

## IDENTIFIKASI PLAT NOMOR KENDARAAN BERMOTOR MENGGUNAKAN METODE MULTI-STEP IMAGE PROCESSING BERBASIS ANDROID

I Gede Susrama Masdiyasa<sup>1)</sup>, Sulianto Bhirawa<sup>2)</sup>, Slamet Winardi<sup>3)</sup>

<sup>1)</sup>Teknik Informatika, UPN Veteran Jawa Timur

<sup>2,3)</sup>Fakultas Ilmu Komputer, Universitas Narotama Surabaya

Email : [gdesusrama@gmail.com](mailto:gdesusrama@gmail.com), [sulianto.bhirawa@gmail.com](mailto:sulianto.bhirawa@gmail.com), [slamet.winardi@narotama.ac.id](mailto:slamet.winardi@narotama.ac.id)

### Abstrak

*Sistem plat nomor di Indonesia merupakan warisan dari Kolonial Hindia Belanda yang membagi berdasarkan wilayah karesidenan. Setelah berjalan sekian lama acuan tersebut masih digunakan hingga sekarang yang terdiri dari dua baris, yaitu baris pertama terdiri dari Kode Wilayah Karesidenan diikuti nomor polisi dari jenis kendaraan bermotor tersebut dan terakhir adalah kode wilayah dari karesidenan tersebut, baris kedua merupakan masa berlaku plat nomor tersebut yang terdiri dari bulan dan tahun. Plat nomor Kendaraan bermotor di Indonesia menggunakan bahan almunium dengan ketebalan 1mm dan mempunyai dimensi 250x105 mm untuk roda dua atau tiga sedangkan untuk roda empat atau lebih mempunyai dimensi 395x135 mm. Peruntukannya berbeda-beda sesuai dengan warna dasar dari plat nomor tersebut, warna yang ada pada plat nomor adalah hitam, merah, kuning, dan putih. Serta ada logo kepolisian sebagai hak cipta dan pemegang merk plat nomor di Indonesia. Keberadaan plat nomor tersebut memberikan identitas dari setiap kendaraan bermotor yang beredar di Indonesia, sehingga mudah mengenali kendaraan tersebut berasal dari wilayah mana sesuai dengan kode plat nomor yang tertera pada posisi depan dan belakang kendaraan bermotor. Identitas kendaraan tersebut kurang optimal penggunaannya untuk keperluan-keperluan tertentu yang membutuhkan identitas secara cepat dan akurat, sehingga untuk keperluan tersebut diperlukan sebuah ID digital yang dapat digunakan untuk identifikasi kendaraan bermotor secara cepat dan akurat. Salah satu teknologi digitalisasi plat nomor kendaraan yang sudah digunakan di beberapa negara yaitu ANPR (Automatic Number Plate Recognition). Teknologi ini menggunakan pengolahan citra digital untuk mengenali dan membaca plat nomor kendaraan. Gambar atau citra yang diolah didapatkan dari kamera pengawas dan juga kamera yang didesain khusus untuk proses pengolahan digital. Mengingat ANPR menggunakan sebuah algoritma dalam pengolahan citra digital, maka memungkinkan untuk mengembangkan perangkat lunak dengan kemampuan serupa namun dalam perangkat yang lebih terjangkau dan murah. Tentu saja juga memungkinkan untuk dikembangkan di perangkat smartphone berbasis sistem operasi Android. Hal ini dikarenakan smartphone memiliki kamera serta kemampuan komputasi yang dibutuhkan untuk pengolahan citra digital. Dengan metode Multi-step image processing ini mampu mengenali plat nomor kendaraan bermotor sampai dengan akurasi 95% dengan kamera minimal 5 megapixel yang dilakukan dalam tiga tahap yaitu threshold, contour dan kNearestNeighbors. Dalam pemindaian jarak menentukan kemampuan untuk mendeteksi segmen plat nomor dan pencahayaan mempengaruhi tingkat akurasi pengenalan karakter sedangkan untuk pengenalan plat nomor yang memiliki dasar warna terang harus dilakukan invers terlebih dahulu.*

