

**PERANCANGAN PERANGKAT LUNAK PENGENALAN KARAKTER
PLAT NOMOR KENDARAAN DENGAN METODE
CONVOLUTIONAL NEURAL NETWORK**

Jenisa Felisa¹, Dhanny Setiawan², Iffa Khalisa³

STMIK LIKMI Bandung
Jl. Ir. H. Juanda no 96 Bandung 40132

E-mail : ¹ jenisafelisa@gmail.com

ABSTRAK

Plat nomor kendaraan diketahui sebagai identitas kendaraan yang memiliki informasi kendaraan dan pemilik kendaraan yang terdaftar di SAMSAT. Dalam Penelitian ini, Penggunaan nomor kendaraan dalam lingkungan transportasi dapat dideteksi dan digunakan dalam pencatatan identitas kendaraan secara otomatis. Pembuatan program deteksi dan pengenalan plat nomor ini berbasis web menggunakan bahasa pemrograman *python* dan *framework flask*, serta menggunakan metode *deep learning* yaitu *Convolutional Neural Network* (CNN) dan *Optical Character Recognition* (OCR) sebagai proses penyempurna gambar untuk mendapatkan informasi. Penelitian ini berfokus pada karakter plat nomor yang dideteksi dan diklasifikasi. Pengujian pada dataset alfanumerik untuk dapat identifikasi plat nomor dengan metode CNN menghasilkan akurasi mencapai 94,20%. Hasil akurasi pengujian dari segmentasi terhadap 131 gambar plat nomor yang telah diperbanyak sebesar 97,17%.

Kata kunci : Optical Character Recognition (OCR), Convolutional Neural Network (CNN), Deep learning.

1. PENDAHULUAN

Perkembangan teknologi yang sangat pesat telah diterapkan dalam kehidupan sehari-hari. Majunya teknologi memberikan dampak positif bagi manusia yang dapat bekerja dengan mudah dan memperoleh informasi. Salah satunya penerapan teknologi di bidang *Deep Learning* dan *Image Processing* dalam lingkungan Transportasi.

Hingga saat ini jumlah kendaraan di Indonesia pertumbuhannya semakin meningkat dari waktu ke waktu sehingga memungkinkan banyak terjadinya pelanggaran lalu lintas, pencurian kendaraan, dan kemacetan. Adapun kasus kejahatan seperti penjualan hasil pencurian motor serta memalsukan Surat Tanda Nomor Kendaraan (STNK) yang dapat disalahgunakan untuk berbagai keperluan. Adanya sistem pengenalan plat nomor kendaraan otomatis atau dengan istilah *Automatic Number Plate Recognition System* (ANPR) dapat menjadi solusi untuk memberikan informasi suatu kendaraan dengan cepat. Teknologi ini merupakan deteksi plat nomor secara otomatis yang dapat diimplementasikan untuk lalu lintas, pelacakan pencurian kendaraan, sistem parkir, keamanan, pengaturan kecepatan, dan pengaplikasian lainnya (Babu & Raghunadh, 2017).

Pada penelitian ini, penulis menerapkan teknologi ANPR berbasis web. Pengambilan plat nomor menggunakan kamera atau dari galeri, lalu dideteksi keberadaan nomor

kendaraan dan selanjutnya diproses dengan memanfaatkan metode *Optical Character Recognition* (OCR) dan metode *Convolutional Neural Network* (CNN). Hasil dari gambar yang telah diproses akan dikonversi menjadi teks. Hasil tersebut dikoneksikan dengan database untuk proses pencarian dan menampilkan informasi yang berformat STNK.

Penulis memilih metode CNN menjadi pengenalan pola untuk dapat mengenali setiap karakter yang terdapat pada gambar berisi teks termasuk citra plat nomor kendaraan yang telah diproses. Metode CNN memiliki akurasi tinggi dari beberapa penelitian sebelumnya seperti penelitian yang dilakukan (Astawa et al., 2018) dengan akurasi terhadap pengujian pengenalan karakter plat nomor mencapai 94%.

2. STUDI LITERATUR

2.1 Tanda Nomor Kendaraan Bermotor

Tanda nomor kendaraan bermotor (TNKB) menjadi identitas penting dan wajib terpasang dalam bentuk plat di setiap kendaraan bagian depan dan belakang. Sebagaimana yang diatur dalam peraturan presiden Republik Indonesia Nomor 5 Tahun 2015 tentang Penyelenggaraan Sistem Administrasi Manunggal Satu Atap Kendaraan Bermotor Bab 1 Pasal 1 ayat 11, menjelaskan bahwa TNKB sebagai bukti legitimasi yaitu pelat yang berisikan kode wilayah, nomor registrasi, serta masa berlaku yang dipasang pada ranmor atau kendaraan (Anonim, 2015).



Gambar 1
Contoh Plat Nomor Kendaraan

STNK (Surat Tanda Nomor Kendaraan) dinyatakan legal dan dapat digunakan jika sudah terdaftar dalam Satuan Administrasi Manunggal Satu Atap (SAMSAT) seperti yang dinyatakan dalam Peraturan Presiden Republik Indonesia Nomor 5 Tahun 2015 tentang Penyelenggaraan Sistem Administrasi Manunggal Satu Atap Kendaraan Bermotor Bab 1 ayat 10. Surat Tanda Nomor Kendaraan Bermotor (STNK) adalah dokumen yang berfungsi sebagai bukti legitimasi pengoperasian Ranmor yang berbentuk surat atau bentuk lain yang diterbitkan Polri yang berisi identitas pemilik, identitas Ranmor dan masa berlaku termasuk pengesahannya.

STNK memiliki format identitas kendaraan dan pemilik terdapat nomor registrasi, nama pemilik, alamat, merk, type, jenis, model, tahun pembuatan kendaraan, isi silinder/daya listrik, nomor rangka/NIK/VIN, nomor mesin, warna, bahan bakar, warna TNKB, tahun registrasi, nomor BPKB, kode lokasi, nomor urut pendaftaran, dan masa berlaku STNK.

2.2 Computer Vision

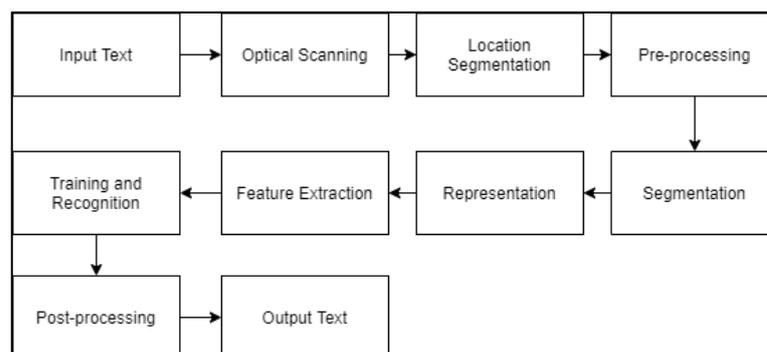
Penerapan *deep learning* khususnya di bidang *computer vision* yang berfokus pada proses visual objek yang terdapat rekaman video maupun gambar dan telah banyak diaplikasikan dalam kehidupan sehari-hari untuk mendapatkan informasi. *Computer vision* seperti halnya dengan penglihatan pada manusia dan hewan. Namun, algoritma dari *computer vision* masih ada kesalahan dan masih dalam tahap pengembangan. Adanya pendekatan data, statistik, dan geometri, permasalahan tersebut dapat diselesaikan dengan *computer vision* (Solem, 2012). *Computer vision* merupakan cara otomatis untuk memperoleh informasi dari gambar yang dikelola oleh komputer.

Menurut Szeliski dalam buku *Computer Vision Algorithms and applications*, *computer vision* telah diaplikasikan berbagai macam aplikasi yaitu (Szeliski, 2011):

1. *Optical character recognition* (OCR) : membaca tulisan tangan dan pengenalan plat nomor secara otomatis (ANPR).
2. *Machine inspection* : pemeriksaan suku cadang untuk jaminan kualitas menggunakan stereo vision dengan iluminasi khusus.
3. Ritel : pengenalan objek untuk jalur *checkout* otomatis.
4. Pemodelan 3D : konstruksi model 3D secara otomatis dari foto yang digunakan dalam sistem seperti *Bing Maps*.
5. *Medical Imaging* : mengetahui perkembangan otak seiring bertambahnya usia dengan dilakukannya studi jangka panjang.
6. *Automotive safety* : mendeteksi hambatan tak terduga seperti pejalan kaki di jalan.
7. *Match move* : melacak titik pergerakan yang terdapat pada video rekaman aksi dan menggabungkan dengan citra CGI.
8. *Motion capture (mocap)* : menggunakan *retro-reflective* dari kamera atau teknik vision lainnya untuk menangkap komputer untuk animasi.
9. *Surveillance* : memantau penyusup, menganalisis lalu lintas jalan raya, dan memantau korban tenggelam.
10. Pengenalan sidik jari dan biometrik : untuk otentikasi akses aplikasi secara otomatis.

