

# EKSTRAKSI CIRI PLAT NOMOR MOBIL MENGGUNAKAN MOMENT PUSAT

(*Extraction of Vehicle License Plate Identification Using Center Moment*)

Nur Wakhidah

Jurusan Teknologi Informasi dan Komunikasi Universitas Semarang

## **Abstract**

*Searching for an image by using form identification can be applied in detecting vehicle license plate from background of an image. Input image that still has noisy is repaired by some methods such as gray level, morphology, and edge detection. The result of an image enhancement is faded by segmentation method using moment to its identification extraction so that it is able to produce vehicle license plate image distinction.*

**Keywords** : *edge detection, moment, identification extraction*

## **1. PENDAHULUAN**

Perkembangan *computer vision* dan teknologi *image processing* hingga saat ini terus diperluas dengan tujuan untuk membantu manusia dalam melakukan pekerjaannya. *Computer vision* mempunyai tujuan untuk membuat keputusan yang berguna tentang objek fisik nyata dan pemandangan (*scenes*) berdasarkan *image* yang didapat dari sensor. Secara sederhana, *computer vision* ingin membangun sebuah mesin pandai yang dapat melihat. Dan *image processing* merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai gambar. Dalam *image processing* gambar yang ada diolah sedemikian rupa sehingga gambar tersebut lebih mudah untuk diproses.

Sistem pengenalan plat nomor mobil secara otomatis telah menjadi suatu aplikasi yang sangat penting dalam bidang *computer vision*. Penelitian tentang pengenalan plat nomor mobil Indonesia belum banyak dilakukan. Dimana pembacaan plat nomor yang dilakukan secara otomatis sangat berguna bagi kehidupan sehari-hari, misalnya manajemen tempat parkir, monitoring lalu lintas, dan pembayaran jalan tol.

Seperti saat ini sistem perparkiran yang telah diterapkan di sebagian besar wilayah Indonesia adalah sistem perparkiran manual dimana pencatatan nomor polisi pada plat nomor mobil dilakukan dengan cara memasukkan nomor polisi tersebut ke dalam komputer yang kemudian diproses untuk dicetak dan dihitung waktu parkirnya. Terdapat pula beberapa dari sistem perparkiran yang sudah mulai menggunakan kamera untuk menangkap gambar dari plat nomor mobil. Namun, pengambilan gambar tersebut hanya sebatas keperluan *database* yang berupa *image* saja.

Selain itu, sistem pembacaan karakter plat nomor mobil mempunyai keuntungan yaitu untuk melawan kejahatan, karena apabila diberikan kamera tambahan yang difokuskan pada wajah driver maka akan sekaligus dapat mengenali identitas driver mobil tersebut.

Teknologi pencarian citra berkembang kearah pencarian data citra berdasarkan isi visual dari citra. Pendekatan ini didasarkan pada fitur fitur yang dimiliki oleh citra seperti fitur warna, bentuk, tekstur atau kombinasi dari unsur-unsur tersebut. Salah satu penerapan pencarian citra berdasarkan isi visual adalah untuk mendeteksi apakah suatu citra itu

termasuk suatu citra porno atau bukan, dengan terlebih dahulu dilakukan training terhadap sistem mengenai ciri-ciri citra yang termasuk kategori porno dan tidak porno.(Wijaya,2004).

Dalam tulisan ini akan berusaha mencari citra plat nomor mobil dari *image* input. Pencarian citra diimplementasikan pada tahap segmentasi dengan ekstraksi ciri menggunakan moment yang dapat mendeskripsikan dari objek yaitu plat nomor mobil tersebut.

## 2. PERMASALAHAN

Bagaimana membuat sebuah sistem cerdas untuk dapat mengekstrak ciri dari plat nomor mobil menggunakan matlab?

## 3. DASAR TEORI

### 3.1. Pengolahan Citra Digital

Pengolahan Citra adalah suatu metode yang digunakan untuk mengolah citra (*gambar/image*) sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Secara umum dan sederhana, citra dapat didefinisikan sebagai representasi visual dari suatu objek. Lebih jauh citra juga dapat diartikan sebagai gambaran yang representatif mengenai suatu objek sedemikian sehingga citra tersebut dapat memberikan kesan yang

mendalam mengenai objek yang dimaksud. Jika ingin mendefinisikannya lebih bebas lagi, citra dapat didefinisikan sebagai bentuk visual yang dapat diterima secara baik oleh indera penglihatan, apapun bentuknya. Dalam bidang komputer, citra atau disebut juga *image* merupakan representasi visual dari suatu objek setelah mengalami berbagai transformasi data dari berbagai bentuk rangkaian numerik.

Sebuah citra (*gambar/image*) diartikan sebagai suatu fungsi kontinyu dalam dua dimensi dari intensitas cahaya ( $x,y$ ). Dimana  $x$  dan  $y$  menyatakan suatu koordinat, dan  $f$  pada setiap titik ( $x,y$ ) menyatakan intensitas atau tingkat kecerahan atau derajat keabuan (*brightness/gray level*).

