# NASKAH PUBLIKASI

# IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK PENGENALAN WARNA KENDARAAN

**Program Studi Informatika**


Disusun oleh:
**TEGUH SETIAWAN**
**5160411059**



**PROGRAM STUDI INFORMATIKA**
**FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO**
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2020**

NASKAH PUBLIKASI

# IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK PENGENALAN WARNA KENDARAAN

Disusun oleh
**TEGUH SETIAWAN**
**5160411059**

Pembimbing

Donny Avianto, S.T., M.T.

Tanggal: 31 - 10 - 2020

# IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK PENGENALAN WARNA KENDARAAN

**Teguh Setiawan[1], Donny Avianto[2]**
*Program Studi Informatika, Fakultas Teknologi Informasi dan Elektro*
*Universitas Teknologi Yogykarta*
*Jl. Ringroad Utara Jombor Sleman Yogyakarta*
*E-mail : [1]baeteguh04@gmail.com, [2]donny@uty.ac.id*

**Abstrak**

*Convolutional Neural Network adalah salah satu metode machine learning dari pengembangan Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. Pengenalan kendaraan memainkan peranan penting dalam aplikasi Sistem Transportasi Cerdas, investigasi kriminal dan sebagainya di. Tugasnya adalah untuk mengenali jenis, warna dan plat nomor kendaraan yang ditargetkan. Warna adalah salah satu atribut dasar kendaraan, Oleh karena itu, pengenalan warna memainkan peran penting dalam pengenalan kendaraan. Cnn banyak digunakan dalam klasifikasi yang data masukannya berupa gambar, didalam penelitian ini digunakan metode cnn untuk klasifikasi warna kendaran. hasil yang diperoleh cukup baik menunjukan akurasi sebesar 78.61 % pada proses training.*

**Kata kunci :** *CNN, Transportasi, MLP, warna kendaraan*

## Abstract

*Convolutional Neural Network is a machine learning method from the development of Multi Layer Perceptron (MLP) which is designed to process two-dimensional data. CNN is included in the type of Deep Neural Network because of its deep network level and is widely implemented in image data. Vehicle recognition plays an important role in the application of the Smart Transportation System, criminal investigation and so on. Its job is to recognize the type, color and number plate of the targeted vehicle. Color is one of the basic attributes of a vehicle. Therefore, color recognition plays an important role in vehicle recognition. CNN is widely used in classifications where the input data is in the form of images, in this study the CNN method is used for vehicle color classification. The results obtained are quite good, showing an accuracy of 76.94% in the training process.*

***Keywords :*** *CNN, Transportation, MLP, vehicle color*

# 1. PENDAHULUAN

## 1.1 Latar Belakang

Departemen pengawasan lalu lintas sering menggunakan plat nomor untuk dilacak informasi kendaraan. Namun, identifikasi kendaraan tidak dapat hanya mengandalkan informasi plat nomor. Beberapa buatan manusia faktor seperti oklusi plat, dek kendaraan palsu dan plat multi-lisensi akan menyebabkan kegagalan plat nomor pengakuan. Warna kendaraan adalah properti penting untuk identifikasi kendaraan dan memberikan isyarat visual untuk hukum tindakan cepat pelaksanaan. Saat ini, pengenalan warna kendaraan otomatis telah menjadi topik penelitian penting dalam Transportasi Cerdas System (ITS) dan telah banyak digunakan, seperti video pengawasan, deteksi kejahatan dan sebagainya.

Pengenalan kendaraan memainkan peranan penting dalam aplikasi Sistem Transportasi Cerdas, untuk investigasi kriminal dan sebagainya. Tugasnya adalah untuk mengenali jenis, warna dan plat nomor kendaraan yang ditargetkan. Warna adalah salah satu atribut dasar kendaraan, Oleh karena itu, pengenalan warna memainkan peran penting dalam pengenalan kendaraan. Namun, karena dampak iluminasi yang kompleks, kondisi cuaca, kebisingan dan kualitas pengambilan gambar, pengenalan warna kendaraan telah menjadi tugas yang menantang. Pertama, pencahayaan, kebisingan dan kondisi cuaca khusus akan menyebabkan perubahan nyata dalam penampilan warna kendaraan. Misalnya, warna kendaraan di bawah iluminasi yang berbeda sangat bervariasi, yang membuat pengakuan sangat sulit. Kedua, mempertimbangkan perbedaan dalam posisi dan parameter kamera, ada perbedaan sudut dan jarak fokus yang signifikan. Selain itu, ukuran dan luasnya kendaraan dalam gambar sangat berubah, yang menimbulkan kesan hebat lainnya tantangan untuk pengenalan warna kendaraan. Akhirnya, gangguan kompleks dan komponen warna non-kendaraan semakin meningkat sulitnya pengenalan warna kendaraan.

Dalam tulisan ini peneliti menggunakan CNN untuk melakukan pengenalan terhadap warna kendaraan mobil, warna yang akan di klasifikasikan atau di kenali yaitu pada empat warna, putih, abu-abu, silver dan hitam. Karena warna kendaraan yang hampir serupa maka cahaya dan iklim alami, mudah menyebabkan kebingungan pengakuan warna. Maka dari itu peneliti menggunakan algoritma CNN, karena Convolutional Neural Network (CNN) adalah pembelajaran yang mendalam, metode yang dirancang khusus untuk klasifikasi gambar dan pengakuan berdasarkan jaringan saraf multi-layer dan CNN memiliki daya tahan tinggi terhadap penskalaan, kemiringan, dan deformasi visual lainnya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang maka rumusan masalah dari penelitian ini adalah, bagaimana mengenali warna kendaraan mobil menggunaka algoritma Convolutional Neural Network berdasarkan tingkat akurasi yang didapatkan dari hasil klasifikasi dari algoritma CNN.

## 1.3 Batasan Masalah

Pembatasan suatu masalah digunakan untuk megindari adanya pelebaran pokok masalah sehingga penelitian lebih terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian tercapai, beberapa batasan masalah dalam penelitian ini adalah sebagai berikut :

a. Data yang digunakan adalah citra kendaraan yang didapatkan dari hasil kamera atau dari internet.
b. Klasifikasi citra ini hanya mencakup 4 warna saja.
c. Metode yang digunakan adalah Convolutional Neural Network.
d. Bahasa pemrograman yang digunakan adalah Python dengan framework Tensorflow.

## 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai pada tugas akhir ini adalah :

a. Mengetahui implementasi metode Deep Learning menggunakan CNN untuk mengklasifikasikan citra warna kendaraan.
b. Mengetahui tingkat akurasi yang didapatkan dari hasil klasifikasi menggunakan CNN.

## 1.5 Manfaat Penelitian

Warna kendaraan adalah properti penting untuk identifikasi kendaraan dan memberikan isyarat visual untuk penegakan hukum tindakan cepat, pengenalan warna juga dapat mempersempit wilayah pencarian kendaraan sehingga dapat memudahkan untuk melakukan pencarian dengan menambahkan parameter warna.

