

Pendeteksian Plat Nomor Kendaraan Menggunakan Metode *Template matching*

Automatic Number Plate Recognition Using Template Matching Method

Ivany Sarief^{1,*}, Harfin Yusuf Biu², Sepryan Ismail Chandra²

¹Program Studi Teknik Elektro, Universitas Sangga Buana

Jl. PH.H. Mustofa No.68, Cikutra, Cibeunying Kidul, Kota Bandung, Jawa Barat 40124

²Program Studi Teknik Elektro, Universitas Komputer Indonesia

Jl. Dipati Ukur No.112-116, Lebakgede, Cobleng, Kota Bandung, Jawa Barat 40132

*Email : ivany.sarief@usbypkp.ac.id

Abstrak - Untuk merancang sebuah sistem agar dapat mengidentifikasi suatu objek plat nomor untuk format Indonesia, maka di rancang sebuah sistem awal yaitu berupa suatu aplikasi pendeteksi plat nomor kendaraan dengan menggunakan metode *template matching*. Tujuan dari pembuatan aplikasi ini yaitu untuk diimplementasikan pada sistem parkir otomatis yang berbasis pada pendeteksian plat nomor kendaraan. Sistem ini menggunakan kamera untuk proses pengambilan gambarnya. Dengan memanfaatkan teknologi pengolahan citra dengan metoda pengenalan *template matching correlation* akan dihasilkan suatu nilai *string* dari gambar tersebut. Sebelum dilakukan proses pengenalan, terlebih dahulu dilakukan tahap *pre-processing* terhadap citra input yang meliputi *grayscale*, binerisasi, sampai tahap segmentasi sebelum dilakukan proses kolerasi/perbandingan terhadap citra *template*. Proses yang berlangsung dalam bagian *pre-processing* dilakukan untuk membuat citra yang akan diproses menjadi lebih ringan dan sederhana. Proses tersebut dapat membuat gambar lebih mudah untuk diproses dan juga kecepatan proses dari sistem akan meningkat. Sebelum mengaplikasikan algoritma *template matching* pada citra target hasil segmentasi, ukuran citra terlebih dahulu disesuaikan dengan ukuran citra *template* dalam basis data. Hal ini dilakukan agar citra target dan citra dalam *template* dapat dibandingkan secara langsung melalui algoritma *template matching*. Keluaran dari sistem yang dibuat adalah suatu nilai *string* yang merupakan nomor dari plat kendaraan yang dibaca melalui kamera. Permasalahan yang timbul dalam proses pengenalan ini adalah bagaimana cara untuk mengidentifikasi berbagai jenis karakter dengan berbagai ukuran dan bentuk sehingga di hasilkan nilai *string* yang sama dengan gambar *text*. Rata-rata tingkat keberhasilan aplikasi ini yaitu sebesar 70% sehingga masih perlu dilakukan penelitian lebih lanjut.

Kata kunci : Pengolahan Citra, *Template Matching*, Kamera, Plat Nomor, Matlab

Abstract - To design a system in order to identify an object number plate for the Indonesian format, an initial system is designed, in the form of a vehicle licence plate recognition application using template matching method. The goal of this application is to be implemented to the parking system by identifying the number plate. This system uses the camera for the image capture process, by utilizing image processing technology with the matching correlation template method for recognition to produce a string value from the image. Before doing recognition process, First, the pre processing stage is performed on the input image which includes grayscale, binary, until the segmentation stage before the correlation / comparison process is carried out on the image of Template. The process that occur in the pre-processing unit done for some reason including to make the image lighter and less complex. This process will make the image easier to be processed and also to increase the proses speed of the system. Before applying the template matching algorithm to the image output from segmentation process, first the image has to be resized to match the size of the template image stored in the data base. This has to done so that the target image and the template image can be compared directly with template matching algorithm. The output of this system is a string value which is represent the value of the license plate captured by the camera used by the system. The problem that arises in the introduction process is how to identify various types of characters with various sizes and shapes so that the string value is the same as the text image. The average success rate of this application is 70% so that further research must be carried out.

Keywords : Image Processing, *Template matching*, Camera, Number Plate, Matlab

I. PENDAHULUAN

A. Latar Belakang

Pada saat ini sistem parkir yang banyak diteliti dan dikembangkan adalah sistem informasi lahan parkir yang bertujuan untuk memberi kemudahan kepada pengguna lahan parkir untuk menemukan lahan yang kosong dan juga untuk memberi kemudahan kepada pengguna untuk menemukan lokasi di mana mereka memarkirkan kendaraan[1] [2]. Namun penelitian dan pengembangan tentang proses otomatisasi yang menjamin keamanan lahan parkir masih jarang dilakukan. Sistem lahan parkir yang ada saat ini masih membutuhkan tenaga petugas untuk kepentingan keamanan. Petugas pada lahan parkir memiliki tugas untuk membandingkan nomor plat pada kendaraan dengan nomor plat pada STNK serta waktu masuk. Untuk menerapkan proses otomatisasi pada lahan parkir, diperlukan suatu sistem yang dapat menggantikan fungsi dari petugas tersebut. Salah satu metode yang paling menjamin yang dapat diterapkan adalah teknologi pengolahan citra.

Pengolahan Citra adalah suatu proses analisa dan manipulasi pada data atau informasi digital dari suatu gambar yang banyak melibatkan persepsi visual. Citra sendiri diartikan sebagai representasi bentuk 2 dimensi dari bentuk fisik 3 dimensi nyata dari suatu objek. Pengolahan citra dilakukan untuk berbagai tujuan, misalnya untuk memperbaiki kualitas gambar, mengambil informasi tertentu dari suatu gambar, dan sebagainya. Salah satu teknik pengolahan citra yang banyak digunakan adalah *template matching*.

Teknik *template matching* dilakukan dengan mencari bagian-bagian kecil pada gambar yang sesuai dengan *template* gambar referensi pada basis data. Pengaplikasian pengolahan citra saat ini juga sedang marak dikembangkan pada teknologi di berbagai bidang ilmu pengetahuan.