2.3 OCR dan ANPR

Optical Character Recognition (OCR) merupakan teknologi *image processing* dalam klasifikasi pola yang terdapat pada citra digital berisikan teks sesuai alfanumerik atau karakter lainnya (Chaudhuri et al., 2017). Teknologi OCR telah banyak diterapkan dalam berbagai aplikasi, namun masih belum mumpuni dengan kemampuan manusia dalam membaca karena akurasi OCR masih rendah rentan terhadap kemiripan karakter saat memprediksi. Umumnya sistem OCR terdapat 3 tahapan penting yaitu *segmentation*, *feature extraction* dan *classification* ditunjukkan pada gambar 3 sebagai berikut :



Gambar 3
Alur Kerja OCR

Menurut Chaudhuri et al., berikut penjelasan dari beberapa tahap alur kerja OCR yaitu (Chaudhuri et al., 2017) :

1. *Optical Scanning*

Optical Scanning merupakan komponen pertama OCR yang terdiri dari mekanisme transportasi dan alat penginderaan yang mengubah intensitas cahaya menjadi tingkat abu-abu. Pemindaian dilakukan terhadap dokumen asli menjadi gambar digital. Dokumen yang dimaksud terdiri dari cetakan berwarna hitam dan berlatar putih.

2. *Location Segmentation*

Location Segmentation dilakukan dalam mengisolasi pada karakter atau kata pada teks. Tetapi terdapat permasalahan utama dalam segmentasi yaitu :

- a. Ekstraksi karakter saling menyentuh dan terfragmentasi.
- b. Membedakan noise pada teks
- c. Salah mengartikan grafik dan geometri dengan teks dan sebaliknya

3. *Preprocessing*

Preprocessing pada image dilakukan dengan tujuan memperhalus image dengan kondisi banyak jumlah noise, sehingga image dapat terdeteksi dengan baik. Selain itu, proses ini mencakup penerapan normalisasi sehingga image mendapat ukuran, kemiringan dan rotasi yang seragam. Pada umumnya proses *preprocessing* terdiri dari proses *grayscale* dan binerisasi dijelaskan sebagai berikut :

a. *Grayscale*

Menurut Lelis Baggio dalam buku *Mastering OpenCV 3* bahwa *grayscale* merupakan proses warna RGB pada *image* menjadi berwarna keabu-abuan menggunakan fungsi dari pustaka *OpenCV* yaitu *cvtColor()*.

b. Binerisasi

Menurut Abdul Kadir menjelaskan bahwa binerisasi merupakan proses image berwarna keabu-abuan menjadi warna biner menggunakan fungsi *cv2.threshold()*. Teknik binerisasi biasanya dengan istilah *threshold*. Adapun penambahan nilai ambang untuk membentuk citra biner dengan menggunakan metode otsu yang dapat mencari nilai ambang yang optimal (Kadir, 2019:191).

4. Segmentasi (*segmentation*)

Menurut Abdul kadir pada segmentasi objek merupakan teknik dalam pemisahan objek satu atau lebih terhadap latar belakang terutama objek yang berhimpitan atau saling bertumpukan (Kadir, 2019). Menurut Wesley dan Hairong dalam buku *Fundamentals of computer vision* bahwa penggunaan teknik segmentasi biasanya menggunakan algoritma otsu. Algoritma ini sangat efektif dalam analisis *histogram* dari gambar bimodal (dua puncak pada *histogram*) dan menambahkan noise gaussian 20dB pada gambar. Jika *histogram* menunjukkan 3 puncak, metode otsu akan gagal menghasilkan hasil segmentasi dengan baik (Snyder & Qi, 2017).

5. *Representation*

Representation merupakan komponen penting dalam sistem pengenalan. Tujuannya untuk meningkatkan akurasi algoritma dan menghindari kompleksitas ekstra pada sistem pengenalan. Metode *Representation* umumnya dikelompokkan menjadi 3 kelompok yaitu *Global transformation* dan *series expansion*, *statistical representation*, dan *geometrical* dan *topological representation*.

6. *Feature Extraction*

Menurut Tyagi bahwa *Feature Extraction* merupakan proses pengambilan informasi visual dari gambar yaitu warna, tekstur, *shape*, dan *edge*. Metode ini dilakukan setelah gambar melewati proses sebelumnya (Tyagi, 2018). Tugas lain dari proses ini yaitu klasifikasi. Ada dua pendekatan klasifikasi untuk OCR yaitu metode teori dan struktural keputusan. Dalam teori, karakter dideskripsikan secara numerik dalam fitur vektor. Pendekatan utama dari teori keputusan yaitu pengklasifikasi jarak minimum, pengklasifikasi statistik, dan jaringan saraf (*neural network*). Sementara untuk struktural keputusan menggunakan metode sintaksis yang merupakan pendekatan yang paling umum.

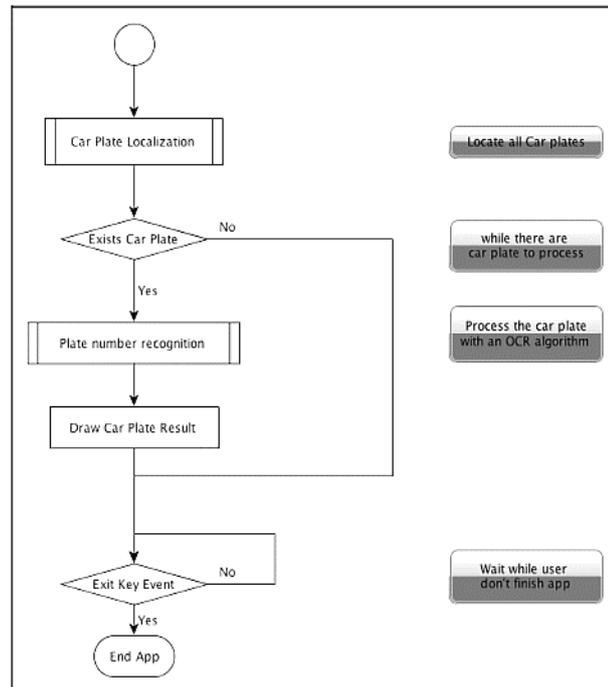
7. *Training and Recognition*

Proses ini menggunakan metodologi pengenalan pola dengan menetapkan sampel yang tidak diketahui ke dalam kelas yang telah ditentukan.

8. Post-processing

Post-processing merupakan tahap akhir dalam OCR. Pada tahap ini, umumnya dilakukannya pengelompokkan serta deteksi dan koreksi kesalahan.

Automatic number plate recognition (ANPR) merupakan teknologi deteksi plat nomor kendaraan dengan dasar menggunakan metode *optical character recognition* (OCR). Ada dua tahap dalam ANPR yaitu deteksi plat dan pengenalan plat yang terdapat pada Gambar 4. Setelah gambar terdapat plat terdeteksi dan tersegmentasi, selanjutnya pengenalan plat menggunakan algoritma OCR atau algoritma klasifikasi untuk menentukan karakter alfanumerik (Baggio, 2017).



Gambar 4
Alur Kerja ANPR

2.4 Deep Learning

Menurut Goodfellow et al. dalam buku *Deep Learning*, definisi deep learning sebagai berikut (Goodfellow et al., 2016) :

Deep learning is a particular kind of machine learning that achieves great power and flexibility by representing the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

Berdasarkan teori di atas, bahwa deep learning merupakan pendekatan machine learning yang mempunyai kekuatan dan fleksibilitas yang tinggi dengan mempresentasikan dunia sebagai konsep-konsep berhirarki. Setiap konsep berkaitan dengan konsep lebih sederhana dan representasi lebih abstrak dihitung dalam istilah kondisi kurang abstrak.

Dilansir dari situs IBM definisi dari deep learning sebagai berikut (IBM Cloud Education 2020) :

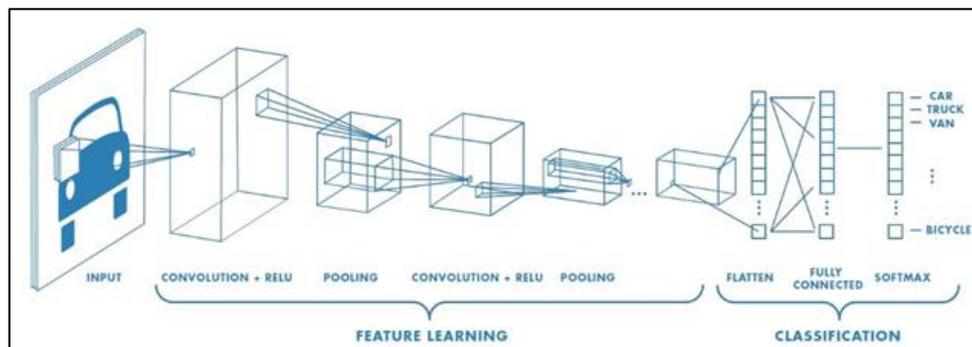
Deep learning is a subset of machine learning in which multi-layered neural networks—modeled to work like the human brain—'learn' from large amounts of data. Within each

layer of the neural network, deep learning algorithms perform calculations and make predictions repeatedly, progressively 'learning' and gradually improving the accuracy of the outcome over time.

Maksud dari teori di atas bahwa *deep learning* bagian dari teknik *machine learning* yang jaringan saraf bermulti-layer dimodelkan untuk bekerja seperti otak manusia untuk mempelajari sejumlah data. Dalam setiap layer jaringan saraf, algoritma *deep learning* melakukan perhitungan dan prediksi berulang kali, *learning* secara progresif dan bertahap meningkatkan keakuratan hasil dari waktu ke waktu.

Dari teori-teori di atas, dapat disimpulkan *deep learning* merupakan representasi data dengan mempelajari data menggunakan *neural network* yang memiliki banyak layer, pembelajaran tersebut dilakukan secara bertahap untuk peningkatan akurasi.

