses *thinning*. Biasanya *pixel-pixel* berbentuk anak tangga. Untuk menghindari terbentuknya lubang baru, ditambahkan sebuah kondisi yang dapat dilihat melalui ekspresi logika di bawah ini :

$$v(C) \wedge \sim (v(N) \wedge ((v(E) \wedge \sim v(NE) \wedge \sim v(SW) \wedge (\sim v(W) \vee \sim v(S))) \vee (v(W) \wedge \sim v(NW) \wedge \sim v(SE) \wedge (\sim v(E) \vee \sim (S))))))$$

Gambar. 7. kondisi yang untuk *Holt's staircase removal* ([8])

Kondisi pada gambar 7 merupakan kondisi yang digunakan untuk anak tangga yang menghadap ke utara atau

atas. Untuk kondisi yang menghadap ke selatan atau bawah hanya menukar posisi utara dan selatan.

D. *Moment Invariant*

Moment Invariant merupakan sebuah metode pengambilan ciri dari sebuah objek. Ciri yang diambil dapat berupa posisi, area, orientasi dan ciri lainnya . Metode ini dikenalkan oleh Hu pada tahun 1962. [3]

Persamaan dasar dari Momen suatu objek didefinisikan pada rumus (1):

$$m_{ij} = \sum_x \sum_y x^i y^j a_{xy} \quad (2)$$

Keterangan :

x = panjang citra

y = lebar citra

a_{xy} = nilai intensitas citra pada titik x dan y

i,j = order momen

Selanjut momen pusat (*central moment*) μ adalah momen yang bersesuaian dengan pusat area.

Momen pusat didefinisikan pada rumus (3) sampai dengan (7).

$$\mu_{ij} = \sum_x \sum_y (x - x')^i (y - y')^j a_{xy} \quad (3)$$

$$x' = \frac{m_{10}}{m_{00}} \text{ dan } y' = \frac{m_{01}}{m_{00}} \quad (4)$$

$$m_{00} = \sum_x \sum_y a_{xy} \quad (5)$$

$$m_{10} = \sum_x \sum_y x \cdot a_{xy} \quad (6)$$

$$m_{01} = \sum_x \sum_y y \cdot a_{xy} \quad (7)$$

Keterangan:

x = panjang citra

y = lebar citra

a_{xy} = nilai intensitas citra pada titik x dan y

$(x', y') = \text{centroid}$ citra dengan koordinat x' dan y'

μ_{ij} = *central moment*

Kemudian momen pusat yang telah dinormalisasi memiliki persamaan seperti pada rumus (8).

$$\eta_{ij} = \frac{\mu_{ij}}{(\mu_{00})^\lambda}; \lambda = \frac{(i+j)}{2} + 1 \quad (8)$$

Keterangan :

μ_{ij} = *central moment*

Hasil dari momen yang sudah ternormalisasi tersebut, dihasilkan 7 Hasil dari momen yang sudah ternormalisasi tersebut, maka dihasilkan 7 *Hu moment invariants* yang dapat dilihat pada rumus (9) sampai dengan (15) [10].

$$\phi_1 = \eta_{20} + \eta_{02} \tag{9}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \tag{10}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{11}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{12}$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2\} \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} \\ & + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2\} \end{aligned} \tag{13}$$

$$\begin{aligned} \phi_6 = & (\eta_{20} - \eta_{02})\{(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2\} \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \tag{14}$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{12} + \eta_{30})^2\} \\ & + (3\eta_{21} - \eta_{03})(3\eta_{21} \\ & + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2\} \end{aligned} \tag{15}$$

E. *K-Nearest Neighbor (KNN)*

K-Nearest Neighbor merupakan salah satu teknik klasifikasi yang populer digunakan dan terbukti menjadi algoritma yang sederhana dan efektif. Tujuan dari algoritma ini adalah mengklasifikasikan data baru berdasarkan jarak dari data latih yang sudah ada. Prinsip kerja *K-Nearest Neighbor* adalah mencari jarak terdekat antara data baru dengan K data terdekatnya dalam data pelatihan [11]. Dekat atau jauhnya jarak data yang akan dievaluasi dengan K tetangga dapat dihitung berdasarkan rumus *Euclidian Distance* pada rumus (16).

