

PENGENALAN PLAT KENDARAAN BERBASIS ANDROID MENGUNAKAN VIOLA JONES DAN KOHONEN NEURAL NETWORK

Aryo Michael

aryo.michael@yahoo.co.id
Universitas Kristen Tentena

Abstrak

Penelitian ini bertujuan untuk membangun sistem pengenalan plat kendaraan roda dua berbasis android. Penelitian ini dilaksanakan di Kota Makassar Indonesia. Metode yang digunakan pada penelitian ini adalah pengenalan pola plat kendaraan dengan metode viola jones, kemudian segmentasi karakter plat menggunakan metode morfologi, dan pengenalan karakter plat dengan kohonen neural network. Hasil penelitian yang diperoleh menunjukkan bahwa pengenalan plat kendaraan menggunakan *kohonen neural network* berdasarkan pengujian yang dilakukan menunjukkan persentase keberhasilan pengenalan karakter pada plat kendaraan bermotor pada kondisi yang baik sebesar 78,57% edangkan pada plat yang kurang baik sebesar 57,14%

Kata Kunci: ANPR, Kohonen Neural Network, OCR, Android.

Copyright © 2016 – Jurnal Ilmiah ILKOM – All rights reserved.

1. Pendahuluan

Dewasa ini, teknologi semakin berkembang dengan sangat pesat. Banyak terobosan-terobosan baru yang diciptakan dengan basis teknologi canggih. Semua itu diciptakan dengan tujuan agar mempermudah dan mempercepat kerja manusia (*assistant system*). Salah satu teknologi yang berkembang saat ini adalah teknologi *smartphone android* yang memberikan kemudahan serta fleksibilitas yang tinggi, yang relative murah dan lebih *flexible* untuk digunakan serta dapat dikembangkan secara massal karena bersifat *opensource*. Dengan adanya perkembangan teknologi *smartphone android* banyak penelitian-penelitian dibidang *computer vision* dilakukan.

Salah satu bidang yang sangat menarik dalam *computer vision* adalah pengenalan plat kendaraan otomatis atau biasa disebut dengan *Automatis Number Plate Recognition* (ANPR). Pengenalan plat nomor kendaraan otomatis merupakan salah satu teknik penting sebagai bagian dari sistem transportasi cerdas yang dapat digunakan untuk mengidentifikasi kendaraan hanya dengan memahami plat nomor. Sistem pengenalan plat kendaraan secara umum terdiri atas 3 (tiga) tahapan penting dalam yaitu deteksi plat nomor, segmentasi karakter, dan pengenalan karakter. Beberapa penelitian mengenai ANPR telah dilakukan dengan menawarkan beberapa metode. Metode saat ini yang banyak digunakan untuk deteksi plat nomor, yaitu metode berbasis *image processing* dan metode berbasis *machine learning*[1].

Di Indonesia sendiri ANPR masih terus dikembangkan terutama untuk tahap pendeteksian objek plat nomor. Meskipun di negara lain ANPR sudah banyak dikembangkan, namun sistem ANPR ini tidak dapat langsung digunakan di negara Indonesia karena perbedaan format atau jenis, warna, dan struktur dari plat nomor. Penelitian tentang ANPR berbasis android yang telah dilakukan pada plat Indonesia, dimana penelitian di fokuskan pada pembacaan karakter plat kendaraan menggunakan Jaringan saraf Tiruan dengan metode pembelajaran *Backpropagation* serta beberapa penelitian mengenai ANPR pada sistem Android [2] [3][4][5].

Berdasarkan uraian di atas, maka pada penelitian ini diusulkan suatu sistem pengenalan plat kendaraan menggunakan metode *viola jones* untuk mendeteksi pola plat, segmentasi karakter plat dengan melihat karakteristik morfologi, dan pengenalan karakter menggunakan *Algoritmakohonen neural network*.

2. Landasan Teori

2.1 Tanda Nomor Kendaraan Bermotor

Setiap kendaraan di Indonesia yang telah melakukan registrasi pasti memiliki sebuah Tanda Nomor Kendaraan Bermotor (TNKB). Aturan atau regulasi mengenai plat kendaraan ini berbeda-beda disetiap satu negara dengan negara yang lainnya. Di Indonesia, Plat nomor kendaraan selalu dibuat seragam atau sama antara satu jenis kendaraan dengan kendaraan lain, baik dalam ukuran plat, hurufpada plat nomor pun selalu sejenis.