**Kata kunci :** plat nomor, pengolahan citra digital, mobile application, android, multi-step image processing

### PENDAHULUAN

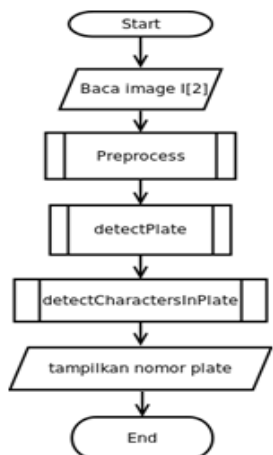
Pada tahun 2014, kendaraan bermotor di Indonesia dari berbagai jenis mencapai 114 juta unit (Biro Pusat Statistik). (Limantara, 2017) Dengan banyaknya kendaraan dan alat identifikasi yang hanya berupa plat nomor kendaraan menimbulkan beberapa masalah, misalnya proses pembuatan baru maupun perpanjangan masa berlaku nomor kendaraan yang memakan waktu yang lama, proses identifikasi kecelakaan kendaraan yang lama, pemalsuan plat nomor hingga identifikasi kasus

pencurian kendaraan yang terkendala pengelolaan data plat nomor kendaraan. (Slamet, Made, Tubagus, Agus, & Sri Wiwoho, 2017) Hal ini terjadi karena semua data nomor kendaraan terpusat dipihak berwenang. Tanda Nomor Kendaraan Bermotor atau sering disebut plat nopol (singkatan dari nomor polisi) adalah plat aluminium yang dipasang pada bagian depan dan belakang kendaraan bermotor sebagai nomor identifikasi kendaraan yang terdaftar di Kantor Bersama Samsat. Plat ini memiliki

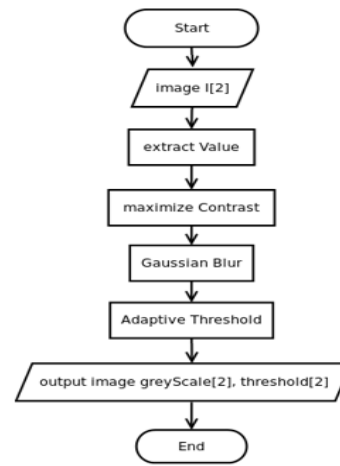
spesifikasi panjang 250 mm dan 105 mm untuk sepeda motor, sedangkan untuk mobil memiliki panjang 395 mm dan lebar 13.5 mm. Salah satu teknologi identifikasi yang sudah digunakan di beberapa negara yaitu ANPR (*Automatic Number Plate Recognition*). Teknologi ini menggunakan pengolahan citra digital untuk mengenali dan membaca plat nomor kendaraan. (Taufiq, Hidayatno, & Isnanto, 2012) Gambar atau citra yang diolah didapatkan dari kamera pengawas dan juga kamera yang didesain khusus untuk proses pengolahan digital. Menjalankan algoritma pengolahan citra digital membutuhkan komputasi yang kompleks. Komputasi yang kompleks secara otomatis membutuhkan *resources* yang besar. Kapasitas memory, kecepatan processor serta perhitungan yang kompleks mempengaruhi kinerja algoritma pengolahan citra digital. Oleh karena itu menjalankan algoritma pengolahan citra digital pada perangkat *smartphone* memerlukan banyak pertimbangan dan perhitungan.

### METODOLOGI PENELITIAN

Pada diagram alir berikut ini, dijelaskan bahwa aplikasi menggunakan 3 subprogram utama untuk membaca dan mengenali plat nomor kendaraan. Sedangkan untuk subprogram pemindaian dijalankan langsung oleh framework Android dan menghasilkan keluaran berupa image binary untuk selanjutnya dijadikan image masukan pada program.



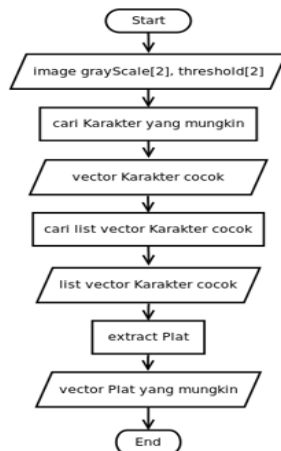
Gambar 1. Flowchart Program Utama



Gambar 2. Flowchart Preprocessing

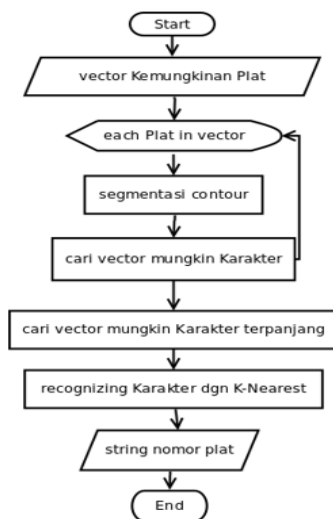
Pada subprogram Preprocessing, algoritma menghasilkan keluaran image dalam format Grayscale dan Threshold. Untuk menghasilkan keluaran tersebut terdapat beberapa langkah, pertama image dikonversi dari format RGB ke dalam format HSV kemudian diambil nilai Value dari image tersebut. Kedua, memaksimalkan kontras menggunakan transformasi morfologi. Ketiga, image difilter dengan Gaussian Blur untuk kemudian langkah terakhir yaitu Adaptive Threshold. Output dari subprogram ini selanjutnya diproses oleh subprogram Detect Plat.

