

APLIKASI PENGENALAN DAN IDENTIFIKASI PLAT NOMOR KENDARAAN BERBASIS ANDROID PADA BAGIAN KEAMANAN UNIVERSITAS BUDI LUHUR DENGAN ALGORITMA *STATIC CLASSIFIER* DAN *ADAPTIVE CLASSIFIER*

Indra¹, Mufti², Nanang Wahyu Hidayat³

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Kebayoran Lama, Jakarta Selatan, 12260
Telp. (021) 5853753, Fax (021) 5866369
¹indra@budiluhur.ac.id, ²muftyhayat@gmail.com, ³11011503354@student.budiluhur.ac.id

ABSTRAK

Pertumbuhan jumlah kendaraan bermotor saat ini sudah sangat pesat. Data mengenai setiap kendaraan sangatlah berguna untuk berbagai keperluan keamanan sampai analisa lalu lintas. Setiap kendaraan memiliki plat nomor sebagai tanda pengenal, dimana plat nomor dapat menunjukkan asal daerah dan juga informasi data kepemilikan. Informasi data kepemilikan biasanya terdapat pada kepolisian, data yang disimpan tersebut dapat dimanfaatkan apabila terjadi kejadian yang melibatkan suatu kendaraan seperti pencurian. Pencarian informasi secara manual akan menghabiskan waktu sehingga dengan memanfaatkan teknologi *Optical Character Recognition (OCR)* pencarian data kepemilikan dapat dilakukan lebih cepat. Pada penelitian ini dibuat sebuah aplikasi pengenalan dan identifikasi plat nomor kendaraan berbasis android dengan algoritma *static classifier* dan *adaptive classifier*. Kedua algoritma tersebut pada dasarnya memiliki cara kerja yang sama, namun menggunakan data training yang berbeda. Aplikasi ini diharapkan dapat membantu proses pencarian data kepemilikan plat nomor kendaraan dengan cepat dan akurat.

Kata kunci : plat nomor kendaraan, *static classifier*, *adaptive classifier*, java

1. PENDAHULUAN

Plat nomor kendaraan selain menunjukkan asal daerah juga dapat diketahui identitas pemilik kendaraan tersebut. Informasi data kepemilikan biasanya tercatat di kepolisian setempat, sehingga apabila suatu kejadian yang melibatkan suatu kendaraan, informasi kepemilikan kendaraan tersebut dapat dilihat. Seiring dengan banyaknya pemilik kendaraan, dan apabila pencarian data kepemilikan dilakukan secara manual maka akan menghabiskan waktu. dengan memanfaatkan teknologi komputer dan pengenalan karakter optik, proses pencarian dapat dilakukan dengan cepat, oleh karena itu penelitian ini akan menghasilkan aplikasi pengenalan dan identifikasi plat nomor kendaraan berbasis pengenalan karakter optik. Aplikasi ini diharapkan dapat membantu petugas keamanan untuk mengidentifikasi plat nomor kendaraan untuk mengetahui pemilik kendaraan yang melakukan pelanggaran.

Pengenalan karakter optik atau *optical character recognition* merupakan teknik untuk mengubah teks non digital menjadi teks digital. Mengingat teknologi yang sedang berkembang saat ini adalah teknologi *mobile*, oleh karena itu pengenalan plat nomor kendaraan ini akan dikembangkan pada perangkat yang menggunakan platform Android.

Aplikasi pengenalan plat nomor kendaraan ini akan memanfaatkan *library Tesseract OCR Engine*. *Tesseract OCR Engine* memiliki akurasi rata-rata mencapai 61% untuk gambar berwarna dan 70% untuk gambar *grayscale*. Rata-rata waktu yang di perlukan *Tesseract OCR Engine* untuk memproses satu

gambar adalah 1 detik untuk gambar berwarna dan 0.82 detik untuk gambar *grayscale*. Dengan pertimbangan diatas maka penelitian ini akan menggunakan algoritma *static classifier* dan *adaptive classifier* yang diterapkan oleh Tesseract.

Tujuan penelitian ini adalah untuk mengetahui tingkat akurasi dari algoritma *static classifier* dan *adaptive classifier*. dan seberapa baik algoritma tersebut mengenali karakter dibandingkan dengan algoritma dari penelitian-penelitian sebelumnya. Dengan adanya penelitian ini maka diharapkan user dapat mengenali dan mengidentifikasi plat nomor kendaraan secara cepat dan akurat.

2. TINJAUAN PUSTAKA

2.1. Tanda Nomor Kendaraan Bermotor

Tanda Nomor Kendaraan Bermotor (TNKB), atau sering kali disebut plat nomor atau nomor polisi (nopol) adalah plat aluminium tanda kendaraan bermotor di Indonesia yang telah didaftarkan pada kantor bersama samsat. Penggunaan tanda nomor kendaraan bermoto di Indonesia, terutama di Jawa, merupakan warisan jaman Hindia Belanda yang menggunakan kode wilayah berdasarkan pembagian wilayah karesidenan [1].