# 2. KAJIAN PUSTAKA DAN TEORI

## 2.1 Kajian Hasil Penelitian

[7]Pembuatan, model, dan pengenalan warna (MMCR) kendaraan sangat menarik dalam beberapa aplikasi seperti penegakan hukum, bantuan pengemudi, pengawasan dan pemantauan lalu lintas. Tugas klasifikasi visual berbutir halus ini secara tradisional merupakan tugas yang sulit untuk komputer. Tantangan utama adalah perbedaan halus antara kelas (mis. BMW seri 3 dan BMW seri 5) dibandingkan dengan beberapa tugas klasifikasi tradisional seperti ImageNet. Baru-baru ini, ada upaya untuk merancang algoritma yang lebih akurat untuk MMCR seperti yang ada dalam karya Sochor et al di

dan Hsieh et al. Selain itu, banyak peneliti telah fokus pada pengumpulan dataset besar untuk memfasilitasi penelitian di bidang ini. Namun, kompleksitas metode saat ini dan / atau ukuran kecil dari dataset saat ini menyebabkan kinerja yang kurang optimal dalam kasus penggunaan dunia nyata. Dengan demikian, masih ada banyak kekurangan untuk agen atau entitas komersial yang ingin menggunakan perangkat lunak yang andal untuk tugas MMCR. Dalam tulisan ini, kami menyajikan sistem yang mampu mendeteksi dan menandai pembuatan, model dan warna kendaraan terlepas dari sudut pandang dengan akurasi tinggi. Model kami dilatih untuk mengenali 59 jenis kendaraan yang berbeda serta 818 model berbeda dalam apa yang kami yakini merupakan set terbesar yang tersedia untuk penggunaan komersial atau non komersial.

[9] Kami mengoptimalkan CNN untuk klasifikasi warna berdasarkan arsitektur ResNet. Evaluasi pertama dari keakuratan klasifikasi warna dilakukan dengan menggunakan dua set data terkontrol dengan kualitas baik dan buruk. Setiap kumpulan data berisi 5.934 gambar dengan semua model / model yang ada. Kedua set data berisi gambar dengan tampilan berbeda. Menunjukkan klasifikasi warna yang benar dan tingkat kesalahan masing-masing kelas. Kami menggunakan sepuluh kelas seperti dalam pelatihan kami. Kami menguji juga kumpulan data Stanford dan kami mendapatkan akurasi yang lebih tinggi daripada akurasi yang disajikan dalam. Kami tidak menggunakan bagian dari set data Stanford dalam pelatihan kami.

[10] Pengenalan data kendaraan adalah bagian kunci dari Intelligent Sistem Transportasi. Pengenalan data kendaraan meliputi pengenalan warna kendaraan, pengenalan plat nomor, dll. Warna memainkan peran yang sangat penting dalam identifikasi kendaraan. Pengenalan warna kendaraan di pemandangan alam akan menawarkan data bermanfaat dalam deteksi kendaraan, pelacakan kendaraan dan mengemudi otomatis sistem. Namun, mengidentifikasi warna kendaraan di lingkungan yang tidak terkontrol bisa menjadi tugas yang sulit. Pengenalan warna kendaraan dapat diperlakukan sebagai pembelajaran yang diawasi masalah terdiri dari preprocessing, ekstraksi fitur gambar dan klasifikasi fitur yang dihasilkan bersama dengan label. Gambar input sistem adalah gambar yang diambil dari kamera pinggir jalan. Terkadang gambar yang diambil di kamera pinggir jalan sepertinya tidak terlihat karena kabut, salju, dll. Penghapusan kabut dari gambar menggunakan teknik penghilangan kabut asap adalah langkah besar untuk mencapai hasil yang baik. Ada banyak algoritma penghapusan kabut gelap saluran sebelumnya, deteksi tepi, dan pelemahan warna sebelum dll. Di sini kita menggunakan algoritma saluran gelap sebelumnya untuk menghilangkan kabut.

[11] Intelligent Transport System (ITS) adalah sistem yang mengelola transportasi dari manajemen lalu lintas ke penegakan hukum. Salah satu objek penting yang banyak dieksplorasi oleh ITS adalah kendaraan dan propertinya, termasuk jenis, warna, dan plat nomor. Warna kendaraan adalah properti penting untuk identifikasi kendaraan dan memberikan isyarat visual untuk penegakan hukum tindakan cepat. Mengenali warna kendaraan adalah tugas yang sangat menantang karena beberapa faktor termasuk kondisi cuaca, kualitas video / akuisisi gambar, dan kombinasi strip kendaraan. Faktor pertama, kondisi cuaca, dapat secara dramatis mengubah iluminasi warna gambar akuisisi. Misalnya, jika gambar / video diambil pada kondisi kabut maka ada banyak white noise "lunak" yang ditambahkan ke gambar. Suara putih lembut berarti bahwa suara itu tidak acak tetapi terus dan dicampur dengan objek latar depan dan latar belakang. Kualitas perolehan video / gambar dipengaruhi oleh keputusan akhir sistem pengenalan warna kendaraan dan ketergantungannya pada sensor optik pada kamera. Kamera yang dapat menangkap objek dengan kecepatan tinggi direkomendasikan untuk ITS, tetapi tidak semua kamera yang terpasang di jalan dapat melakukannya. Banyak kamera yang dipasang di jalan hanya digunakan untuk memantau lalu lintas, pejalan kaki, dan kondisi jalan. Faktor terakhir adalah kombinasi strip kendaraan, yang sangat dipengaruhi oleh sistem pengenalan kendaraan. Pemilihan wilayah sangat penting untuk mengatasi masalah tersebut.

[12] Karena warna kendaraan yang serupa, pengaruhnya cahaya alami dan iklim, mudah menyebabkan kebingungan dalam pengenalan warna. Untuk mengatasi tantangan ini, para sarjana telah mengusulkan banyak karya inovatif, seperti desain manual metode fitur SIFT, histogram RGB standar, dan metode fitur tekstur, yang meningkatkan akurasi pengenalan warna kendaraan dalam berbagai derajat. Namun, dalam skenario yang sangat kompleks, metode ini tidak dapat memenuhi persyaratan aplikasi praktis. Convolutional Neural Network (CNN) adalah pembelajaran yang mendalam metode yang dirancang khusus untuk klasifikasi dan pengenalan gambar berdasarkan jaringan saraf multi-layer. Ini telah banyak digunakan dalam deteksi dan pengenalan . Ini mirip dengan struktur jaringan pembagian berat jaringan saraf biologis, yang dapat menghasilkan efek medan reseptif lokal dalam visi biologis, mengurangi kompleksitas model jaringan dan jumlah bobot. Selain itu, ia memiliki ketahanan tinggi terhadap terjemahan, penskalaan, kemiringan, dan deformasi visual lainnya.

