

# IMPLEMENTASI METODE SOBEL EDGE DAN TEMPLATE MATCHING MENGUNAKAN PERANGKAT RASPBERRY PI SEBAGAI SISTEM PENGENALAN PLAT NOMOR OTOMATIS

## IMPLEMENTATION OF SOBEL EDGE METHOD AND TEMPLATE MATCHING METHOD BY USING RASPBERRY PI FOR AUTOMATIC NUMBER PLATE RECOGNITION

<sup>1</sup>Moch. Erkki Svante N<sup>2</sup>Gelar Budiman,ST,MT<sup>3</sup>Yuli Sun Hariani,ST,MT<sup>1,2</sup>Fakultas Teknik Elektro – Universitas Telkom<sup>3</sup> Fakultas Ilmu Terapan

Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

<sup>1</sup> erkkinfors14@gmail.com<sup>2</sup> gelarbudiman@telkomuniversity.ac.id<sup>3</sup>yulisun@tass.telkomuniversity.ac.id

### ABSTRAK

Selama beberapa tahun terakhir, *intelligent transport system* (ITS) memiliki dampak yang luas dalam kehidupan masyarakat sebagai ruang lingkup untuk meningkatkan keselamatan transportasi, *mobility* untuk meningkatkan produktivitas melalui penggunaan teknologi. Salah satu teknologi dari ITS adalah *automatic number plate recognition* (ANPR) dimana sistem ANPR adalah salah satu bagian dalam bidang *computer vision*. Teknologi ANPR yang biasa digunakan dalam berbagai bidang seperti sistem pembayaran elektronik (pembayaran tol, biaya parkir). Dalam Tugas Akhir ini peneliti menemukan tantangan untuk merancang sistem ANPR menggunakan perangkat raspberry pi karena dengan memori nya yang kecil dan untuk sistem ANPR banyak sekali proses-proses yang harus dilakukan sehingga waktu komputasi dikhawatirkan akan menjadi kendala. Dua metode utama yang digunakan dalam Tugas Akhir ini adalah *sobel edge detection* dan *template matching* untuk pengenalan karakternya. Tahap-tahap yang akan dilakukan meliputi: *pre-processing*, segmentasi, dan *character recognition*. Metode *sobel edge* yang digunakan adalah *vertical edge detection* dan untuk pengenalan akan digunakan *normalized cross correlation* dan *normalized correlation coefficient*. Tugas akhir ini dapat mempunyai tingkat akurasi maksimal sebesar 94.4 persen dengan waktu komputasi maksimal yaitu 5.6 detik

**Kata Kunci :** *Raspberry pi, ITS, ANPR, sobel edge, template matching*

### ABSTRACT

Over the past few years, intelligent transport systems (ITS) have a broad impact in people's lives as the scope for improving transportation safety, mobility to increase productivity through the use of technology. One of ITS technology is the automatic number plate recognition (ANPR) where the ANPR system is one of the sections in the field of computer vision. ANPR technology commonly used in various fields such as electronic payment system (payment of tolls, parking fees). In this Final researchers found a challenge to design the ANPR system using the raspberry pi due to its small memory and for a lot of ANPR system processes which must be done so that the computing time will become a constraint. Two main methods used in this final project is the Sobel edge detection and template matching for recognition karakternya. Tahap-phase will include: pre-processing, segmentation, and character recognition. Sobel edge method used is the vertical edge detection and for the introduction of the normalized cross correlation will be used and normalized correlation coefficient. This system can have a maximum rate of 94.4 percent accuracy with maximum computing time is 5.6 seconds

**Keywords :** *Raspberry pi, ITS, ANPR, sobel edge, template matching*

### 1. Pendahuluan

Perkembangan teknologi yang pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan diberbagai macam aspek. Teknologi dibutuhkan oleh semua orang untuk dapat mempermudah suatu pekerjaan dengan cepat. Salah satu teknologi yang paling banyak digunakan saat ini adalah komputer. Dimana dengan menggunakan komputer semua orang bisa mengakses internet dengan mudah dan mendapat apapun yang diinginkan. Komputer selalu berkembang dari waktu ke waktu. Perkembangan komputer tersebut dapat dilihat dari adanya sistem pengelihan komputer (*computer vision*). Melalui *computer vision* muncul berbagai penelitian di bidang pengolahan citra digital yang salah satunya adalah pembacaan plat nomor kendaraan otomatis atau biasa disebut dengan *automatic plate number recognition* (ANPR). Untuk itu dibutuhkan ketelitian yang sangat tinggi agar bisa menghindari kekeliruan

dalam membaca plat nomor tersebut, namun dalam prosesnya masih banyak yang menggunakan manusia sebagai proses *input*, dimana manusia memiliki ketahanan yang terbatas dan dapat menyebabkan kesalahan dalam proses input.