2.2 Automatic Number Plate Recognition

License Plate Recognition (LPR) atau sering disebut juga dengan *Automatic Number Plate Recognition*[6] merupakan salah satu teknologi atau sistem yang menangani pengenalan nomor polisi yang ada pada plat nomor kendaraan bermotor. Pengenalan nomor polisi yang ada pada plat nomor kendaraan tersebut dapat dilakukan secara otomatis dengan menggunakan teknologi seperti *Optical Character Recognition*. ANPR adalah jenis teknologi, yaitu perangkat lunak yang memungkinkan system komputer dapat membaca nomor kendaraan secara otomatis dari citra digital. Yang dimaksud dengan dengan membaca nomor kendaraan secara otomatis adalah mengubah pixel dari citra digital ke dalam karakter ASCII dari plat nomor.

2.3 Metode Viola Jones

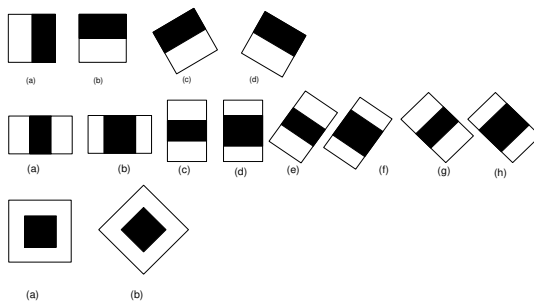
Saat ini telah banyak berkembang aplikasi-aplikasi yang menggunakan fitur deteksi objek. deteksi objek dapat dilakukan dengan berbagai cara, salah satunya menggunakan metode *viola-jones*. metode *viola-jones* diperkenalkan pada tahun 2001. Metode *viola-jones* memiliki empat konsep utama dalam mendeteksi suatu citra [7].

1) Fitur Segi Empat Sederhana *Haar Feature*

Fitur yang digunakan oleh Viola-Jones didasarkan pada *Wavelet Haar*. *Wavelet Haar* adalah gelombang tunggal bujur sangkar yang mempunyai satu interval tinggi dan satu interval rendah. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik disebut *haar feature* atau *haar-like feature*. Setiap *Haar-like feature* terdiri dari gabungan kotak - kotak hitam dan putih seperti pada gambar 1. adanya fitur haar ditentukan dengan mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Secara matematis ditulis dengan rumus :

$$f(x) = \text{SumBlackRectangle} - \text{SumWhiteRectangle}$$

Jikanilai perbedaannya masih diatas nilai ambang atau *threshold*, maka dapat dikatakan bahwa fitur tersebut ada. Terdapat 3 jenis fitur *haar* seperti pada gambar 2.



Gambar 1. Macam-Macam Variasi Fitur Haar [7]

2) Deteksi Fitur dengan *Integral image*

Integral images merupakan metode untuk menghitung kotak *Haar feature*. Metode ini digunakan untuk menghitung secara cepattanpa menghitung keseluruhan nilai pixel yang ada dalam fitur. *Integral images* pada lokasi x,y terdiri dari nilai jumlah pixel diatas dan dikiri lokasi x,y .

Menurut Viola dan Jones *Integral image* pada lokasi x,y , berisikan jumlah dari atas sampai kiri dari x,y perhitungannya menggunakan rumus berikut :

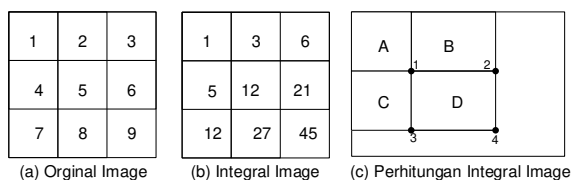
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') = \text{Integral Image adalah original image}$$

Untuk menghitung *integral image* dengan mengabaikan *original image*, menggunakan pasangan rumus berikut ini :

$$\begin{aligned} ii(x, y) &= ii(-1, y) + s(x, y) \\ s(x, y) &= s(x - 1, y) + i(x, y) \end{aligned}$$