## 2.2 Feature Extraction

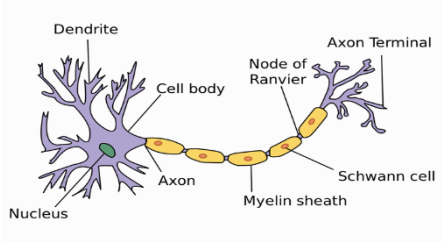
Kemampuan utama CNN adalah arsitektur yang mampu mengenali informasi prediktif suatu objek (gambar, teks, potongan suara, dan sebagainya) walaupun objek tersebut dapat diposisikan dimana saja pada *input*. Kontribusi CNN di *feature extraction layer* adalah pada *convolution* dan *pooling layer*. *Convolution* bekerja dengan prinsip *sliding window* dan *weight sharing* (mengurangi kompleksitas perhitungan). *Pooling layer* digunakan untuk merangkum informasi yang dihasilkan oleh suatu *convolution* (mengurangi dimensi). Sedangkan vektor hasil dari beberapa operasi *convolution* dan *pooling* pada multilayer perceptron dikenal sebagai *fully-connected layer* yang digunakan untuk melakukan suatu pekerjaan (misal klasifikasi).

## 2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independent sehingga menghasilkan hasil yang kurang baik. CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode Deep Learning, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode Machine Learning lainnya seperti SVM pada kasus klasifikasi objek pada citra.

## 2.4 Artificial Neural Network

Artificial Neural Network (ANN) merupakan suatu model komputasi paralel yang meniru fungsi dari sistem jaringan syaraf biologi otak manusia. Dalam otak manusia terdiri dari milyaran neuron yang saling berhubungan. Hubungan ini disebut dengan *Synapses*. Komponen neuron terdiri dari satu inti sel yang akan melakukan pemrosesan informasi, satu akson (*axon*) dan minimal satu dendrit. Informasi yang masuk akan diterima oleh dendrit. Selain itu, dendrit juga menyertasi akson sebagai keluaran dari suatu pemrosesan informasi.




*Gambar 2. 1* Jaringan Syarat Manusia

Cara kerja dari sistem syaraf diatas adalah bermula pada sinyal masuk melalui dendrit menuju cell body. Kemudian sinyal akan di proses didalam *cell body* berdasarkan fungsi tertentu (*Summation Proses*). Jika sinyal hasil proses melebihi nilai ambang batas (*treshold*) tertentu maka sinyal tersebut akan membangkitkan neuron untuk meneruskan sinyal tersebut. Sedang jika dibawah nilai ambang batasnya maka sinyal tersebut akan dihalangi (*inhibited*). Kemudian sinyal yang diteruskan akan menuju ke axon dan akhirnya menuju ke neuron lainnya melewati *synapse*.

## 2.5 Arsitektur Jaringan CNN

JST terdiri dari berbagai layer dan beberapa neuron pada masing-masing layer. Kedua hal tersebut tidak dapat ditentukan menggunakan aturan yang pasti dan berlaku berbeda-beda pada data yang berbeda. Pada kasus MLP, sebuah jaringan tanpa hidden layer dapat memetakan persamaan linear apapun, sedangkan jaringan dengan satu atau dua hidden layer dapat memetakan Sebagian besar persamaan pada data sederhana.

Dengan berkembangnya *deep learning*, ditemukan bahwa untuk mengatasi kekurangan MLP dalam menangani data kompleks, diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu berkembangnya deep learning dimana dalam satu model diberi beberapa layer untuk melakukan transformasi data sebelum data diolah menggunakan metode klasifikasi. Hal tersebut memicu berkembangnya model neural network dengan jumlah layer diatas tiga. Namun dikarenakan fungsi layer awal sebagai metode ekstraksi fitur, maka jumlah layer dalam sebuah DNN tidak memiliki aturan universal dan berlaku berbedabeda tergantung dataset yang digunakan.

Sebuah CNN terdiri dari beberapa layer. Berdasarkan aristektur LeNet5, terdapat empat macam layer utama pada sebuah CNN namun yang diterapkan pada TA ini hanya tiga macam lapisan lantara lain:

1. Convolution Layer
   Convolution Layer melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.
2. Subsampling Layer
   Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, subsampling juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode subsampling yang digunakan adalah max pooling. Max pooling membagi output dari convolution layer menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi.
3. Fully Connected Layer
   Layer tersebut adalah layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap neuron pada convolution layer perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah fully connected layer. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, fully connected layer hanya dapat diimplementasikan di akhir jaringan. Dalam sebuah jurnal oleh Lin et al., dijelaskan bahwa convolution layer dengan ukuran kernel 1 x 1 melakukan fungsi yang sama dengan sebuah fully connected layer namun dengan tetap mempertahankan karakter spasial dari data. Hal tersebut membuat penggunaan fully connectedlayer pada CNN sekarang tidak banyak dipakai.
4. Loss Layer
   Seluruh JST selalu diakhiri dengan sebuah loss layer, yaitu layer yang bertanggung jawab menghitung error dari jaringan untuk proses pelatihan sekaligus melakukan fungsi khusus jaringan. Untuk kasus klasifikasi, yang digunakan pada umumnya adalah softmax classifier. Softmax classifier menerima nilai input sejumlah kelas pada kasus klasifikasi, kemudian mengubahnya menjadi nilai normalized exponential menggunakan operasi softmax:

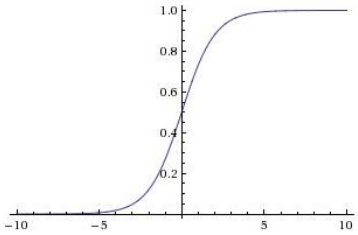
$$\boldsymbol{q} = \frac{e^{f\,yi}}{\sum_j e^{f\,ji}} \qquad (2.1)$$

5. Pooling Layer
   Layer ini ada setelah *convolutional layer*. Terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan bergeser pada seluruh *activation map*. *Pooling* yang digunakan adalah *max* dan *average pooling*. Misal kita gunakan *max pooling* 2 x 2 dengan stride 2, maka disetiap pergeseran filter, nilai maksimum pada area 2 x 2 pixel tersebut yang akan dipilih, sedangkan *average pooling* akan memilih nilai rata-ratanya.

## 2.6 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi non linear yang memungkinkan sebuah JST untuk dapat mentransformasi data input menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan hyperlane sederhana yang memungkinkan dilakukan klasifikasi. Dalam CNN terdapat fungsi aktivasi digunakan yaitu fungsi sigmoid. Fungsi sigmoid mentransformasi range nilai dari input x menjadi antara 0 dan 1 dengan bentuk distribusi fungsi.

a. Sigmoid




*Gambar 2. 2* Grafik Sigmoid

Sigmoid memiliki bentuk formula sebagai berikut:

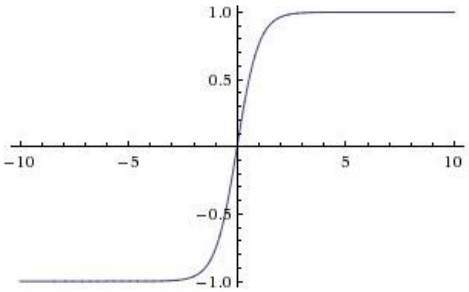
$$S(x) = \frac{1}{1 + e^{x-1}}. \qquad (2.2)$$