B. State of Art

Salah satu penelitian yang dilakukan pada bidang medis tentang pengaplikasian teknik *template matching* adalah "*Breast Cancer Detection Based on Dynamic Template Matching*"[3]. Pada penelitian ini, tumor yang dideteksi tersebut berbentuk *circular* atau *semicircular*. *Template* pada penelitian tersebut juga didefinisikan berdasarkan bentuk dan kecerahan citra dari tumor. Citra tumor tersebut kemudian dibandingkan dengan citra dari organ

yang ingin didiagnosa untuk mencari tahu keberadaan tumor pada organ tersebut. Selain dalam bidang medis, pengolahan citra juga sudah dikembangkan pada sistem keamanan yang berbasis pada pendeteksian wajah. Pendeteksian wajah saat ini banyak diaplikasikan pada sistem keamanan, misalnya sistem keamanan perangkat elektronik yang menggunakan identifikasi wajah untuk mengaksesnya[4], sistem kewanitaan rumah atau gedung dan sebagainya. Identifikasi wajah saat ini juga sudah banyak diaplikasikan pada sistem absensi menggantikan sistem absensi manual dan juga sistem absensi dengan sidik jari[6] pada perusahaan yang memprioritaskan keamanan. Selain pada bidang medis dan sistem keamanan, teknologi pengolahan citra juga sudah diaplikasikan pada sistem lalu lintas salah satunya untuk mendeteksi kepadatan lalu lintas pada persimpangan jalan[7] juga mendeteksi pelanggaran dalam berlalu lintas[8]. Sistem pendeteksian nomor plat kendaraan sudah dibuat diberbagai Negara namun dengan format yang berbeda, karena setiap Negara memiliki format plat nomor kendaraan masing-masing, misalnya saja sistem pendeteksian nomor plat kendaraan dengan format Mesir yang dibuat oleh Khalil[9]. Ada juga sistem pendeteksi plat nomor kendaraan dengan menggunakan pengolahan citra dengan jaringan saraf tiruan yang telah dibuat oleh Ibrahim dan Kirami[10]. Akurasi dari sistem ini bisa dikatakan sangat tinggi namun membutuhkan proses training dan basis data yang besar sehingga tidak cocok diimplementasikan pada sistem parkir sederhana yang banyak ditemui di Indonesia. Penelitian yang serupa juga telah dilakukan oleh Saqib Rasheed yang mendeteksi plat nomor kendaraan untuk format Islamabad. Penelitian tersebut menambahkan algoritma *hough transform* pada sistemnya[11].

C. Tujuan

Jika pada bidang medis pengolahan citra dimanfaatkan untuk mendeteksi penyakit [3] dan pada bidang keamanan digunakan untuk mendeteksi wajah [4-6], penulis dalam paper ini memanfaatkan pengolahan citra untuk mendeteksi nomor plat kendaraan dalam format Indonesia dengan metode *template matching* yang dapat diimplementasikan pada sistem parkir kendaraan.

Hasil dari pendeteksian nomor plat kendaraan ini dapat dimasukkan ke dalam data kendaraan yang terparkir pada lahan parkir tersebut. Saat kendaraan keluar dari lahan parkir, sistem akan mendeteksi plat nomor kendaraan tersebut dan mengeluarkan data kendaraan tersebut dari daftar

kendaraan yang sedang terparkir.

Dengan menambahkan sistem lain seperti sistem pemindai Surat Tanda Nomor Kendaraan (STNK), sistem plang otomatis, dan sistem pembayaran otomatis, sistem lahan parkir tersebut dapat diotomatisasi. Sistem pemindai STNK ini diperlukan untuk membandingkan nomor plat kendaraan dengan nomor plat pada STNK untuk mengkonfirmasi kepemilikan kendaraan. Dengan adanya sistem ini pemilik kendaraan diwajibkan untuk membawa STNK setiap kali memasuki lahan parkir. Dengan adanya sistem ini diharapkan upaya-upaya pencurian kendaraan pada lahan parkir dapat dihindari. Karena proses masuk dan keluar dari lahan parkir dilakukan secara otomatis menggunakan komputer, prosesnya pun akan berlangsung lebih cepat.

II. METODE

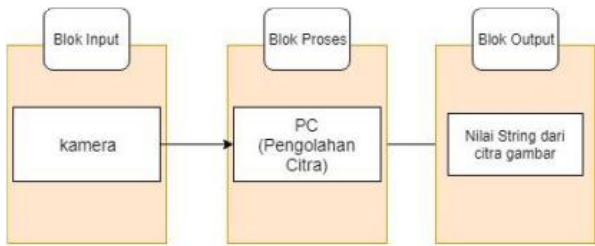
Terdapat dua metode utama yang digunakan pada sistem pendeteksian nomor plat kendaraan ini. Metode-metode tersebut yaitu pre-processing citra digital dan template matching. Pre-processing citra digital terjadi setelah sistem mengambil gambar pada bagian awal dengan menggunakan kamera. Proses pendeteksian diawali dengan proses pengambilan citra dari plat kendaraan yang akan masuk atau keluar dari lahan parkir. Pengambilan citra dilakukan dengan menggunakan kamera yang diposisikan pada area masuk dan keluar lahan parkir. Kamera diposisikan sedemikian rupa sehingga citra yang diambil mengandung semua bagian dari plat nomor kendaraan. Kamera yang digunakan harus beresolusi tinggi agar citra yang ditangkap juga beresolusi tinggi sehingga informasi digital citra tersebut lebih akurat. Setelah citra dari kendaraan yang mengandung plat nomor diambil dan dimasukkan pada suatu folder tertentu yang akan diproses menggunakan *software* matlab, proses selanjutnya adalah meng-*crop* citra dari plat nomor kendaraan tersebut. Proses *cropping* dilakukan untuk mengambil bagian penting dari plat nomor kendaraan tersebut. Bagian-bagian tersebut adalah huruf awal, angka, dan huruf akhir dari plat nomor kendaraan. Proses *cropping* ini juga dilakukan dengan tujuan untuk mengurangi *noise* pada citra yang akan diproses. Setelah proses *cropping* selesai, selanjutnya dilakukan tahapan *preprocessing* yang terdiri dari tiga tahapan yaitu *grayscale*, binerisasi, dan segmentasi. Pada tahap *grayscale* citra yang telah di-*crop* dikonversi kedalam citra *grayscale*. Konversi dari RGB ke dalam *grayscale* memiliki