Ada berbagai macam metode dari jaringan saraf atau *neural network* untuk mengenal objek pada gambar, salah satunya yaitu *Convolutional Neural Network* (CNN). Metode ini termasuk pengenalan pola dengan pendekatan *supervised* untuk memprediksi suatu objek terutama prediksi pada *image*. Metode MLP (*Multi Layer Perceptron*) atau *fully connected* menjadi metode awal dalam pengembangan CNN. Dilansir dari situs cs231n.github.io bahwa metode CNN memiliki kelebihan dibandingkan metode sebelumnya yaitu berdimensi tiga (*width*, *height*, dan *depth*). *Depth* yang dimaksud yaitu dimensi tiga dari *input* volume aktivasi yang merujuk pada jumlah total *layer* dalam jaringan. CNN atau ConvNet merupakan alur yang terdapat banyak *layer*, setiap *layer* mengubah dari satu volume ke volume lainnya dalam fungsi berbeda (Karpathy n.d.). Menurut Suyanto dalam buku *Machine Learning Tingkat Dasar dan Tingkat Lanjut* bahwa *layer* CNN memiliki volume *input* tiga dimensi ditransformasikan ke dalam volume *output* tiga dimensi aktivasi-aktivasi sel saraf (Suyanto, 2018). CNN dibangun tiga *layer* utama yaitu *convolutional layer*, *pooling layer* dan *fully-connected layer*. Sementara arsitektur CNN terdiri dari *convolutional layer*, *rectified linear unit*, *pooling layer*, dan *fully-connected layer* (Karpathy, n.d.). Gambar 5 merupakan ilustrasi arsitektur CNN.



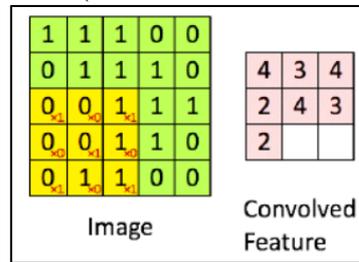
Gambar 5
Arsitektur CNN

Arsitektur CNN dijelaskan sebagai berikut :

1. *Convolutional layer*

Dalam sebuah jurnal oleh Suartika E. P et al menjelaskan bahwa *Convolutional layer* atau *convolution layer* merupakan operasi konvolusi terhadap output dari layer sebelumnya (E.P et al., 2016). *Convolution layer* terdiri dari beberapa bidang neuron dua dimensi yaitu *feature maps*. *Neuron* dari bagian bidang tersebut terhubung dalam *neuron* dari layer sebelumnya (Radzi & Khalil-Hani, 2011). Cara kerja dari konvolusi terhadap citra yang diinput dengan melakukan perhitungan melalui kernel atau filter yang terdapat aksi kotak kuning pada kotak hijau yang merupakan citra (E.P et al., 2016) ditunjukkan pada Gambar 6. Perhitungannya dengan cara menggerakkan filter dari kiri atas hingga

kanan bawah tergantung jumlah *stride* (langkah). Untuk mendapatkan nilai fitur baru, hasil perhitungan konvolusi dijumlahkan dengan istilah bias yang dapat dilatih lalu diteruskan ke *non-linear activation* (Radzi & Khalil-Hani, 2011:46).

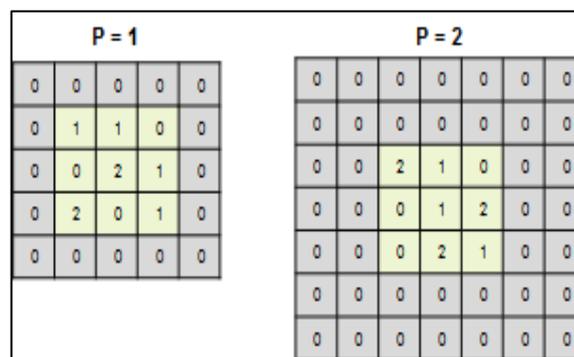


Gambar 6
Operasi Konvolusi

Sebelum melakukan konvolusi, umumnya CNN menggunakan *stride* (langkah) sebesar 1 dengan *zero padding* sebesar :

$$P = ((F - 1))/2$$

Dimana **P** adalah ukuran *padding* dan **F** adalah ukuran bidang reseptif yang sama dengan ukuran filter. Contoh *zero padding* , *stride* = 1 dapat dilihat pada Gambar 7.



Gambar 7
Teknik Zero Padding dengan *stride* =1

Konvolusi pada dasarnya hanya berupa *dot product* antara filter dengan sebuah bidang reseptif kecil pada citra yang berukuran sama dengan ukuran filter (Suyanto, 2018:311). *Conv Layer* menerima citra masukan berukuran :

$$w_1 \times h_1 \times d_1$$

di mana **w₁** dan **h₁** adalah lebar dan tinggi citra sedangkan **d₁** adalah kedalaman (*depth*) dengan jumlah kanal *Red Green Blue* (RGB). CNN memiliki empat *hyperparameter* yaitu jumlah filter **K**, ukuran bidang reseptif atau *spatial extent* (tingkat spasial) **F**, lebar langkah atau *stride* **S**, dan *zero padding* **P**. Terakhir volume *output* dapat dihitung dengan persamaan

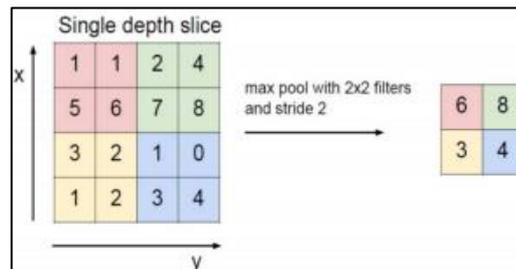
$$w_2 = \frac{w_1 \times h_2 \times d_2}{(w_1 - F + 2P) \times S} + 1,$$

$$h_2 = \frac{(h_1 - F + 2P) \times S}{d_2} + 1,$$

$$d_2 = K,$$

2. Pooling layer

Pooling layer merupakan metode yang mengurangi kompleksitas pada layer selanjutnya, tetapi saat pengurangan tidak mempengaruhi jumlah filter (Albawi, et.al, 2017). Pooling layer disisipkan di antara *layer* konvolusi dalam arsitektur CNN dikarenakan dapat mengontrol overfitting (Karpathy n.d.). Max pooling merupakan type dari metode pooling layer yang umum digunakan dalam filter atau gambar dengan pooling berukuran 2×2 . Cara kerja yang dilakukan max-pooling pada gambar 8 dalam area warna merah melakukan pooling. Dalam area tersebut diambilnya angka paling besar, lalu bergeser sejumlah 2 *stride* (langkah) untuk melakukan pooling kembali (Albawi et al., 2017).



Gambar 8
Operasi Max Pooling

3. Fully connected layer

Fully connected layer terdapat *neuron* memiliki koneksi penuh ke semua aktivasi dalam layer sebelumnya. Aktivasinya dapat dihitung dengan perkalian matriks diikuti dengan offset bias (Karpathy n.d.). Layer ini biasanya diterapkan dalam MLP untuk melakukan tranformasi pada dimensi data. Sebelum memasuki *fully connected layer*, *neuron* pada layer konvolusi harus ditransformasi menjadi data satu dimensi.

3. ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

3.1 Gambaran Umum Perangkat Lunak

Perangkat lunak yang akan dirancang yaitu sistem deteksi plat nomor dan pengenalan karakter pada plat nomor kendaraan. Sistem ini memanfaatkan plat nomor kendaraan untuk pencatatan lebih cepat dan dapat dijalankan di web. Sebelum menampilkan informasi mengenai identitas pengemudi dan kendaraan, akan dilakukannya memasukan gambar dari galeri dengan gambar bagian daerah yang terdapat plat nomor, lalu dideteksi dan klasifikasi karakter pada plat nomor tersebut menjadi teks.

Setelah diproses gambar menjadi teks, teks akan diambil untuk pengambilan informasi kendaraan dari database sesuai format STNK. Jika ada informasi kendaraan tidak dapat ditampilkan, maka dipastikan informasi kendaraan tidak terdaftar di database. Aplikasi ini menggunakan bahasa pemrograman *python*, library *OpenCV* sebagai pemrosesan gambar, algoritma CNN dan library *Keras* sebagai pengenalan pola pada karakter plat nomor serta MySQL sebagai *database*-nya. Untuk perancangan perangkat lunak menggunakan *software jupyter notebook*.

3.2 Spesifikasi Kebutuhan Perangkat Lunak

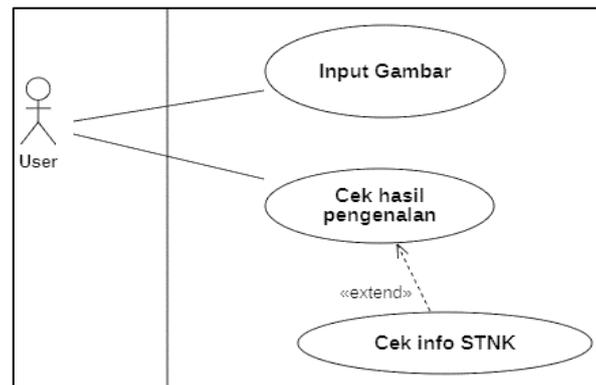
Perangkat lunak ini diharapkan dapat berfungsi antara lain sebagai berikut :

1. Sistem dapat mendeteksi lokasi plat nomor kendaraan setelah menekan tombol proses masukkan gambar.
2. Sistem dapat menghasilkan teks sesuai plat nomor kendaraan.