2.2. Tesseract OCR Engine

Tesseract adalah *engine OCR open source* yang awalnya dikembangkan oleh HP (Hewlett-Packard) antara tahun 1984 dan 1994. Tesseract berawal sebuah proyek penelitian PhD di HP Labs, Bristol oleh Ray Smith. Setelah penelitian bersama

antara HP Labs di Bristol dan divisi Scanner HP di Colorado, Tesseract secara signifikan memimpin dalam hal akurasi atas mesin komersial.

Tahap perkembangan berikutnya berlangsung di HP Labs Bristol sebagai investigasi OCR. Pada tahun 1994, pengembangan berhenti sepenuhnya. Mesin ini dikirim ke UNLV (University Nevada Las Vegas) pada tahun 1995 untuk menjalani tes akurasi tahunan OCR.

2.3. Grayscale Gambar

Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu[2]. Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau, dan biru mempunyai intensitas yang sama.

2.4. Penghalusan Gambar

Sampai saat ini, banyak metode yang telah dicoba untuk mengurangi banyaknya derau pada citra digital dengan tujuan untuk memperbaiki kualitas citra (Image Enhancement). Reduksi derau dapat dilakukan dengan melakukan penapisan / *filtering*. Salah satu metode yang dapat digunakan dengan metode konvolusi. Konvolusi sangat berguna untuk melakukan operasi penapisan (*filtering*) pada citra. Banyak tapis yang diimplementasikan dalam bentuk kernel yang dikonvolusikan dengan citra dengan tujuan untuk perbaikan kualitas citra, diantaranya adalah lolos bawah (*Low Pass Filter*), lolos atas (*High Pass Filter*), Median, Mean dan Gaussian [3].

2.5. Binerisasi Gambar

Gambar biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Secara umum proses binerisasi gambar *grayscale* untuk menghasilkan gambar biner adalah sebagai berikut :

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (1)$$

Dimana $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$ dan T menyatakan nilai ambang atau threshold [4]. Untuk menentukan nilai T atau nilai threshold, penelitian ini menggunakan metode otsu.

2.6. Algoritma Static Classifier dan Adaptive Classifier

Dalam proses pengenalan, Tesseract menggunakan dua *character classifier* yaitu *static classifier* dan *adaptive classifier*. Pada dasarnya *static classifier* dan *adaptive classifier* memiliki cara kerja yang sama, hal yang membedakan antara keduanya adalah hanya pada *data training* yang digunakan. *static classifier* menggunakan *data training* berupa 20 sampel dari 94 karakter dari 8 font dan dengan 4 atribut seperti normal, bold, italic, bold italic. Maka total seluruhnya adalah 60160 *training samples*[5]. Sedangkan *adaptive classifier* menggunakan *data training* dari karakter yang telah dikenali dari *static classifier*.

Proses pengenalan sangat bergantung pada *feature* yang diekstrak. *Feature* adalah komponen dari *polygonal*

approximation yang berasal dari sebuah *outline* suatu karakter. Pada proses *training*, *feature* membentuk vektor 4 dimensi (x, y, arah, panjang) yang berasal dari setiap elemen *polygonal approximation* dan akan dikelompokkan menjadi *prototype*. pada proses pengenalan, elemen-elemen *polygon* dipecah menjadi bagian-bagian yang lebih pendek dengan panjang yang sama, sehingga dimensi panjang dieliminasi dari *vektor feature* [6].

2.7. Penelitian Sebelumnya

Penelitian yang dilakukan pada tahun 2012 menjelaskan bahwa pengenalan plat nomor kendaraan secara otomatis telah menjadi suatu aplikasi yang sangat penting dalam bidang *image processing*. Sistem pengenalan plat ini menerima input citra (*image*) selanjutnya akan dideteksi letak plat nomor serta dikenali secara otomatis karakter angka dan huruf pada plat nomor kendaraan roda empat tersebut, sehingga diharapkan aplikasi dapat mengenali plat nomor tersebut dengan tingkat akurasi yang cukup baik. Penelitian tersebut mengembangkan program yang dapat mendeteksi dan mengenali plat nomor kendaraan roda empat menggunakan algoritma ekstraksi kontur dan OCR dengan tingkat akurasi sebesar 82%. Program tersebut memiliki kehandalan seperti waktu pendeteksian plat nomor beserta pengenalan karakter angka dan huruf yang ada pada plat nomor dapat dikenali dengan cukup cepat dengan waktu rata-rata pengenalan sebesar 227,3047 mili detik. Namun program tersebut juga memiliki kekurangan seperti bila gambar agak jauh, program masih belum dapat mendeteksi plat nomor dan bila posisi gambar miring lebih dari 45 derajat, program masih belum dapat menangkap dan mengenali plat nomor kendaraan dengan baik[7].