Secara umum proses pengenalan dilakukan dalam enam tahap, yaitu lokalisasi plat, orientasi dan mengubah ukuran plat, normalisasi, untuk mengubah tingkat contrast dan kecerahan warna, segmentasi karakter, pengenalan karakter *optic*, serta analisis sintaksis. Tahapan yang dilakukan dapat bervariasi tergantung metode yang digunakan. Algoritma-algoritma tersebut dikembangkan melalui tiga langkah dasar, yakni pencarian area plat nomor, segmentasi karakter dari plat nomor dan pengenalan dari setiap karakter [14] [Angga Kurnia Sasotya]. Penelitian ini mengacu kepada penelitian [14] yang juga menggunakan metode yang sama tetapi dalam penelitian ini akan diimplementasikan pada hardware raspberry pi dengan beberapa parameter yang berbeda. Hasil dari penelitian ini sudah bisa menutupi salah satu kekurangan dari penelitian [14] dimana tidak ada *matching* yang ambigu antara karakter dan angka.

Untuk proses pengenalan suatu karakter, dari gambar maka akan digunakan *template matching*. *Template matching* merupakan sebuah teknik yang dapat digunakan pada pengolahan citra digital untuk menemukan bagian kecil, pada sebuah citra yang cocok dengan citra template yang digunakan. Sedangkan, dalam Tugas Akhir ini, akan digunakan perangkat raspberry pi sebagai hardware dimana image processing akan dilakukan dan data-datanya akan disimpan didalam *memory SD card*. Oleh karena itu, tugas akhir ini akan membahas tentang implementasi metode *sobel edge* dan *template matching* menggunakan perangkat raspberry pi sebagai sistem pengenalan plat nomor otomatis.

## 2. Dasar Teori

### 2.1 Raspberry pi

Raspberry pi adalah sebuah komputer biaya rendah (*low-cost*) berukuran sebesar kartu kredit yang dihubungkan ke monitor komputer ataupun TV, dan menggunakan *keyboard* ataupun *mouse*.

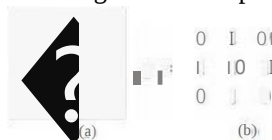
Raspberry Pi Foundation menawarkan 2 versi produk, yaitu model A dan B Raspberry Pi Foundation juga menyediakan distro Linux untuk di download, serta aplikasi pendukung untuk pemrograman yaitu Python, Perl dan BBC untuk BASIC. [1]

### 2.2 Morfologi citra

Kata dari *morphology* umumnya menunjukan cabang dari ilmu biologi yang berhubungan dengan bentuk dan struktur tumbuhan dan hewan. Didalam pengolahan citra istilah dari *mathematic morphology* digunakan sebagai *tools* untuk mengekstraksi komponen citra yang berguna untuk deskripsi dan representasi dari region shape seperti *boundaries*, *skeletons* dan *convex hull*. Proses *morphology* juga berguna untuk *pre* atau *post-processing* seperti *thinning*, *filtering*, *pruning*.

#### 2.2.1 Structuring element

Structuring element (strel) dapat diibaratkan dengan mask pada pemrosesan citra biasa (bukan secara morfologi). Strel juga memiliki titik poros (disebut juga titik origin). Titik origin ditandai dengan tanda titik hitam. Jika tidak ada tanda titik hitam maka diasumsikan titik origin berada di pusat simetri [7].



Gambar 1 Contoh strel (a) titik “O” adalah titik poros, (b) representasi biner strel

#### 2.2.2 Operasi-operasi Morfologi

Dalam morfologi ada beberapa operasi yang dapat dilakukan, yaitu :

##### 1. Dilasi

Dilasi didefinisikan sebagai proses “penumbuhan” atau “penebalan” objek citra biner. Misalkan A dan B adalah himpunan-himpunan piksel. Dilasi A oleh B dinotasikan dengan  $A \oplus B$  dan didefinisikan dengan:

$$A \oplus B = \bigcup_{(u,v) \in B} A \oplus (u,v) \quad (2.2)$$

Ini berarti bahwa untuk setiap titik pada A dilakukan translasi atau pergeseran pada arah  $(u, v)$  dan kemudian menggabungkan seluruh hasil pergeseran (union)

