

## Kajian Pustaka: Deteksi Plat Kendaraan Menggunakan Morfologi Citra dan Implementasinya Menggunakan Android, dan Raspberry Pi

Azwar Annas Azuhri<sup>1</sup>, Setyawan Widyarto<sup>2</sup>

<sup>1</sup>Universitas Budi Luhur

azwarannas\_95@yahoo.com

<sup>2</sup>Centre for Graduate Studies (CGS), Universiti Selangor

swidyarto@gmail.com

### Abstrak:

Pengolahan citra merupakan salah satu teknologi yang sedang tren saat ini, dan banyak di terapkan dalam kehidupan sehari-hari tidak hanya dapat mengolah gambar algoritma dalam pengolahan citra dapat di terapkan juga pada video. Penelitian ini di tujuhan untuk membangun sebuah sistem aplikasi untuk mendeteksi plat nomer kendaraan yang nantinya di harapkan alat ini dapat membantu petugas kepolisian atau lembaga transportasi pemerintah lainnya untuk memantau kendaraan yang ada di jalanan, seperti pelanggaran lalu lintas, kecelakaan, ataupun tindak kriminal lainnya, perancangan sistem deteksi plat kendaraan ini di rancang menggunakan library opencv dengan metode sistem pengenal/pengidentifikasi mobil otomatis atau Automatic Licence Plate Recognition (APLR) dimana system yang kami buat diharapkan dapat berjalan di berbagai perangkat seperti raspberry pi yaitu sebuah mini computer dengan sistem daya yang minimum, dan android kerana dalam perkembangan beberapa tahun belakangan ini android banyak memiliki kemajuan di bidang kamera sehingga diharapkan dengan kamera yang mumpuni dapat mengolah data plat kendaraan dengan sangat baik dan akurat, selain itu system yang minimum ini diharapkan mempermudah penggunanya karena dapat dibawa secara mobile, dan tidak memerlukan server atau resource computer yang berat. Hasil pengujian dengan metode APLR diharapkan dapat mendeteksi tidak hanya plat kendaraan hitam, tapi juga plat kendaraan berwana merah, dan kuning dengan menggunakan morfologi citra.

**Kata kunci:** Pengolahan citra, Morfologi citra, Automatic Licence Plate Recognition (APLR)

### Abstract

Image processing is one of the technologies that are currently trending, and many applied in everyday life can not only process image algorithms in image processing can also be applied to video. This research is aimed at building an application system to detect vehicle license plates which later is expected to help police officers or other government transportation agencies to monitor vehicles on the road, such

as traffic violations, accidents, or other crimes, design This vehicle detection system is designed using the library opencv with the method of the automatic car identification / identification system or Automatic License Plate Recognition (APLR) where the system that we make is expected to run on various devices such as Raspberry Pi which is a mini computer with a minimum power system, and Android because in the development of the past few years Android has a lot of progress in the field of cameras so that it is expected that a capable camera can process vehicle plate data very well and accurately, besides this minimum system is expected to facilitate its users because it can be brought mobile, and does not require heavy servers or computer resources. The results of testing with the APLR method are expected to be able to detect not only black vehicle plates, but also vehicle plates in red and yellow colors using image morphology.

**Keywords:** Image processing, Image morphology, Automatic License Plate Recognition (APLR)

## 1.

### PENDAHULUAN

Plat kendaraan merupakan sesuatu hal yang wajib dimiliki setiap pengguna kendaraan di seluruh dunia termasuk di Indonesia, karena plat nomer kendaraan berguna sebagai nomer unik suatu kendaraan yang dapat mengidentifikasi pemiliknya. Pada tahun 2014, kendaraan bermotor di Indonesia dari berbagai jenis mencapai 114 juta unit (Biro Pusat Statistik). (Limantara, 2017) jumlah ini merupakan jumlah yang sangat besar dan dapat menimbulkan masalah pada saat membuat plat ataupun perbaruan plat kendaraan. Untuk itu jurnal ini akan membahas beberapa review jurnal-jurnal yang membahas tentang identifikasi plat nomer kemudian mencoba mengimplementasikannya pada sebuah smart phone dan raspberry pi.

### 2. METODOLOGI

Penelitian ini merupakan tinjauan sistematis dengan menggunakan metode PRISMA yang dilakukan secara sistematis dengan mengikuti tahapan yang benar, prosedur dari systematic review ini terdiri dari beberapa langkah, yaitu :

- 1) Mendefinisikan kriteria kelayakan
- 2) Mendefinisikan sumber informasi
- 3) Pemilihan literatur
- 4) Pengumpulan data
- 5) Pemilihan item data

#### Kriteria Kelayakan

Adapun kriteria inklusi dalam penelitian ini antara lain sebagai berikut :

- 1) IC1 : artikel harus merupakan riset asli yang telah dikaji dan dituliskan dalam bahasa Indonesia atau bahasa Inggris
- 2) IC2 : artikel penelitian dipublikasikan pada tahun 2015 - 2020
- 3) IC3 : studi penelitian membahas tentang deteksi plat kendaraan bermotor dengan berbagai macam metode dan implementasinya seperti Pengolahan citra, Morfologi citra, Automatic Licence Plate Recognition (APLR)

### Sumber Informasi

Pencarian literatur dilakukan pada database online yang memiliki repositori besar untuk studi akademis seperti ResearchGate, IEEE, Google Scholar, Academia, Semantic Scholar.