Kemudian pada tahun yang sama dikembangkan suatu teknik pengenalan plat nomor kendaraan berbasis teknik pengolahan citra dan jaringan syaraf tiruan *backpropagation*. pengembangan sistem dan pengujian plat nomor kendaraan secara off-line tersebut menghasilkan akurasi sebanyak 96% yang diikuti dalam tahap pelatihan dan 90% dari data citra pada nomor yang tidak diikuti dalam tahap pelatihan. Dari hasil pengujian data citra plat nomor kendaraan didapatkan bahwa sistem beberapa kali tidak dapat mengenali beberapa huruf. Kesalahan tersebut dapat terjadi diantaranya karena beberapa faktor seperti usia plat nomor kendaraan yang diujikan, pengaruh pengolahan citra dan algoritma pelatihan jaringan serta kesamaan sifat dari karakter (huruf atau angka) yang dibaca[8].

Masih pada tahun 2012 sebuah laporan penelitian menjelaskan tentang sistem pendeteksi plat nomor kendaraan menggunakan pengolahan citra. Citra dari kendaraan diolah sedemikian rupa sehingga hanya diperoleh plat nomor saja untuk kemudian dikenali karakternya. Pengenalan plat dilakukan melalui beberapa proses yaitu *preprocessing*, mendeteksi posisi plat, segmentasi, dan pengenalan. Sistem tersebut diujicobakan pada 20 citra uji dengan rincian lima citra dikenali, 15 citra berhasil dikenali dengan rata-rata tingkat akurasi pengujian sebesar 92,80% untuk sepuluh citra yang

diambil dari sisi depan dan sebesar 73,85% untuk lima citra yang diambil dari sisi belakang[9].

Berbagai metode telah digunakan dalam penelitian pengenalan plat nomor seperti *Fuzzy* dan *JST*. Sebuah penelitian mengembangkan sistem pengenalan karakter menggunakan metode *PCA*. *Principal Component Analysis* adalah suatu metode untuk merepresentasikan suatu objek di mana ditekankan pada proses ekstraksi fitur. Dari hasil analisis data menunjukkan sistem pengenalan plat nomor kendaraan dalam ini memiliki kemampuan segmentasi yang baik pada jarak 1 sampai 2m, sudut 30o, 15 o, 0 o, -15o, -30 o serta ketinggian 25-150cm. Sedangkan untuk kemampuan pengenalan yang baik terlihat pada jarak 1m. Sistem pengenalan plat nomor kendaraan ini mampu diaplikasikan dengan kondisi optimal pada jarak 1 m, sudut antara -150 sampai 150 dan ketinggian 25-100cm dengan tingkat keterkenalan 80%-100%[10].

3. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini adalah:

a. Menerima Input Gambar atau Citra

Pada tahap ini aplikasi akan menerima input sebuah gambar RGB, gambar ini dapat dipilih oleh *user* melalui aplikasi kamera dan *file chooser*. Setelah *user* memilih gambar maka akan dilakukan proses *cropping*.

b. Mengkonversi Gambar RBG menjadi Gambar Grayscale

Gambar yang telah dipilih dan di-*crop* selanjutnya akan melalui proses konversi menjadi gambar *grayscale*. Proses ini akan merubah sebuah gambar menjadi abu-abu.

Pada dasarnya proses ini dilakukan dengan meratakan nilai piksel dari 3 nilai RGB menjadi 1 nilai. Tetapi karena mata manusia lebih sensitif terhadap warna hijau daripada warna merah atau biru maka proses konversi ini menggunakan persamaan (1) :

$$Gray = 0.2126R + 0.7152G + 0.0722B \quad (2)$$

Dimana *Gray* adalah nilai kecerahan suatu piksel pada citra abu-abu, dengan persentasi 21% dari warna merah atau red (R), 71% dari warna hijau atau green (G), dan 7% dari warna biru atau blue (B).

c. Proses Penghalusan Gambar

Proses selanjutnya adalah penghalusan. Tahap ini berfungsi untuk mereduksi atau mengurangi *noise* pada gambar yang mungkin akan mengganggu proses pengenalan. Reduksi derau dapat dilakukan dengan melakukan penapisan / *filtering*.

Pada umumnya analisa suatu citra dalam domain frekuensi didasarkan pada teknik konvolusi. Keluaran dari sebuah sistem linear dapat diperoleh dari operasi konvolusi antara respon impuls sistem dengan sinyal masukan. Operasi konvolusi dilakukan dengan menggeser kernel konvolusi piksel per piksel, menghitung piksel keluaran, lalu menyimpannya dalam matriks baru. Konvolusi sangat berguna untuk melakukan operasi penapisan (*filtering*) pada citra. Pada pengolahan citra digital, konvolusi dilakukan secara dua

dimensi pada sebuah citra, ilustrasi nya adalah sebagai berikut :