Pada subprogram Detect Plat, dilakukan segmentasi pada image berdasarkan contour. Hasil segmentasi tersebut berupa *island* atau kelompok-kelompok segmen pada image. Segmen-segmen ini dikelompokkan lagi berdasarkan jarak dan ukuran kedalam satu *rectangle*.



Gambar 3. Flowchart Deteksi Plat Nomor

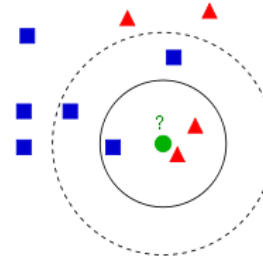
Untuk proses selanjutnya yaitu proses pengenalan atau *recognizing*. Untuk proses ini secara khusus peneliti menggunakan metode K-Nearest Neighbour. Secara umum, dibuat suatu set data berdasarkan sample data yang telah diketahui klasifikasinya. Set data ini merupakan *Training Set* atau data pelatihan. Kemudian berdasarkan data tersebut, program diharapkan untuk dapat memprediksi klasifikasi pada suatu data baru.



Gambar 4. Flowchart Deteksi Karakter

K-Nearest Neighbors atau k-NN adalah metode non-parametric untuk klasifikasi dan regresi. Untuk klasifikasi, output merupakan keanggotaan dalam suatu kelas. Sebuah objek diklasifikasikan berdasarkan *voting* atau suara terbanyak dari objek tetangga terdekatnya

yang paling umum diantara k tetangga terdekat, jika k=1 maka objek diklasifikasikan kepada tetangga terdekat tunggal.



Gambar 5. Ilustrasi Klasifikasi k-Nearest Neighbors

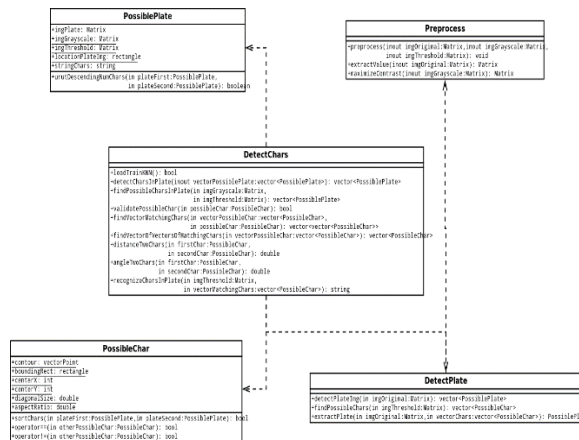
Pada gambar 5 diatas, objek lingkaran hijau harus diklasifikasikan baik untuk kelas pertama (kotak biru) maupun kelas kedua (segitiga merah). Jika k = 3 (pada lingkaran garis tebal) maka objek dikasifikasikan sebagai kelas kedua (segitiga merah) karena terdapat 2 kelas kedua dan 1 kelas pertama. Jika k = 5 (lingkaran garis putus-putus) objek diklasifikasikan sebagai kelas pertama karena terdapat 3 kelas pertama dan 2 kelas kedua.

Mengingat algoritma membutuhkan resource dan komputasi yang kompleks, maka digunakan bahasa pemrograman yang memiliki akses pada level memory (*low level language*) agar sumber daya yang digunakan lebih efisien. Untuk itu algoritma diimplementasikan menggunakan bahasa pemrograman C/C++ yang secara khusus dikemas dalam framework Android Native Development Kit (NDK).

## HASIL DAN PEMBAHASAN

Pada proses penerapan algoritma, dibagi menjadi beberapa subprogram terpisah untuk memudahkan identifikasi jika terdapat kesalahan, baik kesalahan alur program maupun kesalahan logika. Beberapa subprogram tersebut dalam bentuk diagram alir yang kemudian diterapkan ke dalam bentuk psuedocode sederhana. Namun, pengujian dari algoritma sendiri harus dilakukan dengan menggunakan *software* atau *script* mengingat dalam implementasi algoritma ini menggunakan banyak variable,

data sample dan perhitungan yang kompleks. Dalam hal ini, peneliti menggunakan *script* dengan bahasa pemrograman C/C++ pada perangkat laptop.