Dimana :

$$\begin{aligned} i(x, y) &: \text{Original Image} & s(x, -1) &= 0 \\ ii(x, y) &: \text{integral image} & ii(-1, y) &= 0 \\ s(s, y) &: \text{jumlah kumulatif baris} \end{aligned}$$



Gambar 2. Contoh perhitungan *integral image*

Dapat dilihat pada gambar 2, berdasarkan perhitungan jumlah pixel dalam persegi D, dapat dihitung dengan referensi array. Nilai dari *integral image* pada lokasi 1 adalah jumlah dari persegi panjang A. Nilai pada lokasi 2 adalah A+B, nilai pada lokasi 3 adalah A+C, dan pada lokasi 4 adalah A+B+C+D. penjumlahan dalam D dapat dihitung seperti $4+1-(2+3)$. Proses pencarian dilakukan secara iteratif mulai dari ujung kiri atas image hingga ujung kanan bawah *image*.

3) Metode *Machine Learning Adaptive Boosting (AdaBoost)*

Kebanyakan algoritma *boosting* mengikuti sebuah rancangan yaitu menambahkan *weak learner* ke dalam satu *strong learner*. *Weak learner* adalah *classifier* yang hanya memiliki sedikit korelasi dengan klasifikasi sebenarnya. Sementara *strong learner* adalah *classifier* yang memiliki korelasi yang kuat dengan klasifikasi sebenarnya [8]. *Adaboost* merupakan singkatan dari *Adaptive Boosting* yang di formulasi oleh Yoav Freund dan Robert Schapire pada tahun 1995. Algoritma *AdaBoost* berfungsi untuk melakukan pemilihan-fitur-fitur dalam jumlah banyak dengan hanya memilih fitur-fitur tertentu. Algoritma *AdaBoost* merupakan algoritma yang mampu beradaptasi dengan *weak classifier*.

4) *Pengklasifikasian Bertingkat (Cascade Classifier)*

Kombinasi *cascade classifier* merupakan tahap akhir dalam metode *Viola-Jones*. Dengan mengkombinasikan pengklasifikasian dalam sebuah struktur bertingkat, kecepatan dari suatu proses dapat meningkat yaitu dengan cara memusatkan perhatian pada daerah-daerah dalam citra yang berpeluang saja. Hal ini dilakukan untuk menentukan dimana letak objek yang dicari pada suatu citra.

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Hasil dari klasifikasi pertama ini berupa T (*True*) untuk gambar yang memenuhi fitur Haar tertentu dan F (*False*) bila tidak. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Hasil dari klasifikasi kedua berupa T (*True*) untuk gambar yang memenuhi proses *integral image* dan F (*False*) bila tidak. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%. Hasil dari klasifikasi terakhir berupa T (*True*) untuk citra yang memenuhi proses *AdaBoost* dan F (*False*) bila tidak.

2.4 Kohonen Neural Network

Jaringan saraf tiruan dirancang menggunakan suatu aturan yang bersifat secara menyeluruh, dimana semua model jaringan memiliki konsep dasar yang sama. Ada banyak model jaringan yang digunakan sebagai jaringan saraf tiruan, sebagian besar jaringan melakukan penyesuaian bobotnya selama menjalani proses latihan. Salah satu model jaringan Saraf tiruan yang banyak digunakan adalah *Kohonen Neural Network*. *Kohonen Neural Network* merupakan analogi sederhana dari cara kerja otak manusia dalam mengelompokkan informasi. Penelitian menunjukkan bahwa kulit otak manusia terbagi ke dalam bagian-bagian yang berbeda, masing-masing merespon fungsi-fungsi khusus. Sel-sel saraf mengelompokkan dirinya sendiri sesuai dengan informasi yang diterima. Pengelompokan seperti ini disebut *unsupervised learning* [9].

Kohonen Neural network adalah *neural network* berdasarkan kompetisi yang terdiri dari 2 lapisan, lapisan masukan dan lapisan keluaran. Keluaran pada *Kohonen Neural Network* berjumlah sama dengan jumlah pola yang membentuk *neuron* keluaran, dan hanya satu *neuron* keluaran yang menjadi pemenang.