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \tag{16}$$

Keterangan:

x_2 = data uji

x_1 = data sample

p = jumlah dimensi data

Pada penelitian ini, penulis menggunakan bobot dalam metode K-NN. menurut Feray [12] tujuan dari penggunaan bobot ini adalah agar kelas yang memiliki jarak yang jauh

memiliki dampak yang lebih kecil pada perhitungan voting dibanding dengan kelas yang memiliki jarak yang dekat. Setelah mendapatkan nilai jarak dengan menggunakan rumus *Euclidian Distance* lalu dihitung bobot dengan menggunakan rumus (17):

$$w = 1/d \tag{17}$$

Di mana d merupakan nilai jarak dari rumus *Euclidian Distance*. Kemudian dihitung total untuk setiap kelas pada K data terdekat. Kelas yang mempunyai bobot terbesar menang dan merupakan kelas untuk data yang diuji.

III. METODOLOGI PENELITIAN

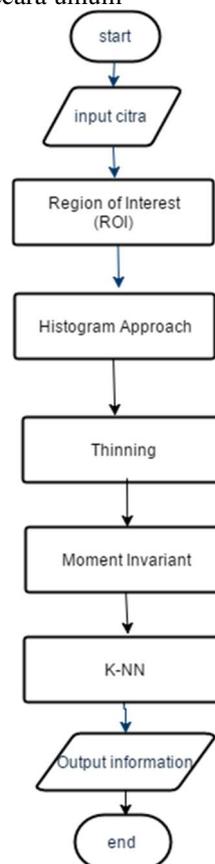
Metode yang digunakan dalam pengembangan aplikasi deteksi kesalahan letak buku perpustakaan UDKDW adalah sebagai berikut:

A. Analisis Kebutuhan

Spesifikasi sistem yang akan digunakan dalam membangun sistem deteksi kesalahan letak buku perpustakaan UKDW adalah menggunakan device Android dengan minimal sistem operasi KitKat 4.4.

B. Perancangan Sistem

Cara kerja sistem secara umum akan dijelaskan dalam bentuk *flowchart*. Gambar 8 merupakan penjelasan dari *flowchart* sistem secara umum



Gambar. 8. *Flowchart* sistem secara umum

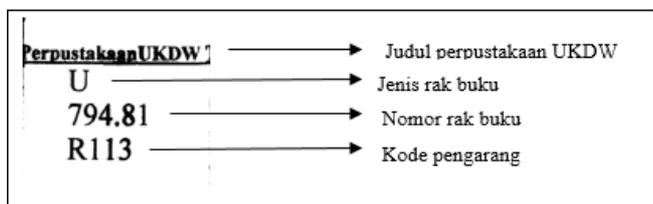
Proses dimulai dengan input citra buku perpustakaan yang akan dicek ke dalam sistem. Kemudian masuk ke proses ROI. Pada proses ROI ini sistem akan mengubah citra buku menjadi citra *grayscale* yang kemudian diubah lagi menjadi citra hitam putih. Tahap terakhir proses ROI adalah mencari posisi baris label dan dari baris label sistem akan memotong menjadi potongan label.

Hasil dari proses ROI digunakan dalam proses *Histogram Approach* yang merupakan proses segmentasi untuk mendapatkan karakter dalam tiap label. Hasil dari proses *Histogram Approach* merupakan citra karakter untuk tiap label. Sebelum memasuki proses pengenalan, citra karakter ini akan melalui proses *thinning* terlebih dahulu untuk kerangka dari citra karakter.

Proses berikutnya adalah proses *Moment Invariant* dan K-NN. Kedua proses ini bertujuan untuk mengenali tiap karakter hasil dari proses *Histogram Approach*. Proses *Moment Invariant* digunakan untuk mengekstraksi fitur dari citra karakter, sedangkan K-NN digunakan untuk mengklasifikasikan citra karakter berdasarkan data latih yang ada. Hasil dari kedua proses tersebut digunakan untuk mendapatkan informasi tentang ada tidaknya kesalahan letak buku pada citra *input* yang kemudian diberikan kepada *user*.