Sigmoid akan menerima angka tunggal dan mengubah nilai **x** menjadi sebuah nilai yang memiliki range mulai dari 0 sampai 1. Belakangan ini Sigmoid tidak disukai dan jarang digunakan, sigmoid memiliki kekurangan berupa:

a) Sigmoid mematikan gradient, property yang paling tidak diinginkan dari Sigmoid adalah ketika activation dari neuron mengeluarkan nilai yang berada pada ekor 0 atau 1, dimana gradient di wilayah ini hampir nol. Karena itu, jika gradient memiliki nilai yang sangat kecil, Sigmoid akan "mematikan" gradient dan kita sangat tidak menginginkan hal ini terjadi saat melakukan backpropagation.
b) Output dari Sigmoid tidak zero-centered. Hal ini berimplikasi pada kedinamisan saat melakukan

gradient descent, karna apabila data yang datang ke neuron selalu positif (e.g. x > 0 elementwise pada **f** = w^Tx+b)) maka gradient pada weights selama backpropagation akan menjadi semua positif atau semua negatif. Hal seperti ini dapat mengganggu proses training, tetapi hal ini tidak separah seperti hal yang ada pada point pertama(mematikan gradient).

b. Tanh



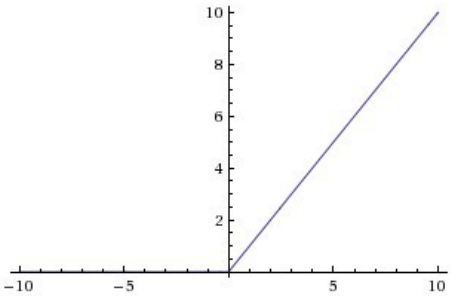

*Gambar 2. 3 Grafik Tanh*

$$\tanh(x)= \frac{2}{1+e^{-2x}} - 1 \qquad (2.3)$$

Tanh akan mengubah nilai input **x** nya menjadi sebuah nilai yang memiliki range mulai dari -1 sampai 1. Sama seperti Sigmoid, Tanh memiliki kekurangan yaitu bisa mematikan gradient, tetapi kelebihannya adalah output yang dimiliki Tanh merupakan zero-centered. Dalam praktiknya Tanh lebih menjadi pilihan jika dibandingkan dengan Sigmoid. Yang perlu dicatat adalah tanh hanyalah merupakan Sigmoid yang dikembangkan.

c. Relu

ReLu merupakan fungsi aktifasi pada Artificial Neural Network yang saat ini banyak digunakan dikarenakan, berikut rumus ReLu.




*Gambar 2. 4 Grafik Relu*

Formulanya dalah sebagai berikut :

$$(x)=\max(0,x) \qquad (2.4)$$

Dimana x adalah input neuron ini juga dikenal sebagai fungsi ramp dan analog dengan rektifikasi half-wave pada teknik elektro. Unit linier yang dikoreksi memiliki output 0 jika input kurang dari 0, dan output mentah dinyatakan. Artinya, jika input lebih besar dari 0, outputnya sama dengan input. Fungsi ReLU lebih mirip neuron seperti pada tubuh manusia. Aktivasi ReLU pada dasarnya sebuah fungsi aktivasi non-linier yang paling sederhana. Bila mendapat input positif, turunannya hanya 1, dengan kata lain, aktivasi hanya men-threshold pada nilai nol. Penelitian menunjukkan bahwa ReLu menghasilkan pelatihan yang lebih cepat untuk jaringan besar.

## 2.7 Adam Optimizer

Adam adalah algoritma pengoptimalan yang dapat digunakan sebagai ganti dari prosedur stochastic descent klasik untuk memperbaharui weight network secara iteratif berdasarkan data training. Adam pertama kali di presentasikan oleh Diederik Kingma dari OpenAI dan Jimmy Ba dari University of Toronto dalam papaer mereka di 2015 ICLR yang berjudul "Adam: A Method for Stochastic Optimization". Adam adalah algoritma yang populer di bidang depp learning karena ia mencapai hasil yang baik dengan cepat. Hasil empiris menunjukan bahwa Adam bekerja dengan baik dalam praktiknya dan lebih baik dibandingan dengan stochastic optimization lainnya. Dalam makalah aslinya, Adam dipergakan secara empiris untuk menunjukan bahwa konvergensi memenuhi harapan analisis teoristis. Adam diaplikasikan pada algoritma regresi logistik pada pengenalan digit MNIST dan dataset analisis sentimen IMDB, algoritma Multilayer Perceptron pada dasets MNIST dan Convolutional Neural Network pada datsets pengenalan gambar CIFAR-10.

## 2.8 Confusion Matrix

Confusion Matrix adalah sebuah metode yang biasa digunakan untuk perhitungan akurasi, recall, precision, dan error rate. Dimana, precision mengevaluasi kemampuan sistem untuk menemukan peringkat yang paling relevan, dan didefinisikan sebagai presentase dokumen yang di retrieve dan benar-benar relevan terhadap query. Recall mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai presentase dokumen yang relevan terhadap query. Accuracy merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus dan error rate merupakan kasus yang diidentifikasi salah dengan jumlah seluruh kasus.

*Table 2. 1 Confusion Matix*

| Class | 1 | 2 |
|---|---|---|
| 1 | **True Positive** | **False Positive** |
| 2 | **False Negative** | **True Negative** |

Berdasarkan table 2.1 Confusin Matrix :

a. True Positive

Merupakan jumlah data positive yang diprediksi benar.

b. False Positive

Merupakan jumlah data negative yang diprediksi benar.

c. False Negative

Merupakan jumlah data negative yang diprediksi sebagai data positive.

d. True Negative

Merupakan jumlah data positive yang diprediksi sebagai data positive.

Bagaimana kita dapat melakukan perhitungan berbagai performance matrix untuk mengukur kinerja model yang telah dibuat :

a. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.5)$$

Keseluruhan data True Positive dan True Negative dibagi dengan keseluruhan data.

b. Precision

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

Keseluruhan data yang diprediksi benar True Positive dibagi dengan jumlah pasion yang diprediksi True Positive dan False Negative.

c. Recall

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

Keseluruhan data yang di prediksi benar True Positive dibagi dengan keseluruhan data sebenarnya.

d. F1 Score

$$F1\ Score = \frac{2(Recall * precision)}{(Recall + Precission)} \quad (2.8)$$

F1 Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan

## 3. METODOLOGI PENELITIAN

### 3.1 Bahan/Data

#### 3.1.1 Data Yang Diperoleh

Untuk dapat melakukan penelitian diperlukan data citra, citra - citra yang dikumpulkan menjadi satu yaitu kumpulan citra dengan kelas tertentu disebut sebagai dataset. Dataset diperlukan untuk proses pelatihan mesin dan proses pengjuian mesin. Dataset yang digunakan dalam pelatihan ini yang didapatkan dari stanford https://ai.stanford.edu/~jkrause/cars/car_dataset.html. Dataset yang diambil dari stanford awalnya digunakan untuk pengenalan kendaraan mobil roda empat dengan 16,185 citra dan 196 kelas. Dataset tersebut digunakan untuk melakukan pengenalan merek kendaraan tetapi peneliti hanya menggunakan 2000 data yang di ambil dari dataset stanford dan membuat dataset baru dengan empat kelas warna yang menjadi objek penelitian.