##### 2. Erosi

Erosi merupakan proses mengecilkan atau menipiskan objek citra biner. Misalkan A dan B adalah himpunan-himpunan piksel. Erosi A oleh B dinotasikan dengan  $A \ominus B$  dan didefinisikan dengan

$$\ominus B = \{w: \ominus \subseteq \ominus\} \quad (2.2)$$

### 3. Closing

Operasi closing adalah kombinasi antara operasi dilasi dan erosi yang dilakukan secara berurutan. Citra asli didilasi terlebih dahulu, kemudian hasilnya dierosi. Proses closing pada sebuah citra A oleh strel B dinotasikan dengan  $A \bullet B$  dan didefinisikan sebagai :

$$A \bullet B = (A \oplus B) \ominus B \quad (2.3)$$

## PERANCANGAN DAN IMPLEMENTASI SISTEM

### 3.1 Pemodelan Sistem

Pada bab ini akan dipaparkan mengenai langkah-langkah yang akan dilakukan selama proses pengerjaan program. Langkah pertama yang dilakukan dalam merancang sistem ini adalah membuat template karakter A-Z dan 0-9. Pembuatan template ini sangat penting diperhatikan karena mengingat metode template matching adalah hanya mencari nilai korelasi piksel sehingga template yang dibuat harus dibuat semirip mungkin dengan karakter aslinya. dalam Tugas Akhir ini digunakan 74 template yang berasal dari 36 font karakter dan 38 template yang dibuat dari plat nomor aslinya dimana 2 tambahan adalah karakter 'D' dan '0' dengan garis diagonal.



Gambar 2 Pemodelan sistem

#### 3.1.1 Akuisisi Citra

Akuisisi citra merupakan tahap awal dalam sistem ini untuk mendapatkan citra digital. Citra mobil di capture melalui handphone smartphone Samsung Galaxy S5 dengan resolusi 16 megapiksel dengan spesifikasi :

- Jarak pengambilan *sample* : 75-100 cm
- Tinggi kamera :  $\pm 40$  cm
- Ukuran citra : 960x540 pixel



Gambar 3 Hasil akuisi citra uji

- Ukuran citra : 960x540 = 518400 piksel
- Luas plat nomor(cm) : 11.8cm x 3.6cm = 42.87  $\text{cm}^2$
- Luas citra(cm): 15.4 cm x 27.3 cm = 420.42  $\text{cm}^2$

Sehingga perhitungan berdasarkan rumus 3.1 adalah menjadi

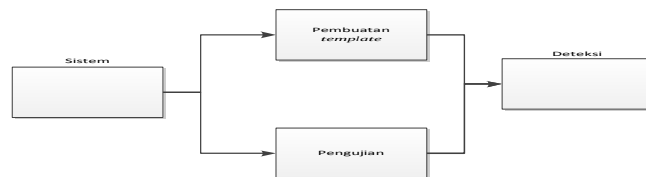
- luas plat nomor =  $518400 \times \frac{42.87 \text{ cm}^2}{420.42 \text{ cm}^2}$
- luas plat nomor =  $518400 \times \frac{1}{10}$
- luas plat nomor = 51840 piksel

Dari hasil diatas didapatkan perkiraan luas plat nomor didalam penelitian ini adalah sekitar 51.840 piksel dan mempunyai perbandingan 1: 10 dengan luas piksel citra.

Karena ukuran dari plat nomor tidak selalu sama maka untuk membuat rentang filter luas area plat nomor peneliti mengurangi dan menambahkan nilai untuk membuat rentang koreksi yaitu 20.000 hingga 75.000