Citra digital adalah citra kontinyu yang diubah ke dalam bentuk diskrit, baik koordinat maupun intensitas cahayanya. Dengan kata lain, citra digital dibuat dengan cara mencuplik suatu citra kontinyu dengan jarak seragam. Suatu titik terkecil pada citra digital sering disebut sebagai *picture element* atau *pixel*. Citra ini dapat berupa citra vektor ataupun citra bitmap. Citra ini mengandung persamaan-persamaan matematis dari bentuk-bentuk dasar yang membentuk citra tersebut. (Gonzales, 2002)

$$f(x, y) = \begin{Bmatrix} f(0,0) & . & . & . & f(0,n-1) \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ f(n-1,0) & . & . & . & f(n-1,n-1) \end{Bmatrix}$$

Gambar 1. Representasi citra digital

Komputer memiliki cara pandang tersendiri terhadap suatu citra. Berbeda dengan citra konvensional yang misalnya dengan melalui proses fotografis seperti pada foto dapat dihasilkan suatu citra nyata yang langsung dapat dinikmati oleh indera penglihatan, citra pada komputer harus melalui beberapa tahapan yang cukup rumit. Tahapan-tahapan

tersebut dapat digambarkan sebagai suatu rangkaian proses dari proses akuisisi data, manipulasi data, visualisasi data, serta proses penyimpanan data.

Diantara proses tersebut diatas, proses penyimpanan data adalah bagian paling pelik dari pengolahan citra pada komputer. Saat ini sudah terdapat banyak metode yang dapat

digunakan untuk menyimpan suatu citra digital ke dalam suatu file. File adalah media penyimpanan standar pada sistem komputer. Metode-metode penyimpanan tadi tentunya memiliki perbedaan yang cukup berarti antara satu dengan yang lainnya. Hal yang paling membedakan biasanya adalah tingkat kompresi data yang ditawarkan oleh masing-masing metode. Beberapa metode yang sudah cukup dikenal adalah BMP, JPEG, PCX, WMF, TIFF, dll.

Operasi pengolahan citra digital umumnya dilakukan dengan tujuan memperbaiki kualitas suatu gambar sehingga dapat dengan mudah diinterpretasi oleh mata manusia dan untuk mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis. (Ahmad, 2005)

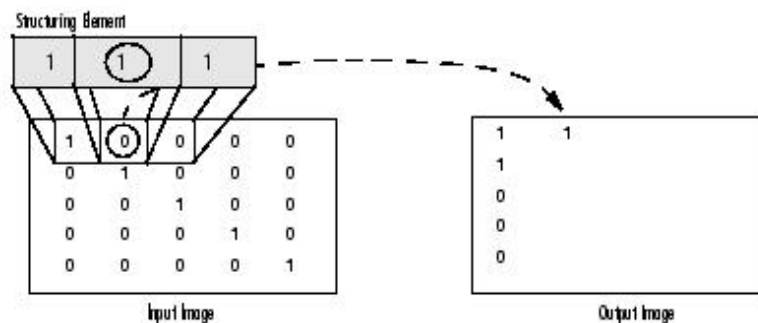
### 3.2. Mathematics Morphology

Morfologi Matematika (*Mathematics Morphology*) adalah sebuah metode untuk analisa *image* berbasis operasi tetangga non-linear (*Nonlinear Neighbourhood Operation*).

Tetangga disini sering disebut dengan *Structuring Element* (SE). Operasi dasar dari morfologi matematika ini adalah erosi dan dilatasi.

Dilatasi menambahkan *pixel* pada batas dari objek di sebuah image, sedangkan erosi mengurangi *pixel* pada batas dari objek. Jumlah *pixel* yang ditambahkan atau dikurangkan tergantung dari besar dan bentuk dari SE yang digunakan untuk mengolah citra. (Matlab, 2008)

Gambar berikut mengilustrasikan bagaimana proses dari dilatasi sebuah citra biner. Dapat dilihat bagaimana SE mengubah tetangga dari *pixel interest* (bagian yang dilingkari). Fungsi dilatasi membuat sebuah aturan kepada *pixel* tetangga dan memberikan sebuah nilai yang dikorespondasikan kepada *pixel* di citra keluaran. Pada gambar, proses dilatasi memberikan nilai *pixel* citra keluaran dengan nilai "1" karena salah satu dari elemen tetangga yang didefinisikan oleh SE dalam posisi aktif.



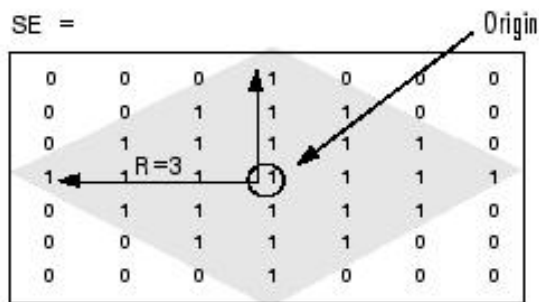
Gambar 2. Proses dilatasi pada citra biner

Bagian penting dari proses morfologi matematika adalah *structuring element* (SE). SE digunakan untuk memodifikasi citra masukan. Pada dasarnya SE adalah sebuah matriks yang terdiri dari "0" dan "1", dimana matriks-matriks tersebut memiliki sebuah ukuran dan bentuk tertentu. *Pixel* yang mempunyai nilai 1 mendefinisikan "tetangga". SE dua dimensi biasanya memiliki ukuran

yang lebih kecil daripada citra yang akan diolah. *Pixel* pusat dari SE, *origin*, mengidentifikasi *pixel of interest* dari *pixel* yang akan diolah. (Matlab, 2008). Jenis-jenis dari SE antara lain adalah:

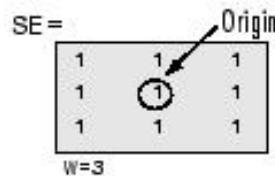
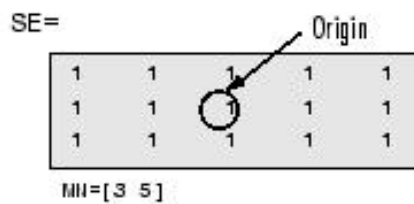
a. Diamond

SE yang berbentuk diamond dengan R adalah jarak dari *origin* ke ujung/tepi dari SE diamond.