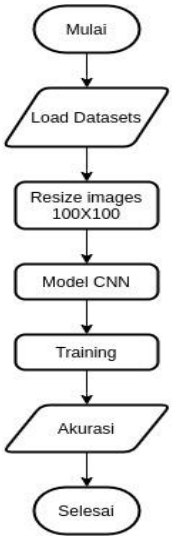
### 3.2 Tahapan Penelitian

#### 3.2.1 Rancangan Datasets

Penggunaan data pada metode CNN yaitu berupa data gambar. Model CNN akan berjalan dengan baik ketika data train gambar yang banyak. Sehingga model dapat belajar mengenali gambar tersebut. Dalam melakukan rancangan datasets untuk penelitian ini diambil dari *stanford cars datasets.* Datasets tersebut mempunyai 16,185 data gambar dengan 192 kelas, data gambar yang ada didalam *datasest* tersebut mempunyai beragam warna mobil dengan jenis dan merk yang sangat beragam. Namum pada penelitian ini hanya menggunakan empat warna mobil yaitu warna putih, abu-abu, hitam, dan silver, diakrenakan ke empat warna tersebut memiliki rentang warna yang berdekatan dan menggunakan 2000 data gambar yang di bagi menjadi 1600 data gambar untuk data *training* dan 400 data gambar sebagai data *validation*. Dalam membuat *datasets* baru dari *datasets* yang sudah ada dilakukan manual dengan memilih gambar yang sesuai untuk penelitian.

#### 3.2.2 Perancangan Sistem

Dalam penelitian ini mengunakan satu program untuk melakukan klasifikasi, praproses dalam penelitian ini hanya melakukan *resize* gambar, dikarenakan gambar yang diperoleh dari *cars datasest* tidak mempunyai ukuran piksel yang sama sehingga jika langsung digunakan untuk proses pelatihan pada program maka hasilnya akan buruk




*Gambar 3. 1Model CNN*

1. Program akan meload data berupa *datasets* gambar yang akan digunakan untuk training data.
2. Pada saat program meload *datasest* maka akan dilakukan *resize* gambar mejadi 100x100 piksel supaya data gambar mempunyai citra yang sama didalam *datasets.*
3. Proses selanjutnya adalah memasukan data inputan berupa *datasets* gambar kedalam model yang sudah di buat, model yang dibuat didalamnya terdapat beberapa layer konvolusi *maxpooling* dan *hidden layer*.
4. Setelah dimasukan dalam model lalu mendapatkan *feature* dari citra, training ini adalah memproses berupa data *feature* gambar. Dalam proses ini proram belajar mengenali pola dari gambar, yang kemudian mendapatkan akurasi dari proses belajar.
5. Setelah training selesai maka program mempunyai nilai *weight*, nilai tersebut digunakan untuk melakukan prediksi terhadap data test.

#### 3.2.3 Rancangan *Convolutional Neural Network*

Setelah dilakukan pembuatan data, langkah selanjutnya adlaah melakukan pelatihan model CNN. Umumnya CNN memiliki 2 tahapan, yaitu tahap *feature learning* dan *classification*. Input gambar pada model CNN menggunakan citra yang berukuran 100x100x3. Angka tiga ynag dimaksud adalah sebuah citra yang memiliki 3 *channel* yaitu *Red, Green, dan Blue* (RGB). Citra masukan akan diproses terlebih dahulu melalui proses konvolusi pada rancangan ini memiliki enam lapisan konvolusi, setiap layer konvolusi memiliki ukuran kernel yang sama dan setiap dua layer konvolusi memiliki jumlah filter yang sama. Kemudian dilakukan proses *flatten* atau proses mengubah *feature map* hasil *pooling layer* kedalam bentuk vector. Proses ini biasa disebut dengan tahap *fully Connected layer*.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Analisis Non Fungsional

Kebutuhan non fungsional yaitu kebutuhan yang diperlukan untuk membangun sistem. Kebutuhan non fungsional dibagi menjadi kebutuhan perangkat lunak dan kebutuhan perangkat keras.

1. Kebutuhan Perangkat Lunak

Merupakan perangkat lunak yang digunakan untuk membangun program. Berikut prangkat lunak yang dibutuhkan :

   a. Bahasa Pemrograman Python 3
   b. Jupyter Lab
   c. Cuda Toolkit 10.1
   d. CUDNN 7.6

2. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk menjalankan sistem. Adapun beberapa perangkat keras yang digunakan antara lain:

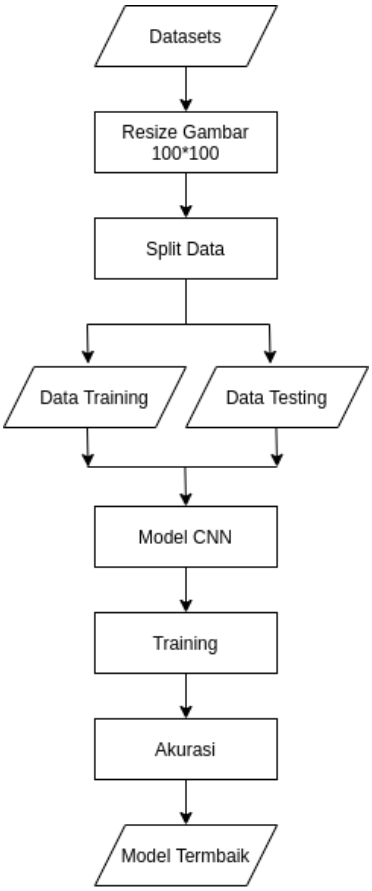
   a. Laptop Acer Swift 3
   b. Processor Core i5 8265U
   c. Ram 8GB
   d. GeForce MX150 2 GB VRam

### 4.2 Desain Logik

Desain Logik digunakan untuk menggambarkan sistem yang akan dibangun. Rancangan sistem pada penelitian ini berfokus pada alur algoritma yang digambarkan melalui *flowchar*. *Flowchart* merupakan diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah atau halannya program dari sistem yang dibuat.

#### 4.2.1 FlowChart Umum CNN

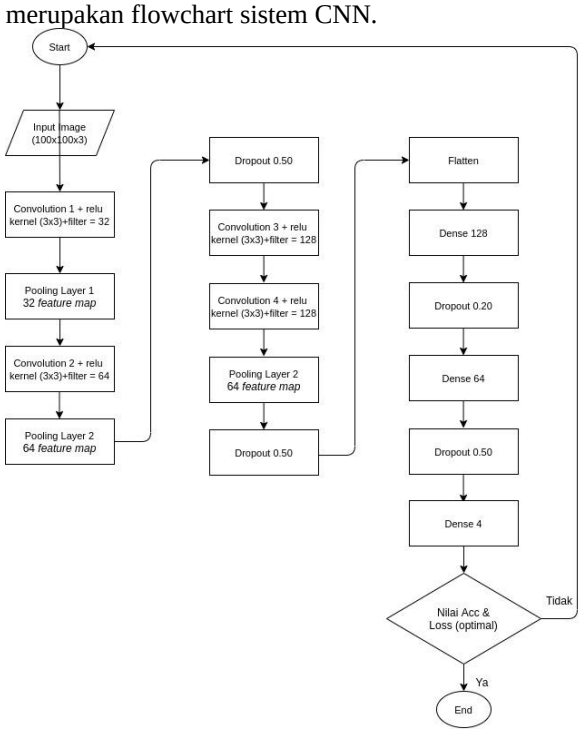
*Flowchart* umum CNN adalah menggambarkan bagaiamana sistem CNN yang di pakai pada penelititan ini secara umum. Berikut Gambar ... merupakan *flowchar* umum dari CNN.