Gambar 6. Diagram Deteksi Plat Nomor

Dari diagram diatas dapat disimpulkan bahwa proses pengenalan plat nomor ditentukan oleh proses deteksi karakter. Namun, untuk optimasi proses deteksi karakter, maka diperlukan juga proses konversi dan thresholding. Jadi, image inputan dilakukan *Preprocessing*, langkah-langkah *Preprocessing* adalah sebagai berikut :

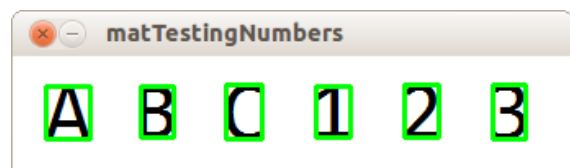
1. Konversi image ke format HSV;
2. Extract Value sebagai image baru dalam format Grayscale;
3. Menambahkan kontras Grayscale.
4. Operasi Gaussian Blur pada image Grayscale;
5. Operasi adaptive threshold dengan image blur sebagai masukan.

Dalam langkah *Preprocess* ini didapatkan output image Grayscale dan image Threshold. Tujuan *Preprocess* ini adalah untuk memaksimalkan contour dan segmentasi pada image. Contour ini nantinya akan memudahkan untuk pengenalan kemungkinan karakter angka dan huruf. Kemungkinan karakter angka dan huruf ini kemudian dikelompokkan menjadi kemungkinan plat nomor. Untuk setiap kemungkinan plat nomor ini kemudian dilakukan pengenalan karakter dan ekstraksi image.

Pada method KNN dilakukan pengujian apakah objek *kNearest* telah “terlatih” dengan mengujinya menggunakan sample data yang sudah diketahui klasifikasinya untuk kemudian dicocokkan dengan hasil klasifikasi *kNearest*.

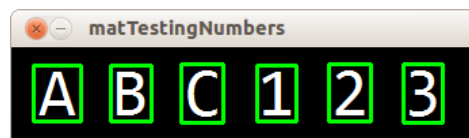
A B C 1 2 3

Gambar 7. Sample data untuk pengujian kNearest.



Gambar 8. Process recognition.

Pada pengujian diatas dihasilkan string hasil pembacaan pengenalan karakter yaitu “ABC123”. Dalam beberapa pengujian dengan sample data yang serupa, program dapat mengenali dengan baik. Namun kesalahan muncul ketika data sample pengujian diubah dengan variasi kebalikan dari data sample yang digunakan untuk *training*. Jika data sample memiliki background dengan warna gelap maka program mengenali karakter namun tidak akurat.



Gambar 9. Data pengujian memiliki warna background lebih gelap dari warna karakter.

Pada pengujian dengan data pengujian kebalikan atau *invert*, program mendeteksi bahwa karakter diatas adalah “RDC889”. Hal ini dikarenakan pada saat *training* data yang digunakan adalah data non-inverse atau warna karakter lebih gelap dari background. Untuk menyiasati ini, maka sebelum dilakukan proses pengenalan maka dilakukan operasi *invert* pada image. Operasi



invert sendiri pada OpenCV adalah operasi `threshold` dengan parameter `CV_THRESH_BINARY_INV`.

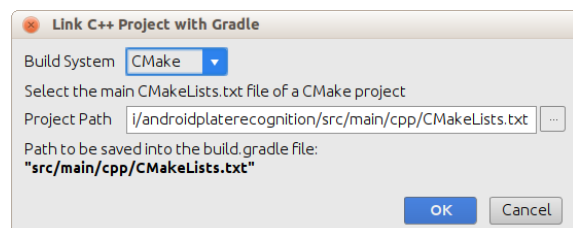
```
cv::threshold(imgOriginalScene,  
imgInverted,100, 255,CV_THRESH_BINARY_INV);
```

Output image yang disimpan pada variable `imgInverted` dengan `imgOriginalScene` sebagai image masukan. Nilai 100 adalah nilai `threshold` dan nilai 255 adalah nilai maximal. Dengan operasi diatas maka data pengujian yang merupakan data kebalikan diubah menjadi data yang sama dengan data *training*.