banyak keuntungan, diantaranya memudahkan program melakukan deteksi sudut untuk memisahkan satu karakter dengan karakter lain, mengurangi kompleksitas dari kode yang dibuat karena besarnya data dari citra RGB 3 kali besar dari data citra *grayscale* sehingga citra *grayscale* lebih sederhana untuk diproses, dan dengan alasan yang sama pula kecepatan proses program lebih cepat dengan citra *grayscale*. Setelah tahap *grayscale*, tahap selanjutnya adalah binerisasi citra. Binerisasi citra dilakukan untuk mengubah data citra ke dalam data yang lebih sederhana lagi sehingga mudah diamati. Data digital dalam suatu citra biner hanya terdiri dari dua nilai saja yaitu 0 dan 1. Tahapan selanjutnya adalah segmentasi yang dilakukan untuk memisahkan karakter satu dengan karakter yang lain. Hasil dari tahapan segmentasi ini adalah beberapa citra yang telah diambil dari citra keseluruhan plat nomor kendaraan. Citra-citra yang merupakan hasil dari proses segmentasi akan di-*resize* sebelum diproses dengan algoritma *template matching*. Ukuran citra dari proses *resize* disesuaikan dengan ukuran citra *template* pada basis data agar nilai-nilai *pixel* pada kedua citra dapat dibandingkan. Proses akhir dari pendeteksian nomor pada plat kendaraan adalah proses *template matching*. Proses *template matching* dilakukan dengan membandingkan nilai *pixel* antara citra yang akan diuji dengan nilai *pixel* dari citra *template* pada basis data. Nilai *pixel* yang dibandingkan adalah nilai *pixel* pada koordinat citra yang sama. Tujuan dari proses *template matching* ini adalah untuk mencari karakter dalam basis data *template* yang bersesuaian dengan karakter pada citra yang sementara diproses.

A. Diagram Blok Sistem

Sistem yang dibuat pada penelitian ini menggunakan teknik pengolahan citra *Optical Character Region* (OCR). Pada **Gambar 1** diperlihatkan diagram blok dari sistem pendeteksian nomor plat kendaraan. Sistem initerdiri dari 3 blok bagian. Blok pertama merupakan blok input yang menggunakan kamera sebagai sensor yang bertugas sebagai alat untuk menangkap gambar dari plat nomor kendaraan yang ingin dideteksi. Blok kedua merupakan blok Proses yang menggunakan *Personal Computer* (PC) sebagai pengolah datanya. Sebagian besar dari proses yang terjadi pada sistem ini dilakukan pada blok kedua ini. Proses yang dilakukan pada PC adalah proses utama yaitu pengolahan citra yang diolah menggunakan matlab. Blok bagian terakhir yang merupakan blok output juga

menggunakan PC sebagai media *interface* antara pengguna dengan sistem. Output dari sistem ini adalah nilai *string* yang diekstrak dari gambar yang ditangkap oleh kamera pada bagian input.



Gambar 1. Diagram blok sistem

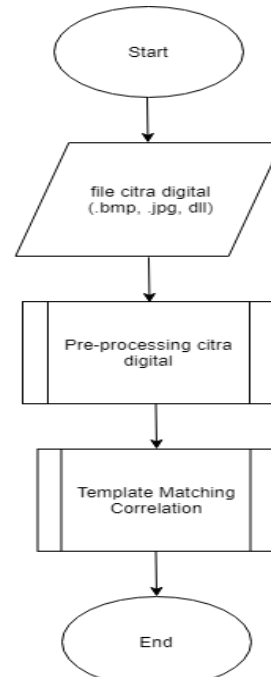
B. Algoritma Sistem

Aplikasi berbasis *Graphical User Interface* (GUI) yang didesain sebagai proses utama dari sistem ini dibuat menggunakan matlab. **Gambar 2** memperlihatkan *flowchart* proses dari sistem tersebut. Proses dimulai dengan memasukkan citra yang telah diambil pada bagian input dengan menggunakan kamera kedalam aplikasi yang telah dibuat. Proses *capture* yang terjadi pada bagian input juga terjadi berdasarkan perintah dari aplikasi yang dibuat. Algoritma dari sistem OCR dengan metode *template matching* yang dibuat adalah sebagai berikut:

- 1) Load gambar
- 2) *Pre-processing* citra digital
 - (2.1) krop bagian karakter dari plat nomor
 - (2.2) ubah citra RGB ke dalam citra *grayscale*
 - (2.3) ubah citra *grayscale* ke dalam citra biner
 - (2.4) menghapus bagian citra biner yang bukan karakter.
 - (2.5) memberi label pada setiap karakter yang terdeteksi.
- 3) Proses deteksi karakter
 - (3.1) Segmentasi untuk memisahkan karakter berdasarkan label yang telah diberikan.
 - (3.2) *Resize* citra karakter hasil segmentasi dengan ukuran citra 42 x 24 untuk dibandingkan pada proses *template matching*.
 - (3.3) membandingkan citra hasil *resize* dengan citra pada basis data
 - (3.4) mengambil karakter dari basis data *template* berdasarkan nilai korelasi yang paling besar.
- 4) Nomor plat teridentifikasi.
- 5) Nomor ditampilkan pada aplikasi berbasis GUI

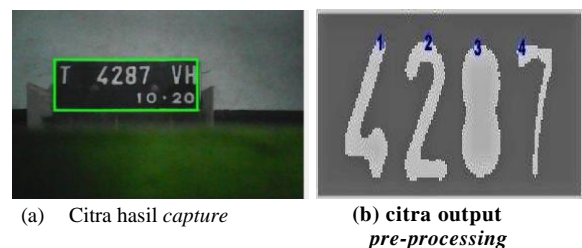
Algoritma sistem yang dibuat membaca gambar dari kamera secara *real-time* dan dikirim ke bagian *pre-processing* untuk diproses. Tugas utama dari bagian *pre-processing* adalah untuk

mengurangi gangguan pada citra dan juga untuk menyesuaikan citra hasil *capture* dengan citra pada basis data *template* agar dapat diproses dengan algoritma *template matching*.