Gambar 1 : Ilustrasi Konvolusi

berdasarkan ilustrasi diatas maka persamaan yang digunakan adalah sebagai berikut :

$$h(x,y) = f(x,y) * g(x,y) \quad (3)$$

$$= A.(x - 1, y - 1) + B.(x, y - 1) + C.(x + 1, y - 1) + D.(x - 1, y) + E.(x, y) + F.(x + 1, y) + G.(x - 1, y + 1) + H.(x, y + 1) + I.(x + 1, y + 1) \quad (4)$$

dimana *f(x,y)* adalah citra asal, *g(x,y)* adalah matriks konvolusi atau kernel dan *h(x,y)* adalah citra hasil konvolusi. Kernel yang digunakan untuk proses *Gaussian blur* adalah sebagai berikut :

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

d. Proses Binerisasi Gambar

Tahap ini merupakan proses terpenting, karena kesuksesan proses pengenalan sangat bergantung pada keberhasilan proses binerisasi. Proses binerisasi akan merubah gambar menjadi hitam putih. Metode yang digunakan adalah metode *otsu*. Tujuan dari metode *otsu* adalah untuk membagi histogram citra gray level kedalam dua daerah yang berbeda secara otomatis tanpa membutuhkan bantuan user untuk memasukan nilai *threshold* atau nilai ambang. nilai ambang *k* dapat ditentukan dengan persamaan sebagai berikut :

$$k = W_b W_f (\mu_b - \mu_f)^2 \quad (5)$$

Dengan :

W_b menyatakan nilai bobot dari *background*

W_f menyatakan nilai bobot dari *foreground*

μ_b menyatakan nilai *mean* dari *background*

μ_f menyatakan nilai *mean* dari *foreground*

e. Proses Pengenalan

Pada tahap ini aplikasi akan mencoba untuk mengenali karakter yang terdapat didalam sebuah gambar. Algoritma *static classifier* dan *adaptive classifier* akan digunakan pada tahap ini. Hasil dari proses pengenalan adalah sebuah karakter yang dapat dikenali.

Proses klasifikasi beroperasi dalam dua tahap. Tahap pertama disebut *class pruner* yang mengelompokkan karakter kedalam *short-list* menjadi 1 sampai 10 karakter. Proses tersebut menggunakan metode yang berkaitan dengan *Locality Sensitive Hashing (LSH)*. Tahap terakhir adalah menghitung *distance* dari karakter yang akan dikenali dengan *prototype* yang termasuk kedalam *short-list*. Pada tahap ini *classifier* menghitung *distance (d_f)* dari setiap *feature* yang berdekatan dengan *prototype*. *distance* kuadrat *eculidean (d)* adalah koordinat *(x,y)* *feature* dari *prototype*. ditambah dengan *weight (w)* dan perbedaan *angle (θ)* dari *prototype*. persamaannya adalah sebagai berikut :

$$d_f = d^2 + w\theta^2 \quad (6)$$

Keterangan :

d_f menyatakan *distance*

d^2 menyatakan kuadrat eculidean pada *distance*

w menyatakan *weight* dari suatu *feature prototype*

θ menyatakan *angle* atau sudut dari *feature prototype*

Setelah *distance* ditemukan, maka proses selanjutnya adalah mengkonversi *distance* menjadi *feature evidence* (E_f).

$$E_f = \frac{1}{1+kd_f^2} \quad (7)$$

Keterangan :

E_f menyatakan *feature evidence*

d_f menyatakan *distance*

k menyatakan nilai konstan

Konstan (k) digunakan untuk mengontrol *rating* karena *evidence* rusak karena *distance*. Sebagai fitur yang cocok dengan *prototype*, *feature evidence* (E_f) disalin ke *prototype* (E_p). Kemudian akan hasilnya dinormalisasi dengan jumlah *feature* dan jumlah panjang *prototype* (L_p), dan hasil seluruhnya dikonversi menjadi *distance* (Ray Smith, Daria Antonova, Dar-Shyang Lee 2009). Berikut ini persamaan nya :

$$d_{final} = 1 - \frac{\sum_f E_f + \sum_p E_p}{N_f + \sum_p L_p} \quad (8)$$

Keterangan :

d_{final} menyatakan *distance* akhir

E_f menyatakan *feature evidence*

E_p menyatakan *prototype evidence*

N_f menyatakan jumlah *feature*

L_p menyatakan panjang *prototype*

f. Menampilkan Data Mahasiswa

Setelah karakter yang telah dikenali sudah didapatkan. Maka proses terakhir adalah menampilkan data mahasiswa. Pada tahap ini data mahasiswa akan tertampil berdasarkan karakter plat nomor yang dikenali.

4. REKAYASA APLIKASI

4.1. Analisa Masalah

Data mengenai setiap kendaraan sangatlah berguna untuk berbagai keperluan dari keamanan sampai ke analisa lalu lintas. Untuk mempermudah pengambilan dan pengelolaan data kendaraan bermotor yang ada maka masing-masing kendaraan ini diberikan plat nomor kendaraan yang unik dan merupakan *identification key* untuk kendaraan bermotor tersebut. Walaupun demikian, karena banyak nya jumlah kendaraan yang ada, sangatlah sulit untuk mendata secara manual setiap plat nomor yang ada.