### Pemilihan Literatur

Pemilihan literatur terdapat empat fase, yaitu :

- 1) Pencarian artikel penelitian yang relevan dengan topik penelitian ini dilakukan dengan menggunakan kata kunci : Pengolahan citra, Morfologi citra, Automatic Licence Plate Recognition (APLR).
- 2) Eksplorasi serta pemilihan judul, abstrak dan kata kunci yang didapatkan dari hasil pencarian berdasarkan kriteria kelayakan yang telah didefinisikan sebelumnya.
- 3) Membaca lengkap atau parsial artikel yang belum tereliminasi pada tahapan sebelumnya untuk menentukan apakah artikel tersebut harus dimasukkan dalam kajian selanjutnya sesuai dengan kriteria kelayakan.
- 4) Daftar referensi dari artikel terpilih dikaji

kembali untuk menemukan studi terkait lainnya. Artikel yang terdapat pada daftar referensi yang berelasi dengan studi ini akan dikaji kembali dengan

mengakukan tahapan 3 sampai dengan 4

### Proses Pengumpulan Data

Pengumpulan data dilakukan secara manual dengan membuat tabel SOTA yang berisi author, problem, metode dan hasil penelitiannya.

Author	Problem	Metode	Hasil Penelitian
(I Dewa Gede Aditya Pemayun, Widyadi Setiawan, Ngurah Indra ER, 2015)	sistem parkir yang memanfaatkan teknik pengenalan pola dalam teknik pengolahan citra digital. Umumnya, pada sistem parkir manual kecepatan pelayanan loket tergantung pada kecepatan petugas dalam membaca dan menginputkan data identitas kendaraaan ke sistem.	Metode Transformasi Hough	Hasil unjuk kerja pada simulasi pendeksi plat nomor kendaraan dari citra kendaraan berdasarkan tiga skenario berbeda didapatkan hasil terbaik menggunakan skenario gabungan vertikal dan horizontal dengan keberhasilan sistem dalam mendeksi 20 citra sampel adalah 95% dan persentase kegagalan 5%.
(Dwi Aulia Priandini, Jumadil Nangi, S.Kom, M.T, Mutmainnah Muchtar, S.T, M.Kom,	Sistem pengenal/pengidentifikasi mobil otomatis atau Automatic Licence Plate Recognition (ALPR) telah menjadi salah satu hal yang penting. ALPR	Morfologi Citra	Hasil dari penelitian ini dari 30 citra uji didapatkan bahwa operasi morfologi citra dapat diterapkan dalam

Jayanti Yusmah Sari, S.T, M.Kom, 2018)	dapat diimplementasikan pada berbagai kebutuhan seperti sistem perparkiran, pengawasan jalan tol, pengawasan lalu lintas dan sebagainya. Tapi system ini hanya		deteksi area plat mobil dengan hasil yang baik.
(Nur Wakhidah, 2016)	Faktor manusia dapat menyebabkan kesalahan dalam mencatat jumlah kendaraan bermotor dan tentu saja ini akan mempengaruhi durasi layanan dari sistem parkir yang ada. Pengenalan pola dapat digunakan untuk mengimplementasikan identifikasi plat nomor otomatis yang sangat berguna untuk kehidupan sehari-hari, seperti manajemen parkir, pemantauan lalu lintas, pengaturan tiket, dan pembayaran jalan tol.	Resize image, deteksi tepi	Tahap preprossesing merupakan tahap yang dapat memudahkan untuk mengenali objek yang akan dikenali/ diklasifikasikan. Resize Image mampu mempercepat proses dengan teknik interpolasinya. Deteksi Tepi dengan operator sobel memberikan hasil yang lebih baik daripada operator lainnya. Luas area maksimum mampu mengidentifikasi plat nomor mobil setelah penyambungan piksel, menghilangkan noise, menghilangkan hole dan

			pengindeksan setiap objek.
(Yasinta Oktaviana Legu Rema, 2019)	Merancang system yang dapat mengenali plat nomer kendaraan.	RGB, Grayscale, Deteksi Tepi (Edge Detection), Segmentasi Plat , Morphology, Thresholding, Metode Otsu.	4.1 Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Kombinasi metode pada segmentasi yaitu konversi RGB ke Grayscale, mengubah citra ke dalam binary image, Edge Detection, Morfology, dan Blob Analysis akan mengubah ketajaman background menjadi semakin gelap dan mempertajam derajat keabuan pada objek utama. Thresholding dengan menggunakan metode Otsu mempertajam derajat keabuan pada background citra dan mempertajam warna gelap pada objek utama. Edge Detection dengan metode Prewitt dan metode Sobel lebih baik