Gambar 2. Flowchart proses

Akurasi pendeteksian suatu karakter pada citra tertentu dipengaruhi oleh keberadaan *noise* pada citra tersebut. Semakin tinggi *noise*, maka akurasi pendeteksian akan semakin kecil dan sebaliknya semakin kecil *noise* akurasi pendeteksian akan semakin besar. Dalam bagian *pre-processing noise* dari citra diusahakan sekecil mungkin dengan cara mengatur sistem agar citra yang diproses dengan *template matching* hanya citra karakter itu saja. Citra karakter diperoleh melalui proses segmentasi karakter-karakter yang sebelumnya telah diberi label. Jumlah label pada citra sebelum segmentasi menunjukkan jumlah karakter yang terdeteksi pada plat nomor kendaraan. **Gambar 3** memperlihatkan citra sebelum dan sesudah diproses pada bagian *pre-processing*.



(a) Citra hasil *capture* (b) citra output *pre-processing*

Gambar 3. Citra sebelum dan setelah melewati bagian *pre-processing*

Deteksi karakter yang dilakukan oleh sistem dilakukan pada proses *template matching*. Pada **Gambar 4**, sebelum citra melewati proses *template matching*, citra yang telah diberi label disegmentasi untuk memisahkan karakter-karakter dalam citra tersebut. Hal ini dilakukan karena proses *template matching* yang akan dilakukan pada dasarnya membandingkan satu karakter dari suatu citra target dengan semua karakter dalam basis data *template* yang berjumlah 36 dengan jumlah huruf 26 (A sampai Z) dan jumlah angka 10 (0 sampai 9).



Gambar 4. Citra hasil segmentasi

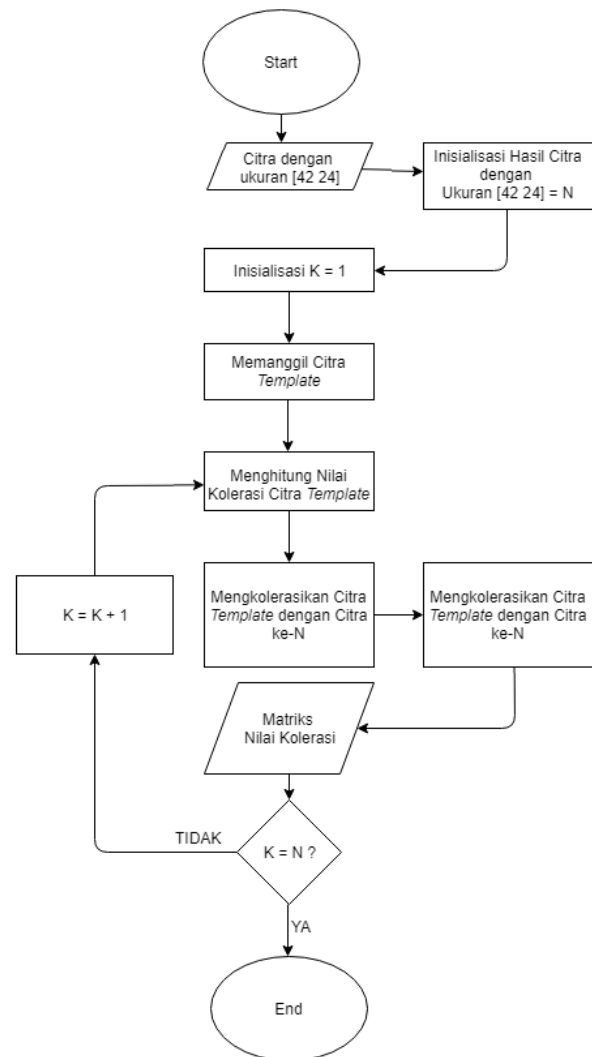
Syarat yang lain yang harus dipenuhi agar proses *template matching* memiliki akurasi pendeteksian yang tinggi adalah ukuran citra target dengan ukuran citra dalam basis data *template* yang akan dibandingkan harus sama. Hal ini dikarenakan proses yang terjadi pada algoritma *template matching* yaitu membandingkan dua buah citra pada pixel yang berada pada kordinat yang sama. **Gambar 5** adalah *flowchart* dari proses *template matching* yang terjadi dalam sistem. Pada dasarnya proses *template matching* adalah membandingkan nilai-nilai *pixel* suatu citra target dengan nilai-nilai *pixel* dari citra gambar dalam basis data *template*.

Citra target dibandingkan dengan seluruh citra dalam basis data *template*. Output dari proses *template matching* adalah karakter yang dinilai sebagai karakter yang terdeteksi pada plat nomor kendaraan target. Output tersebut akan sangat berguna apabila dengan pengembangan dapat dimasukkan ke dalam data base sistem parkir untuk dijadikan sistem informasi pada sistem parkir tersebut. *Flowchart* pada **Gambar 5** menghasilkan ouput satu karakter *string* dari citra hasil segmentasi pada citra keseluruhan dari plat nomor kendaraan. Proses tersebut diulang sebanyak jumlah label yang mengindikasikan jumlah karakter yang terdeteksi pada citra awal hasil *capture* dari kamera.

III. HASIL DAN DISKUSI

Penelitian melalui eksperimen telah dilakukan untuk menguji sistem pendeteksian nomor pada plat kendaraan yang telah diajukan. **Gambar 6** memperlihatkan setup pengujian sistem dari

aplikasi yang telah dibuat. Hasil pengujian sistem pendeteksian nomor plat kendaraan untuk format Indonesia yang telah dibuat disajikan dalam **Tabel I**.



Gambar 5. Flowchart proses *template matching*



(a) Setup hardware system



(b) Setup plat nomor target

Gambar 6. Setup hardware dan plat nomor target

Tabel I memperlihatkan perubahan-perubahan bentuk citra saat melalui proses-proses pengolahan dalam sistem yang telah dibuat beserta dengan hasil pembacaan karakter *string* oleh sistem. Berikut penjelasan tentang proses pengolahan citra seperti yang diperlihatkan pada **Tabel I**.