*Gambar 4. 1Flowchart CNN*

### 4.2.2 Flowchart Sistem CNN

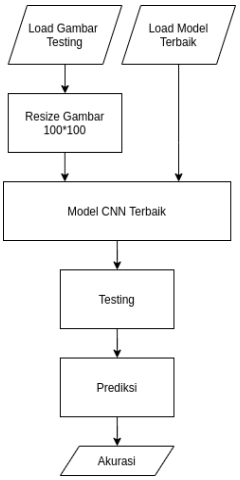
Flowchart Sistem CNN adalah alur lebih detail bagaimana datasets masuk sebagai input pada program di olah oleh layer konvolusi, pooling, dropout, untuk di ambil feature dari masing-masing gambar dalam bentuk matriks, kemudian dari matriks tersebut di ubah menjadi vektor yang kemudian digunakan untuk input pada jaringan syaraf tiruan. Berikut gambar ... merupakan flowchart sistem CNN.




*Gambar 4. 2 Sistem CNN*

### 4.2.3 Flowchart Pengujian

Setelah melakukan pelatihan program terhadap data yang sudah di siapkan, dari hasil training tersebut program mengenali pola dari gambar yang diberikan dengan memunculkan akurasi dan akurasi validasi, dari hasil pelatihan tersebut juga dapat menyimpan model dan nilai bobot terbaik pada saat pelatihan data yang bisa disimpan. Model dan nilai bobot terbaik tersebut yang akan digunakan pada proses pengujian dengan model yang sudah disimpan, maka proses pengujian akan meload data gambar pengujian, model terbaik dari proses pelatihan dan nilai bobot, yang akan digunakan untuk memprediksi data gambar baru pada proses pengujian.




*Gambar 4. 3 Fowchart Pengujian*

## 4.3 Skenario Pelatihan CNN

Pelatihan Convolutional Neural Network dilakukan menggunakan dua algoritma pengoptimalan yaitu adam dan SGD, dalam melakukan uji coba peneliti mencoba menggunakan dua varian learning rate yaitu 0,001 dan 0,0001, kemudian dalam melakukan pengenalan atau pelatihan terhadap mesin peneliti menggunakan beberapa varian epoh untuk mengetahui epoch berapa yang terbaik dan tidak adanya overfitting terhadap sistem, epoch yang digunakan yaitu 10,20,30,40 dan 50.

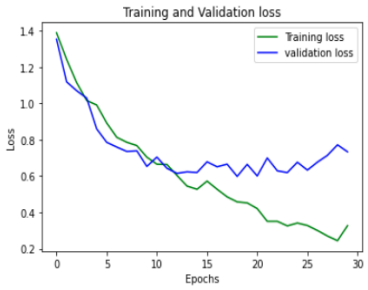
Besaran datasets untuk sistem ini adalah dengan menggunakan 1600 data gambar sebagai data training dan 400 data gambar sebagai data validation yang digunakan juga pada saat proses training. Dan sebanyak 390 data gambar untuk datasets testing. Data validation adalah data yang di ambil dari data training, besaran data training yaitu sebanyak 2000 data gambar dan di split sebanyak 20% untuk data validation. Dan data testing adalah data gambar yang belum pernah digunakan untuk melakukan proses pengenalan pola gambar oleh sistem, gambar pada data testing adalah data gambar baru yang belum pernah dilakukan untuk pelatihan.

Dalam melakukan skema pelatihan, sistem menggunakan dua metode optimasi untuk membandingkan algoritma mana yang baik untuk digunakan pada sistem, epoch pada proses pelatihan sangat penting karena seberapa lama sistem akan terus belajar pola data.

*Table 4. 1 Training Model 1*

| Epoch | Optimizer | Accuracy Training | Accuracy Testing | Learning Rate |
|---|---|---|---|---|
| 10 | Adam | 71,00% | 69,44% | 0,0001 |
| 20 | Adam | 73,00% | 76,94% | 0,0001 |
| 30 | Adam | 75,25% | 74,72% | 0,0001 |
| 40 | Adam | 75,00% | 75,00% | 0,0001 |
| 50 | Adam | 74,50% | 73,33%s | 0,0001 |

Pada table 4.1 Training Model 1, merupakan model yang digunakan untuk melakukan pelatihan terhadap data dengan algoritma optimasi menggunakan adam dan besaran learning rate yang tertera pada table. Table 2.2 membandingkan besaran epoch akan mempengaruhi akurasi dan overfitting terhadap proses pelatihan, accuracy training tertinggi terdapat pada epoch 30 sebesar 75,25% namun pada accuracy terhadap data testing accuracy justru menurun pada angka 74,72%. Dapat dilihat pada gambar 4.4 Training Model Epoch 30, pada saat melakukan pelatihan data, terjadi overfitting terhadap data validation sehingga acuraccy pada saat pelatihan cukup tinggi namun ketika dilakukan uji coba terhadap data testing accuracy menurun.




*Gambar 4. 4 Training Model 1 Epoch 30*

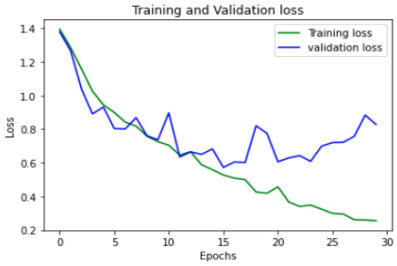
Kemudian accuracy testing tertinggi terdapat pada epoch 20 sebesar 76,94%, namun pada training mempunyai accuracy sebesar 73,00% yang berarti cukup bagus karena adanya peningkatan sebesar 3,94% terhadap data testing. Dapat dilihat pada gambar 4.4 Model Training Epcoh 30, terlihat pada saat epoch menyentuh angka 15 sampai 20, grafik antara training loss dan validation loss tidak seimbang, dan validation loss mempunyai nilai yang cukup jauh perbedaannya dengan training loss. Maka ketika kita menemukan hal seperti itu dinamakan overfitting pada data training, sehingga menyebabkan accuracy pada data testing kebanyakan akan tidak lebih baik dari accuracy training. Hal tersebut juga bisa di lihat pada gambar 4.5 Training Model Epoch 20, hal yang akan menunjukan overfitting terhadap data training juga dapat dilihat sudah mulai adanya perbedaan nilai yang cukup jauh antara training loss dengan validation loss.