Metode pembelajaran adalah proses latihan mengenali data atau pola tertentu dan menyimpan pengetahuan atau informasi yang didapat ke dalam bobot-bobot [11]. Proses pembelajaran dilakukan melalui beberapa *epoch* (jangkauan waktu) sampai kesalahan dari *Kohonen Neural network* di bawah *level* yang ditentukan atau batas *epoch* terlewati. Proses pembelajaran *Kohonen Neural network* bersifat kompetitif atau bersaing. Untuk setiap pelatihan terdapat satu *neuron* keluaran dalam kondisi *firing* (*neuron* keluaran yang bernilai paling besar). Bobot yang terkoneksi pada *neuron* keluaran tersebut akan disesuaikan sehingga nilainya lebih kuat pada latihan berikutnya.

Algoritma metode pembelajaran Kohonen [12] adalah sebagai berikut:

1. Inisialisasi seluruh bobot dengan nilai random: w_{ij}
normalisasi seluruh bobot dan vektor masukan \mathbf{x} , set parameter *learning rate*
2. Kerjakan a-f sampai kondisi berhenti bernilai *TRUE*
 - a. evaluasi kesalahan untuk setiap vektor masukan \mathbf{x}
 - b. simpan bobot dengan kesalahan paling minimal

- c. cek neuron keluaran yang telah firing atau aktif, jika terdapat neuron keluaran yang tidak pernah firing, maka *forcewin* dan kembali ke a). Jika tidak lanjutkan langkah e)
 - d. *Forcewin*
 - hitung aktivasi setiap vektor masukan dan ambil indeks vektor masukan dengan aktivasi yang paling kecil
 - hitung setiap *neuron* keluaran dengan vektor masukan yang didapat pada a) dan pilih indeks *neuron* keluaran dengan nilai terbesar yang tidak pernah aktif selama latihan.
 - modifikasi bobot dari *neuron* keluaran pada b)
 - e. Sesuaikan bobot dengan persamaan $w' = \alpha(x - w)$
 - f. Perbaiki *learning rate*
 - g. Test kondisi berhenti.
3. Ambil bobot terbaik
 4. Normalisasi bobot

3. Metode

3.1 Pengumpulan Data dan Informasi

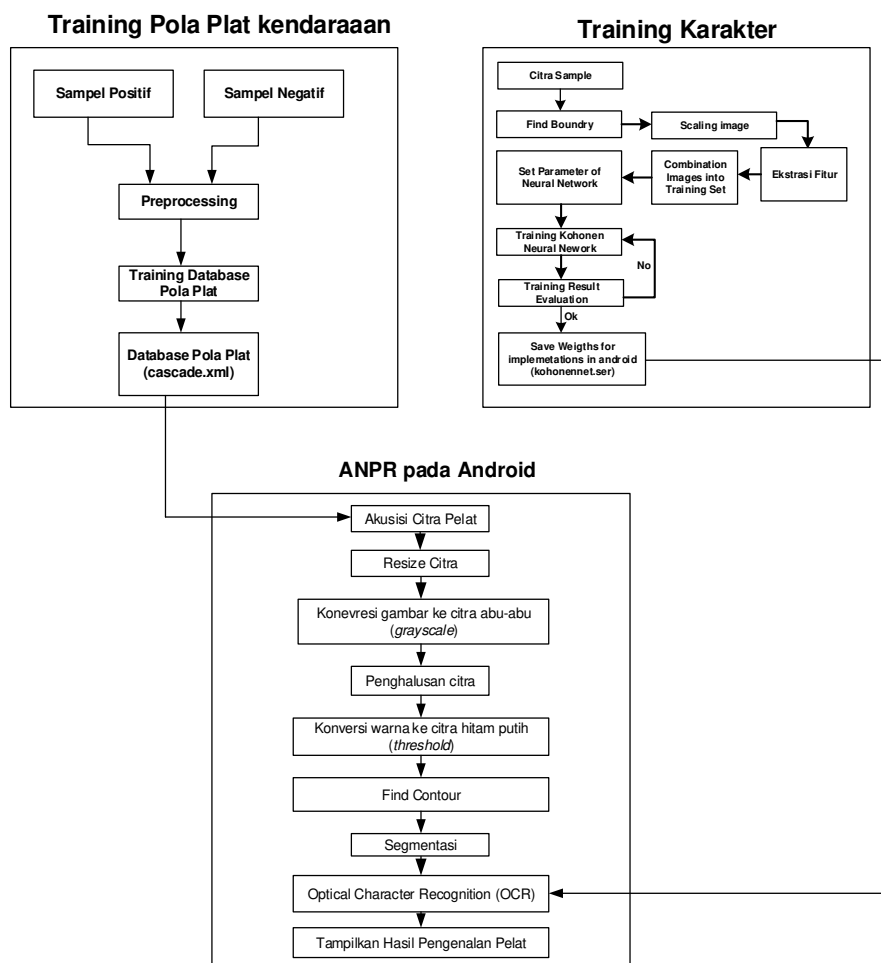
1. Studi literatur yaitu Mempelajari buku referensi yang berkaitan dengan tema penelitian.
2. **Pengambilan Sampel** yang akan digunakan dalam penelitian ini adalah *imagesplat* kendaraan yang diambil dengan menggunakan kamera *smartphone* menggunakan resolusi kamera 0,9 MP.