Dengan banyaknya jumlah kendaraan saat ini membuat lahan parkir cepat penuh. Keadaan ini terkadang membuat pengendara kendaraan bermotor tidak tertib dalam berparkir. Selain itu, hal ini bisa dimanfaatkan untuk tindak kejahatan seperti pencurian kendaraan bermotor. dengan cara manual, proses identifikasi yang melibatkan kendaraan bermotor akan

membutuhkan waktu yang tidak sedikit. Permasalahan yang muncul adalah kesulitan mengidentifikasi dan mengenali secara cepat dan akurat plat nomor kendaraan tersebut.

4.2. Penyelesaian

Penggunaan sistem secara otomatis pada identifikasi dan pengenalan plat nomor kendaraan bermotor dapat menjadi solusi dari kekurangan-kekurangan sistem manual saat ini. Pendeteksian dan pengenalan plat nomor kendaraan menjadi salah satu aplikasi yang sangat penting dalam meningkatkan suatu sistem keamanan. Oleh karena itu, akan dilakukan penelitian mengenai teknik pengenalan karakter yang dapat mendukung terciptanya sistem yang dapat menerima masukan data dari plat nomor kendaraan. Data tersebut akan diperoleh melalui kamera yang dapat menangkap gambar dari plat nomor kendaraan. Melalui citra atau gambar plat nomor kendaraan tersebut akan dilanjutkan dengan pemrosesan secara otomatis oleh sistem ini. Namun, sistem ini cenderung hanya berlaku untuk wilayah tertentu atau spesifik, karena variasi plat nomor kendaraan dari masing-masing daerah atau Negara mungkin saja berbeda.

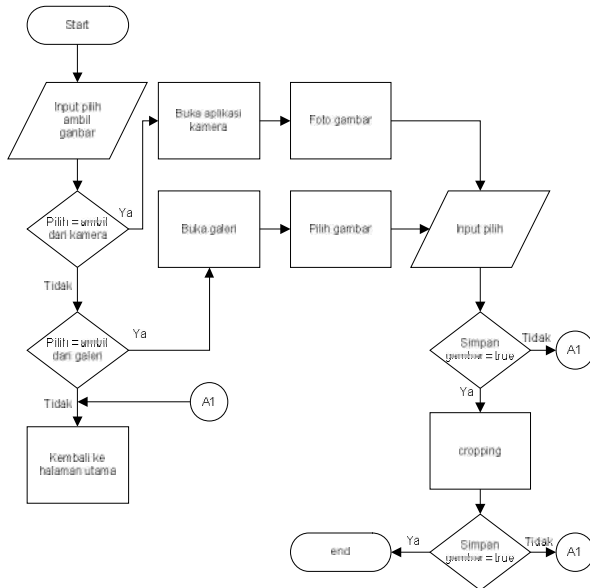
4.3. Perancangan Aplikasi

Rancangan antarmuka (*User Interface*) da menu pada Android menggunakan XML. Antarmuka dirancang dengan menggunakan objek-objek XML seperti *LinearLayout*, *RelativeLayout*, *TextView*, *EditText*, *ImageView* dan *Button*. Antarmuka pengguna pada program ini terdiri dari halaman utama, halaman hasil pengenalan, halaman cari plat, halaman mahasiswa, halaman bantuan dan halaman tentang.

4.4. Proses Sistem

a. Proses Pengambilan Gambar dari Kamera dan File Chooser

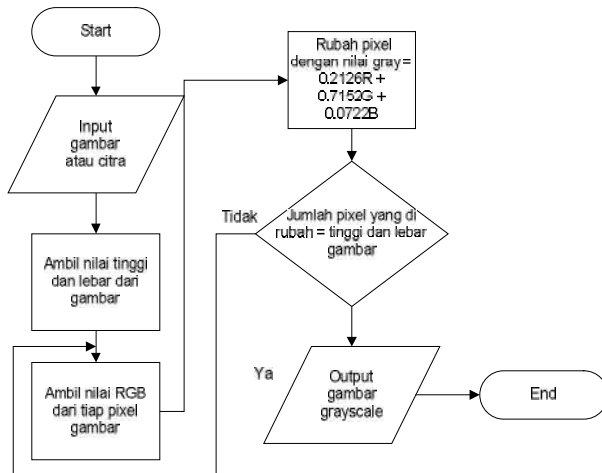
Pengambilan gambar dapat dilakukan melalui kamera dan *file chooser*. Pilihan pertama untuk mendapatkan gambar yang diinginkan yaitu dengan menggunakan kamera untuk *capture* image yang berisi plat nomor kendaraan. Program ini memanfaatkan kamera yang terdapat didalam *device* android secara *default*. Dengan cara ini semua fitur yang terdapat pada aplikasi kamera android dapat dimanfaatkan. Pilihan kedua adalah menggunakan *File Chooser* atau *Browse file*. Baik pilihan pertama atau pun kedua sebuah gambar yang sudah dipilih akan melalui proses *cropping* terlebih dahulu. Proses *cropping* diperlukan agar hanya bagian yang di-*crop* saja yang akan diproses. Bagian gambar yang di-*crop* adalah bagian gambar yang diketahui sebagai plat nomor kendaraan saja. Proses ini akan mengurangi waktu pengenalan karakter karena tidak seluruh gambar yang diproses ke proses selanjutnya dan dapat membantu meningkatkan akurasi dari proses pengenalan karakter plat nomor kendaraan. Berikut ini adalah *flowchart* dari proses pengambilan gambar melalui kamera dan *file chooser*:



Gambar 2 : Flowchart Proses Pengambilan Gambar dari Kamera dan File Chooser

b. Proses Grayscale Gambar

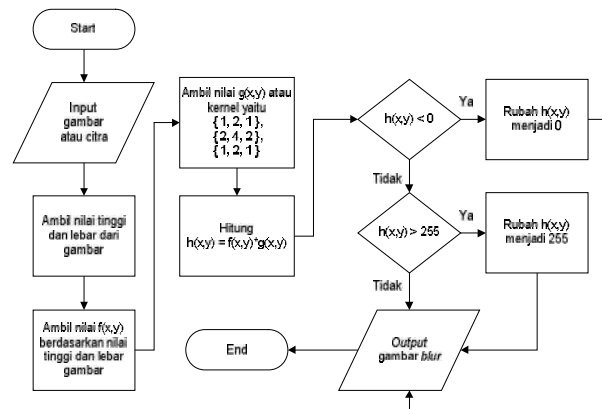
Pada proses pengkonversian gambar grayscale, program akan mengambil nilai tiap pixel dari gambar dengan melakukan proses scan gambar per-pixel pada koordinat (x,y) sebesar ukuran panjang dan lebar gambar. Setiap pixel pada koordinat (x,y) akan diubah menjadi warna keabuan dengan nilai grayscale. Metode yang digunakan adalah *luminosity*. metode ini mengambil nilai rata-rata tiap pixel namun nilai untuk warna hijau lebih tinggi dikarenakan mata manusia lebih sensitif terhadap warna hijau dibandingkan warna lain. derajat keabuan pada masing-masing pixel tergantung pada nilai RGB pada pixel tersebut. Dibawah ini adalah *flowchart* dari proses grayscale:



Gambar 3 : Flowchart Proses Grayscale Gambar

c. Proses Penghalusan Gambar

Penghalusan gambar berfungsi sebagai *noise filtering*. Penghalusan gambar pada penelitian ini menggunakan *Gaussian Blur* dengan metode matriks konvolusi 3x3. Penghalusan gambar bertujuan untuk menghaluskan gambar dan mereduksi atau mengurangi *noise* pada gambar sehingga diharapkan dapat meningkatkan persentasi akurasi dari program pengenalan plat nomor kendaraan ini. Konvolusi merupakan proses perkalian antara dua fungsi yaitu $f(x,y)$ dan $g(x,y)$ yang akan menghasilkan $h(x,y)$. dimana $f(x,y)$ merupakan nilai pixel dari gambar atau citra dan $g(x,y)$ merupakan kernel. Kernel merupakan sebuah filter yang berupa matriks yang umumnya berukuran 3x3 yang berfungsi untuk menghasilkan pixel yang baru berdasarkan hasil perkalian dengan $f(x,y)$. Dibawah ini merupakan *flowchart* dari proses penghalusan gambar:



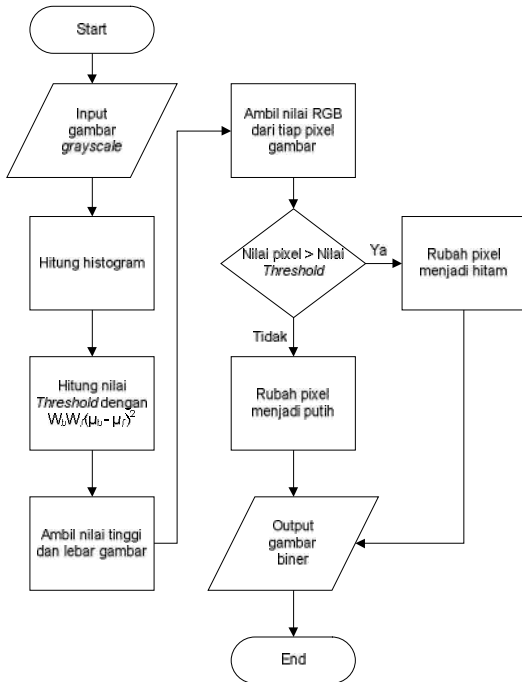
Gambar 4 : Flowchart Penghalusan Gambar

d. Proses Binerisasi Gambar

Binerisasi gambar adalah proses untuk merubah gambar menjadi hitam putih. Gambar ini yang akan diproses untuk pengenalan. Binerisasi gambar adalah sebuah proses untuk mengubah gambar grayscale menjadi gambar yang hanya mempunyai dua kemungkinan yaitu 0 untuk hitam dan 255 untuk putih. Proses binerisasi gambar merupakan bagian terpenting dalam pengenalan karakter karena keberhasilan binerisasi gambar sangat menentukan akurasi dan tingkat keberhasilan suatu proses pengenalan karakter. Secara umum proses binerisasi citra grayscale untuk menghasilkan citra biner adalah sebagai berikut.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (5)$$

