

**DETEKSI DAN KLASIFIKASI JENIS KENDARAAN BERBASIS
PENGOLAHAN CITRA DENGAN METODE *CONVOLUTIONAL*
NEURAL NETWORK (CNN)**

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



Disusun oleh:

Ahmad Tri Nurolan

14524075

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta**

2020

LEMBAR PENGESAHAN

DETEKSI DAN KLASIFIKASI JENIS KENDARAAN BERBASIS PENGOLAHAN
CITRA DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

TUGAS AKHIR
ISLAM
UNIVERSITAS INDONESIA
Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

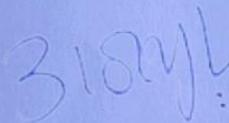
Disusun oleh:


Ahmad Tri Nurolan
14524075

الجامعة الإسلامية
Yogyakarta, 24 Juli 2020

Menyetujui,

Pembimbing



Elvira Sukma Wahyuni S.Pd.,M.Eng

NIK : 155231301

LEMBAR PENGESAHAN

SKRIPSI

DETEKSI DAN KLASIFIKASI JENIS KENDARAAN BERBASIS PENGOLAHAN CITRA DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

Dipersiapkan dan disusun oleh:

Ahmad Tri Nurolan

14524075

Telah dipertahankan di depan dewan penguji

Pada tanggal: 13 Agustus 2020

Susunan dewan penguji

Ketua Penguji : Elvira Sukma Wahyuni, S.Pd.,M.Eng.

Anggota Penguji 1: Sisdarmanto Adinandra, ST, M.Sc., Ph.D.

Anggota Penguji 2: Ida Nurcahyan, ST, M.Eng.

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 26 Agustus 2020

Ketua Program Studi Teknik Elektro



Yusuf Aziz Amrullah, S.T., M.Sc., Ph.D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 13 Agustus 2020



Ahmad Tri Nurolan

KATA PENGANTAR



Assalamualaikum. Wr. Wb

Puji syukur penyusun panjatkan atas kehadiran Allah SWT, yang telah memberikan rahmat dan karunia nya sehingga penulis dapat menyelesaikan laporan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Elektro pendidikan Strata Satu (S1) Fakultas Teknologi Industri Universitas Islam Indonesia.

Sholawat beserta salam senantiasa penyusun haturkan kepada junjungan kita Nabi besar Muhammad SAW keluarga serta para sahabat-nya, karena dengan syafa'atnya kita dapat hijrah dari zaman jahiliyah sehingga menuju ke zaman yang terang benderang. Semoga kita dapat menjadi umat-umatnya yang mendapat syafaat Nabi Muhammad SAW di yaumul akhir nanti. Selama penulisan laporan ini, penulis banyak mendapat bantuan dari berbagai pihak, untuk itu dengan kerendahan hati, penulis ingin berterima kasih kepada :

1. Bapak Yusuf Aziz Amrulloh, ST., M.Eng., Ph.D selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Medilla Kusriyanto, ST., M.Eng selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Elvira Sukma Wahyuni, S.Pd., M.Eng selaku dosen pembimbing yang telah banyak memberikan arahan dalam proses pengerjaan tugas akhir ini.
4. Orangtua tercinta Ibu Kusnayati dan Bapak Budi Usman yang selalu mendoakan dan membantu segala sesuatunya.
5. Kakak dan Adik penulis Ega, Eliza, dan Dela yang telah memberikan semangat dan dukungan kepada penulis dalam mengerjakan skripsi.
6. Teman-teman penulis dari Suroto Squad Ahmad, Bayu, Danang, Faiz, Gilang, Gatot, Hardiansyah, Hendri, Rahmat, Rendy, Ridho, Riduan, Sabil, Wisnu dan Yoga terimakasih banyak atas keceriaannya, kebersamaan ketika susah maupun senang. selalu menghibur penulis dalam mengerjakan skripsi ini.
7. Teman-teman penulis dari Unit Kegiatan Mahasiswa Teater Djemuran Andi, Faiq, Ghozi, Galih, terimakasih banyak karena telah menemani dan memberi dukungan serta semangat di saat penulis mengerjakan skripsi ini.