*Gambar 4. 5 Training Model 1 Epoch 20*

*Table 4. 2 Training Model 2*

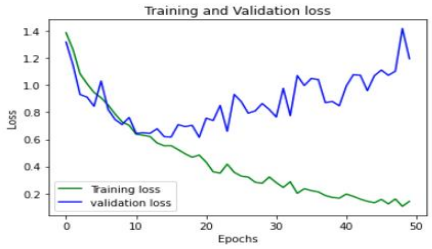
| Epoch | Optimizer | Accuracy Training | Accuracy Testing | Learning Rate |
|---|---|---|---|---|
| 10 | Adam | 67,25% | 65,56% | 0,001 |
| 20 | Adam | 76,00% | 72,78% | 0,001 |
| 30 | Adam | 77,25% | 74,44% | 0,001 |
| 40 | Adam | 77,00% | 70,00% | 0,001 |
| 50 | Adam | 76,00% | 75,83% | 0,001 |

Pada training model kedua pada table 4.2 yaitu menggunakan adam sebagai algoritma optimasi namun dengan merubah learning rate menjadi lebih besar, hasil yang di dapat untuk accuracy training epoch 30 memiliki accuracy tertinggi sebesar 77,25% namun pada saat dilakukan pengujian terhadap data testing accuracy yang didapat kurang bagus yaitu hanya sebesar 74,44% ada penurunan accuracy antara training dan testing. Jika melihat accuracy testing maka yang paling besar hasil accuracynya adalah dengan menggunakan epoch 50, dan accuracy testing sebesar 75,83% namun tetap adanya penurunan jika di bandingkan dengan accuracy training yang menandakan model tersebut kurang bagus.




*Gambar 4. 6 Training Model 2 Epoch 30*

Pada saat melakukan proses training dengan menggunakan learning rate yang lebih tinggi yaitu sebesar 0,001 accuracy testing tidak lebih baik dari pada accuracy training oleh karena itu learning rate dengan besaran 0,001 tidak cocok untuk data yang peneliti gunakan dikarenakan accuracy hanya bagus pada saat proses trainig tetatpi tidak untuk proses testing.




*Gambar 4. 7 Training Model 2 Epoch 50*

Kemudian pada table 4.3 Training Model 3 yaitu peneliti menggunakan algoritma *Stochastic Gradient Descent* atau biasa disebut dengan SGD, dengan menggunakan learning rate sebesar 0,0001 maka hasilnya tidak lebih baik menggunakan algoritma

Adam dengan learning rate sebesar 0,0001. Namun accuracy selalu menunjukan peningkatan, accuracy testing konsisten lebih besar daripada accuracy training pada semua epoch yang di ujikan.

*Table 4. 3 Training Model 3*

| Epoch | Optimizer | Accuracy Training | Accuracy Testing | Learning Rate |
|---|---|---|---|---|
| 10 | SGD | 50,25% | 56,94% | 0,0001 |
| 20 | SGD | 55,75% | 58,61% | 0,0001 |
| 30 | SGD | 57,50% | 63,89% | 0,0001 |
| 40 | SGD | 54,75% | 61,94% | 0,0001 |
| 50 | SGD | 63,25% | 66,39% | 0,0001 |

Jika ketika menggunakan algoritma adam sebagai pengoptimalan pada sistem, maka pada epoch mulai dari ke-15 sehingga seterusnya menunjukan perbedaan angka yang cukup jauh antara training loss dengan validation loss. Namun ketika menggunakan algoritma SGD cenderung tidak adanya overfitting terhadap data pelatihan seperti menggunakan Adam. Training loss dan validation loss memiliki nilai yang konsisten cenderung selalu mengecil, hanya saja accuracy yang didapat tidak lebih baik daripada menggunakan algoritma Adam sebagai algoritma pengoptimalan.

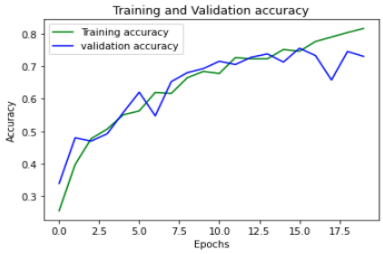
*Gambar 4. 8 Training Model 3 Epoch 50*

Jika dilihat dari hasil perbandingan maka peneliti menggunakan model dengan menggunakan algoritma Adam sebagai optimasi dengan learning rate sebesar 0,0001 dengan menggunakan epoch sebanyak 20, karena model tersebut tidak mengalami overfitting dalam melakukan training memang akurasi pada saat training bukan yang paling besar namun jika model tersebut digunakan untuk testing maka akurasi dari model tersebut yang tertinggi.

## 4.4 Hasil

Penelitian ini melakukan klasifikasi terhadap empat warna mobil yaitu warna putih, abu-abu, silver, dan putih. Proses program ini diawali dengan memuat data gambar lalu dilakukan training menggunakan CNN, proses training adalah untuk menghasilkan model dan nilai yang baik, yang akan digunakan pada bagian pengujian. Parameter untuk mengukur tigkat keberhasilan model adalah nilai akurasi. Proses training menggunakan packages Keras pada python dengan back-end tensorflow. Keras merupakan salah satu modul yang dibuat oleh Google untuk mempermudah dalam research mengenai neural network dan mampu berjalan diatas tensorflow, theano, MXNet.

*Gambar 4. 9 Training and Validation Loss*




*Gambar 4. 10 Training and Validation Accuracy*

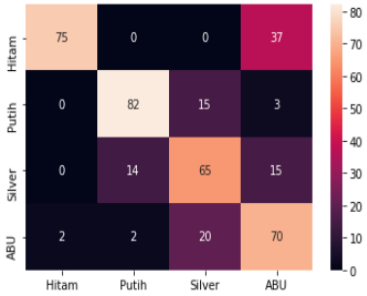
Penelitian dengan menggunakan datasest yang sudah dibuat, pada saat program pelatihan dijalankan menggunakan sebanyak 20 epoch, dengan *batch size* sebesar 32, *batch size* merupakan banyak data yang dimasukan ke proses pelatihan dalam satu iterasi. Ukuran *batch size* mempengaruhi akurasi dan penggunaan memory pada GPU. Dengan menggunakan *adam* sebagai metode pengoptimalan dengan *learning rate* sebesar 0.0001 .

*Table 4. 4 Table Classification Report Training*

| Class | Precision | Recall | F1-score | Support | Accuracy Class |
|---|---|---|---|---|---|
| Hitam | 0,97 | 0,67 | 0,79 | 112 | 66.9% |
| Putih | 0,84 | 0,82 | 0,83 | 100 | 82% |
| Siver | 0,65 | 0,69 | 0,67 | 94 | 69.1% |
| Abu | 0,56 | 0,74 | 0,64 | 94 | 74,4% |
| Overall accuracy | | | | | 73,00% |

Melihat dari data *clasification report* pada saat *training* nilai pada setiap kelas yang berbeda, kelas mobil dengan warna putih merupakan kelas paling baik dalam proses *training* dapat dilihat pada table 4.4 *Table Classification Report*, nilai *precision*, *recal* dan *f1-score* memiliki nilai rata rata yang paling baik diantara semua kelas. Pada gambar 4.11 *Confusion Matrix Training* yang menunjukan *confusion matrix* kelas mobil berwarna putih pada *validation* memiliki jumlah data gambar yang paling banyak menghasilkan *True Positif* dan kelas terbaik kedua adalah mobil dengan warna hitam, kelas dengan warna silver dan abu-abu memiliki nilai yang cukup baik, karena niali

*True Positif* nya masih lebih tinggi. Akurasi yang didapat pada saat training data mengasilkan sebesar 73,00% .