3. Sistem dapat terhubung dengan database dalam pengambilan informasi identitas kendaraan.

3.3 Use Case Diagram

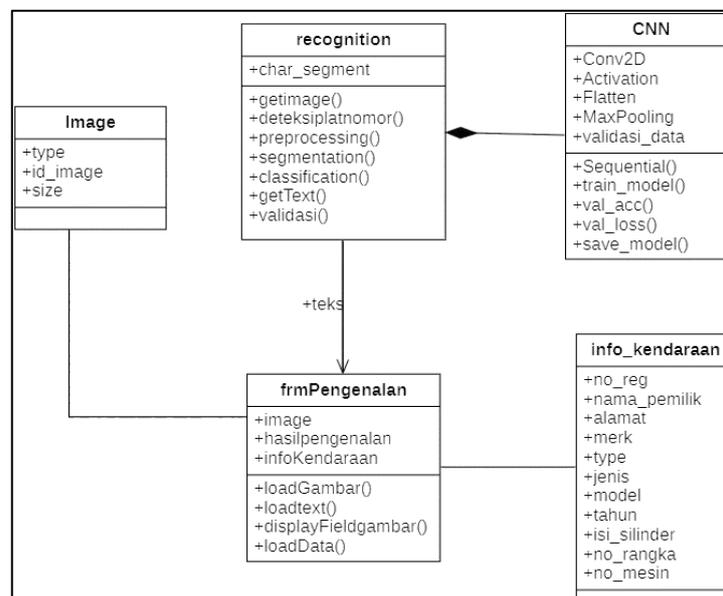
Use case dimodelkan interaksi antara pengguna dan sistem dapat dilihat pada gambar 9. Sistem aplikasi perangkat lunak yang dirancang memiliki dua fungsi utama (key behaviour) yaitu input gambar dan cek hasil pengenalan, dengan fungsi cek info STNK sebagai fungsi yang menyertai proses cek hasil pengenalan.



Gambar 9
Use Case Diagram

3.4 Class Diagram

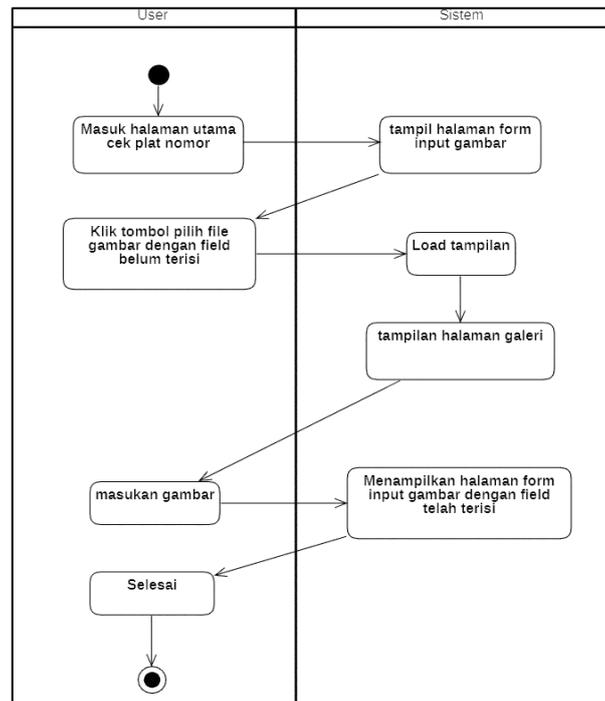
Class diagram aplikasi perangkat lunak dapat dilihat pada gambar 10.



Gambar 10
Class Diagram

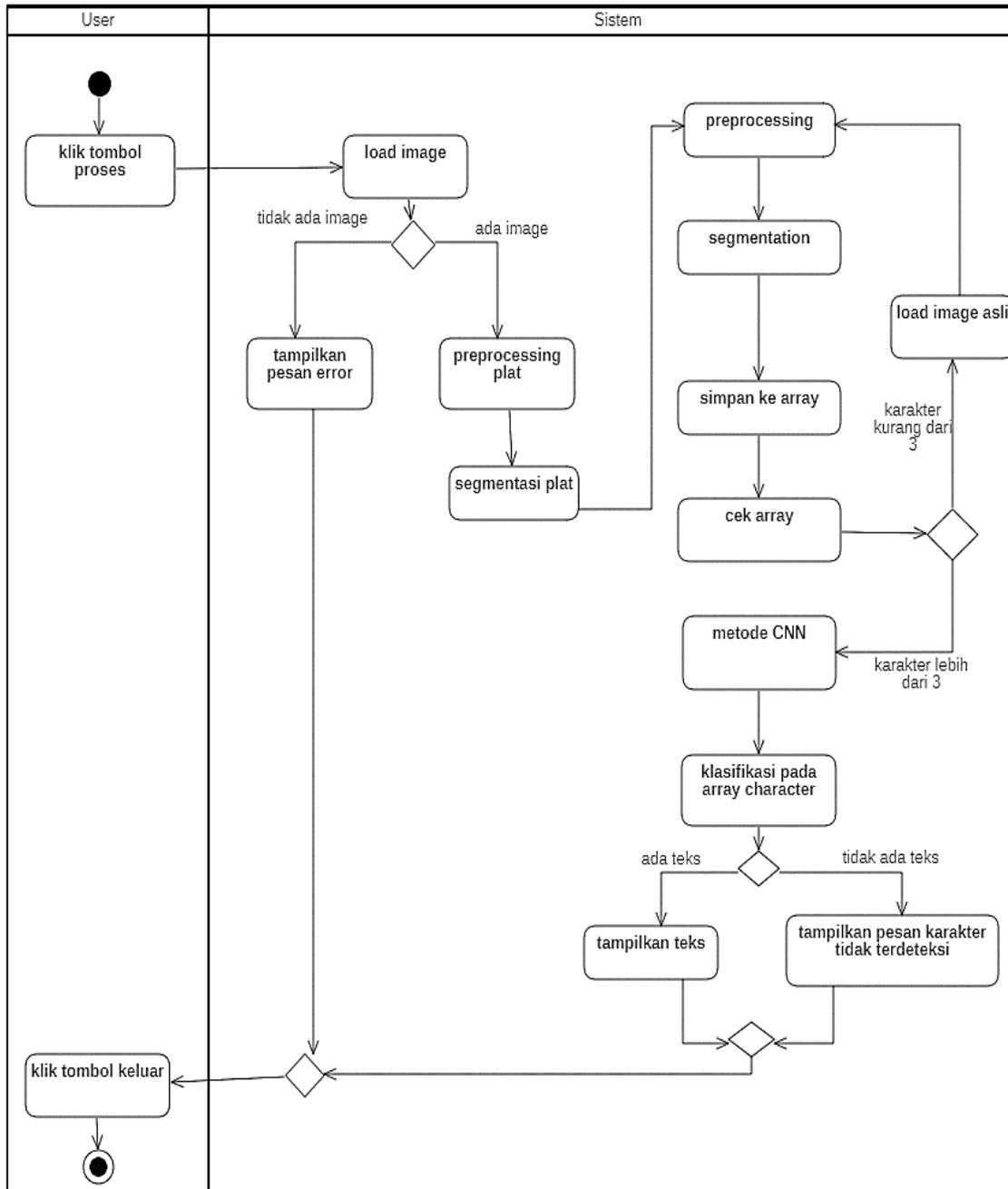
3.5 Activity Diagram

Activity diagram merupakan penggambaran alur proses aktivitas terhadap suatu sistem. Pada Gambar 11 menggambarkan *user* melakukan aktivitas memasukan gambar pada *field* yang belum terisi dalam *form input* gambar. Pada halaman *form input* gambar, ditampilkan *field* gambar yang belum terisi, tombol pilih *file* atau berkas dan tombol proses pengenalan. Ketika user memilih tombol file, akan muncul tampilan galeri untuk memasukan gambar.



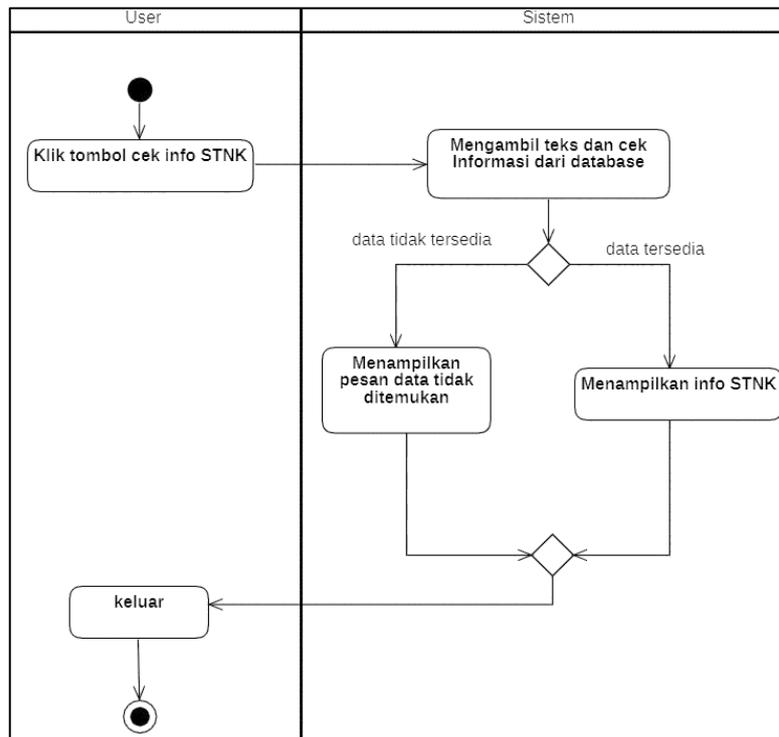
Gambar 11
Activity Diagram Input Gambar

Pada *activity diagram* cek pengenalan menggambarkan aktivitas sistem melakukan deteksi dan pengenalan pada plat nomor kendaraan setelah *user* menekan tombol proses. Tahapan pertama yang dilakukan sistem yaitu tahap load image di mana gambar tersebut tersedia pada *field* gambar atau tidak. Selanjutnya tahap deteksi plat dan dilakukannya preprocessing untuk menyempurnakan image, lalu proses segmentasi untuk memisahkan karakter pada plat. Terakhir tahap pengenalan atau klasifikasi dengan metode CNN hingga menjadi teks. Aktivitas tersebut ditunjukkan pada Gambar 12.