8. Seluruh keluarga besar Teknik Elektro Universitas Islam Indonesia yang telah membantu secara langsung maupun tidak langsung dalam proses pengerjaan Tugas Akhir ini.
9. Pihak-pihak lain yang tidak dapat disebutkan satu persatu yang telah membantu secara langsung maupun tidak langsung dalam proses Tugas Akhir ini.

Penulis menyadari sepenuhnya bahwa dalam pembuatan skripsi ini banyak kekeliruan dan kekurangan, . Pada akhirnya penulis mohon maaf yang sebesar-besarnya apabila penulis masih memiliki kekurangan baik yang disengaja maupun tidak disengaja. Semoga Allah SWT selalu meridhoi kita semua Aamiin.

Wassalamualaikum Wr.Wb

Yogyakarta, 26 Agustus 2020

Ahmad Tri Nurolan

ARTI LAMBANG DAN SINGKATAN

AI	:	<i>Artificial Intelligence</i>
CNN	:	<i>Convolutional Neural Network</i>
CPU	:	<i>Central Processing Unit</i>
GPU	:	<i>Graphics Processing Unit</i>
MLP	:	<i>Multi-Layer Perceptron</i>
MVP	:	<i>Multi Purpose Vehicle</i>
Open CV	:	<i>Open Source Computer Vision Library</i>
PC	:	<i>Personal Computer</i>
R-CNN	:	<i>Regional Convolutional Neural Network</i>

ABSTRAK

Kendaraan adalah alat transportasi yang sering digunakan masyarakat dalam kegiatan sehari-hari. Seiring berjalannya waktu jumlah populasi kendaraan di Indonesia meningkat sangat tinggi. Akibatnya terjadi banyak kemacetan di mana-mana, apalagi di pintu masuk dan keluar gerbang tol yang menggunakan sistem penggolongan jenis kendaraan yang dirasa kurang efisien untuk saat ini. Untuk itu dilakukan perancangan sistem yang mampu mengatur lajur kendaraan yang akan masuk dan keluar gerbang tol secara otomatis tanpa harus dilakukan penggolongan jenis kendaraan sehingga dapat mengurangi kemacetan yang ada pada gerbang tol serta dapat menghitung volume jumlah kendaraan secara otomatis berdasarkan jenisnya. Perancangan pada sistem ini menggunakan metode *Convolutional Neural Network* (CNN) sebagai pengolah data citra. Seluruh *database* penelitian menggunakan citra gambar sebagai data input dan data output yang di antaranya adalah Mobil MVP, Mini bus, Bus, dan Mobil pickup. Metode *Convolutional Neural Network* (CNN) akan menghasilkan nilai parameter *training* sebagai acuan dalam deteksi dan klasifikasi jenis kendaraan.. Seluruh *database* penelitian menggunakan 200 gambar dari jenis kendaraan, data tersebut merupakan data training dan data uji, di mana data training menggunakan 200 gambar yang terdiri dari 40 gambar (Minibus), 40 gambar (Bus), 40 gambar (Mobil MVP), 40 gambar (Pick Up), dan 40 gambar campuran dari (Minibus, Bus, Mobil MVP, dan Pick Up) dan data uji 25 gambar (Minibus, Bus, Mobil MVP, Pickup, dan gambar campuran). Pada penelitian ini menunjukkan rata-rata akurasi sebesar 81,94% pada seluruh pengujiannya dengan *iterasi* sebesar 10.000

Kata Kunci: *Convolutional Neural Network*, Deteksi Objek, *Droupout Regularization*, Jenis Kendaraan, *Tensorflow*

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN	iii
KATA PENGANTAR	iv
ARTI LAMBANG DAN SINGKATAN	vi
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Studi Literatur	4
2.2 Tinjauan Teori.....	5
2.2.1 <i>Convolutional Neural Network</i>	5
2.2.2 <i>Convolutional Layer</i>	6
2.2.3 <i>Pooling Layer</i>	8
2.2.4 <i>Activation Function</i>	8
2.2.5 <i>Fully Connected Layer</i>	8
2.2.6 <i>Dropout Regularization</i>	9
2.2.7 Rumus perhitungan objek terprediksi	9