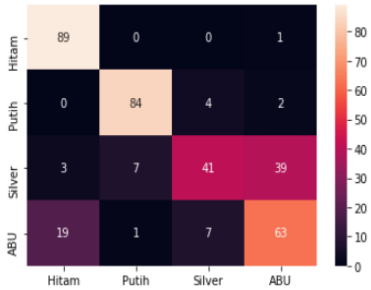

*Gambar 4. 11 Confusion Matrix Training*

Proses pengujian dapat dilihat hasinya melalui Table 4.8 *clasification report testing*, nilai akurasi pada proses pengujian memiliki akurasi sebesar 76.94% dari 360 data gambar yang dijadikan input sebagai data pengujian. Nilai *precision* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf, kelas warna kendaraan putih memiliki nilai yang paling tinggi, pada nilai recall adalaah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Dan nilai *f1-score* yang merupakan perbandingan rata-rata *precision* dan *recall*.

*Table 4. 5 Table Classification Report Testing*

| Class | Precision | Recall | F1-score | Support | Accuracy Class |
|---|---|---|---|---|---|
| Hitam | 0,83 | 96 | 0,89 | 90 | 95,5% |
| Putih | 0,90 | 0,91 | 0,91 | 90 | 91.1% |
| Siver | 0,77 | 0,53 | 0,63 | 90 | 53.3% |
| Abu | 0,62 | 0,72 | 0,67 | 90 | 72.2% |
| Overall accuracy | | | | | 76,94% |

Pada gambar 4.12 *Confusion Matrix Testing* kelas mobil berwarna hitam memiliki jumlah *True Positif* yang paling besar, terprediksi seluruhnya dengan benar, dan kelas yang memiliki nilai yang rendah pada kelas mobil berwarna silver hanya memprediski 41 data gambar yang memiliki *True Positif* dari keseluruhan data pengujian. Namun hasil yang di hasilkan dari proses pengujian masih cukup baik karena memiliki nilai *True Positif* dari masing masing kelas yang masih lebih tinggi dari hasil yang tidak terprediksi tidak benar.




*Gambar 4. 12 Confusion Matrix Testing*

## 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan hasil analisis selama penelitian dari hasil uji coba pada proses pelatihan dan pengujian menghasilkan beberpa kesimpulan yaitu :

1. Dalam membangun model convolutional neural network untuk melakukan klasifikasi terhadap warna kendaraan mobil memperoleh akurasi yang cukup tinggi, model dapat mengenali warna kendaraan dengan cukup baik.
2. Penggunaan GPU dalam melakukan komputasi dengan CUDA dan CuDNN menghaislkan percepatan proses pelatihan dan pengujian.
3. Model *convolutional neural network* yang dibuat pada penelitian ini menggunakan input data gambar dengan ukuran 100 x 100 dengan 3 *channel* warna, menggunakan adam sebagai optimasi dengan nilai *learning rate* sebesar 0.0001, dengan 4 layer konvolusi dengan ukuran kernal 3 x 3, dan menggunakan *pooling layer* dengan ukuran 2 x 2.
4. Proses pelatihan model menggunakan *input* data gambar sebanyak 2000 data, dengan data *validation* sebanyak 400 yang diatur menggunakan *train test spilt.* Menghasilkan akurasi klasifikasi warna kendaraan sebesar 73,00%.
5. Proses pengujian model menggunakan input data menggunakan data gambar sebanyak 360 dengan masing masing kelas berisi 90 data gambar, dan menghasilkan akurasi sebesar 76.94%.
6. Besaran *memory* juga mempengaruhi dalam pelatihan dan pengujian terutama pada input piksel pada setiap data dan besaran input gambar pada setiap epochnya.

### 5.2. Saran

Adapun saran yang diberikan untuk pengembangan program ini adalah :

1. Melakukan pemotongan gambar inputan sehingga hanya gambar kendaraan yang berada pada citra.

2. Melakukan *preprocessing* untuk memperbaiki kualitas gambar citra atau untuk menghilangkan noise pada citra dengan lebih baik.
3. Menggunakan memori GPU yang lebih besar sehingga dapat membuat model CNN dengan kedalaman lebih besar dari saat program sekarang. Inputan ukuran citra juga dapat menjadi lebih besar dan inputan citra pada setiap iterasi juga dapat diperbesar.
4. Melakukan eksplorasi terhadap hyperparameter padding, maxpooling, average pooling, dropout, stride.

## Daftar Pustaka


[1] Dong, M., He, D., Luo, C., Liu, D. and Zeng, W. (2017), A CNn-based approach for automatic license plate recognition in the wild, British Machine Vision Conference 2017, BMVC 2017, 1–12.

[2] Harani, N.H., Prianto, C. and Hasanah, M. (2019), Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network ( CNN ) Berbasis Python, , 11(3), 47–53.

[3] Li, H. and Shen, C. (2016), Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs, Accessed from http://arxiv.org/ abs/1601.05610.

[4] Selmi, Z., Ben Halima, M. and Alimi, A.M. (2018), Deep Learning System for Automatic License Plate Detection and Recognition, Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 1(November), 1132–1138.

[5] Susanto, Kevin Nyoto; Gunadi,Kartika; Setyati, E. (2019), Pengenalan Karakter pada Plat Nomor Indonesia dengan Tilt Corection dan Metode Faster RCNN, Jurnal Infra Petra, 62.

[6] Xie, L., Ahmad, T., Jin, L., Liu, Y. and Zhang, S. (2018), A New CNN-Based Method for Multi-Directional Car License Plate Detection, IEEE Transactions on Intelligent Transportation Systems, 19(2), 507–517.

[7]Dehghan, A., Masood, S.Z., Shu, G. and Ortiz, E.G. (2017), *View Independent Vehicle Make, Model and Color Recognition Using Convolutional Neural Network*, , 1–7Accessed from http://arxiv.org/abs/1702.01721.

[8] Zou, L., Zhao, M., Gao, Z., Cao, M., Jia, H. and Pei, M. (2018), License Plate Detection with Shallow and Deep CNNs in Complex Environments, Complexity, 2018.

[9] Nafzi, M., Brauckmann, M. and Glasmachers, T. (2019), *Vehicle Shape and Color Classification Using Convolutional Neural Network*, Accessed from http://arxiv.org/abs/1905.08612.

[10] Aarathi, K.S. and Abraham, A. (2017), *Vehicle color recognition using deep learning for hazy images, Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, (Icicct), 335–339.

[11]Rachmadi, R.F. and Purnama, I.K.E. (2015), *Vehicle Color Recognition using Convolutional Neural Network*, , 2–6Accessed from http://arxiv.org/abs/1510.07391.

[12] Zhang, Q., Zhuo, L., Li, J., Zhang, J., Zhang, H. and Li, X. (2018), *Vehicle color recognition using Multiple-Layer Feature Representations of lightweight convolutional neural network*, *Signal Processing*, *147*, 146–153.