### Implementasi pada Android Studio

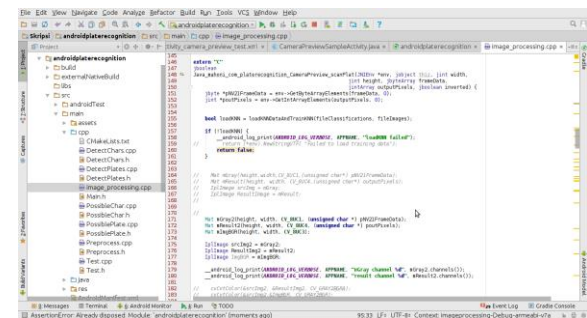
Mengingat penelitian ini bertujuan untuk melakukan komputasi yang kompleks dan menggunakan bahasa pemrograman *low-level* yaitu C/C++, maka pada saat implementasi pada Android Studio harus menggunakan *plugin* untuk mendukung native development yaitu CMake. CMake adalah sistem yang mengelola proses kompilasi pada sistem operasi yang tidak tergantung pada *compiler* tertentu. Dalam hal CMake membantu peneliti untuk melakukan kompilasi framework OpenCV menjadi *library* yang nantinya digunakan dalam aplikasi.

Langkah pertama yaitu membuat project pada Android Studio, kemudian dibuat deklarasi kelas-kelas yang sudah dirancang sebelumnya kedalam bentuk C/C++.(Chen & Luo, 2012) Secara default, project belum mendukung untuk development menggunakan native framework. Oleh karena itu, perlu dilakukan penyesuaian yaitu "Link C++ project With Gradle".



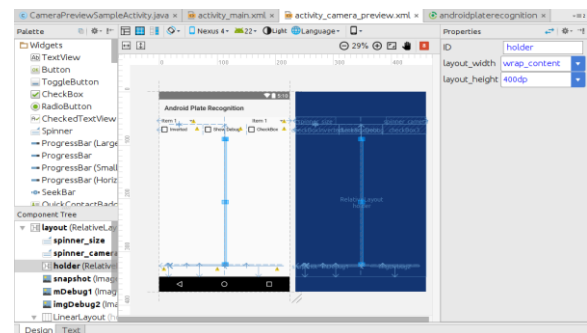
Gambar 10. Pop up window Link C++ Project With Gradle

Setelah project mendukung native development, maka secara otomatis semua source dalam format C/C++ dapat dikenali sebagai bagian dari project berdampingan dengan source Java dan XML yang merupakan source default.



Gambar 11. Tampilan project pada Android Studio.

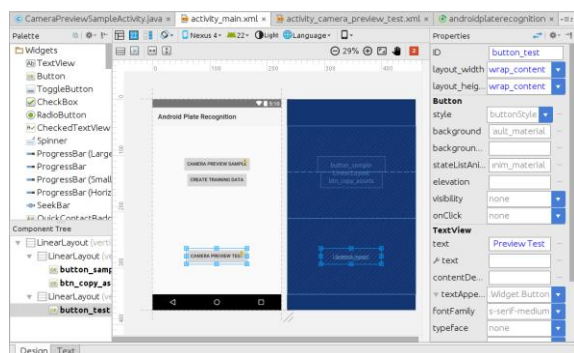
Langkah selanjutnya adalah membuat layout tampilan aplikasi. Aplikasi ini dibagi menjadi 3 *activity*. Dari 3 *activity* tersebut didesain 2 layout, 1 layout sebagai *main activity* dan 1 layout sebagai template scanner *activity*. Layout ini dibuat dengan XML untuk mendeklarasikan setiap komponen yang digunakan seperti *RelativeLayout*, *ImageView*, *Button* dan sebagainya.



Gambar 12. Layout untuk main activity.

Dalam layout *main activity* ini terdapat 3 button yaitu button untuk import data klasifikasi ke media penyimpanan perangkat, button untuk scanner plat nomor kendaraan dan button untuk scanner plat nomor kendaraan dengan variasi resolusi image yang dapat diubah. Layout berikutnya yaitu

*activity\_camera\_preview.xml*. Ini merupakan template untuk 2 activity yaitu scanner plat nomor dan scanner plat nomor dengan resolusi yang berbeda. Karena kedua *activity* tersebut memiliki fungsi utama yang sama maka layoutnya digabung menjadi satu namun beberapa komponen akan di non-aktifkan jika tidak digunakan. Seperti spinner untuk mengubah resolusi dan spinner untuk mengganti default kamera.



Gambar 13. Layout untuk activity scanner.