C. Pengujian Sistem

Pengujian sistem dilakukan dengan 3 tahap. Tahap pertama yaitu mencari ukuran citra karakter yang optimal pada proses *Moment Invariant* dan ukuran yang diuji adalah 60x60, 80x80 dan 100x100 dengan nilai K sebesar 1. Tahap kedua yaitu mencari nilai K pada K-NN yang optimal dalam melakukan pengenalan citra karakter. Nilai K yang diuji adalah 1, 3 dan 5 dengan ukuran citra karakter menggunakan hasil pengujian sebelumnya. Citra uji yang digunakan dalam 2 pengujian pertama adalah 30 citra uji di mana terdapat *template* yang diambil dari citra uji. Tahap ketiga yaitu pengujian dengan menggunakan 20 citra uji baru di mana tidak ada *template* yang diambil dari citra uji dan pengujian ini menggunakan nilai k dan ukuran citra karakter dari hasil pengujian sebelumnya. Data latih/ *template* yang digunakan diambil dari karakter dalam label buku perpustakaan yaitu angka 0 hingga 9 dan huruf U dan R. Gambar 9 merupakan contoh dari label buku perpustakaan serta keterangan untuk tiap barisnya.



Gambar. 9. contoh label buku perpustakaan

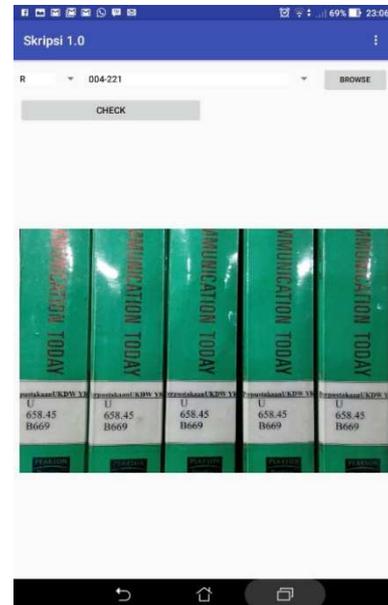
D. Evaluasi Kinerja

Evaluasi kinerja sistem diukur melalui rata-rata akurasi pengenalan pola karakter buku perpustakaan UKDW menggunakan metode *Moment Invariant* dan *K-Nearest Neighbor* (K-NN). Akurasi tersebut dihitung dengan menggunakan rumus persentase yaitu (Jumlah karakter yang terdeteksi benar/jumlah seluruh karakter)*100.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Sistem deteksi kesalahan letak buku perpustakaan UKDW dibuat dan dijalankan pada *platform* Android. Sistem ini memiliki satu halaman utama yang mencakup semua fungsi sistem. Gambar 10 merupakan tampilan halaman utama sistem.



Gambar. 10. Halaman utama

Terlihat gambar 10, halaman utama memiliki beberapa komponen yaitu sebagai berikut: *dropdown list* untuk memilih jenis rak dan nomor rak, tombol untuk memilih citra yang ingin dideteksi yang kemudian ditampilkan pada halaman utama, serta terdapat tombol *check* yang digunakan untuk memulai proses.

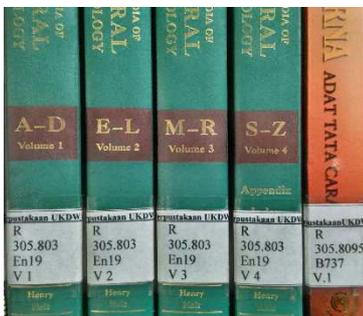


Gambar. 11. Halaman utama ketika menampilkan hasil proses

Pada gambar 11 memperlihatkan tampilan halaman utama ketika memperlihatkan informasi hasil dari proses deteksi dalam bentuk *message box*. Kemudian pada citra buku yang dipilih akan muncul angka untuk memudahkan user dalam mengetahui buku mana yang salah letak.