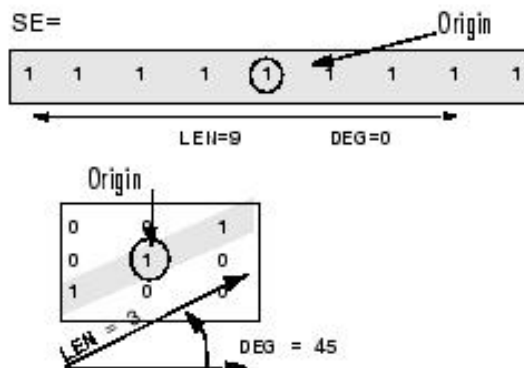
Gambar 3. SE Diamond

- b. Rectangle/Square dari dua buah elemen vector *nonnegative integers*. M adalah ukuran untuk baris dan N adalah ukuran untuk kolom.
- SE yang berbentuk persegi atau kotak. MN merepresentasikan ukuran dari SE. MN terdiri



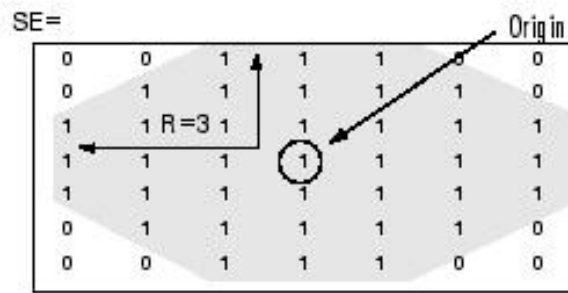
Gambar 4. SE Rectangle/Square

- c. Line yang diukur dari arah sumbu horizontal. LEN dapat diartikan jarak dari titik ujung SE ke ujung SE lainnya.
- Sebuah SE yang *flat* dan linier, dimana LEN merepresentasikan panjang dan DEG merepresentasikan sudut (dalam derajat) *line*



Gambar 5. SE Line

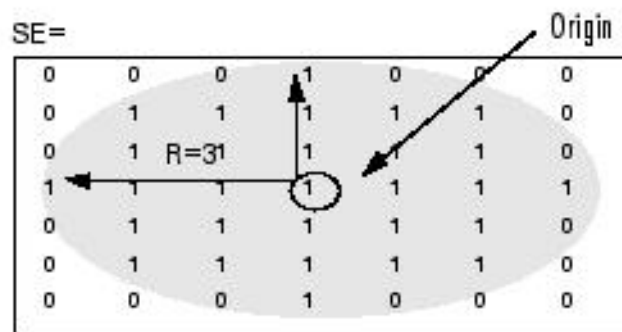
- d. Octagon SE yang berbentuk segi-8, dimana R adalah jarak dari *origin* SE dengan tepian dari segi-8, diukur dari sumbu x dan sumbu y.



Gambar 6. SE Octagon

e. Disk

SE berbentuk lingkaran/disk, dimana R adalah jari-jari yang diukur dari *origin* ke tepi dari lingkaran.



Gambar 7. SE Disk

Erosi dan dilatasi sendiri jarang digunakan secara terpisah. Biasanya dilakukan kombinasi antara keduanya:

- Opening: Kombinasi dari erosi-dilatasi dengan SE yang sama. Operasi ini akan menghapus "lubang" putih pada objek yang gelap (hitam).
- Closing: Kombinasi dari dilatasi-erosi dengan SE yang sama. Operasi ini akan menghapus "lubang" hitam pada permukaan terang/putih.

### 3.3. Ekstraksi Ciri

Ekstraksi ciri merupakan langkah awal dalam melakukan klasifikasi dan interpretasi citra. Proses ini berkaitan dengan kuantisasi karakteristik citra ke dalam sekelompok nilai ciri yang sesuai. Salah satu ciri yang dapat digunakan adalah momen, dimana momen dapat mengkombinasikan properti luas area objek, kepadatan objek, ketidakteraturan menjadi satu. (McAndrew, 2004)

#### 3.3.1. Momen

Momen dapat menggambarkan suatu objek dalam hal area, posisi, orientasi dan parameter terdefinisi lainnya. Persamaan dasar dari momen suatu objek didefinisikan sebagai berikut

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

dengan order dari momen adalah  $(i + j)$ .  $x$  dan  $y$  menyatakan koordinat titik, sedangkan  $a_{xy}$  menyatakan intensitas titik.