Dalam layout *activity* scanner terdapat beberapa komponen penting yaitu *RelativeLayout* sebagai *placeholder* untuk preview camera. *ImageView* untuk menampilkan debugging proses pengenalan plat nomor. *TextView* untuk menampilkan *string* hasil pengenalan plat nomor. Beberapa checkbox untuk invert image serta menampilkan *debug process*.

Dalam aplikasi ini, inputan image didapatkan langsung dari camera yang terdapat pada perangkat. Android sudah menyediakan API untuk memanipulasi dan memanfaatkan fitur camera. Untuk itu dibuat suatu *class* khusus untuk mendapatkan image dari camera dan secara *realtime* memprosesnya secara digital. *Class* ini mengimplementasikan *callback* *onPreviewFrame()* camera untuk mendapatkan image berupa *byte array* untuk selanjutnya memprosesnya ke dalam native method. Native method memproses *byte array* sebagai image matrix dan menghasilkan *bitmap* untuk ditampilkan

sebagai *debug process* serta *array String* sebagai hasil proses pengenalan plat.

Dalam cuplikan *onPreviewFrame* diatas, dapat dilihat bahwa dalam pengolahan image hanya memproses image dengan format *YCrCb* digunakan untuk gambar, yang menggunakan format *encoding* *NV21*. Hal ini untuk memastikan bahwa aplikasi dapat berjalan di semua perangkat Android karena *default format* camera adalah *NV21*.

Langkah selanjutnya adalah melakukan kompilasi di perangkat Android. Hal ini dilakukan untuk memastikan bahwa layout, native *library* serta *activity* berjalan dengan lancar pada perangkat. Perlu diketahui bahwa variasi perangkat Android sangat beragam, mulai dari ukuran layar, *density* atau kerapatan pixel pada layar hingga *architecture processor*. Hal ini akan menimbulkan perbedaan antara layout yang sudah didesain di Android Studio dengan yang tampil pada perangkat.

Untuk mencoba tampilan yang berbeda pada variasi perangkat yang berbeda, selain menggunakan perangkat sebenarnya penulis juga menggunakan *emulator* untuk memudahkan pengujian.



Gambar 14. Tampilan main activity pada perangkat HP dengan ukuran layar 600x1024 pixel

Pengujian dilakukan dengan perangkat yang ukuran layarnya 600x1024 pixel dan tampilan sesuai dengan rancangan

pada Android Studio. Pengujian juga berjalan lancar pada emulator dengan ukuran layar 1600x2560 pixel. Namun pengujian gagal ketika membuka activity scanner dikarenakan gagal untuk mengakses camera pada emulator. Pada emulator fitur camera sengaja di non-aktifkan untuk mengurangi pemakaian resource oleh emulator. Alasan lainnya yaitu untuk mengetahui error jika aplikasi tidak mendapat *permission* untuk mengakses kamera pada perangkat sebenarnya yang menggunakan API 23.

Pengujian selanjutnya adalah proses import data klasifikasi dari *assets directory* aplikasi ke media penyimpanan perangkat. Hal ini untuk memastikan bahwa proses pengenalan karakter dapat menggunakan data klasifikasi tersebut. File klasifikasi berupa file dengan ekstensi .xml berisi data training dan data klasifikasi yang dimuat oleh kelas DetectChar.cpp sebagai parameter untuk konstruksi object kNearest.

Pada pengujian import data klasifikasi dan data training, aplikasi berjalan lancar pada perangkat, namun proses import gagal pada emulator. Hal ini dikarenakan secara default aplikasi meng-import file ke *external storage*, namun pada emulator *external storage* secara default tidak tersedia. Untuk itu perlu dilakukan pengecekan terlebih dahulu sebelum proses import apakah *external storage* tersedia atau tidak.

Pengujian pengenalan plat nomor dilakukan dengan perangkat dan beberapa sample plat nomor kendaraan sebagai objek scanning. Proses scanning dilakukan langsung menggunakan perangkat kepada plat nomor kendaraan yang terpasang pada kendaraan.