B. Pengujian Sistem

Pengujian sistem ini akan menampilkan hasil dari 3 pengujian yang telah dilakukan. Dua pengujian pertama yaitu mencari ukuran optimal citra karakter pada proses *moment invariant* dan mencari nilai K yang optimal pada proses K-NN. Citra uji yang digunakan dalam setiap pengujian adalah 30 citra di mana *template* / data latih berasal dari citra uji tersebut, contoh citra uji dapat dilihat pada gambar 12. Hasil pengujian pertama yaitu ukuran citra karakter akan digunakan dalam pengujian kedua yaitu pengujian nilai K pada K-NN. Kemudian pengujian ke 3 yaitu pengujian dengan menggunakan 20 citra uji di mana tidak ada *template* yang berasal dari citra uji tersebut.



Gambar. 12. contoh citra uji

1) Pengujian Keakuratan Sistem Berdasarkan Ukuran Citra Karakter

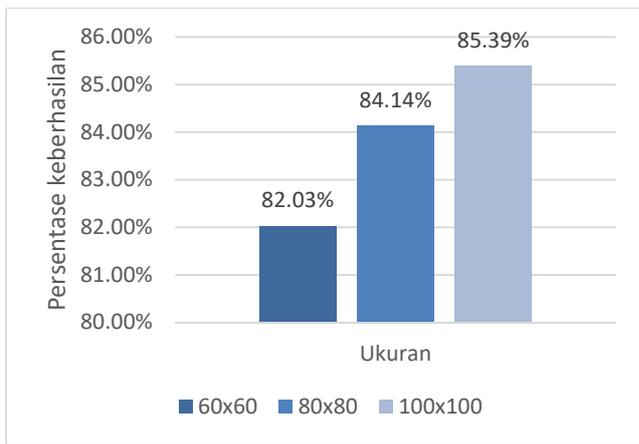
Pengujian pertama adalah pengujian untuk mencari ukuran citra karakter yang paling optimal yang digunakan dalam proses *Moment Invariant*. Pengujian ini akan dilakukan 3 kali yaitu menggunakan ukuran citra karakter 60x60, 80x80 dan 100x100. Kemudian citra karakter yang sudah diubah ukurannya digunakan dalam proses *Moment Invariant*. Nilai K pada proses K-NN yang digunakan dalam pengujian ini adalah 1. Tabel I merupakan hasil pengujian keakuratan berdasarkan ukuran citra karakter.

TABEL I
TABEL PENGUJIAN KEAKURATAN SISTEM BERDASARKAN UKURAN CITRA KARAKTER

No	Keakuratan Berdasarkan Ukuran		
	60x60	80x80	100x100
1	16/20(80%)	20/20(100%)	19/20(95%)
2	20/24(83.33%)	21/24(87.5%)	19/24(79.17%)
3	20/20(100%)	18/20(90%)	18/20(90%)
4	18/20(90%)	19/20(95%)	18/20(90%)

5	18/20(90%)	19/20(95%)	18/20(90%)
6	14/20(70%)	15/20(75%)	17/20(85%)
7	16/20(80%)	14/20(70%)	14/20(70%)
8	18/20(90%)	20/20(100%)	20/20(100%)
9	20/20(100%)	20/20(100%)	19/20(95%)
10	18/20(90%)	18/20(90%)	19/20(95%)
11	19/20(95%)	19/20(95%)	20/20(100%)
12	20/20(100%)	19/20(95%)	20/20(100%)
13	15/20(75%)	15/20(75%)	16/20(80%)
14	17/20(85%)	15/20(75%)	18/20(90%)
15	16/20(80%)	16/20(80%)	17/20(85%)
16	17/20(85%)	14/20(70%)	15/20(75%)
17	16/20(80%)	20/20(100%)	20/20(100%)
18	11/20(55%)	11/20(55%)	13/20(65%)
19	20/20(100%)	19/20(95%)	18/20(90%)
20	15/20(75%)	15/20(75%)	14/20(70%)
21	9/20(45%)	13/20(65%)	12/20(60%)
22	18/20(90%)	20/20(100%)	19/20(95%)
23	15/20(75%)	17/20(85%)	16/20(80%)
24	15/20(75%)	17/20(85%)	16/20(80%)
25	21/24(87.5%)	22/24(91.67%)	21/24(87.5%)
26	14/20(70%)	14/20(70%)	16/20(80%)
27	16/20(80%)	16/20(80%)	18/20(90%)
28	14/20(70%)	13/20(65%)	12/20(60%)
29	18/20(90%)	14/20(70%)	19/20(95%)
30	15/20(75%)	19/20(95%)	18/20(90%)
Rata-rata	82.03%	84.14%	85.39%