			daripada metode Edge Detection yang digunakan pada penelitian ini.
(Yuda Puspito, F. X. Arinto Setyawan, Helmy Fitriawan, 2017)	Bagaimana cara mendeteksi plat nomer kendaraan dengan matlab.	Metode Transformasi Hough Dan Hit Or Miss	Berdasarkan hasil penelitian didapatkan bahwa sistem berhasil mendeteksi posisi plat nomor kendaraan dengan tingkat keberhasilan pendekripsi sebesar 76% untuk nilai threshold 0,75, 72% untuk nilai threshold 0,8 dan 48% untuk nilai threshold 0,85. Hasil penelitian juga menunjukkan nilai rata-rata recall sebesar 54% untuk nilai threshold 0,75, 50% untuk nilai threshold 0,8 dan 40% untuk nilai threshold 0,85, sedangkan nilai rata-rata precision sebesar 14% untuk nilai threshold 0,75, 14% untuk nilai threshold 0,8 dan 12% untuk nilai threshold 0,85.

(Muhammad Nanda Kurniawan, Dudit Widiyanto, 2015)	Bagaimana membuat drone yang dapat mendeteksi plat kendaraan.	algoritma pengolahan citra (Centroid-Contour Distance (CCD)), algoritma ekstraksi fitur (Principal Component Analysis (PCA)), dan algoritma jaringan syaraf tiruan (Generalized Learning Vector Quantization (GLVQ))	Hasil akhir dari penelitian ini adalah sebuah program untuk AR. Drone yang berfungsi untuk menjelaki sebuah objek bergerak di lantai dengan respon waktu yang cepat dibawah satu detik.
(Dharu Wihartasih, dan Hari Wibawanto, 2015)	Prototipe sistem deteksi plat kendaraan bermotor diharapkan mampu mendeteksi plat kendaraan meskipun plat sudah dimodifikasi.	Morfologi	Hasil dari penelitian ini menyatakan bahwa prototipe sistem mempunyai akurasi yang tinggi untuk mendeteksi plat motor dan mobil. Prototipe sistem diuji dengan 130 citra kendaraan, sehingga menghasilkan akurasi 93,03%. Pengenalan karakter pada prototype sistem ini menggunakan

			metode OCR template matching. Sehingga akurasi pengenalan karakter terhadap plat yang terdeteksi adalah 74,29%. Akurasi pengenalan karakter tergolong cukup baik.
(Juniman Arief, Fitri Utaminingsrum, Yuita Arum Sari, 2018)	bagaimana mengenali sebuah kendaraan, sistem transportasi cerdas saat ini membutuhkan sistem yang dapat mengenali kendaraan untuk melakukan pendataan nomor kendaraan seperti pada sistem parkir dan sistem pengawasan maupun penegakan hukum dalam berlalu lintas (Yasin, 2014).	Selective Ratio Bounding Box	Dari hasil keseluruhan pengujian yang telah dilakukan didapatkan rata-rata tingkat akurasi pada penggunaan bounding box yaitu sebesar 54% pada keseluruan pengujian. Pada penggunaan selective ratio bounding box didapatkan rata-rata tingkat akurasi tertinggi pada pengujian ke-2 yaitu sebesar 92% dan rata-rata tingkat akurasi terendah pada pengujian ke-4 yaitu sebesar 68%. Sedangkan pada pengujian lainnya didapatkan rata-rata akurasi

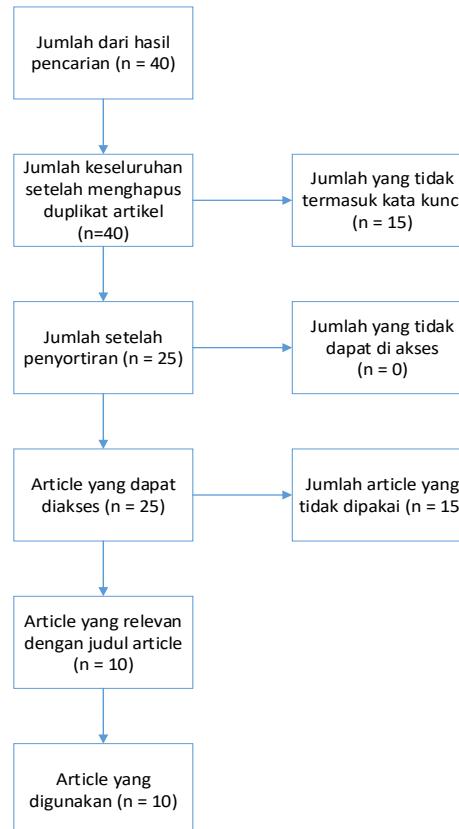
			yang sama yaitu sebesar 88%. Kata
(Farida, Zahir Zainuddin, Supriadi Sahibu, 2019)	Bagaimana mendeteksi kendaraan mengganggu penguna kendaraan.	Cara plat tanpa penguna	<p>Dari hasil peneltian yang telah dilakukan pada citra plat nomor kendaraan pribadi (plat nomor warn hitam) dengan menggunakan metode KNearest Neighbour dapat diperoleh kesimpulan sebagai berikut:</p> <p>Citra dapat diidentifikasi jika cahaya yang ada pada citra tidak terlalu besar maupun kecacatan tidak terlalu parah. Karena akan menimbulkan kesalahan dalam identifikasi.</p> <p>Dalam pengidentifikasi citra dibutuhkan banyak data set sehingga dalam pengidentifikasi tidak terjadi kesalahan. Karena dalam proses pengidentifikasi ini mencari kemiripan dengan data set yang mirip dengan data set yang sudah ada jika</p>