Tabel I: Hasil pengujian sistem

No	Plat Nomor Target			Output String	Keterangan
	Citra awal	Citra biner	Citra terlabel		
1				Z2093BV	Berhasil Terdeteksi
2				F3696UH	Berhasil Terdeteksi
3				T4287VH	Berhasil Terdeteksi
4				T4822UM	Berhasil Terdeteksi
5				26470FO	Tidak Berhasil Terdeteksi
6				T4046VZ	Berhasil Terdeteksi
7				Z6389FL	Berhasil Terdeteksi
8				7L4934UL4	Tidak Berhasil Terdeteksi
9				T6577UJ	Tidak Berhasil Terdeteksi
10				T3456X	Berhasil Terdeteksi

1) Citra awal
 Citra awal citra dalam format RGB yang merupakan hasil *cropping* dari gambar keseluruhan hasil tangkapan kamera. Citra dikrop pada bagian plat nomornya saja untuk mempermudah proses selanjutnya. *Cropping* juga membatasi pixel yang ingin diproses

dengan demikian ukuran data akan lebih kecil yang mengakibatkan proses akan berlangsung dengan lebih cepat.
 2) Citra biner
 Citra biner merupakan citra yang dihasilkan dari dari pengolahan citra awal yang dirubah menjadi gambar biner. Binerisasi citra

dilakukan untuk mengubah data citra ke dalam data yang lebih sederhana lagi sehingga mudah diamati. Data dari citra biner ini lebih sederhana karena nilai dari pixel citra hanya bernilai 1 atau 0 saja. Data yang hanya terdiri dari 2 nilai akan memudahkan dan mempercepat proses perbandingan dengan *template* pada basis data.

3) Citra terlabel

Pelabelan pada citra dilakukan untuk mempermudah proses segmentasi untuk memisahkan satu karakter dengan karakter yang lain. Pada proses ini jika diamati dengan mata manusia, **Tabel I** sudah mulai memperlihatkan kesalahan pada pembacaan jumlah karakter oleh sistem. Pada citra target ke 8 yang diperlihatkan pada **Gambar 7**, jumlah karakter pada plat nomor yang ingin dideteksi ada sebanyak 7 karakter, sedangkan sistem melabeli citra dengan 9 label, 2 lebih banyak dari yang seharusnya. Hal ini dapat terjadi karena beberapa faktor, misalnya rendahnya kaulitas dari citra yang ditangkap oleh kamera dan bisa juga disebabkan oleh penentuan *threshold* yang tidak tepat yang mengakibatkan satu karakter terputus sehingga sistem melabeli satu karakter tersebut dengan lebih dari 1 label sehingga satu karakter akan dibaca oleh sistem menjadi lebih dari satu karakter. Dari 10 plat nomor yang diujikan terdapat 1 plat nomor yang dilabeli dengan nilai yang salah oleh sistem. Dengan demikian persentase kesalahan pelabelan dapat dihitung dengan persamaan (1).

$$\% \text{ Kesalahan} = \frac{\text{Jumlah Kesalahan}}{\text{Jumlah Target Uji}} \times 100 \% \quad (1)$$

Maka kesalahan pendeteksian jumlah karakter oleh sistem adalah sebesar

$$\frac{1}{10} \times 100 \% = 10 \%$$



Gambar 7. Kesalahan pelabelan karakter

Terlihat pada **Gambar 7** karakter ‘Z’ terputus pada bagian tengah sehingga sistem melabeli karakter tersebut menjadi dua bagian. Beda halnya yang terjadi pada karakter ‘4’, bagian bawah karakter 4 merupakan bagian lubang

skrup dari plat nomor yang dibaca oleh sistem sebagai satu karakter.

4) Hasil Segmentasi

Dari data pada **Tabel I** terdapat 10 plat nomor kendaraan dengan total karakter yang diuji sebanyak 69 karakter. Dari 69 karakter yang dibandingkan dengan metode *template matching*, terdapat 4 karakter yang dideteksi sebagai karakter yang berbeda. Maka persentase kesalahan pendeteksian adalah sebesar

$$\frac{4}{69} \times 100 \% = 0.58 \%$$

Nilai tersebut bisa dikatakan kecil namun sangat berpengaruh jika target yang ingin dideteksi adalah kumpulan karakter seperti nomor plat kendaraan. **Gambar 8** memperlihatkan perbandingan citra asli hasil segmentasi dengan karakter yang dideteksi oleh sistem.



Gambar 8. Hasil segmentasi citra dan output *template matching*

Kesalahan pembacaan karakter oleh sistem dimana karakter ‘W’ diterjemahkan menjadi karakter ‘U’ terjadi karena nilai pixel pada *template* karakter ‘U’ lebih banyak bersesuaian dengan karakter ‘W’ yang menjadi target dibandingkan dengan karakter ‘W’ dalam *template* pada basis data. Hal ini merupakan salah satu kelemahan sistem yang diajukan dalam penelitian ini.

5) Output String

Ke-4 kesalahan karakter yang dideteksi dari 69 karakter yang diujikan terbagi kedalam 3 kesalahan pembacaan oleh sistem dari total 10 pembacaan. Kesalahan pembacaan tersebut bermula dari kesalahan penentuan jumlah karakter yang ada dalam citra pada proses pelabelan. **Gambar 9** memperlihatkan perbandingan citra hasil pelabelan dengan nilai *string* hasil dari pemrosesan oleh sistem.



Gambar 9. Citra yang dilabeli dan hasil pembacaan sistem

Hasil pembacaan oleh sistem menyalahartikan karakter 'Z' pada plat nomor kendaraan menjadi karakter '7L'. Hal ini dikarenakan sistem telah melabeli karakter Z ke dalam 2 label. Jika diperhatikan dengan seksama karakter Z pada citra yang dilabeli terpotong menjadi 2 objek karakter yang sekilas bagian atasnya terlihat mirip dengan angka 7 dan bagian bawah terlihat mirip dengan karakter L. Hal ini mengakibatkan pada bagian output, *string* yang dikeluarkan oleh sistem membaca karakter 'Z' dengan karakter '7L'.