Gambar 12
Activity Diagram Cek Pengenalan

Activity diagram pada Gambar 13 menggambarkan aktivitas *user* menggunakan hasil pengenalan berupa teks untuk pencarian yang dapat menghasilkan info kendaraan berformat STNK yang tersimpan dari *database*.

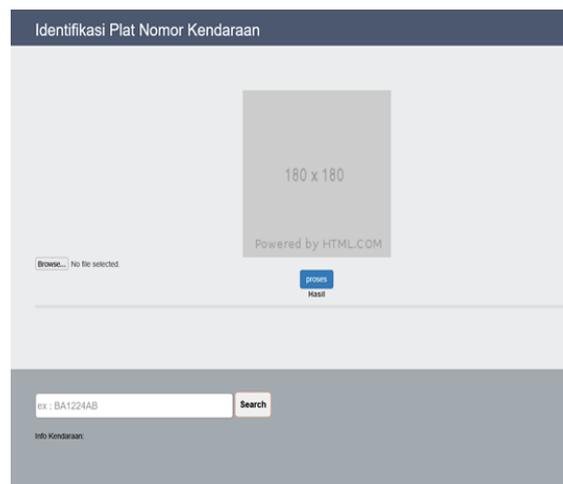


Gambar 13
Activity Diagram Cek Info STNK

4. IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

4.1 Pengujian Antar Muka

Pengujian antarmuka terhadap perangkat lunak yang dibuat terdiri dari tampilan pengenalan karakter dan pengecekan data diambil dari *database*. Pada gambar 14 terdapat antarmuka identifikasi plat yang dapat menginput gambar untuk dideteksi dan menampilkan informasi kendaraan.



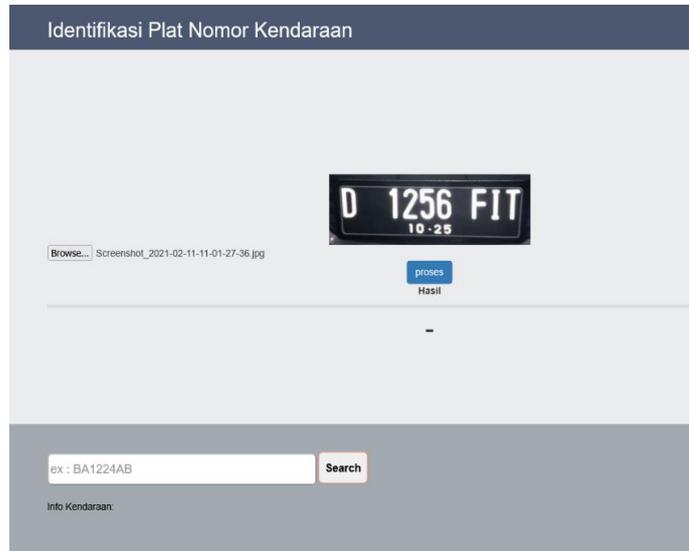
Gambar 14
Antar Muka Halaman Form Deteksi

Pada halaman deteksi dalam gambar 14 terdapat komponen-komponen yaitu:

1. *Button choose file* untuk memasukkan gambar.
2. *Button proses* untuk memproses identifikasi gambar yang telah dimasukkan.
3. *Text input field* plat nomor untuk menampung hasil teks yang telah diproses.

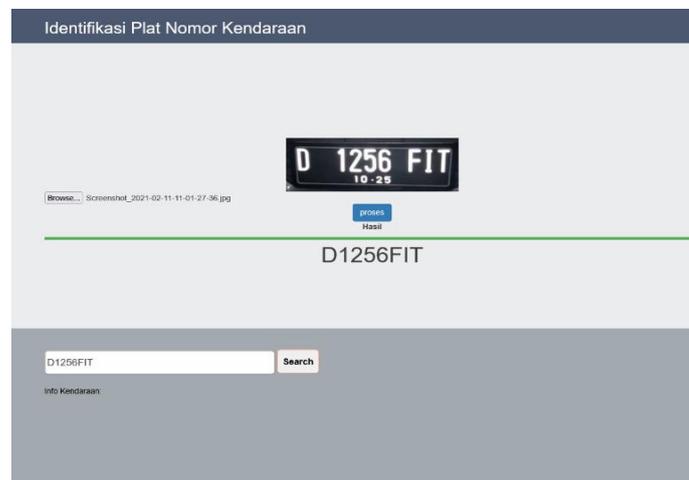
4. *Choose file field*, memberi keterangan nama *file*.
5. *Button search* untuk mengambil data dari database sesuai input di *text input* plat nomor.
6. Tabel info kendaraan berisikan informasi kendaraan yang dihasilkan dari *button* cek info.
7. *Image preview* untuk menampilkan gambar setelah di-*input*.

Pada gambar 15 merupakan antar muka halaman form deteksi setelah dimasukan gambar plat.



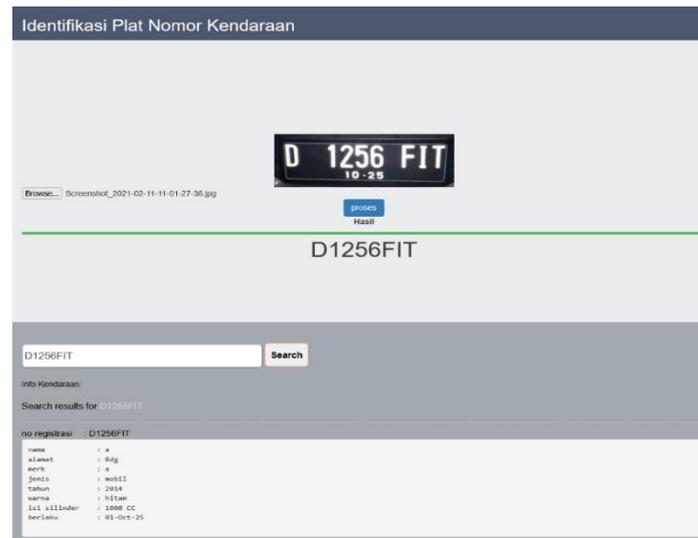
Gambar 15
Tampilan Setelah Input Gambar

Pada gambar 16 merupakan tampilan hasil plat setelah menekan *button* proses.



Gambar 16
Tampilan Ketika Menekan Button Proses

Pada gambar 17 ditampilkan informasi kendaraan dalam bentuk tabel setelah menekan *button* cek info. Jika field text plat nomor tidak sesuai dengan data pada database, maka informasi tidak ditampilkan.



Gambar 17
Tampilan Info Kendaraan

4.2 Pengujian Fungsi

Untuk menganalisa program yang diuji menggunakan pengujian black box untuk mendapatkan hasil dan memenuhi kebutuhan Perangkat lunak. Hasil pengujian diperlihatkan dalam tabel 1.

Tabel 1
Pengujian Black Box Halaman Form Deteksi

No	Data yang diuji	Alternatif Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Button choose file	-	Menampilkan galeri setelah klik tombol	Menampilkan galeri setelah klik tombol	Valid
2	Button proses	Memasukkan gambar dan field dari choose file berisi nama file	Menampilkan hasil prediksi dari gambar yang telah diproses	Menampilkan hasil prediksi dari gambar yang telah diproses	Valid
		Tidak memasukkan gambar	Menampilkan pesan <i>error</i>	Menampilkan pesan <i>error</i>	Valid
3	Text field input plat nomor	File gambar terdeteksi	Menampilkan teks setelah klik button proses	Menampilkan teks setelah klik button proses	Valid
		File gambar tidak terdeteksi	Tidak menampilkan teks setelah klik button proses	Tidak menampilkan teks setelah klik button proses	Valid
4	Button search	Text input plat nomor terdaftar dalam database	Menampilkan tabel informasi kendaraan	Menampilkan tabel informasi kendaraan	Valid
		Text input plat nomor tidak terdaftar dalam database atau field text input dalam keadaan kosong	Tidak menampilkan tabel informasi kendaraan	Tidak menampilkan tabel informasi kendaraan	Valid

4.3 Hasil Performa *Testing*

Pengujian yang dibuat terdiri dari proses *training* data dan proses *testing*. Pengujian menggunakan *dataset* alfanumerik yang diperoleh dari github terdapat 36352 *file* gambar biner. Lalu data tersebut dibagi dengan perbandingan 80:20 *data train* dan *data test*. Sehingga menghasilkan jumlah *data train* sebesar 29064 data dan *data test* sebesar 7288 data. Proses data train ditrainingkan dengan parameter iterasi sebesar 5 *epoch* dan *step per epoch* dengan 32 *batch size data train* yang berjumlah 908 data. Terdapat 4 konfigurasi CNN untuk melakukan pengujian pada tabel 2:

Tabel 2
Konfigurasi CNN

Model A	Model B	Model C	Model D
<i>10 weight layers</i>	<i>6 weight layers</i>	<i>3 weight layers</i>	<i>3 weight layers</i>
<i>input 28 x 28 RGB image</i>			
			<i>conv2-32 + relu</i>
			<i>Batch normalization</i>
			<i>maxpool</i>
<i>conv2-64 + relu</i>	<i>conv2-64 + relu</i>	<i>conv2-64 + relu</i>	<i>conv2-64 + relu</i>
<i>Batch normalization</i>	<i>Batch normalization</i>	<i>Batch normalization</i>	<i>Batch normalization</i>
<i>conv2-64 + relu</i>	<i>conv2-64 + relu</i>	<i>conv2-64 + relu</i>	
<i>Batch normalization</i>	<i>Batch normalization</i>	<i>Batch normalization</i>	
<i>maxpool</i>	<i>maxpool</i>	<i>maxpool</i>	<i>maxpool</i>
<i>conv2-128 + relu</i>	<i>conv2-128 + relu</i>		
<i>Batch normalization</i>	<i>Batch normalization</i>		
<i>conv2-128 + relu</i>	<i>conv2-128 + relu</i>		
<i>Batch normalization</i>	<i>Batch normalization</i>		
<i>maxpool</i>	<i>Maxpool</i>		
<i>conv2-256 + relu</i>			
<i>Batch normalization</i>			
<i>conv2-256 + relu</i>			
<i>Batch normalization</i>			
<i>conv2-256 + relu</i>			
<i>Batch normalization</i>			
<i>maxpool</i>			
<i>Flatten</i>			
<i>dropout(0.2)</i>	<i>dropout(0.2)</i>	<i>dropout(0.2)</i>	<i>dropout(0.2)</i>
<i>dense(512)</i>	<i>dense(512)</i>	<i>dense(512)</i>	
<i>dropout(0.2)</i>	<i>dropout(0.2)</i>	<i>dropout(0.2)</i>	
<i>dense(256)</i>	<i>dense(256)</i>		<i>dense(256)</i>
<i>dropout(0.2)</i>	<i>dropout(0.2)</i>		<i>dropout(0.2)</i>
<i>dense(128)</i>			
<i>dropout(0.2)</i>			

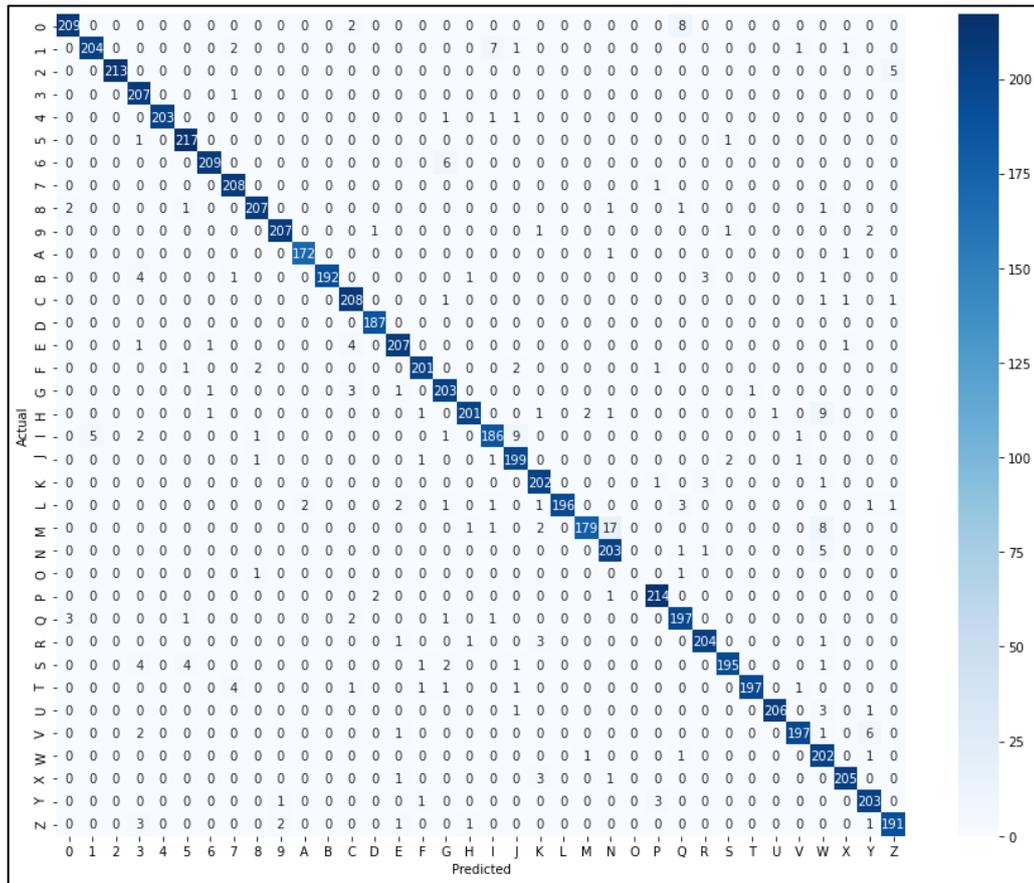
Konfigurasi CNN di atas memiliki jumlah parameter berbeda disetiap modelnya. Semakin banyak jumlah *layer*, proses *training* semakin lama. Hasil *training* dari konfigurasi model A, B, C dan D terdapat pada Tabel 3.

Tabel 3
Proses *Training* Model A-D terhadap 36352 data

Model A					
	<i>Accuracy</i>	<i>Loss</i>	<i>val_loss</i>	<i>val_accuracy</i>	Waktu
<i>Epoch 1</i>	0.7335	0.9593	0.4256	0.8938	279 detik
<i>Epoch 2</i>	0.9083	0.3517	0.1861	0.9535	284 detik
<i>Epoch 3</i>	0.9264	0.2800	0.1747	0.9547	282 detik
<i>Epoch 4</i>	0.9368	0.2457	0.1566	0.9588	286 detik
<i>Epoch 5</i>	0.9420	0.2283	0.2333	0.9453	289 detik
Model B					
	<i>Accuracy</i>	<i>Loss</i>	<i>val_loss</i>	<i>val_accuracy</i>	Waktu
<i>Epoch 1</i>	0.7414	0.9523	0.3174	0.9115	179 detik
<i>Epoch 2</i>	0.8893	0.4024	0.1742	0.9512	183 detik
<i>Epoch 3</i>	0.9136	0.3079	0.1423	0.9602	182 detik
<i>Epoch 4</i>	0.9252	0.2729	0.1290	0.9617	183 detik
<i>Epoch 5</i>	0.9304	0.2475	0.1219	0.9647	187 detik
Model C					
	<i>Accuracy</i>	<i>Loss</i>	<i>val_loss</i>	<i>val_accuracy</i>	Waktu
<i>Epoch 1</i>	0.7116	1.1347	0.2750	0.9273	120 detik
<i>Epoch 2</i>	0.8659	0.4898	0.2347	0.9366	124 detik
<i>Epoch 3</i>	0.8919	0.3976	0.2132	0.9410	162 detik
<i>Epoch 4</i>	0.9037	0.3426	0.1724	0.9549	168 detik
<i>Epoch 5</i>	0.9152	0.2987	0.1793	0.9491	165 detik
Model D					
	<i>Accuracy</i>	<i>Loss</i>	<i>val_loss</i>	<i>val_accuracy</i>	Waktu
<i>Epoch 1</i>	0.7625	0.8618	0.3026	0.9260	27 detik
<i>Epoch 2</i>	0.8881	0.3966	0.2132	0.9448	30 detik
<i>Epoch 3</i>	0.9067	0.3318	0.1721	0.9550	31 detik
<i>Epoch 4</i>	0.9163	0.2926	0.1670	0.9557	29 detik
<i>Epoch 5</i>	0.9238	0.2664	0.1608	0.9572	32 detik

Hasil pengujian pada model A dan C terjadi kenaikan pada akurasi *loss* data validasi dari salah satu *epoch*, sehingga pada model tersebut tidak optimal dan terjadinya *overfitting*. Sementara model B dan D merupakan model yang optimal dari konsistennya penurunan akurasi *loss* data validasi. *Filter* yang tepat untuk *training* data tersebut maksimal dengan nilai 128 dari model B memperoleh akurasi sebesar 93,04% dan *loss* sebesar 24,75%. Pada data *test* yang diuji menghasilkan akurasi sebesar 96,47% dan nilai *loss* sebesar 12,19%.

Evaluasi prediksi data *test* dari model B divisualisasikan pada gambar 18 dan tabel 4 menunjukkan perhitungan *confusion matrix*. Terdapat kategori huruf O yang tidak dapat dievaluasi karena jumlah data test yang sedikit. Pada evaluasi tersebut, terdapat banyak huruf yang mirip seperti huruf Y mirip huruf V, dan huruf N dan M yang diprediksikan saling mirip.