### 3.2 Diagram blok sistem

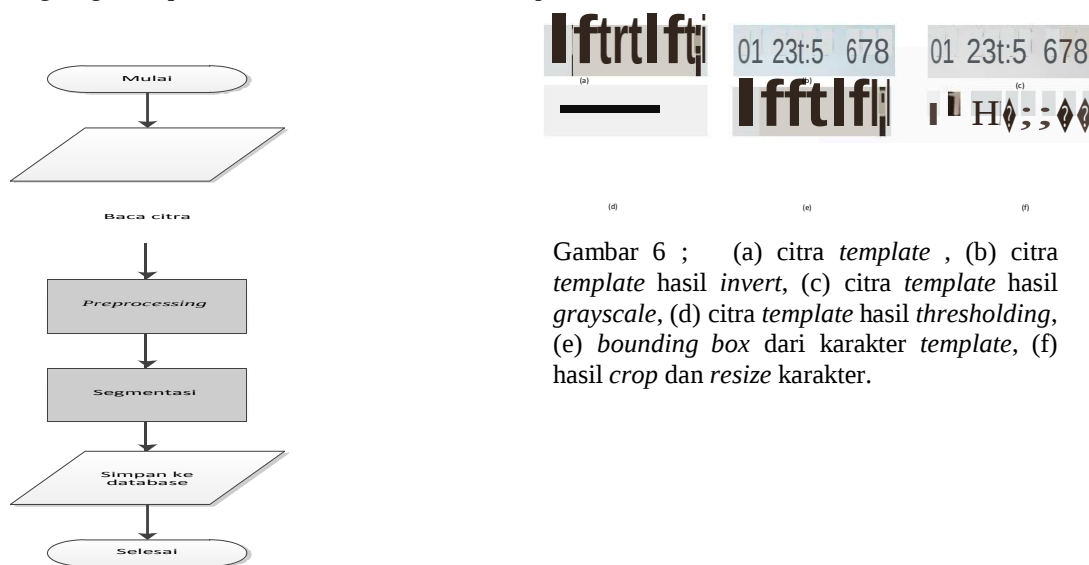
Proses identifikasi dibagi menjadi dua proses, yaitu proses pembuatan template dan proses uji. Proses pembuatan template yaitu proses menyimpan template karakter yang menjadi acuan untuk database program, dimana citra tersebut yang akan dicocokkan dengan citra uji untuk mendeteksi plat nomor. Tahapan proses identifikasi untuk proses pembuatan template dan tahap uji dapat digambarkan sebagai berikut.



Gambar 4 Diagram alir identifikasi

#### 3.2.1 Pembuatan template karakter

Citra hasil akuisisi akan langsung dilakukan proses *preprocessing* setelah itu dilakukan proses segmentasi yang bertujuan untuk memotong citra plat nomor utuh menjadi citra per karakter. Hasil dari proses segmentasi akan langsung disimpan ke dalam *database home/user/pi*



Gambar 6 ; (a) citra *template* , (b) citra *template* hasil *invert*, (c) citra *template* hasil *grayscale*, (d) citra *template* hasil *thresholding*, (e) *bounding box* dari karakter *template*, (f) hasil *crop* dan *resize* karakter.

Gambar 5 Dialgram alir pembuatan template

Pada proses *preprocessing* ini citra pertama akan melalui proses inversi yang bertujuan untuk menukar *foreground* dan *background* dimana '1' yaitu piksel putih akan menjadi bagian *pattern* dan piksel '0' yaitu hitam akan menjadi bagian dari *background*.

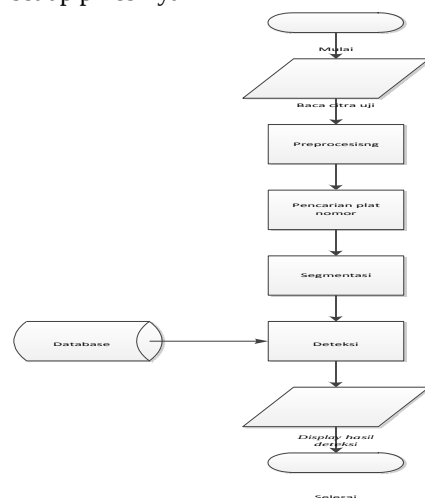
Untuk pencarian kemudian citra diubah kedalam derajat keabuan (*grayscale*) setelah itu dilakukan proses *thresholding* dengan metode *adaptive threshold*. Yang terakhir adalah mengubah ukuran karakter menjadi 79x157 piksel seragam untuk ke 74 citra *template*.

Pada proses ini citra plat nomor mobil ini akan dilakukan proses segmentasi karakter dari citra plat nomor yang peneliti buat untuk di segmentasi menjadi citra per karakter yang nantinya akan di simpan di *database*. Setelah citra dibaca maka akan dilakukan proses *preprocessing* untuk mempermudah pencarian kontur karakter.

Akan dilakukan validasi dari sisi tinggi karakter, karena plat nomor yang digunakan sebagai *template* dikhawatirkan masih ada noise sisa cat yang kurang sempurna maka validasi tinggi yang dilakukan adalah harus lebih dari 20 piksel dan luas nya lebih dari 50 piksel. Setelah proses ini citra plat nomor tersebut akan tersegmentasi menjadi citra sejumlah karakter plat nomor yang disimpan di *database* yang terdapat pada *root/home/pi*.