			tidak ditemukan maka akan terbaca sebagai data set yang lain sehingga terjadi kesalahan. Hal ini merupakan karakter dari metode K-Nearest neighbor. Dalam proses ini diperoleh tingkat akurasi sebesar 88 % dengan jarak 40 CM dan akurasi 85 % dengan jarak 50 CM dan 55 % dengan jarak 60 CM. Penggunaan metode k-nearest neighbor lebih tepat Karena metode ini tidak melakukan perulangan dalam proses pencocokan sehingga lebih efisien dalam melakukan pengujian.
	.		

### 3.

#### **HASIL DAN PEMBAHASAN**

#### **Jumlah dan Sumber Artikel yang Termasuk Kriteria Inklusi**



**Gambar 3.1**  
**alur pengumpulan data**

Berdasarkan penulusuran artikel dengan kata kunci diatas, populasi dalam penelitian adalah semua artikel yang diterbitkan dalam jurnal nasional dan internasional yang memiliki topik tentang implementasi penolahan citra, morfologi citra atau dengan metode lainnya untuk mendeteksi plat kendaraan menunjukan bahwa terdapat 40 artikel. Langkah berikutnya adalah peninjauan abstrak. Setelah meninjau abstrak dari 40 artikel yang dipilih, 25 artikel yang dimasukkan untuk

langkah berikutnya yang merupakan tinjauan fulltext. Pencarian untuk fulltext dari 25 artikel yang dipilih, sebanyak 10 artikel yang dipilih dimasukkan untuk dianalisis.

## Pembahasan

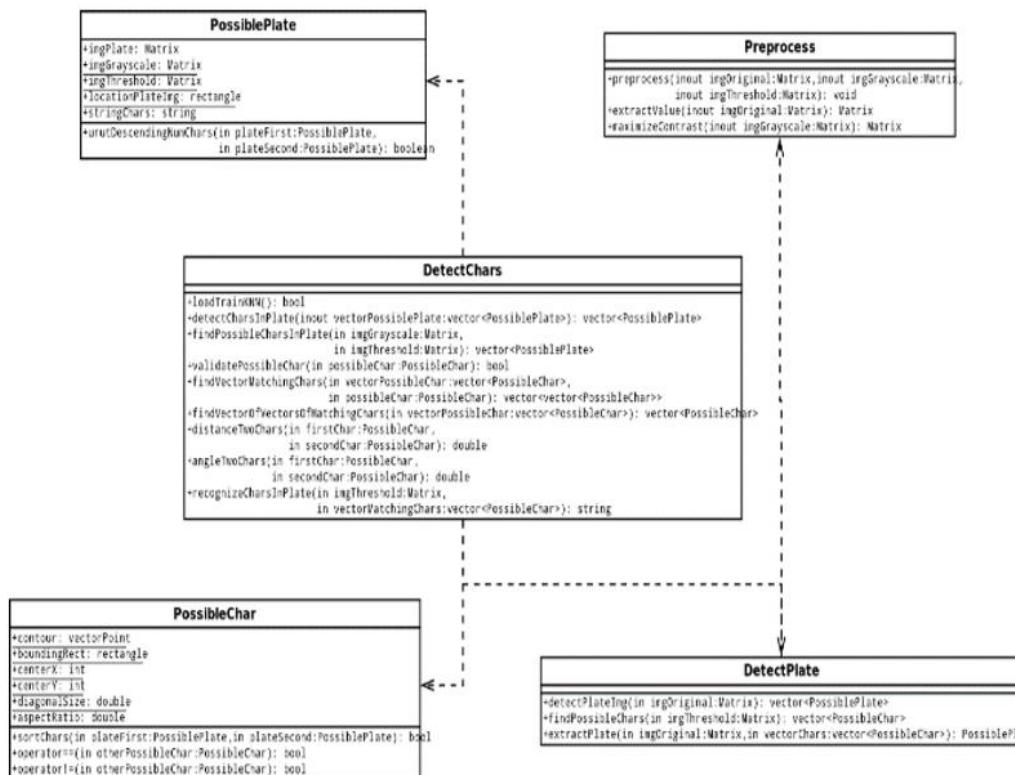
Dalam jurnal penelitian yang dilakukan oleh (I Gede Susrama Masdiyasa, Sulianto Bhirawa, Slamet Winard, 2019) dengan judul “Identifikasi plat nomor kendaraan bermotor menggunakan metode multi-step image processing berbasis