3.2 Pengembangan Sistem

1. Melakukan *training* pola plat kendaraan dengan metode *viola-jones*,
2. Melakukan *Training* karakter plat kendaraan dengan algoritma *kohonen neural network*.
3. Perancangan sistem pengenalan plat kendaraan pada android menggunakan Bahasa pemrograman java.

4. Hasil dan Pembahasan

4.1 Perancangan Sistem



Gambar 3. Desain system yang diusulkan

1. Training Pengenalan Pola

1) Input Gambar

Images yang digunakan untuk *training database* pola plat adalah *image* dengan *extensi file* .BMP dan jenis color image RGB. Sampel yang digunakan pada *training database* pola plat kendaraan dibagi menjadi 2 (dua) bagian yaitu sampel positif dan sampel negatif. Sampel positif adalah sampel yang merupakan pola positif dari sebuah plat, sedangkan sampel negatif adalah sampel yang diambil dari sembarang pola yang tidak mengandung objek.

2) Preprocessing

Pada tahapan ini akan dilakukan *resize* pada sampel positif. Tujuan dari proses *resize* untuk mempercepat proses komputasi pada saat *training*.

3) Training Database

Metode *Viola-Jones* mengadopsi *cascade objek detector* untuk melatih data input sebagai database dengan *extensi file* *.XML. Ada dua cara yang dapat dilakukan untuk melatih sampel positif, yang pertama yaitu menentukan *bounding boxes* pada ROI (*region of interest*) dan yang kedua yaitu melakukan *cropping* hanya pada gambar plat kendaraan.

Training database pola plat kendaraan pada penelitian ini menggunakan *OpenCV Traincascade*. *Setting Parameter* yang digunakan untuk melakukan *training* pola plat kendaraan pada sistem ini adalah :

-numPos (Jumlah Sampel Positif)	: 1087	-maxfalsealarm	: 0.5 [default]
-numNeg (Jumlah Sampel Negatif)	: 2880	-Feature Type	: Haar [default]
-numStages (Jumlah Stages)	: 20	-Weigth	: 40
-minhitrate (True Positif Rate)	: 0.995	-Heigth	: 16

2. Training Karakter Kohonen Neural Network

Pengumpulan sampel untuk *training* karakter diambil plat kendaraan bermotor. Adapun tahapan-tahapan yang dilakukan dalam training karakter plat kendaraan adalah sebagai berikut :

1) Preprocessing

yang dimaksud disini adalah melakukan *cropping* secara manual untuk memisahkan masing-masing karakter plat kendaraan. Karakter yang telah dilakukan *cropping* secara manual. Karakter yang digunakan sebagai sampel *training* sebanyak 36 karakter yaitu 0 - 9 dan A - Z dengan *extensi file* *.jpg.

2) Find Boundry

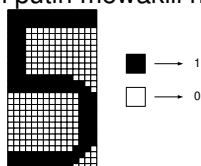
Proses *Find Boundry* karakter dilakukan dengan mencari batas daerah gambar, mulai dari batas atas, batas bawah, batas kanan dan batas kiri. Proses *find boundry* diawali dengan menscan gambar dari kiri ke kanan dan dari atas ke bawah. Proses *scan* adalah untuk mengetahui awal terdapatnya piksel hitam sampai selanjutnya bertemu pada piksel hitam yang terakhir pada gambar. Setelah proses selesai, akan dilanjutkan dengan melakukan pengkotakan atau pembatasan daerah gambar yang memiliki piksel hitam.