Momen tingkat ke-0 dan ke-1 (*zero- and first-order moments*) didefinisikan sebagai berikut.

$$m_{00} = \sum_x \sum_y a_{xy}$$

$$m_{10} = \sum_x \sum_y x a_{xy}$$

$$m_{01} = \sum_x \sum_y y a_{xy}$$

Pada citra biner yang mana  $a_{xy}$  akan bernilai 0 atau 1, momen tingkat ke-0 ( $m_{00}$ ) adalah sama dengan area dari objek.

Pusat dari area atau massa (*centroid*) adalah parameter yang baik untuk menyatakan lokasi dari objek. Pusat area dari objek didefinisikan sebagai berikut.

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

dengan  $(x', y')$  merupakan pusat koordinat dari objek.

Momen pusat (*central moment*)  $m$  adalah momen yang bersesuaian dengan pusat area, didefinisikan sebagai berikut.

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

dan momen pusat yang ternormalisasi dinyatakan dengan persamaan berikut.

$$\mu_{ij} = \frac{\mu_{ij}}{(\mu_{00})^\lambda}$$

dengan  $\lambda = \frac{(i+j)}{2} + 1$  dengan  $(i+j) \geq 2$  (momen tingkat ke-1 adalah selalu invarian).

Dari momen ternormalisasi di atas, sekumpulan momen-momen invarian (*invariant moments*) dapat didefinisikan. Momen-momen ini sangat berguna dalam membuat vektor ciri untuk pengenalan objek. Berikut ini adalah persamaan dari momen-momen invarian.

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \end{aligned}$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

$$\phi_6 = (\eta_{20} - \eta_{02})\{(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} + 4\eta_{11}(\eta_{30} + \eta_{12}) + (\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

Orientasi dari sebuah objek dapat didefinisikan sebagai berikut.

$$\theta = \frac{1}{2} \tan^{-1} \left[ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right]$$

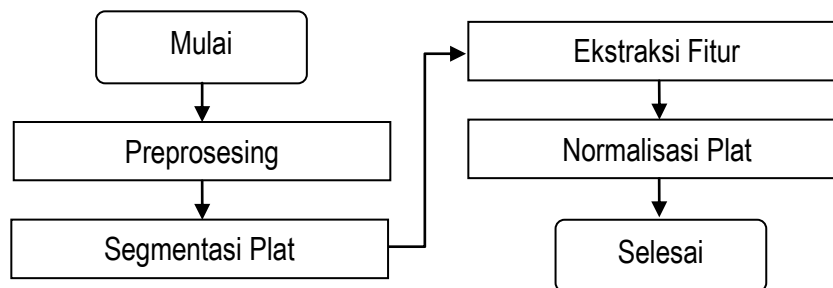
dengan  $\theta$  adalah besar sudut yang menyatakan orientasi dengan sumbu x. Besarnya penguluran/pemanjangan (*eccentricity*) suatu objek dapat diukur sebagai

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{\mu_{00}}$$

#### 4. PEMBAHASAN

Implementasi program dalam menyelesaikan permasalahan digunakan pemrograman MATLAB. Data citra yang digunakan dalam paper ini terdiri dari beberapa citra yang berukuran 3264 x 2448 pixel dan komponen warna truecolor.

Untuk menyelesaikan permasalahan tersebut, tahap-tahap yang dilakukan adalah :



Gambar 8. Tahapan Ekstraksi Fitur Plat Nomor Mobil

#### 4.1. Preprocessing

Tahap preprocessing dilakukan untuk mengolah citra yang diambil agar noise yang

ada bisa dihilangkan semaksimal mungkin, sehingga berfungsi dapat mengkondisikan gambar untuk ditemukan platnya dan memisahkannya plat dari latarbelakangnya.

Tahap preprocessing, meliputi:

#### 4.1.1. Perubahan Ukuran Image

Pembacaan image dilakukan terlebih dahulu untuk mengetahui ukurannya, dengan perintah

```
data = 'm2.jpg';
asli1 = imread(data);
ukuran = size (asli1);
```

Kemudian image diubah ukurannya sebesar 0.4 kali dari besar image semula bertujuan untuk mempercepat proses, dengan perintah

```
asli3 = imresize(asli1,0.4,'nearest');
```

proses `imresize` akan mengubah ukuran image dengan menggunakan metode interpolasi, dimana interpolasi adalah proses yang dikerjakan oleh perangkat lunak untuk melakukan pembuatan ulang (*resample*) dari contoh image untuk menentukan nilai-nilai antara piksel-piksel yang ditetapkan. Metode interpolasi yang disediakan oleh matlab 7.7.0 (R2008b) adalah Nearest-neighbor interpolation, Bilinear interpolation, dan Bicubic interpolation. Dalam paper ini akan menggunakan metode nearest neighbor interpolation.

Perbedaan untuk ketiga metode diatas adalah :

Metode interpolasi tetangga terdekat, nilai piksel keluaran ditetapkan dari nilai piksel dari suatu titik yang ditentukan letak posisinya yang terdekat. Adapun piksel-piksel yang lain tidak diperhitungkan

Metode interpolasi *Bilinear*, nilai piksel keluaran adalah nilai rata-rata pembobotan piksel-piksel tetangga berukuran 2x2 (berarti ada 4 nilai piksel) yang terdekat.