android". Pada proses penerapan algoritma, dibagi menjadi beberapa subprogram terpisah untuk memudahkan identifikasi jika terdapat kesalahan, baik kesalahan alur program maupun kesalahan logika. Beberapa subrrogram 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,

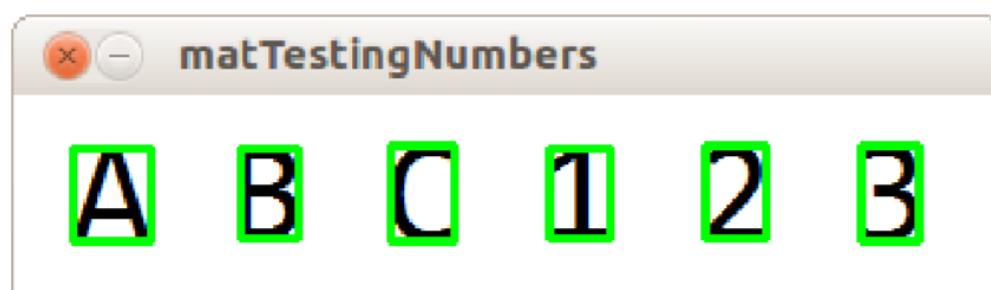
untuk optimasi proses deteksi karakter, maka diperlukan juga proses konversi dan thresholding. Jadi, image inputan dilakukan Preprocessing , langkah-langkah Preprocessing adalah sebagai berikut : 1. Konversi image ke format HSV; 2. Extract Value sebagai image baru dalam format Grayscale; 3. Menambahkan 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

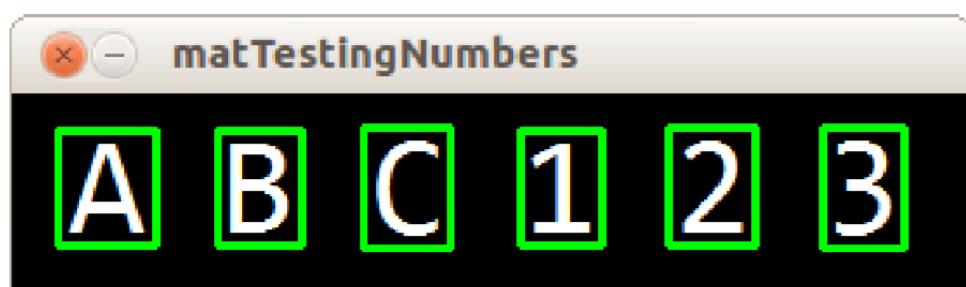
Gambar 7. Sample data untuk pengujian kNearest.



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.

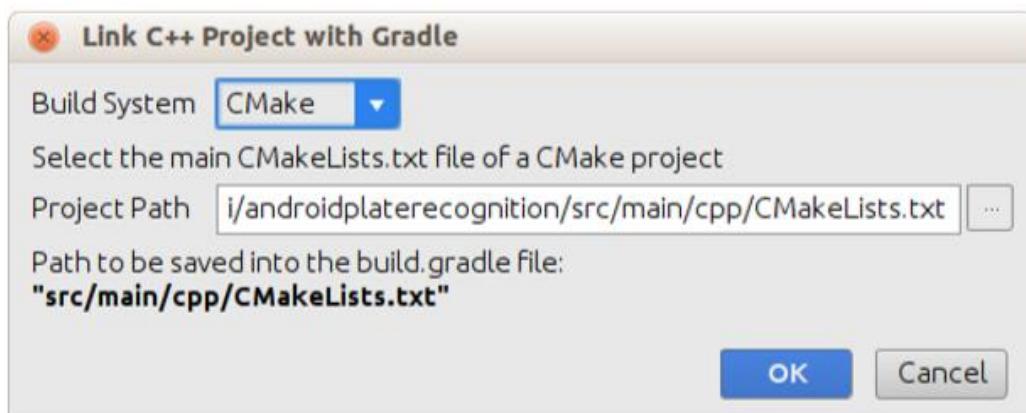
```
cv::threshold(imgOriginalScene,
imgInverted,100,
255,CV_THRESH_BINARY_INV);
```

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.