3) Scaling

Setelah proses *find boundry* langkah selanjutnya adalah *scaling*, dimana proses *scaling* adalah proses untuk mengubah ukuran suatu image ke dalam ukuran yang sama. *Scaling* merupakan hal yang sangat penting dalam pengenalan karakter, dimana hal yang menjadi pertimbangan adalah apabila sampel karakter yang akan digunakan terlalu kecil akan mempercepat proses training, akan tetapi akan semakin memperbesar tingkat kesalahan deteksi. Sebaliknya, sampel yang terlalu besar akan mengurangi tingkat kesalahan deteksi akan tetapi akan semakin lama dalam proses training karakter. Pada penelitian ini semua ukuran sampel *image* karakter diubah ke dalam ukuran matrix 20 x 50.

4) Ekstraksi Fitur

Ekstraksi fitur bertujuan untuk mendapatkan karakteristik suatu karakter yang membedakan dari karakter yang lain. Hasil proses *scaling* kemudian ditransformasi ke dot matrix, dimana pixel hitam mewakili nilai 1 dan pixel putih mewakili nilai 0.



Gambar 4 Ekstraksi Fitur

Hasil ekstraksi fitur kemudian disimpan dalam sebuah dataset yang akan dijadikan sebagai *dataset training* karakter.

Setelah proses penyatuan sampel training dalam sebuah dataset, maka dilakukan training dengan *Kohonen Neural Network*. Parameter yang digunakan dalam algoritma kohonen meliputi bobot acak dengan random 0 dan 1, *learning rate*= 0.5, pengurang *learning rate* = 0.3, MaxEpoch = 10000.

3. Sistem ANPR pada Android

1) Akusisi Citra Plat

Dalam proses akusisi citra tidak dibatasi oleh resolusi kamera yang ada pada *smartphone*. Citra yang akan diambil merupakan hasil deteksi pola plat menggunakan metode *Viola-Jones*. Hasil training pengenalan pola dengan *viola jones* dimasukan sebagai database pada sistem pengenalan plat pada android.

2) Rezise Citra

Citra yang telah dikenali polanya oleh sistem, kemudian disimpan dan *resize* ke ukuran 450 x 670.

3) Konversi Gambar Citra Keabuan

Citra yang akusisi oleh kamera *smartphone* merupakan citra RGB. Tahapan yang pertama adalah mengubah citra RGB menjadi citra keabuan (*grayscale*). Pada *openCv* digunakan syntax "*imgproc.cvtColor (src, dst, RGB2GRAY);*"



Gambar 5 Citra Grayscale

4) Penghalusancitra

Penghalusan citra berfungsi untuk mengurangi atau menghilangkan noise yang ada pada citra. Sebagian besar citra yang diambil pastinya tidak akan terlepas yang *noise*. Banyaknya *noise* akan sangat berpengaruh pada proses selanjutnya karakter nantinya. Berdasarkan hal tersebut, untuk mengatasi gangguan *noise* pada citra *grayscale* tersebut maka digunakan *median filter* untuk penghalusan citra pada citra *grayscale*.

5) Thresholding

Metode yang digunakan untuk mengubah citra keabuan (*grayscale*) ke citra hitam putih pada sistem ini adalah *Otsu Threshold*. Tujuan dari metode *otsu* adalah membagi histogram citra keabuan (*grayscale*)kedalam dua daerah yang berbeda secara otomatis tanpa membutuhkan bantuan *user* untuk memasukkan nilai ambang. Nilai *global threshold* yang digunakan adalah 125. Setelah *otsu threshold* dilakukan maka nilai *threshold* dikembalikan sebagai nilai normal antara 0 dan 1.



Gambar 6 Citra Hitam Putih

6) Deteksi Kontour

Setelah proses *thresholding* selesai, maka tahapan selanjutnya adalah deteksi kontour dari citra *threshold*. Pada *OpenCV* fungsi deteksi tepi dengan teknik deteksi kontur dapat dilakukan dengan syntax "*imgproc.findContours(src, contours, hierarchy, imgproc.Chain_Approx_Simple, imgproc Retr_List)*". Setelah proses pencarian *contours* maka proses tersebut dilanjutkan dengan proses labeling dengan menggunakan *boundingrect*.