### 3.2.2 Diagram alir pengujian sistem

*Preprocessing* dalam citra uji sedikit berbeda karena didalam pengujian citra yang sudah menjadi *grayscale* akan dilakukan *blurring*, proses ini menggunakan kernel Gaussian untuk membuat citra tampak lebih *blur*, efek ini akan meminimalisir *noise* yang tidak diinginkan untuk proses selanjutnya. Kernel yang digunakan untuk Gaussian *blur* ini adalah 5x5, setelah itu dilakukan deteksi tepi sobel hanya untuk arah-x / vertikal agar garis vertikal dari semua karakter terlihat lebih dominan dan mempermudah untuk membentuk kontur tertutup. Yang terakhir proses binerisasi yang digunakan pada proses pengujian ini adalah *Otsu method* dimana proses ini mencari nilai threshold optimal pada setiap pikselnya



Gambar 7 Diagram alir pengujian sistem

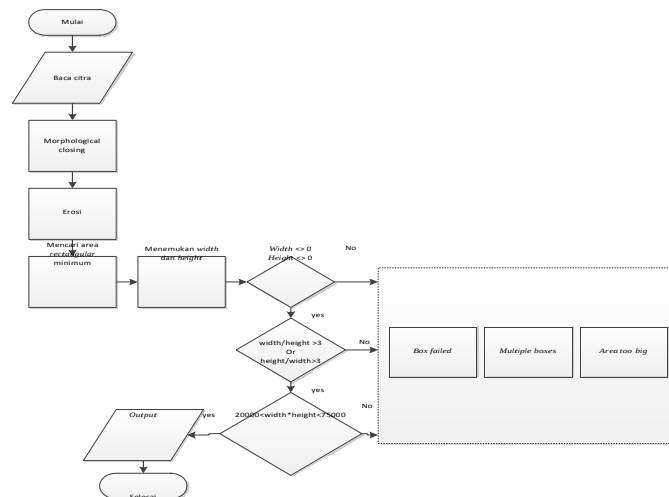


Gambar 8 (a) citra uji (b) citra yang melalui crop bagian atas (c) citra setelah Gaussian blur 5x5 (d) citra setelah vertical edge detection (e) citra setelah otsu threshold

#### 3.2.2.1 Sub flowchart pencarian plat nomor

Setelah citra mengalami preprocessing, citra sudah menjadi citra biner dan siap untuk di manipulasi dengan *morphological operation*. Didalam penelitian ini digunakan *morphological closing* dan dilasi. Pemotongan citra bagian atas sudah melalui beberapa percobaan sehingga tidak ada plat nomor yang terpotong. Pemotongan ini berguna untuk mengurangi beban prosesing citra dan memperkecil kemungkinan untuk mencari ROI (*region of interest*). Perbedaan sebelum dan sesudah pemotongan dapat dilihat pada gambar 3.10(a) dan 3.10(b). ada beberapa filter lagi yang dilakukan sebelum pencarian kontur plat nomor dilakukan yaitu filter Aspect ratio dimana hasil bagi dari lebar dan tinggi plat yang relative terhadap ukuran piksel harus lebih dari tiga begitu juga sebaliknya hasil bagi dari tinggi dan lebar harus lebih dari tiga

Selanjutnya yang terakhir adalah filter luas dimana luas dibawah 20000 piksel dan diatas 75000 akan dianggap sebagai bukan plat nomor.



Gambar 9 Sub diagram alir pencarian plat nomor

### 3.3 Template matching

Metode *template matching* akan digunakan untuk pendeteksian karakter, hasil dari pembuatan template yaitu sebanyak 74 template akan dibandingkan dengan hasil segmentasi dari data uji dan yang dianggap mempunyai kemiripan yang paling besar akan dianggap sebagai output yang akan ditampilkan saat *running* program dan juga akan langsung di simpan ke dalam *file .txt*. Keuntungan dari plat nomor untuk format Negara Indonesia adalah plat nomor mempunyai format yang konstan dan dibagi kedalam tiga bagian kelompok. Ketiga kelompok tersebut yaitu :

1. Kelompok pertama : 1 – 2 huruf
2. Kelompok kedua : 1 – 4 angka
3. Kelompok ketiga : 1 – 3 huruf

Sehingga berdasarkan penelitian, jumlah kombinasi plat yang mungkin terjadi adalah 21 kombinasi. Untuk meningkatkan akurasi dan menghindari ambiguitas dari hasil *template matching* terhadap karakter 0 dengan D, 1 dengan I, 8,3 dengan B, 7 dengan Z maka peneliti membuat algoritma untuk membedakan jenis plat nomor dengan yang lainnya dengan jarak per karakter. Diharapkan tidak akan ada ambiguitas korelasi antara huruf dan angka, sehingga kelompok pertama hanya dimatchingkan dengan template huruf, kelompok kedua dengan angka, kelompok ketiga dengan huruf.