BAB 3 METODOLOGI	10
3.1 Perancangan sistem penelitian	11
3.2 <i>Database</i> dan Klasifikasi	11
3.3 <i>Training Data</i>	12
3.3.1 <i>Konversi File XML ke CSV dan Generate TFRecord</i>	12
3.3.2 <i>Label Map</i>	12
3.3.3 <i>Konfigurasi Pipeline</i>	13
3.3.4 <i>Training Database</i>	13
3.3.5 <i>Export Graph</i>	13
3.4 Evaluasi.....	13
3.4.1 Praproses Sebelum Evaluasi	14
BAB 4 HASIL DAN PEMBAHASAN.....	16
4.1 Pelabelan Data Bus, Mobil MVP, Pick Up dan Minibus	16
4.2 <i>Database Hasil Training</i>	17
4.2.1 <i>Training Step</i>	17
4.2.2 <i>Total Loss</i>	18
4.2.3 <i>Model</i>	18
4.3 Hasil deteksi Mobil MVP, Mini Bus, BUS, Pick Up.....	19
4.3.1 Hasil Pengujian CNN.....	19
5.1 Kesimpulan	27
5.2 Saran	27
DAFTAR PUSTAKA	28

DAFTAR GAMBAR

Gambar 2.1 CNN layer.....	5
Gambar 2.2 Konvolusi Neural Network	6
Gambar 2.3 Proses Konvolusi Neural Network	7
Gambar 2.4 Average Pooling	8
Gambar 2.5 Perbandingan Dengan Dropout.....	9
Gambar 3.1 Bagan alir metode penelitian.....	10
Gambar 3.2 Tampilan Konfigurasi Klasifikasi Objek.....	14
Gambar 3.3 Tampilan Konfigurasi Untuk Jumlah <i>Iterasi Training Database</i>	14
Gambar 3.4 Tampilan Konfigurasi yang digunakan untuk GPU Komputer.....	14
Gambar 4.1 Pelabelan Database.....	16
Gambar 4.2 Training Step Pada Command Window.....	17
Gambar 4.3 Grafik Total Loss	18
Gambar 4.4 Hasil Model Training	18
Gambar 4.5 Gambar Hasil Training Database.....	19
Gambar 4.6 Perbandingan Jumlah Data Pengujian.....	26

DAFTAR TABEL

Tabel 3.1 Konfigurasi data Uji dan Training.....	11
Tabel 4.1 Pengujian 1	20
Tabel 4.2 Pengujian 2.....	21
Tabel 4.3 Pengujian 3.....	21
Tabel 4.4 Pengujian 4.....	22
Tabel 4.5 Pengujian 5.....	22
Tabel 4. 6 Pengujian 6.....	23
Tabel 4.7 Pengujian 7.....	24
Tabel 4.8 Pengujian 8.....	24
Tabel 4.9 Perbandingan Akurasi Pengujian	25

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Kendaraan adalah alat transportasi yang sering digunakan masyarakat dalam kegiatan sehari-hari. Terdapat beberapa jenis kendaraan yang sangat umum digunakan masyarakat untuk menempuh perjalanan jarak jauh contohnya Mobil, Motor, dan Bus. Dalam beberapa tahun terakhir, jumlah volume kendaraan di Indonesia meningkat pesat dengan tidak diiringi pembangunan infrastruktur yang memadai. Menurut data dari Badan Pusat Statistik (BPS) jumlah seluruh jenis kendaraan bermotor (Sepeda Motor, Mobil Penumpang, Mobil Bis, Mobil Barang) di Indonesia pada tahun 2018 sebesar 146.858.759 juta. Data tersebut didapatkan melalui pendaftaran registrasi kendaraan yang masuk[1]. Bertambahnya jumlah kendaraan setiap tahunnya, dapat menyebabkan kemacetan lalu lintas di jalan raya maupun pintu masuk dan pintu keluar gerbang tol.