Gambar 15. Plat Nomor Kendaraan Sebagai Objek Scanning

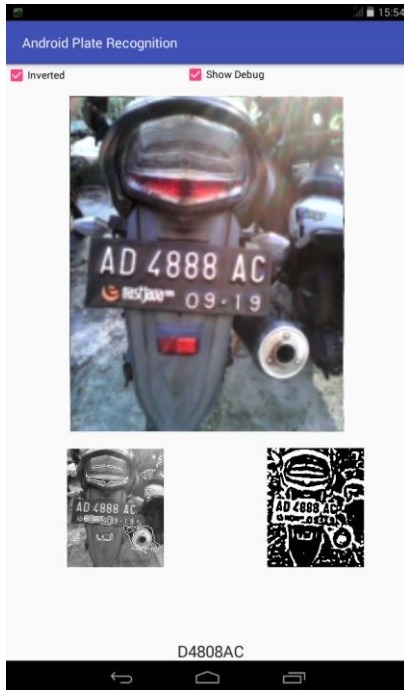
Pengujian dilakukan terhadap 10 plat nomor kendaraan dan smartphone dengan spesifikasi kamera 2 Megapixel hasil pengujian sebagai berikut. Dari tabel dibawah dapat disimpulkan bahwa kemampuan deteksi plat pada pencahayaan matahari ~80% dengan akurasi 23.37%. Hal ini disebabkan oleh beberapa faktor yaitu pada beberapa plat nomor terjadi pantulan sehingga mempengaruhi *white balance* kamera, bentuk nomor "6", "0", "8", "9" yang identik.

Tabel 1. Hasil pengujian dengan pencahayaan matahari dan smartphone dengan resolusi kamera 2 Megapixel

No	Nomor Plat diuji	Terdeteksi	Hasil deteksi	Akurasi
1	AG 5352 EE	YA	AG53LJ2I2	50%
2	AD 4888 AC	YA	4808AC	62.5%
3	L 5707 TS	YA	5I0I	14.2%
4	L 6658 ZI	YA	JD7	0%
5	L 5337 RB	YA	IVI	0%
6	L 6980 ME	YA	L6888	50%
7	L 5453 DX	YA	5A50	28.5%
8	L 5600 DJ	TIDAK	-	0%
9	L 6379 RY	YA	6070R	0%
10	L 6227 Y5	TIDAK	-	28.5%

Pada pengujian yang dilakukan dengan pencahayaan matahari, jarak pemindaian ternyata tidak mempengaruhi

akurasi pembacaan. Namun jarak optimal pemindaian adalah 80-100 cm.



Gambar 16. Pemindaian dari jarak ±1m

Pengujian selanjutnya menggunakan perangkat dengan resolusi kamera 5 Megapixel. Hal ini ditujukan untuk menguji apakah dengan resolusi lebih tinggi dapat meningkatkan akurasi pembacaan plat nomor, dan hasilnya sebagai berikut :

Tabel 2. Hasil pengujian dengan pencahayaan matahari dan smartphone dengan resolusi kamera 5 Megapixel

No	Nomor Plat diuji	Terdeteksi	Hasil deteksi	Akurasi
1	AG 5352 EE	YA	AG5352EF	87,5%
2	AD 4888 AC	YA	AD4888AC	100%
3	L 5707 TS	YA	5107T8	14,2%
4	L 6658 ZI	YA	L6658Z1	95%
5	L 5337 RB	YA	L5337RB	100%
6	L 6980 ME	YA	L6888	50%
7	L 5453 DX	YA	L5458DX	95%
8	L 5600 DJ	TIDAK	-	0%
9	L 6379 RY	YA	L6879RY	95%
10	L 6227 YS	TIDAK	-	0%

Dari tabel pengujian diatas dapat disimpulkan bahwa resolusi kamera dapat meningkatkan akurasi pembacaan plat nomor yaitu hingga ~95%. Namun dalam beberapa kasus, aplikasi kesulitan mengenali plat nomor ketika terjadi lekukan atau goresan tertentu pada area kotak plat sehingga tidak terdeteksi sebagai plat nomor.

Dari hasil pengujian pada plat nomor yang sama namun dengan variasi jarak didapatkan tabel sebagai berikut:

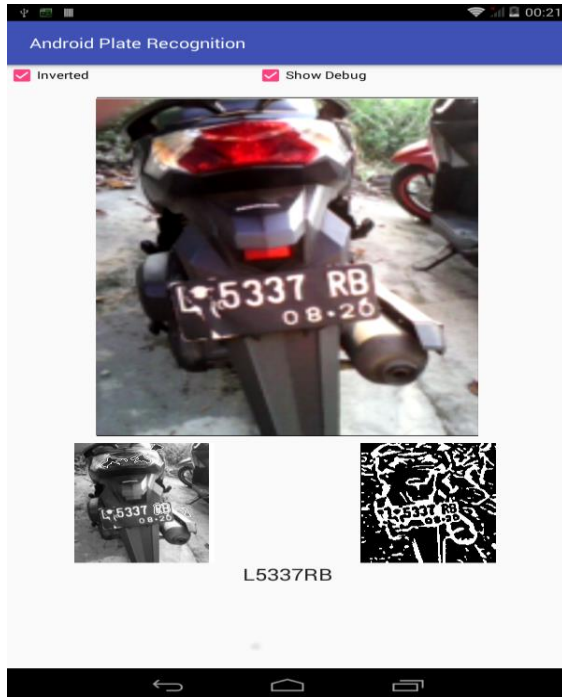
Tabel 3. Hasil pengujian berdasarkan jarak