Metode interpolasi *Bicubic*, nilai piksel keluaran adalah nilai rata-rata pembobotan piksel-piksel tetangga berukuran 4x4 yang terdekat (berarti ada 16 nilai piksel)

Jumlah piksel yang akan diikutsertakan dalam perhitungan akan mengakibatkan

kompleksitas dari komputasi komputer. Sehingga metode interpolasi bilinear akan lebih lambat daripada metode interpolasi tetangga terdekat, dan metode interpolasi Bicubic akan lebih lambat daripada metode interpolasi bilinear. Akan tetapi, semakin banyak jumlah piksel yang diperhitungkan, maka hasilnya akan semakin akurat. Dalam hal ini, ada pertimbangan antara waktu proses dan kualitas image.

#### 4.1.2. Greylevel

Suatu citra digital dapat diwakili oleh ruang warna RGB (*Red-Green-Blue*) untuk setiap titiknya, dimana setiap komponen warna memiliki batasan sebesar 1 byte (Gonzalez, 2002). Jadi untuk masing-masing komponen R, G, dan B mempunyai variasi dari 0 sampai 255. Total variasi yang dapat dihasilkan untuk sistem dengan format warna RGB adalah  $256 \times 256 \times 256$  atau 16.777.216 jenis warna. Karena setiap komponen warna memiliki batasan sebesar 1 byte atau 8 bit, maka total untuk mempresentasikan warna RGB adalah  $8+8+8 = 24$  bit.

*Graylevel* adalah tingkat warna abu-abu dari sebuah piksel, dapat juga dikatakan tingkat cahaya dari sebuah piksel. Maksudnya nilai yang terkandung dalam piksel menunjukkan tingkat terangnya piksel tersebut dari hitam ke putih. Biasanya ditetapkan nilainya antara 0 hingga 255 (untuk 256-graylevel), dengan 0 adalah hitam dan 255 adalah putih. Karena hanya terbatas 1 byte saja maka untuk mempresentasikan nilai piksel cukup 8 bit saja. Grayscale adalah gambar yang memiliki graylevel sebagai nilai dari tiap pikselnya.

*Grayscale* adalah proses perubahan nilai piksel dari warna (RGB) menjadi graylevel (Gonzalez, 2002). Pada dasarnya proses ini dilakukan dengan meratakan nilai piksel dari 3 nilai RGB menjadi 1 nilai. Tetapi karena ketiga warna pokok tersebut dianggap tidak seragam dalam hal kemampuan kontribusi pada kecerahan, ada yang berpendapat bahwa cara konversi yang lebih tepat adalah dengan

menggunakan persamaan sebagai berikut (Ahmad, 2005) :

$$Y = 0.299R + 0.587G + 0.114B$$

dimana Y adalah nilai kecerahan suatu piksel pada citra abu-abu, dengan persentasi 29,9% dari warna merah (R), 58,7% dari warna hijau (G), dan 11,4% dari warna biru (B).

Perintah yang diberikan :

```
j = rgb2gray(asli3);
```

#### 4.1.3. Edge Detection

Operasi ini digunakan untuk menentukan lokasi titik-titik yang merupakan tepi obyek citra. (Ahmad, 2005). Secara umum, tepi suatu obyek dalam citra dinyatakan sebagai titik yang nilai warnanya berbeda cukup besar dengan titik yang ada disebelahnya. Ada beberapa mask yang telah dirancang untuk deteksi tepi, salah satunya adalah operator Sobel. Perintahnya adalah

```
bw2=edge(j,'sobel');
```

Rumusan pengaturan pixel di sekitar pixel (x,y) adalah:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator Sobel adalah magnitudo dari gradien yang dihitung dengan

$$M = \sqrt{(s_x^2 + s_y^2)}$$

dan turunan parsial dihitung dengan

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

dengan konstanta c=2. Dalam bentuk mask, sx dan sy dinyatakan sebagai:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

#### 4.1.4. Opening dan Closing

Langkah selanjutnya adalah menyambung pixel agar menjadi sebuah objek dan menghilangkan noise. Dilatasi dapat menambahkan *pixel* pada batas dari objek di sebuah image, sedangkan erosi mengurangi *pixel* pada batas dari objek. Erosi dan dilatasi sendiri jarang digunakan secara terpisah.

Biasanya dilakukan kombinasi antara keduanya:

- Opening: Kombinasi dari erosi-dilatasi dengan SE yang sama. Operasi ini akan menghapus "lubang" putih pada objek yang hitam
- Closing: Kombinasi dari dilatasi-erosi dengan SE yang sama. Operasi ini akan menghapus "lubang" hitam pada permukaan putih.