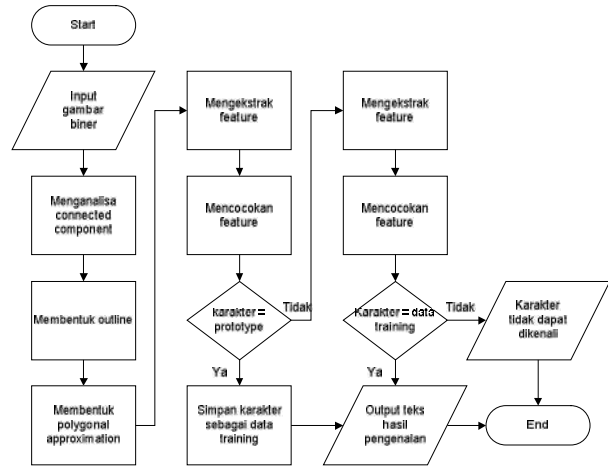
Dimana $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$ dan T menyatakan nilai ambang atau thresholding. Aplikasi ini menggunakan metode otsu untuk melakukan proses binerisasi. Metode otsu diciptakan oleh Nobuyuki Otsu. Berikut ini adalah *flowchart* dari proses binerisasi gambar:



Gambar 5: Flowchart Proses Binerisasi Gambar

e. Proses Pengenalan Karakter

Proses pengenalan pada tesseract dimulai dengan menganalisa *connected component* pada sebuah gambar. *Connected Component Analysis* mampu menandai setiap kemungkinan karakter yang terdapat dalam gambar inputan, yang sudah menjadi gambar biner tentunya. Dengan dilakukan metode ini maka *outline* dari sebuah karakter dapat diketahui. Kemudian *outline* karakter akan terbentuk berdasarkan *connected component*, dan *outline* yang terbentuk akan dirubah menjadi *polygonal approximation*. *Polygonal approximation* digunakan untuk menghemat waktu karena mengurangi sudut atau *angle* yang terdapat pada sebuah karakter. Sudut tersebut nantinya yang akan dijadikan sebagai salah satu *feature* oleh *classifier*. Proses selanjutnya adalah *static classifier*. *static classifier* akan mengekstrak *feature* dari *polygonal approximation*, *feature* yang diekstrak berupa 3 dimensi dari posisi x, posisi y, dan *angle*. *Feature* ini yang akan dilakukan perbandingan *prototype*. *Prototype* sendiri merupakan *data training* yang digunakan pada proses *static classifier*. Sebuah karakter yang memiliki kecocokan dengan *prototype* maka akan dikirim ke *adaptive classifier* sebagai *data training*. Dengan ini maka *prototype* pada *adaptive classifier* merupakan *output* dari *static classifier*. Dengan cara ini maka karakter yang gagal dikenali pada proses *static classifier* dapat dilanjutkan ke *adaptive classifier* untuk dilakukan proses pengenalan kembali. Berikut ini adalah *flowchart* dari proses pengenalan:



Gambar 6 : Flowchart Proses Pengenalan Karakter

5. HASIL DAN PEMBAHASAN

5.1. Implementasi Program

a. Perangkat Keras (*Hardware*)

- 1) Qualcomm APQ8064 Snapdragon
- 2) Quad-core 1.5 GHz Krait
- 3) Adreno 320
- 4) Monitor
- 5) 16 GB storage, 2 GB RAM

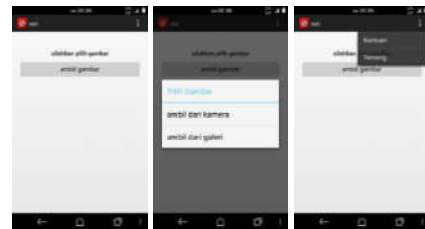
b. Perangkat Lunak (*Software*)

- 1) Android OS, v4.3 (Jelly Bean)
- 2) Java Android
- 3) JDK 1.6
- 4) SQLite

5.2. Tampilan layar

a. Tampilan Halaman Utama

Gambar dibawah ini adalah tampilan halaman utama. halaman ini memiliki sebuah *button* dan sebuah *menu* yang akan menampilkan halaman sesuai dengan yang *user* pilih. *Button* ambil gambar memiliki pilihan ambil dari kamera dan ambil dari galeri. *Button* ini berfungsi untuk mengambil atau memilih gambar yang akan dijadikan objek untuk pengenalan. Selain itu terdapat juga *menu* yang berfungsi menampilkan halaman bantuan dan halaman tentang.