Pada gambar 13 merupakan grafik hasil dari pengujian ukuran citra karakter. Dari grafik diatas dapat dilihat bahwa ukuran 100x100 memiliki persentase keberhasilan tertinggi yaitu 85.39%. Persentase keberhasilan terendah adalah 82.03% yaitu didapatkan dari pengujian dengan menggunakan ukuran 60x60. Sedangkan untuk hasil pengujian menggunakan ukuran 80x80 menghasilkan persentase keberhasilan 84.14%. Berdasarkan grafik di atas semakin besar ukuran citra karakter semakin tinggi persentase keberhasilannya. Dengan demikian ukuran yang dipakai adalah 100x100 dan ukuran ini akan dipakai untuk pengujian selanjutnya yaitu pengujian nilai K untuk proses K-NN.



Gambar. 13. Grafik rata-rata persentase pengujian keakuratan berdasarkan ukuran citra karakter

2) Pengujian Nilai K pada Proses K-NN

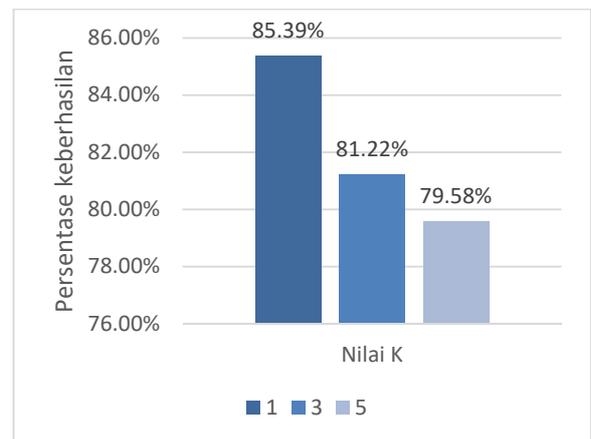
Pengujian kedua adalah pengujian untuk mencari nilai K yang memiliki persentase keberhasilan tertinggi pada proses K-NN. Pengujian akan dilakukan 3 kali dengan menggunakan nilai K pada proses K-NN yang berbeda-beda yaitu 1, 3 dan 5 dengan ukuran citra karakter yaitu 100x100 yang didapat dari pengujian sebelumnya. Tabel II merupakan hasil pengujian nilai K.

TABEL II
PENGUJIAN NILAI K PADA PROSES K-NN

No	Nilai K		
	1	3	5
1	19/20(95%)	17/20(85%)	17/20(85%)
2	19/24(79.17%)	15/24(62.5%)	15/24(62.5%)
3	18/20(90%)	14/20(70%)	15/20(75%)
4	18/20(90%)	18/20(90%)	19/20(95%)
5	18/20(90%)	16/20(80%)	17/20(85%)
6	17/20(85%)	16/20(80%)	13/20(65%)
7	14/20(70%)	14/20(70%)	14/20(70%)
8	20/20(100%)	20/20(100%)	20/20(100%)
9	19/20(95%)	17/20(85%)	14/20(75%)
10	19/20(95%)	19/20(95%)	18/20(90%)
11	20/20(100%)	20/20(100%)	19/20(95%)
12	20/20(100%)	20/20(100%)	20/20(100%)
13	16/20(80%)	16/20(80%)	15/20(75%)
14	18/20(90%)	17/20(85%)	18/20(90%)
15	17/20(85%)	17/20(85%)	17/20(85%)
16	15/20(75%)	16/20(80%)	17/20(85%)
17	20/20(100%)	20/20(100%)	19/20(95%)