No	Jarak pemindaian	Terdeteksi	Keterangan
1	10-50 cm	YA	Karakter plat terpotong
2	50-100 cm	YA	Terbaca
3	100 - 150 cm	TIDAK	Tidak terdeteksi

Dari hasil pengujian diatas, dapat disimpulkan bahwa algoritma yang diterapkan pada proses simulasi memiliki kemampuan yang berbeda jika diterapkan kepada perangkat smartphone. Pada saat simulasi image yang digunakan sebagai objek pengujian merupakan image yang sudah terlebih dulu di-capture dan berupa file, sehingga pixel yang dibaca merupakan data statis. Sedangkan pada aplikasi perangkat Android, image masukan merupakan image hasil pembacaan langsung / *real time* sehingga pixel pada image selalu berubah. Peneliti juga melakukan beberapa modifikasi pada beberapa parameter untuk menyesuaikan dengan image



masukam namun belum terjadi perubahan yang signifikan.



Gambar 17. Screenshot dengan menggunakan kamera 5 MP.

## KESIMPULAN

Dari hasil perancangan dan pengujian terhadap system identifikasi plat nomor kendaraan bermotor adalah sebagai berikut :

1. Dengan menggunakan smartphone Android identifikasi lebih fleksible dilakukan karena smartphone selalu dibawa sehingga kapanpun dan dimanapun dapat melakukan identifikasi plat nomor kendaraan.
2. Penggunaan algoritma dalam pengembangan aplikasi ini yaitu algoritma *threshold*, *contour* dan *kNearestNeighbors*.
3. Dari penggunaan algoritma tersebut, aplikasi memiliki karakteristik sebagai berikut:
  - a) Aplikasi digitalisasi plat nomor ini dapat membaca dan mengenali segmen tertentu pada image sebagai plat nomor.

- b) Pada saat pemindaian, jarak mempengaruhi kemampuan untuk mendeteksi segmen plat nomor. Sedangkan pencahayaan mempengaruhi tingkat akurasi pengenalan karakter.
- c) Tingkat akurasi pengenalan plat mencapai 95% dengan menggunakan perangkat smartphone yang memiliki fitur kamera beresolusi 5 Megapixel atau lebih.
- d) Untuk mendeteksi plat nomor dengan warna dasar terang harus dilakukan *invers* dulu agar algoritma pengenalan karakter dapat lebih optimal.

## DAFTAR PUSTAKA

- Avianto, D. (2016). Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropagation Neural Network. *Jurnal Informatika*, 10(1), 1199–1209.
- Chen, R., & Luo, Y. (2012). An Improved License Plate Location Method Based On Edge Detection. *Physics Procedia*, 24, 1350–1356. <https://doi.org/10.1016/j.phpro.2012.02.201>
- Limantara, A. D. (2017). Pemanfaatan Internet of Things (IoT) Sebagai Solusi Manajemen Transportasi Kendaraan Sepeda Motor, (August).
- Slamet, W., Made, K., Tubagus, P., Agus, S., & Sri Wiwoho, M. (2017). Internet of Things (IoT) as Green City Economic Development Smart Transportation System. *MATEC Web of Conferences*, 138, 07015. <https://doi.org/10.1051/mateconf/201713807015>
- Taufiq, M., Hidayatno, A., & Isnanto, R. (2012). Sistem Pengenalan Plat Nomor Polisi Kendaraan Bermotor Dengan Menggunakan Metode Jaringan Saraf Tiruan Perambatan Balik. *Electro. Undip. Ac. Id*, 7. Retrieved from [http://www.elektro.undip.ac.id/el\\_kpta/wp-content/uploads/2012/05/L2FO08057\\_MT A.pdf](http://www.elektro.undip.ac.id/el_kpta/wp-content/uploads/2012/05/L2FO08057_MT A.pdf)