Pemerintah melakukan usaha pencegahan kepadatan lalu lintas dengan menetapkan jalur khusus yang boleh dilewati kendaraan roda dua, empat atau lebih. Penggolongan kendaraan telah dilakukan di pintu masuk tol, ada dua tipe gardu tol ataupun gerbang tol yaitu gardu tol otomatis dan manual. Gardu tol otomatis hanya dikhususkan untuk kendaraan golongan 1 yang mengacu pada BPJT (Badan Pengatur Jalan Tol) yaitu, foto Mobil MVP (*Multi Purpose Vehicle*), Bus, Pick Up, dan Mini Bus[2] dengan metode pembayaran secara otomatis, sedangkan gardu tol manual dengan metode dilayani oleh petugas, khususnya kendaraan besar seperti Truk, dan Truk gandeng yang melewati gardu tersebut.

Dikelompokkannya antara gardu tol manual dan gardu tol otomatis ini gunanya sebagai pemisah antara golongan kendaraan dengan kapasitas yang berbeda, pengelompokan jenis kendaraan yang dipisah manual dan otomatis dapat menyebabkan penumpukan kendaraan di gardu manual ketika volume kendaraan yang melewati tol tersebut sedang pada puncaknya. Untuk itu dibutuhkan sebuah sistem klasifikasi jenis kendaraan secara otomatis sehingga tidak perlu adanya pengelompokan jenis kendaraan di setiap gerbang tol dan juga dapat digunakan untuk menghitung volume kendaraan berdasarkan jenisnya [1].

Pada saat ini teknologi digital mengalami perkembangan yang sangat cepat, di mana kita dapat mempelajari bagaimana menduplikasikan kemampuan manusia dalam memahami suatu citra informasi, agar komputer dapat mengenali sebuah objek pada citra selayaknya manusia, problem dalam visi komputer yang telah lama dicari solusinya adalah klasifikasi objek pada

citra secara umum. Proses yang digunakan pada umumnya sangat terbatas hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi pada jenis citra apapun. Hal ini dikarenakan berbagai perbedaan citra antara lainnya perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek, dan sebagainya.

Deep Learning saat ini telah menjadi salah satu topik hangat dalam dunia *Machine Learning* karena kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti citra dan suara. Metode *Deep Learning* pada saat ini yang memiliki hasil yang paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network (CNN)*. Hal ini dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual manusia sehingga dapat memiliki kemampuan mengolah informasi citra. Namun CNN, seperti *Deep Learning* lain pada umumnya memiliki kelemahan pada proses pelatihan data yang lama, hal tersebut dapat diatasi dengan menggunakan teknologi *Graphical Processing Unit (GPU)* pada perangkat komputer[3].

Hal inilah yang mendasari penelitian ini menggunakan metode CNN karena mampu membaca pengolahan citra (*image processing*) dengan tingkat akurasi yang tinggi dalam mendeteksi objek maupun *image classification*. Sehingga pada penelitian ini dikembangkan metode klasifikasi jenis kendaraan berbasis citra dengan menggunakan algoritma CNN.

1.2 Rumusan Masalah

1. Bagaimana merancang dan mengimplementasikan sistem klasifikasi jenis kendaraan menggunakan pemrosesan citra
2. Bagaimana arsitektur jaringan yang digunakan untuk mendeteksi jenis kendaraan menggunakan algoritma Convolutional Neural Network?
3. Bagaimana hasil pengujian dari pendeteksian jenis kendaraan menggunakan algoritma *Convolutional Neural Network* ?
4. Seberapa tinggi tingkat akurasi pendeteksian jenis kendaraan menggunakan algoritma *Convolutional Neural Network* ?

1.3 Batasan Masalah

1. Sistem klasifikasi jenis kendaraan ini menggunakan metode *Convolutional Neural Network (CNN)*.
2. Penelitian ini menggunakan gambar sebagai input dan output.