18	13/20(65%)	12/20(60%)	11/20(55%)
19	18/20(90%)	14/20(70%)	15/20(75%)
20	14/20(70%)	14/20(70%)	14/20(70%)
21	12/20(60%)	10/20(50%)	9/20(45%)
22	19/20(95%)	19/20(95%)	19/20(95%)
23	16/20(80%)	16/20(80%)	14/20(70%)
24	16/20(80%)	15/20(75%)	13/20(65%)
25	21/24(87.5%)	19/24(79.17%)	18/24(75%)
26	16/20(80%)	14/20(70%)	15/20(75%)
27	18/20(90%)	18/20(90%)	18/20(90%)
28	12/20(60%)	13/20(65%)	13/20(65%)
29	19/20(95%)	19/20(95%)	19/20(95%)
30	18/20(90%)	18/20(90%)	17/20(85%)
Rata-rata	85.39%	81.22%	79.58%

Gambar 14 merupakan grafik hasil dari pengujian nilai K. Dari grafik di atas dapat dilihat bahwa pengujian dengan nilai K dengan nilai 1 memiliki persentase keberhasilan tertinggi yaitu 85.39%. Persentase keberhasilan terendah adalah 79.58% yaitu pengujian dengan menggunakan nilai 5 sebagai nilai K pada proses K-NN. Sedangkan nilai K dengan nilai 3 menghasilkan persentase keberhasilan sebesar 81.22%. Nilai K dengan nilai 3 dan 5 memiliki persentase hasil lebih rendah dibanding nilai K dengan nilai 1, hal ini dikarenakan adanya satu atau lebih *template* lain yang memiliki jumlah bobot lebih besar dari bobot *template* yang benar.



Gambar. 14. Grafik rata-rata persentase pengujian nilai K

3) Pengujian dengan menggunakan 20 citra uji baru tanpa ada *template* yang berasal dari citra uji

Pengujian ketiga bertujuan untuk mengetahui persentase keakuratan jika menggunakan citra uji baru di mana *template* atau data latih bukan berasal dari citra uji baru tersebut. Pengujian ini menggunakan 20 citra uji yang baru yang berbeda dari 30 citra uji sebelumnya. Untuk nilai K dan

ukuran citra karakter akan menggunakan hasil dari pengujian sebelumnya yaitu 1 untuk nilai k dan 100x100 untuk ukuran citra karakter. Tabel III merupakan hasil dari pengujian. Berdasarkan tabel, jika dilakukan pengujian dengan menggunakan citra uji baru di mana *template* tidak berasal dari citra uji baru akan menghasilkan rata-persentase yang lebih rendah dibanding dengan pengujian sebelumnya yaitu 81.54%.

TABEL III
PENGUJIAN DENGAN MENGGUNAKAN 20 CITRA UJI BARU TANPA ADA TEMPLATE YANG BERASAL DARI CITRA UJI

No	Persentase Keakuratan
1	19/20(95%)
2	13/20(65%)
3	13/20(65%)
4	14/20(70%)
5	17/20(85%)
6	13/20(65%)
7	15/20(75%)
8	16/20(80%)
9	16/20(80%)
10	19/20(95%)
11	15/20(75%)
12	18/20(90%)
13	20/20(100%)
14	20/20(100%)
15	17/20(85%)
16	17/20(85%)
17	17/20(85%)
18	18/20(90%)
19	14/24(58.33%)
20	21/24(87.5%)
Rata-rata	81.54%

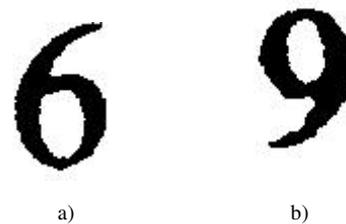
C. Analisis Sistem

Sesuai dengan pengujian diatas, dengan menggunakan citra karakter dengan ukuran 100x100 dan nilai K sebesar 1, sistem deteksi kesalahan letak buku perpustakaan UKDW dengan menggunakan metode *Moment Invariant* dapat mengenali karakter label dengan persentase keberhasilan sebesar 85.39% dari 30 citra uji yang di dalamnya terdapat citra karakter yang dipakai sebagai *template*. Dari 30 citra uji yang telah diuji terdapat beberapa citra memiliki tingkat keberhasilan 100%. Pengujian ketiga yaitu pengujian dengan menggunakan 20 citra uji baru di mana di dalam citra uji tersebut tidak terdapat citra karakter yang digunakan sebagai

template. Pengujian ini memiliki tingkat keberhasilan sebesar 81.54%.