Perintah yang digunakan untuk menyambung pixel yang mempunyai jarak kurang dari 10 piksel menjadi satu objek yang dijadikan kandidat area plat adalah

```
se = strel('disk',10);
```

```
closeBW2 = imclose(bw2,se);
```

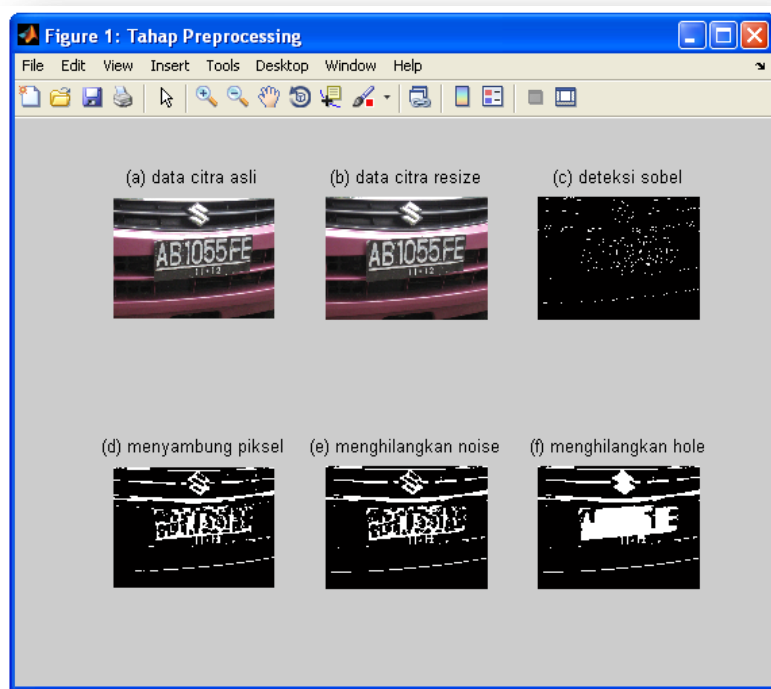
Kemudian untuk menghilangkan noise yaitu luasan area yang mempunyai luas kurang dari 10 pixel, sehingga luasan yang tidak digunakan akan dibersihkan. Terdapat pada perintah

```
bwvv2=bwareaopen(closeBW2,10);
```

dan yang terakhir dalam proses preprocessing adalah menghilangkan hole yang ada pada citra biner tersebut, yaitu dengan perintah

```
bw22=imfill(bwvv2,'holes');
```





Gambar 9. Tahap-tahap preprocessing

#### 4.2. Segmentasi Plat

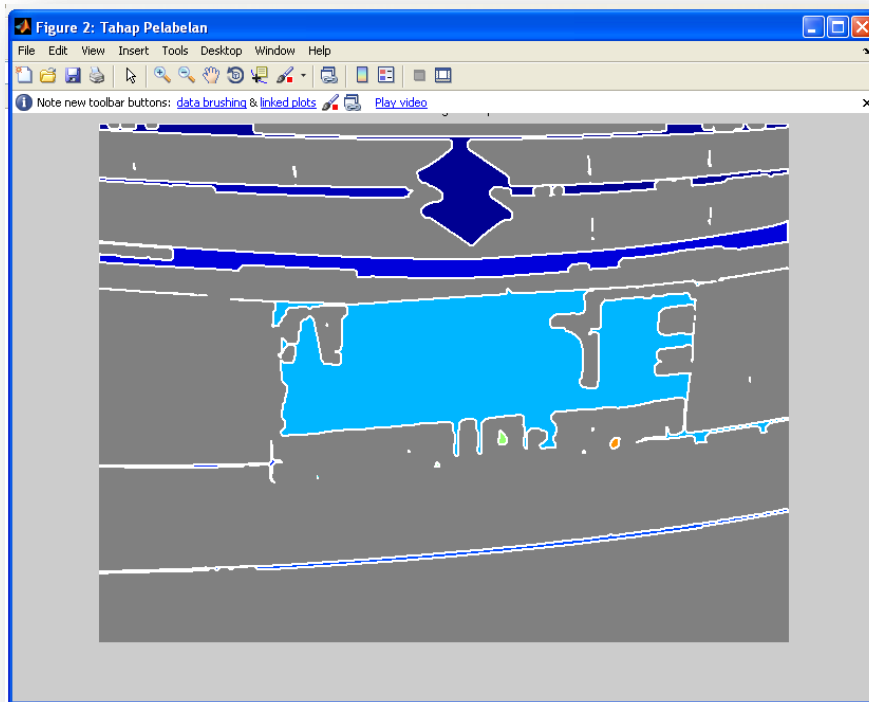
Tahapan segmentasi adalah tahap memisahkan objek yaitu plat nomor mobil dari latar belakangnya. Tahap-tahap yang dilakukan dalam segmentasi plat adalah :

- a. Memberi label pada semua objek yang terdapat dalam citra

Citra yang diperoleh setelah tahap preprocessing memiliki beberapa objek. Untuk memisahkan objek yang dinyatakan sebagai objek plat, maka perlu melakukan pelabelan

pada tiap-tiap objek. Pelabelan komponen merupakan suatu proses pemberian label yang sama pada sekumpulan piksel pembentuk objek yang saling berdekatan. Untuk melakukan *labeling* tersebut, terlebih dahulu yang dilakukan adalah menelusuri tepian dari tiap-tiap objek dengan menggunakan perintah

```
[B,L] = bwboundaries(bw22,'noholes');
for k=1:length(B)
    boundary = B{k};
end
```



Gambar 10. Tahap Pelabelan

b. Menentukan objek yang terluas sebagai plat nomer

Setelah mencari tepian dari masing-masing objek yang merupakan kandidat-kandidat dari plat nomer, yang dilakukan berikutnya adalah mencari luas areanya.