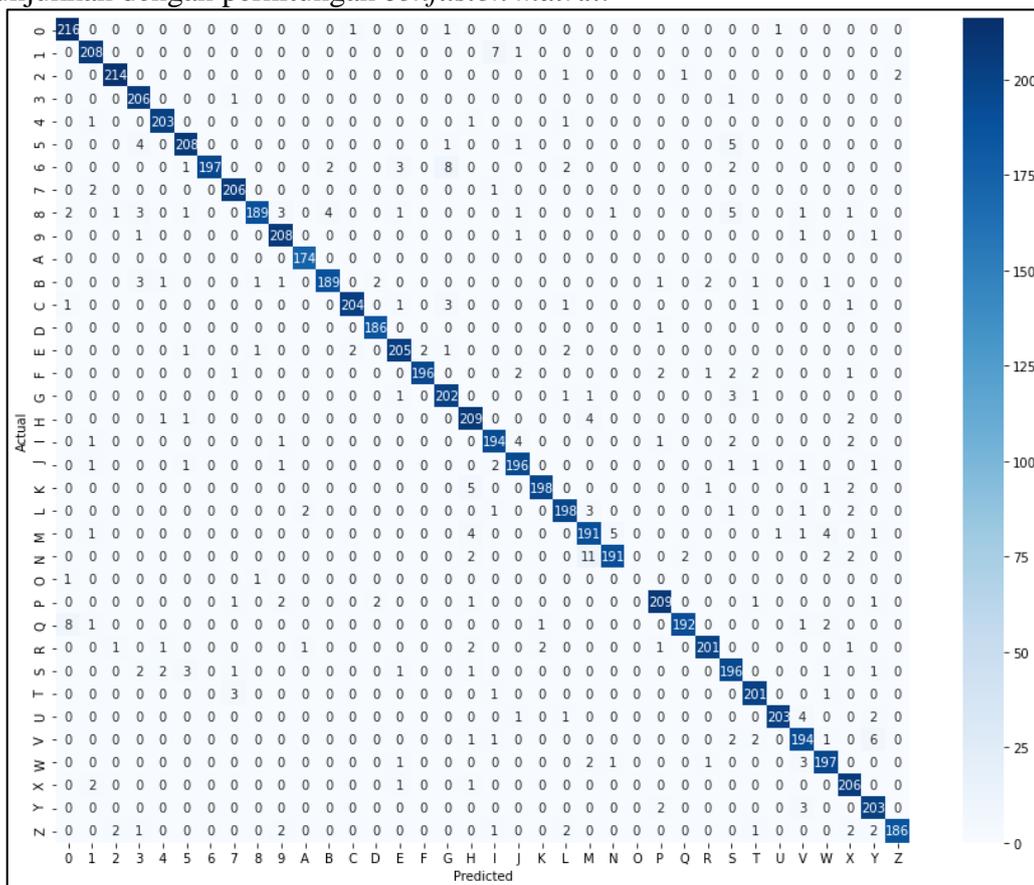
Gambar 18
 Visualisasi Prediksi 7288 Data Test Model B

Tabel 4
 Perhitungan Confusion Matrix untuk data test model B dengan total 7288 data

Kategori	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
0	0.98	0.95	0.97	219
1	0.98	0.94	0.96	216
2	1.00	0.98	0.99	218
3	0.92	1.00	0.96	208
4	1.00	0.99	0.99	206
5	0.97	0.99	0.98	219
6	0.99	0.97	0.98	215
7	0.96	1.00	0.98	209
8	0.98	0.97	0.97	213
9	0.99	0.98	0.98	212
A	0.99	0.99	0.99	174
B	1.00	0.95	0.97	202
C	0.95	0.98	0.96	212
D	0.98	1.00	0.99	187
E	0.87	0.97	0.97	214
F	0.98	0.97	0.97	207
G	0.94	0.97	0.95	209
H	0.98	0.93	0.95	217
I	0.94	0.91	0.92	205

J	0.93	0.97	0.95	205
K	0.95	0.98	0.96	207
L	1.00	0.94	0.97	208
M	0.98	0.86	0.92	208
N	0.90	0.97	0.93	210
O	0.00	0.00	0.00	2
P	0.97	0.99	0.98	217
Q	0.93	0.96	0.94	205
R	0.97	0.97	0.97	210
S	0.98	0.94	0.96	208
T	0.99	0.96	0.98	206
U	1.00	0.98	0.99	211
V	0.98	0.95	0.97	207
W	0.86	0.99	0.92	205
X	0.98	0.98	0.98	210
Y	0.94	0.98	0.96	208
Z	0.96	0.96	0.96	199
akurasi			0.96	7288
macro avg	0.94	0.94	0.94	7288
weighted avg	0.97	0.96	0.96	7288

Evaluasi prediksi data *test* dari model D divisualisasikan pada gambar 19 dan tabel 5 menunjukkan dengan perhitungan *confusion matrix*.



Gambar 19
Visualisasi Prediksi 7288 Data Test Model D

Tabel 5
 Perhitungan Confusion Matrix untuk data test model D dengan total 7288 data

Kategori	Precision	Recall	F1-score	Support
0	0.95	0.99	0.97	219
1	0.96	0.96	0.96	216
2	0.98	0.98	0.98	218
3	0.94	0.99	0.96	208
4	0.98	0.99	0.98	206
5	0.96	0.95	0.96	219
6	1.00	0.92	0.96	215
7	0.97	0.99	0.98	209
8	0.98	0.89	0.93	213
9	0.95	0.98	0.97	212
A	0.98	1.00	0.99	174
B	0.97	0.94	0.95	202
C	0.99	0.96	0.97	212
D	0.98	0.99	0.99	187
E	0.96	0.96	0.96	214
F	0.99	0.95	0.97	207
G	0.94	0.97	0.95	209
H	0.92	0.96	0.94	217
I	0.93	0.95	0.94	205
J	0.95	0.96	0.95	205
K	0.99	0.96	0.97	207
L	0.95	0.95	0.95	208
M	0.90	0.92	0.91	208
N	0.96	0.91	0.94	210
O	0.00	0.00	0.00	2
P	0.96	0.96	0.96	217
Q	0.98	0.94	0.96	205
R	0.98	0.96	0.97	210
S	0.89	0.94	0.92	208
T	0.95	0.98	0.96	206
U	0.99	0.96	0.98	211
V	0.92	0.94	0.93	207
W	0.94	0.96	0.95	205
X	0.93	0.98	0.95	210
Y	0.93	0.98	0.95	208
Z	0.99	0.93	0.96	199
Akurasi			0.96	7288
macro avg	0.93	0.93	0.93	7288
weighted avg	0.96	0.96	0.96	7288

Pengujian selanjutnya terhadap data segmentasi dari gambar-gambar plat yang diperoleh dari *kaggle*, terdeteksi karakter sebanyak 131 gambar dan menghasilkan data gambar karakter alfanumerik sebanyak 859 data. Data karakter yang telah disegmentasi, diperbanyak dengan teknik augmentasi, sehingga data yang dihasilkan 3817 data. Data segmentasi dibagi sama seperti pengujian sebelumnya dengan perbandingan 80:20 data train dan data test. Data yang dibagi menghasilkan data train sebanyak 3040 data dan data test sebanyak 777 data. Proses data train dengan iterasi dan *step per epoch* sama seperti data

Tabel 6
 Perhitungan Confusion Matrix untuk data test dengan total 777 data

Kategori	Precision	Recall	F1-score	Support
0	0.62	0.95	0.75	21
1	0.87	1.00	0.93	20
2	1.00	0.95	0.98	21
3	1.00	1.00	1.00	20
4	1.00	1.00	1.00	20
5	1.00	0.96	0.98	23
6	1.00	1.00	1.00	23
7	1.00	0.82	0.90	22
8	1.00	1.00	1.00	19
9	0.96	1.00	0.98	24
A	1.00	1.00	1.00	22
B	1.00	1.00	1.00	21
C	1.00	1.00	1.00	22
D	1.00	0.91	0.95	22
E	1.00	0.95	0.97	19
F	1.00	1.00	1.00	22
G	1.00	1.00	1.00	22
H	1.00	1.00	1.00	22
I	1.00	1.00	1.00	22
J	1.00	1.00	1.00	22
K	1.00	1.00	1.00	22
L	1.00	1.00	1.00	23
M	1.00	1.00	1.00	22
N	1.00	1.00	1.00	22
O	1.00	1.00	1.00	22
P	1.00	0.90	0.99	21
Q	1.00	0.52	0.98	21
R	1.00	1.00	1.00	22
S	0.95	1.00	0.95	21
T	0.88	1.00	0.96	22
U	0.96	1.00	0.89	22
V	1.00	1.00	1.00	22
W	1.00	1.00	1.00	22
X	1.00	1.00	1.00	22
Y	1.00	1.00	1.00	21
Z	0.95	1.00	0.97	21
akurasi			0.97	777
macro avg	0.98	0.97	0.97	777

4.4 Pengujian ANPR

Pengujian ANPR terhadap gambar yang diuji total 391 gambar dengan berbagai kondisi menghasilkan gambar yang terdeteksi sebesar 49 plat nomor yang dapat mendeteksi seluruh karakter, sementara gambar yang tidak dapat terdeteksi dikarenakan kondisi cahaya tidak rata, banyak gangguan dan kondisi rusak dengan jumlah 342 gambar plat nomor. Sebagian hasil pengujian tersebut dapat tidak akurat karena hasil *preprocessing* plat kurang baik, sehingga gambar bukan karakter ikut tersegmentasi.