Karena terdapat 3 kesalahan pembacaan dari total 10 target yang diuji, maka persentase kesalahan pembacaan oleh sistem adalah

$$\frac{3}{10} \times 100 \% = 30 \%$$

Persentase kesalahan ini tergolong masih sangat tinggi jika dibandingkan dengan sistem hasil penelitian oleh Saqib Rasheed dengan persentase kesalahan yang hanya sebesar 10,3% [11]. Jika terdapat 100 kendaraan yang ingin dideteksi, dengan persentase kesalahan 30% akan terdapat 30 kendaraan yang dideteksi dengan plat nomor yang berbeda dari yang seharusnya. Hal ini tentunya akan sangat berpengaruh jika sistem ini diimplementasikan pada sistem parkir otomatis. Kelebihan sistem yang dibuat oleh Rasheed dibandingkan sistem yang penulis buat dalam penelitian ini berada pada tambahan algoritma *hough transformation* yang memberikan akurasi yang lebih pada proses pendeteksiannya.

Pada **Gambar 10** karakter 'T' tidak terbaca dengan baik oleh sistem namun dapat diterjemahkan oleh algoritma *template matching* dengan akurat. Kelemahan dari sistem ini adalah saat terdapat 2 karakter yang mirip (hampir sama), kemungkinan kesalahan pendeteksian akan meningkat. Seperti pada **Gambar 8** Bentuk font W dan U pada plat kendaraan yang diujikan hampir sama sehingga ketika ada *noise* atau karakter tidak tercapture secara sempurna, maka sistem kemungkinan besar akan menerjemahkannya dengan karakter yang berbeda seperti yang sudah terjadi pada penelitian ini. Karakter lain yang juga memiliki kemungkinan kesalahan pendeteksian yang tinggi adalah '0' (nol) dan 'O'. akan sangat sulit bagi sistem untuk membedakan antara kedua karakter ini. Oleh sebab itu penulis mengakali sistem ini dengan membagi citra kedalam tiga bagian yaitu

bagian huruf awal, bagian angka, dan bagian huruf akhir. Dengan ini kita tidak perlu membandingkan bagian huruf awal dengan *template* yang berisi angka dan juga sebaliknya.

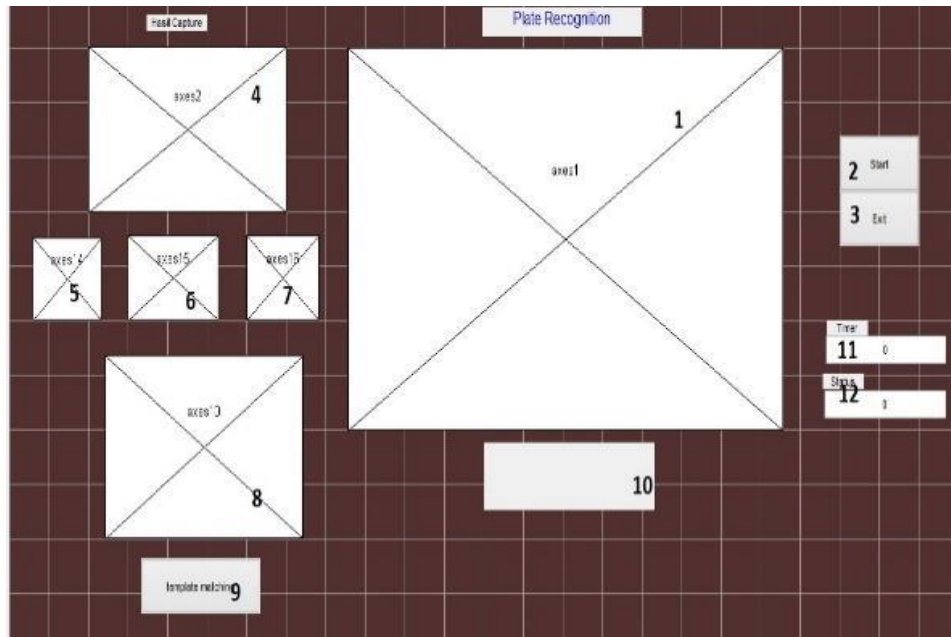


(a) Citra hasil segmentasi (b) Output *string* citra
Gambar 10. Citra hasil segmentasi dan Output *string*

Sebagai interface antara pengguna dengan sistem, penulis merancang GUI yang diperlihatkan pada **Gambar 11**. Aplikasi dalam bentuk GUI ini dibuat menggunakan *software* matlab dan dijalankan pada komputer dengan *processor* Intel Core i3. Karena tidak memerlukan basis data yang besar seperti pada penelitian dengan *neural network*[10], sistem tidak terlalu berat untuk ditangani oleh komputer.

Berikut adalah penjelasan mengenai tampilan yang ada pada **Gambar 11**.

- 1) axes 1 merupakan tampilan dari gambar yang ditangkap oleh kamera yang terhubung dengan PC secara *real-time*. Kualitas gambar ini tergantung dari resolusi kamera yang digunakan. Semakin tinggi resolusinya maka tingkat *noise* pada gambar akan berkurang. Namun semakin tinggi resolusinya maka semakin berat pekerjaan PC dalam mengolah gambar.
- 2) *Push button* start digunakan untuk memberi perintah kepada sistem untuk memulai proses *capturing* gambar.
- 3) *Push button* Exit digunakan untuk mengakhiri program.
- 4) axes 2 merupakan tampilan dari gambar hasil capture yang dilakukan oleh sistem pada gambar yang ada pada axes 1.
- 5) axes 4 merupakan tampilan karakter huruf bagian awal dari plat nomor kendaraan.
- 6) axes 5 merupakan tampilan karakter angka bagian tengah dari plat nomor kendaraan.
- 7) axes 6 merupakan tampilan karakter huruf bagian akhir dari plat nomor kendaraan.
- 8) axes 8 merupakan tampilan gabungan dari axes 5, 6, dan 7. Tampilan ini merupakan tampilan setelah citra pada axis-axis tersebut selesai diproses pada bagian *pre-processing*.
- 9) *Push Button template matching* digunakan untuk memberi perintah kepada sistem untuk melakukan proses *template matching* pada karakter yang telah disegmentasi.