Area suatu objek adalah jumlah piksel penyusun objek tersebut dan unit umum yang digunakan adalah piksel, karena sejumlah piksel tadi membentuk suatu luasan. Area dapat mencerminkan ukuran atau berat objek sesungguhnya. Hal ini berlaku untuk benda pejal dengan bentuk yang hampir seragam, tetapi tidak demikian untuk benda yang berongga (Ahcmad, 2005)

Perimeter adalah bagian terluar dari suatu objek yang bersebelahan dengan piksel latar belakang (Ahcmad, 2005). Oleh karena itu, perimeter mempunyai beberapa definisi yang berbeda, namun sebenarnya mempunyai maksud yang sama.

Ketiga definisi tersebut adalah:

1. Jumlah panjang garis yang memisahkan sepanjang piksel  $p$  dan  $q$  dimana  $p \in S$  dan  $q \in S$
2. Jumlah langkah yang diambil dalam menemukan batas daerah
3. Jumlah piksel dari batas daerah

dari ketiga definisi diatas, definisi ketiga lebih mudah dilakukan dan lebih banyak digunakan. Jadi nilai perimeter dapat dicari dengan menghitung banyaknya piksel yang merupakan piksel-piksel yang berada pada perbatasan dari objek-objek tersebut.

Kekompakan (*compactness*) suatu objek dapat diukur melalui persamaan berikut:

$$C = \frac{4\pi A}{P^2}$$

Dimana  $C$ ,  $A$  dan  $P$  adalah Kekompakan, Area dan Perimeter. Dengan menganalisis factor bentuk kekompakkan, objek-objek dengan tepi yang rata akan memperlihatkan nilai yang berbeda dengan objek-objek dengan tepi bergerigi. Hal ini dapat digunakan misalnya untuk mengidentifikasi bentuk dan

ukuran objek yang sama, tetapi dengan profil tepi yang berbeda. Objek dengan tepi yang rata akan memberikan nilai C sekitar 1, dan semakin mengecil bila tepinya tidak rata atau membentuk lintasan yang berliku-liku (Achmad,2005).

Perintah yang digunakan dalam tahap ini adalah:

```
stats=
regionprops(L,'Area','Centroid','BoundingBox');
threshold = graythresh(asli1);
% loop boundaries
for k = 1:length(B)
% Mencari koordinat (X,Y)label 'k'
    boundary = B{k};
% object's perimeter
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq,2)));
% Mencari luas area
    area = stats(k).Area;

% Menghitung roundness metric
    metric = 4*pi*area/perimeter^2;
% Menampilkan hasil
    areatulis = sprintf('%2.2f',area);
    if area>maksluas
        maksluas=area;
```

```
end
% Menandai
    if metric > threshold
        centroid = stats(k).Centroid;
    end
end
```

Setelah ekstraksi ciri telah ditemukan, maka mencari luas maksimum area yang dianggap sebagai plat nomor mobil, perintahnya adalah :

```
%Mencari luas area yang terluas
for k = 1:length(B)
    area=stats(k).Area;
    if maksluas==area
        lihat=stats(k).BoundingBox;
    end
end
```

maka nilai boundingbox yang tersimpan pada variable lihat merupakan koordinat-koordinat dari area yang terluas dan akan disimpan dalam database sebagai hasil ekstraksi cirri yang diperoleh.

```
conn = database('finalPCD','');
insert(conn,'fitur',{luas'},maksluas);
close(conn);
```

ID	luas	pusat	batas
4	165047		
5	266119		
6	128484		
7	265966		
8	186147		
9	161992		
10	100544		
*	(New)		

Gambar 11. Database

kemudian dilakukan pemotongan untuk memperoleh citra yang berupa plat nomor mobil. Perintah yang diberikan :

```

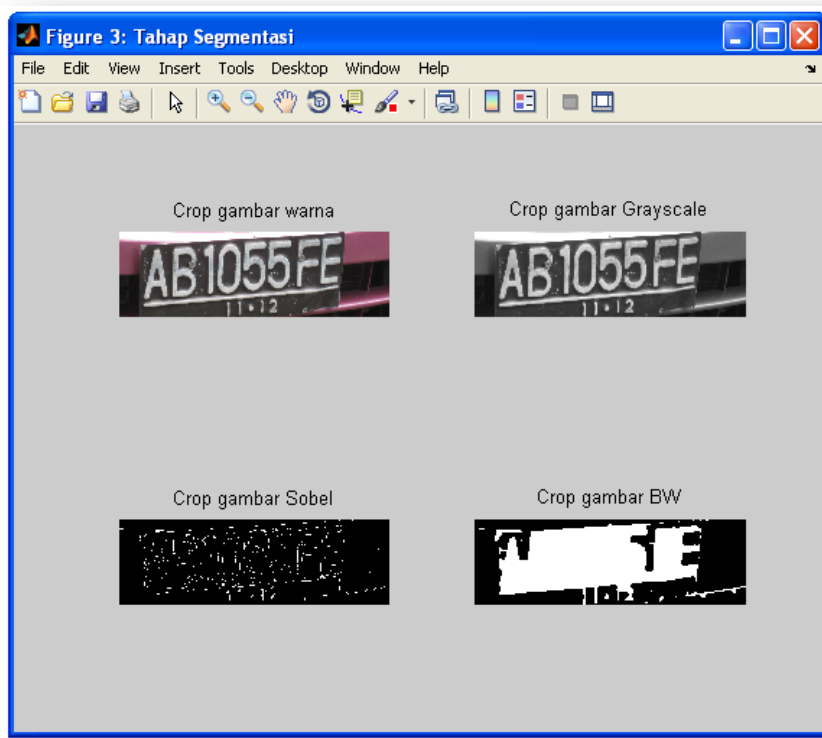
I12 = imcrop(asli3,lihat);
I13 = imcrop(j,lihat);
I14 = imcrop(bw2,lihat);
I15 = imcrop(bw22,lihat);
Figure;
subplot(2,2,1);imshow(I12);
title('Crop gambar warna');