Gambar 10 Jarak antar karakter

Untuk contoh diatas jarak antara karakter “D” dan “1” dapat dihitung dengan selisih dari  $(x_2 + w_1)$  dan  $(x_1, x_2)$ , sehingga rumus untuk mencari jarak antar karkter adalah

$$J = x_2 - (x_1 + w_1) \quad (3.2)$$

Dimana :

$J$  = jarak antar karakter

$x_2$  = koordinat x pada karakter n

$x_1$  = koordinat x pada karakter pertama

$w_1$  = lebar dari karakter pertama

Proses selanjutnya adalah menentukan *threshold* hasil selisih. Pada penelitian ini digunakan *threshold* 24 yang artinya jika hasil selisih lebih dari 24 maka itu artinya sudah berbeda kelompok. Dan jika hasil selisih kurang dari 24 maka itu adalah dari kelompok yang sama.

#### 4 PENGUJIAN SISTEM DAN ANALISIS

Untuk mengetahui performansi sistem yang telah dirancang, maka perlu dilakukan pengujian terhadap sistem yang telah dikembangkan. Dalam pengujian akan diukur seberapa besar tingkat keberhasilan sistem yang dirancang dengan melakukan analisis terhadap beberapa parameter.

##### 4.1 Mengetahui *structur element* (SE) yang tepat

Tujuan dari pengujian ini adalah menentukan SE yang tepat untuk proses *morphological closing* yang berguna untuk menyatukan karakter didalam plat nomor. Peneliti menguji tiga SE dalam bentuk matrix horizontal dalam ukuran yang berbeda yaitu 86, 96 dan 106. Selanjutnya peneliti menganalisis ketiga ukuran SE tersebut kedalam tiga kategori yaitu; *not connected*, *multiplebox*, dan *area too big*. hasil dari pemetaan gambar dapat dilihat pada gambar 11

| <br>citra closing<br>multiple box   | <br>multiple box   | <table><tr><th>SE</th><th>NOT CONNECTED</th></tr><tr><td>86</td><td>4</td></tr><tr><td>96</td><td>2</td></tr><tr><td>106</td><td>0</td></tr></table> | SE | NOT CONNECTED | 86 | 4 | 96 | 2 | 106 | 0 | <table><tr><th>MULTIPLE BOX</th><th>AREA TOO BIG</th></tr><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>9</td></tr><tr><td>2</td><td>14</td></tr></table> | MULTIPLE BOX | AREA TOO BIG | 1 | 10 | 2 | 9 | 2 | 14 |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----|---------------|----|---|----|---|-----|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------|---|----|---|---|---|----|
| SE                                                                                                                   | NOT CONNECTED                                                                                       |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 86                                                                                                                   | 4                                                                                                   |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 96                                                                                                                   | 2                                                                                                   |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 106                                                                                                                  | 0                                                                                                   |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| MULTIPLE BOX                                                                                                         | AREA TOO BIG                                                                                        |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 1                                                                                                                    | 10                                                                                                  |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 2                                                                                                                    | 9                                                                                                   |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| 2                                                                                                                    | 14                                                                                                  |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| <br>too big                         | <br>area too big   |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |
| <br>citra closing not<br>connected | <br>not connected |                                                                                                                                                      |    |               |    |   |    |   |     |   |                                                                                                                                                             |              |              |   |    |   |   |   |    |

Tabel 1 uji coba SE pada tiga parameter

Tabel 1 uji coba SE pada tiga parameter

Gambar 11 Tiga kategori SE analisis cloisng  
abel 1 uji coba SE pada tiga parameter