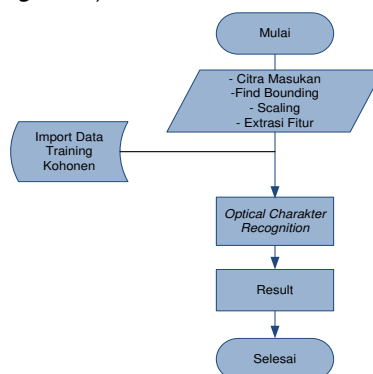
7) Segmentasi Karakter

Pemisahan karakter dilakukan dengan melihat ukuran *bounding box* yang ada pada karakter. Oleh karena ukuran *bounding box* bervariasi maka untuk itu *segmentasi* karakter menggunakan ukuran minimal dan maximal *bounding box* yang dideteksi.



Gambar 7 Hasil segmentasi karakter

8) OCR (Optical Karakter Recognition)



Gambar 8 Diagram Alir OCR dengan Kohonen Neural Network

Proses pengenalan karakter pada android tidak jauh berbeda dengan proses pengenalan karakter pada saat *training*. Citra masukan adalah citra karakter hasil segmentasi. Proses selanjutnya adalah mencari *boundry* pada masing-masing karakter, proses *Find Boundry* karakter pada android dilakukan dengan mencari batas daerah citra, mulai dari batas atas, batas bawah, batas kanan dan batas kiri. Proses *find boundry* diawali dengan menscan gambar dari kiri ke kanan dan dari atas ke bawah. Proses *scan* adalah untuk mengetahui awal terdapatnya piksel hitam sampai selanjutnya bertemu pada piksel hitam yang terakhir pada gambar. Setelah proses selesai, akan dilanjutkan dengan melakukan pengkotakan atau pembatasan daerah gambar yang memiliki piksel hitam.

Setelah masing-masing karakter diberi *boundry*, selanjutnya untuk menyamakan ukuran karakter maka semua citra karakter yang telah diboundry diubah keukuran 20 x 50 pixel sesuai dengan sampel *training* karakter. Extraksi fitur dilakukan untuk mengambil nilai pixel yang ada pada citra masukan dimana pixel hitam mewakili nilai 1 dan pixel putih mewakili 0. Parameter yang digunakan untuk pengenalan karakter dengan kohonen network, sama dengan yang digunakan pada saat training karakter. Parameter yang digunakan dalam algoritma kohonen meliputi bobot acak dengan random 0 dan 1, *learning rate*= 0.3, pengurang *learning rate* = 0.99, MaxEpoch = 10000. Tahapan selanjutnya ada melakukan pencocokan dengan hasil *training* karakter yang ada. Hasil pengenalan karakter akan ditampilkan pada layar *smartphone*.

4.2 Pengujian

Setelah sistem selesai dibuat maka dilakukan pengujian. Pengujian dilakukan dengan melakukan percobaan terhadap 28 plat berbeda yang terbagi dalam 2 (dua) bagian 14 plat dengan kondisi baik dan 14 plat dengan kondisi yang kurang baik. Plat kendaraan diuji pada jarak yang berbeda-beda yaitu 40cm-60cm. Pengujian dilakukan menggunakan *smartphone* android Samsung V dengan spesifikasi Processor 1,2 Ghz, RAM 512 Megabyte, Kamera 3,2 Megapixel.

Tabel 1. Hasil Pengujian Pada Plat Kondisi Baik

	No Plat	Hasil Pengenalan	Keterangan
1	DD3407UE	DD3407UE	SUKSES
2	DN5269VK	DN5269VK	SUKSES
3	DN4584MD	DN4S84M0	GAGAL
4	DN4505VT	DN4505VT	SUKSES
5	DN3990MG	DN3990MG	SUKSES
6	DN3742ND	DN3742N0	GAGAL
7	DN3142MF	DN3142MF	SUKSES
8	DN4771YS	0N4771YS	GAGAL
9	DN4112YE	DN4112YE	SUKSES
10	DN2106NI	DN2106NI	SUKSES
11	DN4839YC	DN4839YC	SUKSES
12	DN5685VH	DN5685VH	SUKSES
13	DN6330VM	DN6330VM	SUKSES
14	DN3913NE	DN3913NE	SUKSES
	Persentase Keberhasilan		$\frac{11}{14} = 78,57\%$