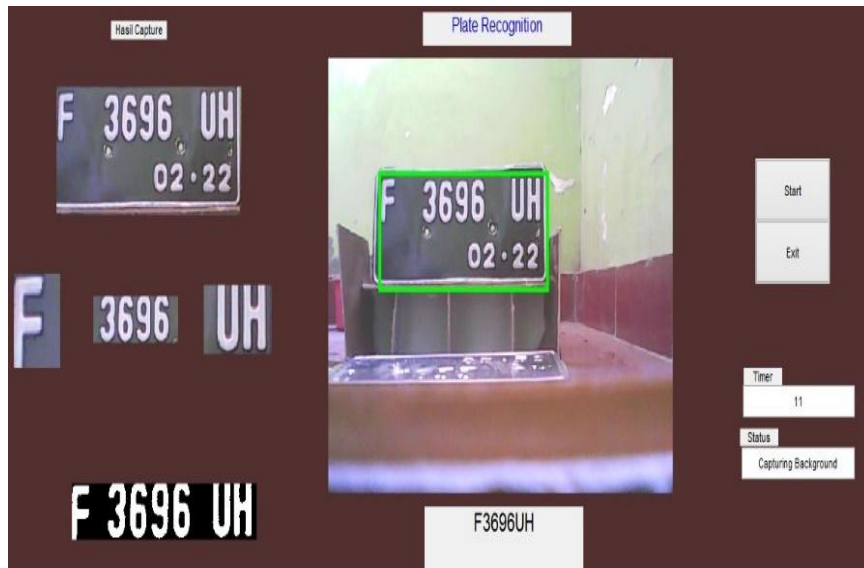
Gambar 11. Tampilan GUI sistem

- 10) Tampilan nilai *string* yang menunjukkan hasil pembacaan sistem terhadap plat nomor kendaraan target. Nilai ini menjadi yang paling penting dalam sistem yang dibuat ini, karena nilai ini merupakan output akhir dari sistem yang dibuat.
- 11) Penanda waktu proses digunakan sebagai indikator lamanya waktu yang digunakan oleh sistem untuk memproses satu loop pendeteksian plat nomor kendaraan.
- 12) Indikator bahwa sistem sedang melakukan proses *capturing* gambar.

Tampilan GUI yang dibuat pada sistem ini dapat dikatakan sebagai persepektif sistem yang dapat diamati oleh manusia. Dengan menggunakan aplikasi berbasis GUI kesalahan-kesalahan pada sistem dapat dicari dan diamati dengan lebih mudah. **Gambar 12** memperlihatkan bagaimana aplikasi GUI yang telah dibuat mendeteksi citra pada kondisi cahaya dengan intensitas yang sedang. Pada Intensitas cahaya yang sedang, sistem memiliki tingkat akurasi yang baik dalam mendeteksi karakter pada plat nomor kendaraan. Intensitas cahaya juga berpengaruh terhadap kualitas citra yang ditangkap oleh kamera. Jika intensitas cahaya terlalu kecil, maka kamera dengan resolusi yang rendah akan mengambil gambar dengan *noise* yang tinggi. Namun jika intensitas cahaya juga terlalu tinggi, maka hal yang sama pun akan terjadi.

Gambar 13 memperlihatkan bagaimana aplikasi GUI yang telah dibuat mendeteksi citra pada kondisi cahaya dengan intensitas yang agak rendah. Pada Intensitas cahaya yang agak rendah

ini, sistem masih memiliki tingkat akurasi yang baik dalam mendeteksi karakter pada plat nomor kendaraan. Intensitas cahaya yang ada pada lingkungan target mempengaruhi *threshold* yang akan digunakan untuk merubah citra *grayscale* ke dalam citra biner. Jika penentuan *threshold* tidak sesuai dengan yang seharusnya, maka hasil gambar biner akan menunjukkan jumlah pixel bernilai 1 yang jauh dari jumlah pixel gambar pada *template* yang ada dalam data base. Pada penelitian yang dilakukan oleh Saqib Rasheed, plat nomor kendaraan yang dideteksi warnanya berkebalikan dengan plat nomor yang dideteksi pada penelitian ini[11]. Warna latar pada nomor plat penelitian Rasheed adalah warna hitam dengan warna karakter putih sedangkan penelitian ini latar dari nomor plat berwarna hitam dengan karakter berwarna putih. Pada dasarnya hal ini sama saja dan tidak terlalu mempengaruhi akurasi dari pendeteksian. Hal tersebut terjadi karena kemampuan sistem membaca karakter pada plat nomor kendaraan tersebut ditentukan oleh besarnya kontras warna antara latar dengan karakter pada plat nomor tersebut. Semakin jelas perbedaan warna antara latar dengan karakter, maka sistem akan lebih mudah untuk mendeteksi karakter-karakter yang ada di dalamnya. Lagi pula pada matlab kita dapat melakukan inverse warna pada citra biner dengan satu baris kode saja yaitu kode dengan fungsi *complement(x)*. Kode tersebut akan membuat data pada *pixel* yang bernilai 0 menjadi 1 dan data pada *pixel* yang bernilai 1 menjadi 0, dengan kata lain warna hitam dijadikan putih dan warna putih dijadikan hitam.



Gambar 12. Tampilan aplikasi GUI pada pendeteksian dengan intensitas cahaya yang tinggi



Gambar 13. Tampilan aplikasi GUI pada pendeteksian dengan intensitas cahaya yang rendah

Pengaplikasian sistem yang serupa dengan pendeteksian plat kendaraan, tidak selamanya target yang diamati akan tegak lurus dengan kamera. Masalah ini sebenarnya dapat diatasi dengan menambahkan beberapa baris algoritma pada program matlab yang dibuat. Algoritma tersebut adalah algoritma projective transformation. Dengan demikian masalah kemiringan sudut nomor plat kendaraan dapat diatasi. Masalah lain juga yang mungkin muncul pada sistem ini adalah ketika permukaan dari plat nomor yang dideteksi bertekstur glossy yang biasanya terjadi ketika plat nomor kendaraan dilapis lapisan pelindung seperti akrilik. Hal ini akan menyulitkan karena saat cahaya mengenai permukaan plat dari samping, permukaannya akan memantulkan cahaya sehingga tidak ada cahaya yang mengenai karakter pada plat nomor kendaraan tersebut dan hasilnya citra dari plat nomor tidak akan menunjukkan citra seharusnya.