##### 4.2 Mengetahui kernel erosi yang tepat

Dari 45 percobaan data uji, hasil yang didapatkan pada pencarian ukuran SE, disimpulkan ukuran yang tepat ialah 106. Namun pada SE 106 masih ditemukan beberapa kesalahan yaitu *multiplebox* dan *area too big*. peneliti mencoba menggunakan 4 kernel matriks berukuran axb yaitu 4x1, 4x2, 4x3, dan 4x4, dai percobaan dengan 4 kernel erosi didapat kernel 4x4 yang memiliki akurasi tertinggi. Hasil dari analisis uji erosi tersebut dapat dilihat pada gambar tabel 2

Tabel 2 Uji coba kernel erosi terhadap SE 106

|           | SUCCESS | FAILED |
|-----------|---------|--------|
| EROSI 4X1 | 39      | 6      |
| EROSI 4X2 | 41      | 4      |
| EROSI 4X3 | 42      | 3      |
| EROSI 4X4 | 44      | 1      |

##### 4.3 Pengujian akurasi dengan menggunakan metode *normalized cross correlation*

Setelah melalui proses *preprocessing* data uji akan dibandingkan dengan data *template* yang ada di dalam *database*, dengan menggunakan metode *normalized cross correlation*. Hasil pengujian akurasi sistem dengan metode *normalized cross correlation* ditunjukkan pada tabel berikut :

Tabel 3 Akurasi dengan metode normalized cross correlation

|                     | JUMLAH BENAR | AKURASI |
|---------------------|--------------|---------|
| PENCARIAN BOX       | 44/45        | 99%     |
| SEGMENTASI KARAKTER | 43/45        | 95%     |
| OCR                 | 40,95/45     | 91%     |

#### 4.3.1 Pengujian akurasi sistem dengan metode *normalized correlation coefficient*

Setelah melalui proses *preprocessing* data uji akan dibandingkan dengan data *template* yang ada di dalam *database*. Hasil pengujian akurasi sistem dengan metode *normalized cross correlation* ditunjukkan pada tabel berikut :

Tabel 4 Akurasi dengan metode normalized correlation coefficient

|                     | JUMLAH   | AKURASI |
|---------------------|----------|---------|
| PENCARIAN BOX       | 44/45    | 99%     |
| SEGMENTASI KARAKTER | 43/45    | 95%     |
| OCR                 | 42,58/45 | 94,02%  |

#### 4.3.2 Pengujian akurasi dengan citra yang diberi noise salt and pepper 0.02 dan 0.04

Citra uji yang semula normal di tambahkan *noise salt & pepper* dengan standar deviasi 0.02 dan 0.04. . Penambahan *noise* pada citra uji ini dimaksudkan untuk melihat ketahanan sistem terhadap *noise salt & pepper* yang biasa dihasilkan oleh malfungsi elemen piksel di sensor kamera, kekeliruan didalam alokasi memori, kesalahan didalam proses digitalisasi. Hasil pengujian sistem dapat dilihat dalam tabel dan grafik dibawah :

Tabel 5 Citra uji dengan STD noise salt &amp; pepper 0.02

| STD = 0.02          | JUMLAH   | AKURASI |
|---------------------|----------|---------|
| PENCARIAN BOX       | 43/45    | 96%     |
| SEGMENTASI KARAKTER | 41/45    | 91.11%  |
| OCR                 | 40.44/45 | 89.97%  |

Tabel 6 Citra uji dengan STD noise salt &amp; pepper 0.04

| STD = 0,04          | JUMLAH   | AKURASI |
|---------------------|----------|---------|
| PENCARIAN BOX       | 42/45    | 93%     |
| SEGMENTASI KARAKTER | 39/45    | 86.6%   |
| OCR                 | 38.02/45 | 84.49%  |

#### 4.3.1 Pengujian akurasi dan waktu komputasi berdasarkan dimensi piksel *template*

Waktu komputasi juga dipengaruhi karena kesamaan banyaknya jumlah piksel yang dibandingkan. Seluruh ukuran citra, untuk citra *template* ukuran piksel yaitu 79 x 157 dan citra uji memiliki ukuran yaitu 960 x 540 piksel. Semakin besar ukuran citra maka semakin banyak pula nilai piksel yang dibandingkan. Hal ini akan mempengaruhi lamanya waktu komputasi sistem. Berdasarkan pengujian yang telah dilakukan dengan citra uji dengan ukuran piksel *template* tertentu didapatkan hasil sebagai berikut:

Tabel 7 Dimensi piksel *template* terhadap akurasi dan waktu komputasi

| Dimensi piksel <i>template</i> | Akurasi | Waktu komputasi |
|--------------------------------|---------|-----------------|
| 960x540                        | 94.31%  | 25.6 detik      |
| 20x40                          | 94.60%  | 9.6 detik       |