Tabel 2. Hasil Pengujian Pada Plat Kondisi Kurang Baik

	No Plat	Hasil Pengenalan	Keterangan
1	DN4370MA	D4370MA	GAGAL
2	DN4235VN	DN423SVN	GAGAL
3	DN6330VM	DN6330VM	SUKSES
4	DN3468GO	DN3468GO	SUKSES
5	DP3829FC	0P3829FC	GAGAL
6	DN4112VV	DN4112VV	SUKSES
7	DN3735VO	DN3735VO	SUKSES
8	DD4550AI	DD4550U1	GAGAL
9	DD2210EB	DD2210EB	SUKSES
10	DD5558MM	DD5558MM	SUKSES
11	DN2579NG	DN2578N6	GAGAL
12	DN3290NI	DN3290NI	SUKSES
13	DN2289NC	DN2289NC	SUKSES
14	DN5896JK	DN5896UK	GAGAL
Persentase Keberhasilan			$\frac{8}{14} = 57,14\%$

5. Kesimpulan

Tingkat keberhasilan pengenalan karakter plat kendaraan dengan menggunakan Kohonen Neural Network sebesar 78,57 % pada kondisi plat kendaraan yang baik sedangkan 57,14% pada kondisi plat yang kurang baik. Kegagalan pada proses pengenalan karakter disebabkan oleh pencahayaan yang tidak merata pada plat sehingga ada beberapa karakter yang tidak bisa tersegmentasi dengan baik serta kemiripan karakter pada plat kendaraan seperti D dan 0, N dan H, S dan 5.

Daftar Pustaka

- [1] Muhammad Faisal Amin, Romi Satria Wahono. Penerapan Reduksi Region Palsu Berbasis Mathematical Morphology pada Algoritma Adaboost Untuk Deteksi Plat Nomor Kendaraan Indonesia. *Journal of Intelligent Systems*, Vol. 1. SN 2356-3982. 2015.
- [2] Mohammed Ali Alshahrani, 2013. *Real Time Vehicle License Plate Recognition on Mobile Devices (Thesis)*. Master of Science in Computer Science. The University of Waikato. New Zealand.
- [3] Amita Mundhe, Jayashree Otari. *Automatic Number Plate Recognition Using Smart Phones*. *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181. .2013
- [4] Rizqi K. Romadhon, Muhammad Ilham, Nofan I. Munawar, Sofyan Tan, Rinda Hedwig. *Android-Based Licence Plate Recognition Using pre-Trained Neural Network* *Internetworking Indonesia Journal*, ISSN: 1942-9703. . 2012
- [5] Tuan Nguyen, Don Nguyen, Phu Nguen. *Supervised Learning Method for Mobile International Number Plate Recognition*. *Proceedings of the 16th International Conference on Information Itegration and Web-based Application & Services* Pages 212-216, ISBN: 978-1-4503-2644-5. 2014.
- [6] S. Shashidhar , S.V Krishna , 2014. *FPGA Implementation of Various Security Based Tollgate system Using ANPR Technique*. *International Journal of Engineering Trends and Technology (IJETT)* Vol.15 .ISSN 2231-5381
- [7] Paul Viola, Michael Jones. 2001, *Rapid Object Detection using a Boosted Cascade of Simple Features*, *Conference On Computer Vision and Pattern Recognition*.
- [8] Andoko, dkk. *Perancangan Program Simulasi Deteksi Wajah Dengan Support Vector Machines- Viola Jones*. 2007
- [9] Kusumadewi, S., *Artificial Intelligence Teknik dan Aplikasinya*, Graha Ilmu, Yogyakarta. 2003.,
- [10] Heaton J. *Introduction to Neural Network with Java*, <http://www.heatonresearch.com/articles/6/page2.html>, (diakses terakhir tanggal 18 April 2014). 2003
- [11] Zamasari, Reki., *Pengenalan Pola Karakter Aksara Jawa dengan Algoritma Kohonen Neural Network*, Universitas Diponegoro, Semarang. 2005.