The screenshot shows the Android Studio interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, and Window. The left sidebar shows the project structure under 'Project' with modules like androidplaterecognition, build, external/nativeBuild, lib, and src. The src folder contains subfolders like androidTest, main, and cpp, which further contain files such as image\_processing.cpp, Main.h, PossibleChar.cpp, PossibleChar.h, PossiblePlate.cpp, PossiblePlate.h, Preprocess.cpp, Preprocess.h, Text.cpp, and Text.h. The code editor displays image\_processing.cpp with several lines of C++ code related to image processing and neural network loading. The bottom status bar shows the message 'AssertionError: Already disposed: Module: "androidplaterecognition" (moment ago)'.

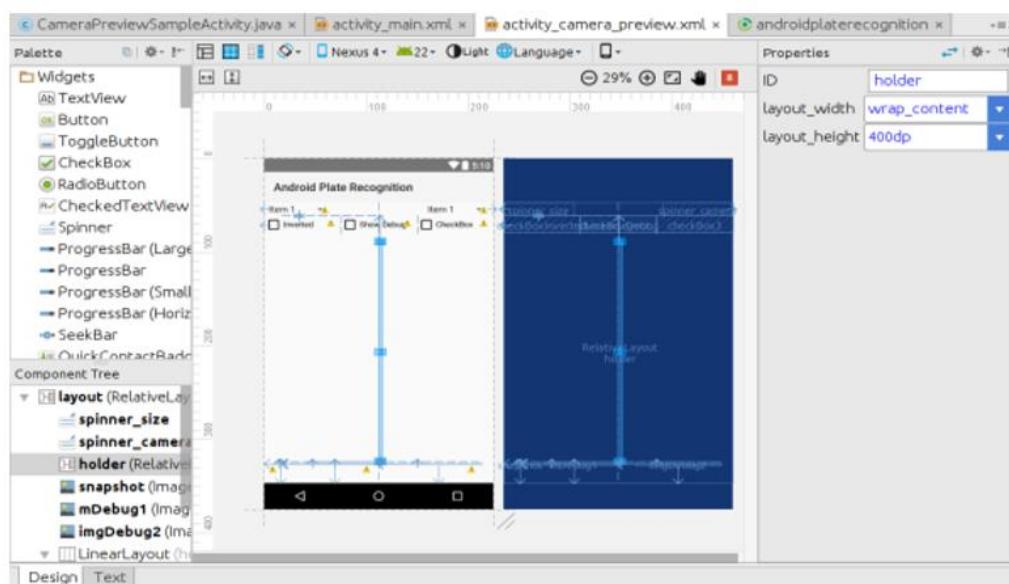
```

145     extern "C"
146     {
147         positive Java_xplatia_com_platerecognition_CameraPreview_startPlatesIniv(JavaVM *env, jobject thiz, jint width,
148             jint height, jobjectArray frameData,
149             jintArray outputPixels, jboolean inverted)
150         {
151             byte *pW2IFrameData = env->GetByteArrayElements(frameData, 0);
152             int *pOutPixels = env->GetIntArrayElements(outputPixels, 0);
153             jboolean inverted;
154         }
155         bool LoadNN = loadNNModelAndTrainNNWithClassification_filePages();
156         if (!loadNN)
157             __android_log_print(ANDROID_LOG_WARN, APPNAME, "(LoadNN failed)");
158         // return false;
159         // return true;
160         // return false;
161         // return true;
162         // return false;
163         // return true;
164         // return false;
165         // return true;
166         // return false;
167         // return true;
168         // return false;
169         // return true;
170         // return false;
171         // return true;
172         // return false;
173         // return true;
174         // return false;
175         // return true;
176         // return false;
177         // return true;
178         // return false;
179         // return true;
180         // return false;
181         // return true;
182         // return false;
183         // return true;
184         // return false;
185         // return true;
186         // return false;
187         // return true;
188         // return false;
189         // return true;
190     }
191 
```