Di samping memiliki persentase keberhasilan yang cukup tinggi, sistem ini memiliki beberapa keterbatasan. Tidak semua citra uji berhasil dikenali. Masih terdapat beberapa karakter pada citra uji yang tidak dikenali. Gagalnya sistem dalam mengenali karakter disebabkan oleh beberapa hal akan dijelaskan di bawah ini.

- Metode *Moment Invariant* merupakan sebuah metode yang menyimpan ciri dari sebuah citra tanpa memperhatikan posisi, ukuran dan arah. Hal ini menyebabkan sering terjadinya kesalahan dalam mengenali beberapa karakter seperti angka 6 dan 9. Gambar 15 merupakan contoh gambar untuk angka 6 dan 9 yang tertukar dengan angka 9 merupakan karakter yang benar dan karakter 6 merupakan karakter yang salah.



Gambar. 15. Contoh gambar a) angka 6 b) angka 9

Kemudian dari kedua gambar tersebut didapat nilai *Moment Invariant* yang dapat dilihat Tabel IV.

TABEL IV
NILAI MOMENT INVARIANT UNTUK GAMBAR 6 DAN 9

Moment ke	Angka	
	9	6
1	3.502888	3.604196
2	3.598339	3.385482
3	5.554393	5.1551
4	0.836843	0.834139
5	-0.19995	-0.20031
6	-0.22641	-0.16729
7	-1.79309	-1.71808

Dari Tabel IV dapat dilihat bahwa nilai moment dari kedua gambar tersebut sangat mirip dan jika dihitung jaraknya dengan menggunakan rumus *Euclidian Distance* maka akan menghasilkan nilai yang cukup kecil. Berdasarkan contoh di atas, angka 6 dan angka 9 sering tertukar disebabkan karena bentuk yang hampir sama sehingga menghasilkan nilai *Moment Invariant* yang mirip pula dan nilai *Moment* tersebut lebih kecil dibanding *template* benar lainnya sehingga angka 9 salah dikenali sebagai angka 6.

- Kegagalan pengenalan karakter juga dapat disebabkan oleh kegagalan dalam *cropping*. Kegagalan *cropping* sendiri dapat disebabkan oleh faktor pencahayaan.
- Perlu ditambahkan proses *pre-processing* seperti *normalization* untuk mengatasi beberapa masalah seperti pencahayaan yang terlalu gelap atau terang.
- Penambahan kemampuan sistem sehingga dapat mendeteksi buku perpustakaan dalam keadaan miring dan label yang tidak sejajar.



Gambar. 16. Contoh citra karakter yang gagal pada proses cropping

Gambar 16 merupakan contoh dari kegagalan *cropping*. Karena pencahayaan yang gelap mengakibatkan karakter menjadi satu dengan karakter lain pada proses binarisasi. Hal ini menyebabkan proses *Vertical Histogram* gagal dalam memotong kedua karakter tersebut dan menyebabkan terjadinya kesalahan pengenalan karakter.