## 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Implementasi metode sobel edge dan *template matching* mampu mendeteksi dan mengenali karakter plat nomor

2. Tingkat akurasi tertinggi sebesar 94.9% dicapai dengan metode *template matching normalized correlation coefficient* dengan ukuran *template* 20x40 piksel dengan waktu komputasi 5.6 detik
3. Tingkat akurasi hanya tercapai sebesar 91% pada metode *template matching normalized cross correlation*
4. Tingkat akurasi hanya tercapai sebesar 89.97% pada citra dengan salt and pepper noise STD = 0.02 dan akurasi sebesar 84.47% pada citra dengan salt and pepper noise STD = 0.04
5. Algoritma pembagian jarak mampu mengatasi korelasi yang ambigu antara huruf dan angka
6. Konsistensi pengambilan citra uji sangat penting untuk mempermudah proses pencarian plat nomor.
7. Kernel erosi 4x4 mampu untuk mengatasi sebagian besar kesalahan *area too big* dan *multiple box*
8. Raspberry pi kurang cocok untuk implementasi sistem ANPR dengan metode *template matching* karena memory nya yang kecil dan menghasilkan waktu komputasi yang masih relative lama.

## 5.2 Saran

1. Dibutuhkan algoritma pencarian plat nomor mobil yang lebih bagus untuk citra uji mobil dengan struktur *bumper* yang kompleks seperti pada citra-uji *cobacoba41.jpg*
2. Dibutuhkan penelitian lebih lanjut untuk segmentasi karakter plat nomor yang terlalu tipis
3. Dibutuhkan penelitian yang lebih lanjut mengenai *storing* data *template* dengan format seperti *.txt* agar waktu komputasi lebih cepat

## DAFTAR PUSTAKA

- [1] Raspberrypi.org. <http://www.Raspberrypi.org/help/what-is-raspberrypi/>. 24 April: 12.45 WIB
- [2] Munir,Renaldi. Agustus 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika Bandung
- [3] Sigit,Riyanto dkk. Maret 2005. Step by Step Pengolahan Citra Digital. Yogyakarta: C.V. ANDI OFFSET (Penerbit Andi)
- [4] Fauzi,Hadidh M,Tjandrasa Handayani. Implementasi Thresholding Citra menggunakan Algoritma Hybrid Optimal Estimation. Surabaya; Institut Teknologi Sepuluh November, Surabaya
- [5] Noor Santi, Candra. 2001. Mengubah Citra Berwarna menjadi Gray-Scale dan Citra biner. Semarang: Universitas Stikubank Semarang.
- [6] Lecturer. eepis. Institut Teknologi Surabaya. <http://lecturer.eepis-its.edu/~riyanto/citra-bab8.pdf>. 22 April: 12.35 WIB
- [7] R.C Gonzales and R. E. Wood,Digital Image Processing (second edition), Prentice Hall. New Jersey.2002
- [8] Ozbay, Serkan and Ercelebi, Ergun. Automatic Vehicle Identification by Plate Recognition. World Academy of Science, Engineering and Technology. International Journal of Electrical, Electronic Science and Engineering Vol:1 No.9.2007
- [9] Lecture 4 Thresholding, Bryan S. Morse, Brigham Young University, 2000.[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/threshold.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf) 13 Desember : 09.12 AM
- [10] Opencv 3.0.0-dev documentation, Image Thresholding. [http://docs.opencv.org/trunk/doc/py\\_tutorials/py\\_imgproc/py\\_thresholding/py\\_thresholding.html](http://docs.opencv.org/trunk/doc/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html). 13 Desember : 10.12 AM
- [11] Intel Corporation All Rights Reserved Copyright © 1999-2001, Open Source Computer Vision Library Reference Manual, U.S.A.Order Number: 123456-001
- [12] Nanang Trisnadik, Achmad Hidayatno dan R. Rizal Isnanto. Pendeteksian Posisi Plat Nomor Kendaraan Menggunakan Metode Morfologi Matematika. Jurusan Teknik Elektro, Universitas Diponegoro Semarang.
- [13] Gary bradski dan Adrian Kaehler, Copyright 2008, Learning OpenCV Computer Vision With The OpenCV Library, O'REILLY
- [14] Angga, Kurnia, Sasotya. *Metode Sobel Edge dan Template Matching Dalam Pengenalan Plat Nomor Otomatis*. Teknik Informatika, Universitas Telkom