Akurasi pengujian gambar terdeteksi mencapai 12,28% dan tidak terdeteksi dengan benar mencapai 87,72%. Pengujian terhadap karakter plat nomor kendaraan yang terdeteksi dan tidak terdeteksi dapat dilihat sample hasil pengujian pada Tabel 7

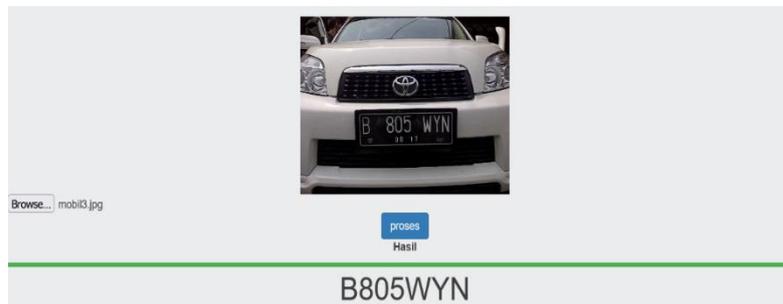
Tabel 7
Hasil Pengujian Terhadap 20 Gambar Plat Nomor Kendaraan

No.	Nama File	No. Plat	Hasil Pengenalan	Kesimpulan
1	mobil3.jpg 	B805WYN	B805WYN	Benar
2	plat4.jpg 	AD4526KU	AD4526KU	Benar
3	Screenshot_2021-02-11-11-01-08-54.jpg 	AD9121FF	AD9121FF	Benar
4	14.E\ 1129\ RP-07-18.jpg 	E1129RP	F1129RP	Salah
5	144.E\ 3063\ RS-07-22.jpeg 	E3063RS	E3063RS	Benar
6	152.E\ 3772\ SV-05-16.jpeg 	E3772SV	E3772SV	Benar
8	163.E\ 6804\ RV-02-19.jpeg 	E6804RV	4E6804RV	Salah

No.	Nama File	No. Plat	Hasil Pengenalan	Kesimpulan
9	169.E\ 2637\ WM-02-21.jpg 	E2637WM	E2637WM	Benar
10	313.E 5971 TO-08-22.jpg 	E5971TO	6591T6	Salah
11	314.E 3016 SB-10-18.jpeg 	E3016SB	E3SE	Salah
12	315.E 6091 TV-04-18.jpg 	E6091TV	E609TV	Salah
13	317.E 5064 SE-05-19.jpeg 	E5064SE	E5064S19E	Salah
14	320.E 3796 PY-05-20.jpg 	E3796PY	37965P20	Salah
15	237.E 3243 QN-12-17.jpeg 	E3243QN	tidak terdeteksi	Salah
16	81.E 3115 SN-06-20.jpg 	E3115SN	tidak terdeteksi	Salah
17	82.E 4993 TI-01-22.jpg 	E4993TI	tidak terdeteksi	Salah

No.	Nama File	No. Plat	Hasil Pengenalan	Kesimpulan
18	83.E 4147 QJ-06-19.jpeg 	E4147QJ	WE1K7QW	Salah
19	87.E 6673 ZQ-04-18.jpg 	E6673ZQ	E30618	Salah
20	290.E 2157 PAO-06-22.jpeg 	E2157PAO	NGNA	Salah

Pada gambar 22 adalah tampilan plat nomor terdapat pinggiran putih dapat dideteksi.



Gambar 22
Hasil dari Plat Pinggiran Putih

Pada gambar 23 adalah tampilan plat nomor yang tidak memiliki pinggiran putih dapat dideteksi.



Gambar 23
Hasil dari Plat Nomor tanpa Pinggiran Putih

Pada gambar 24 adalah tampilan hasil plat nomor terdapat kesalahan klasifikasi karakter. Hanya sebagian karakter diklasifikasikan dengan benar.



Gambar 24
 Hasil Karakter terklasifikasi sebagian

Pada gambar 25 adalah tampilan hasil deteksi melebihi karakter pada plat nomor karena kondisi plat kurang baik saat *preprocessing*.



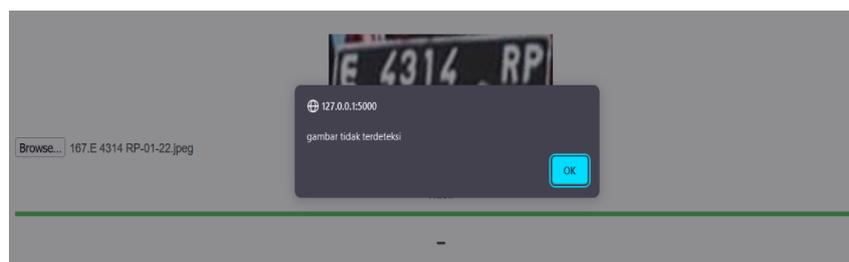
Gambar 25
 Hasil Karakter melebihi Karakter dari Gambar Asli

Pada gambar 26 terlihat hasil karakter plat nomor yang tidak sesuai dengan gambar plat. Kondisi plat nomor kurang baik.



Gambar 26
 Tampilan Hasil Karakter yang Salah

Pada gambar 27 terdapat notifikasi gambar tidak terdeteksi pada gambar plat dengan kondisi resolusi gambar kurang baik.



Gambar 27
 Hasil Karakter Tidak Terdeteksi

5. KESIMPULAN

Dari penelitian yang telah dilakukan, disimpulkan beberapa hal sebagai berikut:

- a. Penelitian ini terdapat tiga skenario pengujian. Pengujian pertama yaitu penggunaan CNN terhadap dataset alfanumerik, setelah di-*training* memperoleh akurasi dari konfigurasi CNN model B sebesar 93,04% dan model D sebesar 92,38%. Model B dan D merupakan model yang optimal karena tidak terjadi *overfitting* dan filter yang digunakan maksimal dengan nilai 128. Pengujian kedua terhadap segmentasi 131 plat nomor kendaraan menggunakan konfigurasi CNN model A memperoleh akurasi sebesar 97,17% dan loss sebesar 8,07%. Pengujian ketiga dari implementasi CNN pada web untuk deteksi gambar karakter plat nomor kendaraan yang berjumlah 391 gambar dengan model B yang memiliki filter 128. Hasil pengujian tersebut memperoleh akurasi karakter terdeteksi dengan benar sebesar 12,28% dan terdeteksi salah sebesar 87,72%. Gambar karakter yang banyak terdeteksi salah dikarenakan banyak gangguan sehingga beberapa karakter tidak dapat terklasifikasi dengan baik.
- b. Pada perangkat lunak yang dibuat memberikan hasil yang baik dari penginputan gambar menghasilkan informasi. Pembuatan web menggunakan framework flask sebagai *client-server* cukup baik dalam mengkoneksikan *database* dan mengelola pengenalan dengan model CNN.

6. DAFTAR PUSTAKA

- Albawi, Saad et al., 2017, "Layers of a Convolutional Neural Network." Ieee.
- Astawa, I. Nyoman Gede Arya et al., 2018, "Convolutional Neural Network Method Implementation for License Plate Recognition in Android." Proceedings - 2nd East Indonesia Conference on Computer and Information Technology: Internet of Things for Industry, EIconCIT 2018 176–79.
- Babu, K. Mahesh, Raghunadh, M. V., 2017, "Vehicle Number Plate Detection and Recognition Using Bounding Box Method." Proceedings of 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2016 (978):106–10.
- Chaudhuri, Arindam et al., 2017, "Optical Character Recognition System for Different Languages with Soft Computing, Studies in Fuzziness and Soft Computing", Springer.
- Gonzales, Rafael C., Woods, Richard E., 2018, "Digital Image Processing", Vol. 19, Pearson.
- Kadir, Abdul, 2019, "Langkah Mudah Pemrograman OpenCV & Python".
- Karpathy, Andrej, "CS231n Convolutional Neural Networks for Visual Recognition." Standford University., <https://cs231n.github.io/convolutional-networks/>, diakses tanggal 4 November 2020.
- P, I. Wayan Suartika E. et al., 2016, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101." Jurnal Teknik ITS 5(1):76.
- Paulus, Erick et al., 2019, "Analisa Dan Pengolahan Citra Digital - Studi Kasus Tulisan Tangan Sunda Kuno", Bitread Publishing.
- Perkap, 2015, "PERATURAN PRESIDEN REPUBLIK INDONESIA NOMOR 5 TAHUN 2015 TENTANG PENYELENGGARAAN SISTEM ADMINISTRASI MANUNGGAL SATU ATAP KENDARAAN BERMOTOR".
- Radzi, Syafeeza Ahmad, Khalil-Hani, Mohamed, 2011, "Character Recognition of License Plate Number Using Convolutional Neural Network." Lecture Notes in Computer

Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7066 LNCS(PART 1):45–55.

Snyder, Wesley E., Qi, Hairong, 2017, "Fundamentals of Computer Vision", CAMBRIDGE UNIVERSITY PRESS.

Solem, Jan Erik, 2012, "Programming Computer Vision with Python." Programming Computer Vision with Python 264.

Solichin, Achmad, Rahman, Zulfikar, 2015, "Aplikasi Identifikasi Nomor Kendaraan Berbasis Android Dengan Metode Learning Vector Quantization." Teknik Informatika 3(3):216–22.

Suyanto, 2018, "Machine Learning Tingkat Dasar Dan Tingkat Lanjut", INFORMATIKA.

Szeliski, Richard, 2011, "Computer Vision Algorithms and Applications", Springer.

Tyagi, Vipin, 2018, "Understanding Digital Image Processing", CRC Press.