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

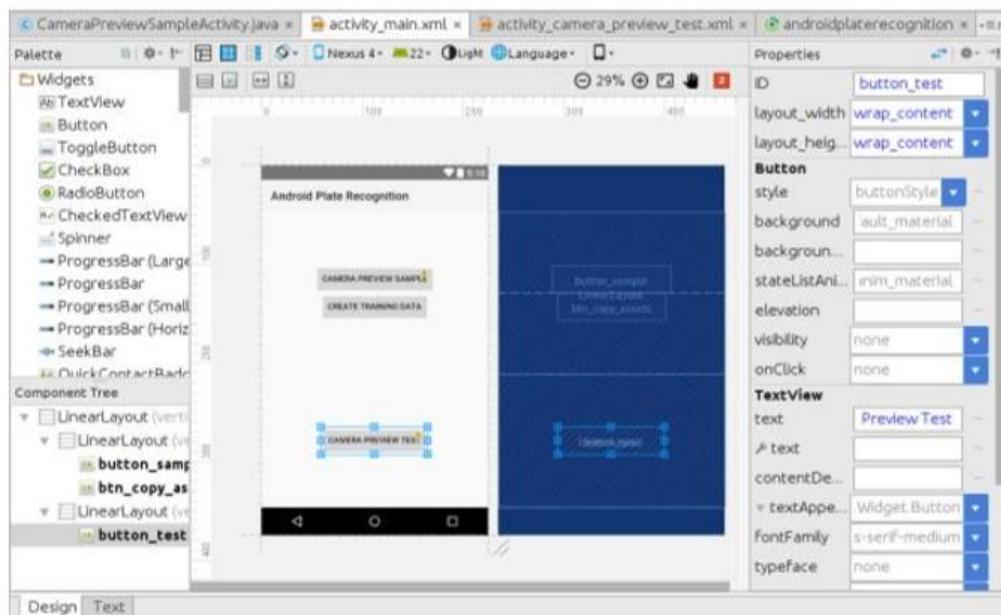
activity. Layout ini dibuat dengan XML untuk mendeklarasikan setiap komponen yang digunakan seperti RelativeLayout, ImageView, Button dan sebagainya.



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

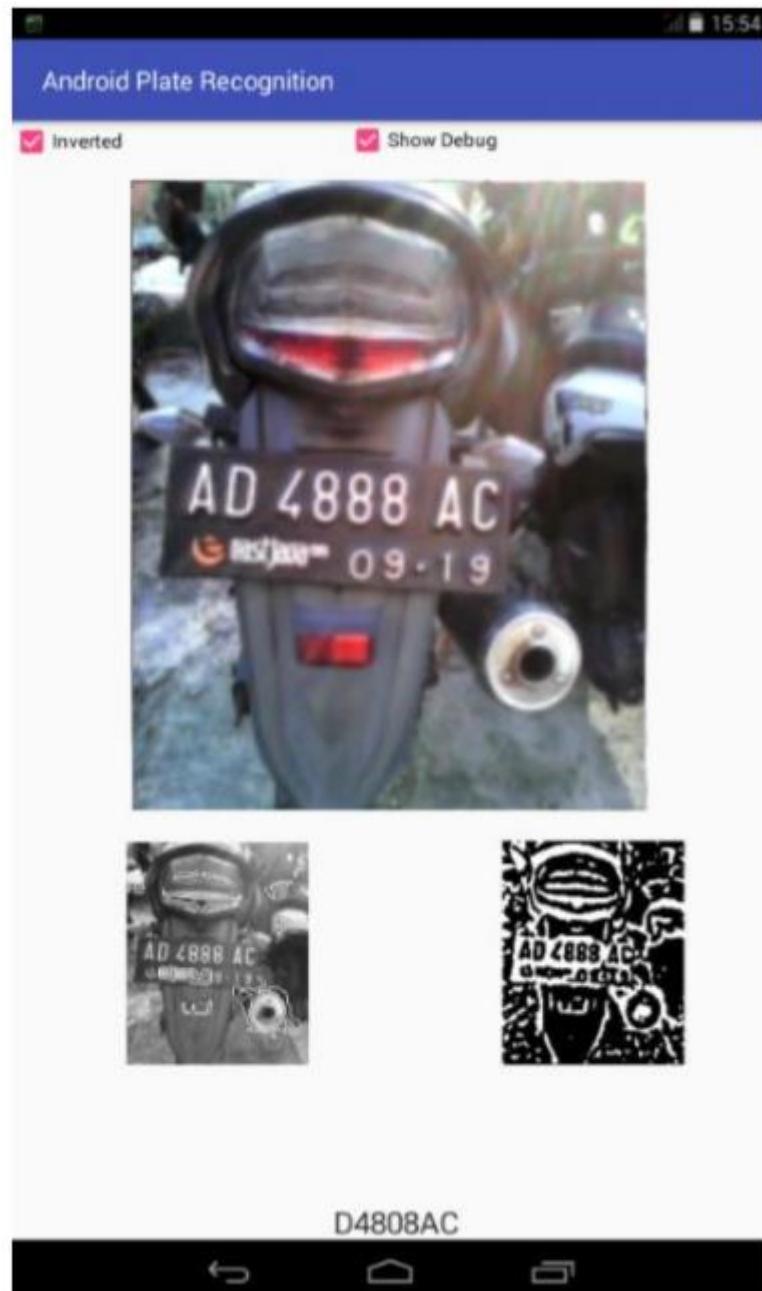
Pengujian dilakukan dengan perangkat yang ukuran layarnya 600x1024 pixel dan tampilan sesuai dengan rancangan pada Android Studio. Pengujian juga berjalan lancar pada emulator dengan ukuran layar 1600x2560 pixel. Namun pengujian gagal ketika membuka activity scanner dikarenakan gagal untuk mengakses 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 a s e t s d i r e c t o r y

aplikasi ke media penyimpanan perangkat. Hal ini untuk memastikan bahwa proses pengenalan karakter dapat menggunakan data klasifikasi tersebut. File klasifikasi berupa file dengan extensi .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. 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 :