Gambar 7 : Tampilan Halaman Utama

b. Tampilan Halaman Hasil Pengenalan

Dibawah ini merupakan tampilan halaman hasil pengenalan



Gambar 8 : Tampilan Halaman Hasil Pengenalan

c. Tampilan Halaman Cari Plat

Dibawah ini merupakan tampilan halaman cari plat.



Gambar 9 : Tampilan Halaman Cari Plat

d. Tampilan Halaman Mahasiswa

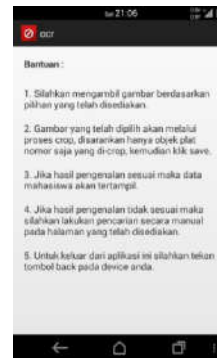
Halaman mahasiswa berfungsi menampilkan data mahasiswa berdasarkan pencarian yang dilakukan pada halaman cari plat. Sesuai dengan gambar dibawah ini, data mahasiswa dapat tertampil beserta foto dan informasi singkat mengenai kendaraan yang digunakan mahasiswa tersebut.



Gambar 10 : Tampilan Halaman Mahasiswa

e. Tampilan Halaman Bantuan

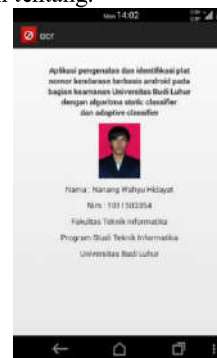
Gambar dibawah ini merupakan halaman bantuan. Halaman ini berfungsi untuk memberikan informasi cara penggunaan program aplikasi pengenalan plat nomor kendaraan ini.



Gambar 11 : Tampilan Halaman Bantuan

f. Tampilan Halaman Tentang

Halaman tentang berfungsi menampilkan profil singkat mengenai pembuat aplikasi ini. Gambar dibawah ini merupakan halaman tentang.



Gambar 12 : Tampilan Halaman Tentang

6. KESIMPULAN

Adapun kesimpulan yang diperoleh dari perancangan, pembuatan, serangkaian uji coba dan analisa program aplikasi pengenalan plat nomor kendaraan ini, maka dapat dibuat suatu kesimpulan antara lain:

- Tingkat akurasi program aplikasi pengenalan plat nomor kendaraan ini mencapai 96.25%. relatif lebih akurat dibandingkan dengan aplikasi yang menggunakan algoritma ekstraksi kontur yang memiliki tingkat akurasi rata-rata 82% dan sedikit lebih akurat dibandingkan dengan metode jaringan syaraf tiruan *backpropagation* yang memiliki tingkat akurasi 96%.
- Algoritma manapun yang digunakan masih sangat bergantung pada kualitas gambar yang dijadikan input. Contohnya apabila plat nomor rusak maka proses pengenalan menjadi tidak akurat.

DAFTAR PUSTAKA

- [1] Wikipedia, Tanda Nomor Kendaraan Bermotor (*online*), dilihat 12 Oktober 2013, <http://id.wikipedia.org/wiki/Tanda_nomor_kendaraan_bermotor>
- [2] Ruslianto, Ikhwan, Harjoko, Agus. (2011). *Pengenalan Karakter Plat Nomor Mobil Secara Real Time*. Yogyakarta: FMIPA UGM.
- [3] Novita, Rika Wardhani, Kartika, Mera Delimayanti. *Analisis Penerapan Metode Konvolusi untuk Reduksi Derau pada Citra Digital*. Jakarta: Politeknik Negeri Jakarta.
- [4] Putra, Darma. (2004). *Binerisasi Citra Tangan Dengan Metode Otsu*. Universitas Udayana.
- [5] Smith, Ray. (2007). *An Overview of the Tesseract OCR Engine*. Google Inc.
- [6] Aprilia, Priska. (2012). *Perancangan Aplikasi Pengenalan Karakter Korea pada Platform Android menggunakan Metode OCR Adaptive Classifier*. Jakarta: Universitas Bina Nusantara.
- [7] Setiadi, Himawan. (2012). *Perancangan Program Deteksi dan Pengenalan Plat Nomor Kendaraan Menggunakan Algoritma Ekstraksi Kontur dan OCR*. Jakarta: Universitas Bina Nusantara.
- [8] Fitriawan, Helmy, Pucu, Ouriz, Baptista, Yohanes. (2012). *Identifikasi Plat Nomor Kendaraan Secara Off-line Berbasis Pengolahan Citra dan Jaringan Syaraf Tiruan*. Lampung: Universitas Lampung
- [9] Badruzzaman, Moh. Iqbal. (2012). *Sistem Pendeteksi Plat Nomor Kendaraan Menggunakan Pengolahan Citra*. Yogyakarta: Universitas Islam Indonesia.
- [10] Yulida, Selpha, Kusumawardhan, Apriani, Setijono, Heru. (2013). *Perancangan Sistem Pengenalan Plat Nomor Kendaraan Menggunakan Metode Principal Component Analysis*. Surabaya: ITS Surabaya Indonesia