IV. KESIMPULAN

Dalam paper ini penulis telah memaparkan metode template matching yang digunakan dalam mendeteksi nomor plat kendaraan dalam format yang ada di Indonesia. Akurasi pendeteksian nomor plat kendaraan format Indonesia yang dilakukan oleh sistem berada pada nilai rata-rata 70%. Nilai pendeteksian tersebut masih termasuk rendah jika ingin diimplementasikan pada sistem parkir kendaraan otomatis. Melalui penelitian ini, penulis mendapatkan informasi mengenai beberapa faktor yang mempengaruhi akurasi sistem pendeteksian plat nomor kendaraan pada format Indonesia dengan menggunakan metode template matching. Faktor-faktor tersebut diantaranya noise pada gambar hasil capture, kecerahan gambar, dan sebagainya. Untuk menghasilkan pendeteksian dengan akurasi yang lebih tinggi serta proses yang lebih cepat,

diperlukan juga device penunjang yang lebih baik. Dalam hal ini kamera dengan resolusi yang lebih tinggi akan menghasilkan kualitas citra yang lebih baik dengan rendahnya noise. Pengolahan proses akan berjalan lebih baik jika PC yang digunakan memiliki spesifikasi sistem yang lebih baik. Penulis dalam hal ini menyarankan penggunaan GPU (Graphical Processing Unit) dengan spesifikasi tinggi dalam memproses program yang dibuat jika ingin sistem menjalankan proses dengan lebih cepat.

Selain faktor dari luar yang mempengaruhi kualitas gambar seperti kondisi cahaya lingkungan dan kemiringan target yang ingin dideteksi, ada juga faktor lain yang mempengaruhi akurasi dari hasil deteksi oleh sistem. faktor tersebut menurut pengamatan penulis adalah penulisan algoritma yang efektif serta efisien. Dengan algoritma yang benar, noise dalam gambar dapat dikurangi salah satunya dengan cara yang telah penulis terapkan dalam penelitian ini, yaitu dengan menghiraukan objek kecil yang terdeteksi sebagai karakter dengan cara menghitung jumlah pixel-nya dan meletakkan threshold sebagai pembatas untuk memisahkan objek yang ditentukan sebagai sebuah karakter dan yang tidak ditentukan sebagai sebuah karakter.

Untuk pengembangan penelitian ini selanjutnya, sistem ini perlu ditambahkan algoritma lain seperti algoritma hough transform untuk meningkatkan akurasi pendeteksian melalui metode template matching. Sistem pendeteksian nomor plat kendaraan ini dapat diimplementasikan pada sistem parkir yang ada di Indonesia. Dengan menambahkan sistem lain, seperti sistem pemindai STNK kendaraan, sistem plang otomatis, dan sistem pembayaran otomatis, sistem lahan parkir tersebut dapat diotomatisasi. Sistem pemindai STNK ini diperlukan untuk membandingkan nomor plat kendaraan dengan nomor plat pada STNK untuk mengkonfirmasi kepemilikan kendaraan. Dengan adanya sistem ini pemilik kendaraan diwajibkan untuk membawa STNK setiap kali memasuki lahan parkir. Karena proses masuk dan keluar dari lahan parkir dilakukan secara otomatis menggunakan komputer, prosesnya pun akan berlangsung lebih cepat.

Dengan teknik yang sama kita juga bisa mengimplementasikan sistem ini pada sistem tilang online. Pelanggaran yang dapat dideteksi dari citra plat nomor adalah pelanggaran

pembayaran pajak kendaraan yang dapat dilihat pada bagian plat kendaraan yang menunjukkan tanggal kadaluarsa pajak kendaraan tersebut. Proses akuisisi gambar pada implementasi ini mirip dengan proses yang telah dilakukan pada penelitian ini. Bedanya adalah sistem yang dibuat ini hanya mampu mendeteksi plat nomor yang tegak lurus dengan kamera sehingga akan sulit diimplementasikan secara langsung pada persimpangan jalan atau pada lampu merah. Oleh karena itu untuk pengembangan selanjutnya sistem ini dapat dilengkapi dengan algoritma projective transformation sehingga sistem dapat mendeteksi plat nomor kendaraan yang berada dalam posisi miring terhadap kamera.

DAFTAR PUSTAKA

- [1] H. Al-Kharusi and I. Al-Bahadly, "Intelligent Parking Management System Based on Image Processing," *World J. Eng. Technol.*, vol. 02, no. 02, pp. 55–67, 2014.
- [2] M. A. Therib, "Republic of Iraq Design and Implementation of Smart Car Parking System," *International Journal of VLSI Design & Communication Systems*, 2017.
- [3] O. R. Shahin, H. M. Kelash, G. Attiya, and O. S. Farag, "Breast Cancer Detection Based on Dynamic Template Matching," *Wulvenia Journal Klagenfurt Australia*, vol. 20, no. 12, pp. 193–205, 2013.
- [4] M. Owayjan, A. Dergham, G. Haber, N. Fakhri, A. Hamoush, and E. Abdo, "Face Recognition Security System," *Lect. Notes Electr. Eng.*, vol. 312, pp. 343–348, 2015.
- [5] D. A. Chowdhry, A. Hussain, M. Z. U. Rehman, F. Ahmad, A. Ahmad, and M. Pervaiz, "Smart security system for sensitive area using face recognition," *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET)*, pp. 11–14, 2013.
- [6] M. O. Oloyede, A. O. Adedoyin, and K. S. Adewole, "Fingerprint Biometric Authentication for Enhancing Staff Attendance System," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 3, pp. 19–24, 2013.
- [7] N. Abbas, M. Tayyab, and M. T. Qadri, "Real Time Traffic Density Count using Image Processing," *International Journal of Computer Applications*, vol. 83, 2013.
- [8] X. Wang, L. M. Meng, B. Zhang, J. Lu, and K. L. Du, "A video-based traffic violation detection system," *Proceedings - 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013*, pp. 1191–1194, 2013.
- [9] M. I. Khalil, "Car Plate Recognition Using the Template Matching Method," *Int. J. Comput. Theory Eng.*, vol. 2, no. 5, pp. 683–687, 2013.
- [10] İ. Türkyılmaz and K. Kaçan, "License Plate Recognition System Using Artificial Neural Networks," *ETRI Journal*, vol. 39, no. 2, 2017.
- [11] S. Rasheed, A. Nameem, O. Ishaq, "Automated Number Plate Recognition Using Hough Lines and Template Matching," *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2012.