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 VS	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

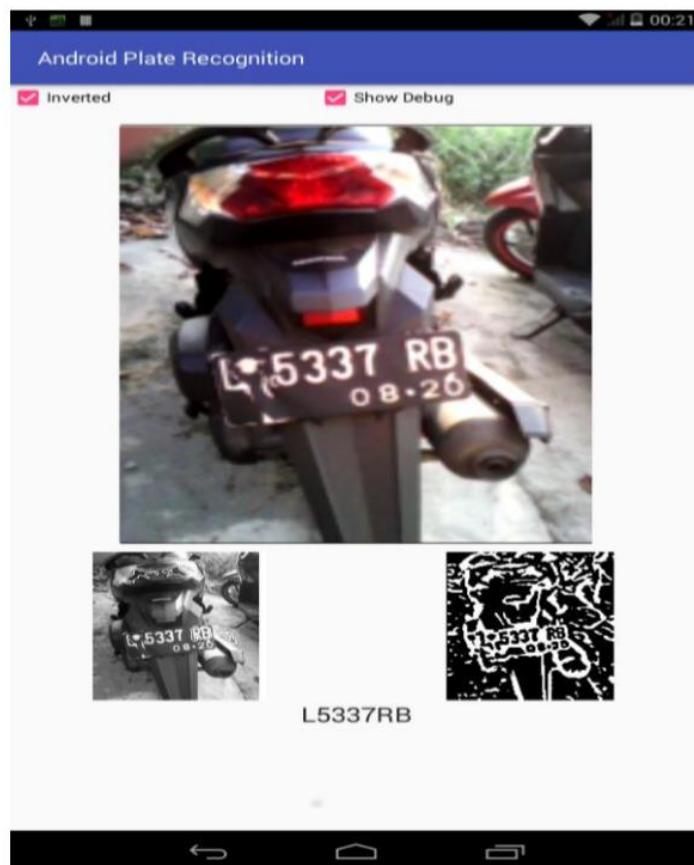
goresan tertentu pada area kotak plat sehingga tidak terdeteksi sebagai plat nomor. Dari hasil pengujian pada plat nomor yang sama namun dengan variasi jarak didapatkan tabel sebagai berikut:

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

Tabel 3. Hasil pengujian berdasarkan jarak

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 dicapture 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 masukan namun belum terjadi perubahan yang signifikan.



Gambar 17. Screenshot dengan menggunakan kamera 5 MP.

#### 4.

### 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 semartphone selalu dibawa sehingga kapanpun dan dimanapun dapat melakukan

identifikasi plat nomor kendaraan. 2.

Penggunaan algoritma dalam pengembangan aplikasi ini yaitu algoritma t h r e s h o l d, c o n t o u r 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.

## Daftar Pustaka

- [1]. I Dewa Gede Aditya Pemayun, Widyadi Setiawan, Ngurah Indra ER, 2015. Analisa Sistem Pendekripsi Posisi Plat Kendaraan Dari Citra Kendaraan.
- [2]. Dwi Aulia Priandini, Jumadil Nangi, S.Kom, M.T, Mutmainnah Muchtar, S.T, M.Kom, Jayanti Yusmah Sari, S.T, M.Kom, 2018. Deteksi Area Plat Mobil Menggunakan Operasi Morfologi Citra.
- [3]. Nur Wakhidah, 2016. Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area pada Image Segmentation.
- [4]. Yasinta Oktaviana Legu Rema, 2019. Deteksi Plat Nomor Kendaraan Bermotor dengan Segmentasi Gambar.
- [5]. Yuda Puspito, F. X. Arinto Setyawan, Helmy Fitriawan, 2017. Deteksi Posisi Plat Nomor Kendaraan Menggunakan Metode

Transformasi Hough Dan Hit Or Miss.

- [6]. Identifikasi Plat Nomor Kendaraan bermotor menggunakan Metode Multi-Step Image Processing Berbasis Android.
- [7]. Muhammad Nanda Kurniawan, Dudit Widiyanto, 2015. Implementation Of Image Processing Algorithm And Generalized learning Vector Quantitazion To Track An Object Using Drone.
- [8]. Dharu Wihartasih, dan Hari Wibawanto, 2015. Pembuatan Prototipe Sistem deteksi Plat Kendaraan Bermotor di Indonesia.
- [9]. Juniman Arief, Fitri Utaminingrum, Yuita Arum Sari, 2018. Penentuan Jumlah Karakter pada Plat Nomor Kendaraan dengan menggunakan Selective Ratio Bounding Box.
- [10]. Farida, Zahir Zainuddin, Supriadi Sahibu, 2019. Sistem Deteksi Plat kendaraan Dengan Menggunakan Metode K-Nearest Neghbouur.