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil pengujian dan analisis sistem yang dibuat, maka dapat disimpulkan bahwa:

- Sistem berhasil dalam mengimplementasikan metode *Moment Invariant* dan K-NN untuk mengenali karakter label buku. Implementasi dilakukan dengan mengubah ukuran citra menjadi 100x100 yang digunakan pada implementasi *Moment Invariant* dan nilai K sebesar 1 dalam implementasi K-NN. Untuk pengujian dengan menggunakan 30 citra uji di mana *template* berasal dari citra uji tersebut memiliki persentase keakuratan sebesar 85.39% sedangkan pengujian dengan menggunakan 20 citra uji di mana di dalamnya tidak ada citra karakter yang digunakan sebagai *template* memiliki persentase keakuratan sebesar 81.54%.
- Implementasi metode *Moment Invariant* pada kasus ini memiliki beberapa kelemahan dalam mengenali karakter yang mirip seperti 6 dan 9 yang membuat sistem seringkali tertukar dalam mengenali karakter tersebut. Hal ini disebabkan karena *Moment Invariant* menyimpan ciri tanpa memperhatikan ukuran, posisi dan arah dari objek.

B. Saran

Sistem deteksi kesalahan letak buku perpustakaan UKDW ini masih dapat dilakukan pengembangan lebih lanjut lagi. Penulis memberikan saran sebagai berikut:

- Penambahan fitur selain 7 fitur dari *Moment Invariant* seperti fitur yang berhubungan dengan orientasi. Sehingga karakter seperti 6 dan 9 dapat dibedakan dan dikenali dengan tepat.

UCAPAN TERIMA KASIH

Terima kasih kepada dosen-dosen Jurusan Teknik Informatika yang telah membimbing dan memberi saran sehingga penelitian ini bisa diselesaikan. Serta penulis juga berterima kasih kepada keluarga dan teman-teman penulis yang selalu memberi motivasi dan dukungan selama penelitian berlangsung.

DAFTAR PUSTAKA

- [1] A. W. Pujiyanto, "Implementasi Template Matching untuk Pengenalan Label dan Deteksi Kesalahan Letak Buku Perpustakaan Berbasis Android", Undergraduate Thesis, Universitas Kristen Duta Wacana, 2015.
- [2] H. A. D. Rani, E. Supriyati dan T. Khotimah, "Deteksi Iris Mata untuk Menentukan Kelebihan Kolesterol menggunakan Ekstraksi Ciri Moment Invariant dengan K-Means Clustering," *Prosiding Seminar Nasional Teknologi dan Informatika*, vol. 1, pp. 287-292, 2014.
- [3] H. V. Rizanti, Suprpto dan R. Regasari, "Pengenalan Citra Alphabet Berdasarkan Paramater Momen Invarian dengan Metode Case-Base Reasoning," *Repositori Jurnal Mahasiswa PTIIK UB*, vol. 1, 2013.
- [4] R. Brinkmann, *The Art and Science of Digital Compositing*, San Diego: Elsevier, 1999, p. 184.
- [5] C. N. Santi, "Mengubah Citra Berwarna Menjadi GrayScale," *Jurnal Teknologi Informasi DINAMIK*, pp. 14-19, 2011.
- [6] V. J. Dongre dan V. H. Mankar, "Devnagari Document Segmentation Using Histogram Approach," *International Journal of Computer Science, Engineering and Information Technology (IJCSIEIT)*, p. 47, August 2011.
- [7] L. B. A. Prabangkoro, "Algoritma Thinning stentiford untuk mendapatkan bentuk dasar huruf", Undergraduate Thesis, Universitas Sanata Dharma, 2008.
- [8] J. Parker, *Algorithms for Image Processing and Computer Vision*, Second Edition, Indianapolis: Wiley Publishing, Inc., 2011.
- [9] N. Jawas dan N. Suciati, "Image Inpainting using Erosion and Dilation Operation," *International Journal of Advanced Science and Technology*, pp. 128-134, 2013.
- [10] Z. Huang dan J. Leng, "Analysis of Hu's Moment Invariants on Image Scaling and Rotation," *2010 2nd International Conference on Computer Engineering and Technology*, pp. 476-480, 2010.
- [11] R. N. Whidhiasih, N. A. Wahanani dan S. , "Klasifikasi Buah Belimbing Berdasarkan Citra Red-Green-Blue Menggunakan KNN dan LDA," *Jurnal Penelitian Ilmu Komputer*, p. 31, 2013.
- [12] F. Özsakabasi, "Classification Of Forest Areas By K Nearest Neighbor Method:Case Study, Antalya.", Thesis, Middle East Technical University, 2008.