```

```

subplot(2,2,2);imshow(I13);
title('Crop gambar Grayscale');
subplot(2,2,3);imshow(I14);
title('Crop gambar Sobel');
subplot(2,2,4);imshow(I15);
title('Crop gambar BW');
B=im2bw(I13,threshold);
figure;
imshow(B);title ('Plat Nomor Mobil Hasil Segmentasi');

```



Gambar 12. Tahap Segmentasi

#### 4.3. Normalisasi Plat

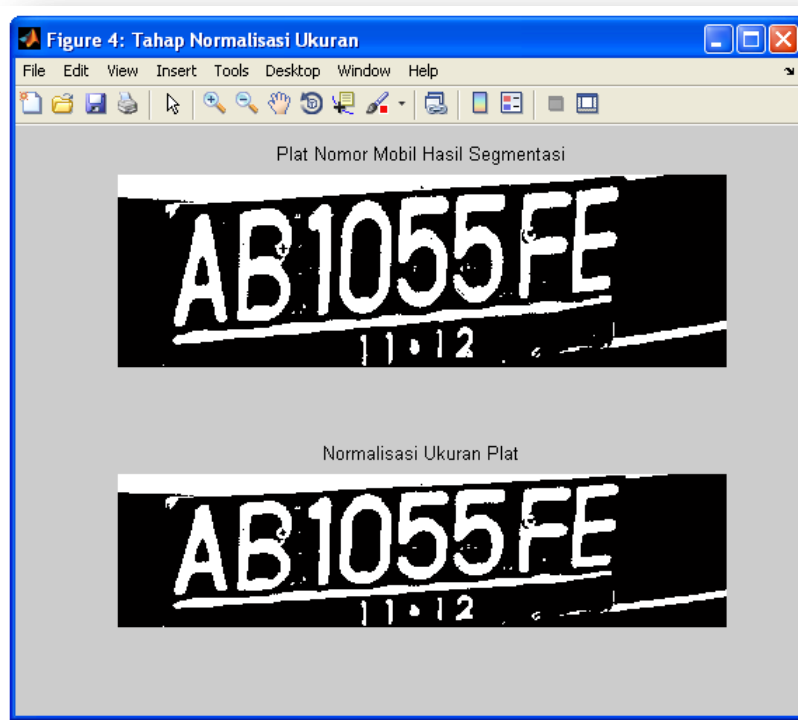
Tahapan yang dilakukan pada normalisasi plat adalah normalisasi ukuran, dimana ukuran plat nomot yang telah disegmentasi menjadi

ukuran standart yaitu 200 x 800 pixel. Perintah yang diberikan adalah

```

bw1=imresize(B,[200 800],'bilinear');
figure;imshow(bw1); title('Normalisasi Ukuran Plat');

```



Gambar 13. Tahap Normalisasi

## 5. KESIMPULAN

Dari hasil ujicoba di atas dapat ditarik kesimpulan sebagai berikut :

1. Untuk pencarian citra dengan menggunakan pengenalan bentuk dapat menggunakan 2 buah descriptor, yaitu deskripsi border (berdasarkan garis batas objek) dan deskriptor region (berdasarkan area objek)
2. Secara analisa open dan closing cukup baik digunakan sebagai pemrosesan awal citra dan momen mampu sebagai alat penajaman informasi untuk pengenalan objek, yaitu plat nomor mobil

## DAFTAR PUSTAKA

Alasdair McAndrew, 2004 , *An Introduction to Digital Image Processing with Matlab*, School of Computer Science and Mathematics, Victoria University of Technology

Agus Harjoko, Kusri, *Pencarian Citra Visual Berbasis Isi Citra Menggunakan Fitur Warna Citra*,

Rafael C. Gonzalez, Richard E. Woods, 1993, *Digital Image Processing*, Addison-Wesley Publishing Company Inc., USA

Usman Ahmad, 2005, *Pengolahan Citra Digital*, Graha Ilmu, Yogyakarta

I Gede Pasek Suta Wijaya, IBK Widiartha, 2004, *Pengenalan Citra Porno Berbasis Kandungan Informasi Citra (Image Content)*, Jurnal Teknik Elektro Vol.4 No.2

License Plate Recognition - A Tutorial  
[www.licenseplaterecognition.com](http://www.licenseplaterecognition.com)  
 (diakses tanggal 1 Januari 2013)

Image Processing & Computer Vision Indonesia.  
[www.wibirama.com/dip](http://www.wibirama.com/dip)  
 (diakses tanggal 1 Januari 2013)

Pengolahan Citra Digital.  
[http://viplab.if.its.ac.id/pcd\\_online](http://viplab.if.its.ac.id/pcd_online)  
 (diakses tanggal 5 Januari 2013)