3. Penelitian ini menggunakan foto dengan posisi pengambilan yang sudah ditentukan.
4. *Database* yang digunakan pada penelitian ini terdiri dari beberapa foto golongan 1 jenis kendaraan yang mengacu pada BPJT (Badan Pengatur Jalan Tol) yaitu, foto Mobil MVP (*Multi Purpose Vehicle*), Bus, Pick Up, dan Mini Bus.

1.4 Tujuan Penelitian

Berikut adalah tujuan dari penelitian yang ingin dicapai sebagai berikut :

1. Dapat mengklasifikasikan jenis kendaraan berbasis pengolahan citra dengan metode *CNN*.
2. Adanya sistem klasifikasi jenis kendaraan agar dapat diimplementasikan pada jalan raya dan tol untuk menghindari kepadatan jumlah kendaraan.
3. Mendapatkan hasil pengujian dari pendeteksian jenis kendaraan menggunakan algoritma *Convolutional Neural Network*.
4. Mendapatkan tingkat akurasi tinggi pada pendeteksian jenis kendaraan menggunakan algoritma *Convolutional Neural Network*.

1.5 Manfaat Penelitian

Manfaat dalam penelitian yang ingin dicapai adalah untuk meminimalisir jumlah kemacetan yang terjadi di jalan raya maupun gerbang masuk dan keluar pintu tol.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang klasifikasi jenis kendaraan telah banyak dilakukan, salah satunya dengan metode *Covolutional Neural Network* (CNN). Penelitian klasifikasi citra menggunakan *Covolutional Neural Network* (CNN) pada *Caltech 101* oleh [3]. Pada penelitian ini proses pengerjaannya dibagi menjadi beberapa tahap. Tahap pertama yaitu pra proses dan pengolahan data input dengan cara *Wrapping* dan *Croppping*, tahap kedua training dan tahap ketiga yaitu testing. Hasil akhirnya merupakan data akurasi dan klasifikasi.

Penelitian berikutnya masih berkaitan dengan klasifikasi jenis kendaraan yaitu dengan metode *Gabor Filtering* dan *Naive Bayes* oleh [4], penelitian ini bertujuan untuk membuat suatu sistem secara otomatis yang dapat mengelompokkan kendaraan, sehingga tidak perlu menggunakan pembatas/palang untuk melakukan transaksi di gerbang tol dan lebih efisien dalam segi waktu. *Gabor Filter* merupakan teknik ekstraksi ciri yang mengambil representasi yang optimal dari arah orientasi dan frekuensi spesial. Untuk pencocokan data digunakan teknik *Naive Bayes* sebagai classifier karena merupakan salah satu metode klasifikasi. Sistem kerjanya membedakan kendaraan ke dalam tiga bagian yaitu Car, Bus, dan Truk, skema yang dilakukan *training* dan *testing*. Dari penelitian ini didapat hasil akurasi maksimal sebesar 81,73% pada pengujian nilai dengan *Naive Bayes*, akurasi sebesar 81,73% dengan pengujian jumlah *Gabor Filter*.

Penelitian selanjutnya oleh [5] tentang sistem klasifikasi kendaraan berbasis pengolahan citra digital dengan metode *Multilayer Perceptron*. Tujuan penelitian ini untuk menyediakan informasi seputar kendaraan yang melintas agar dapat dipantau dengan memanfaatkan kamera dapat dilakukan secara otomatis. Metode penelitian ini terdiri dari dua tahapan pengolahan citra digital yaitu proses deteksi dan klasifikasi. Proses deteksi menggunakan *Haar Cascade Classifier* dengan data training berupa citra kendaraan dan data uji berupa citra keadaan di jalan tol yang diambil secara acak, proses klasifikasi menggunakan *Multilayer Perceptron* dengan memanfaatkan hasil dari deteksi yang diproses. Hasil penelitian menunjukkan sistem klasifikasi memiliki rata-rata nilai akurasi sebesar 87,60%.

Penelitian oleh [6] berikut ini berkaitan dengan deteksi jenis kendaraan di jalan menggunakan *OpenCV*. Tujuan dari penelitian ini untuk memantau kondisi jalan dengan program yang bisa mendeteksi jenis kendaraan di jalan raya dan menghitung jumlah kendaraan

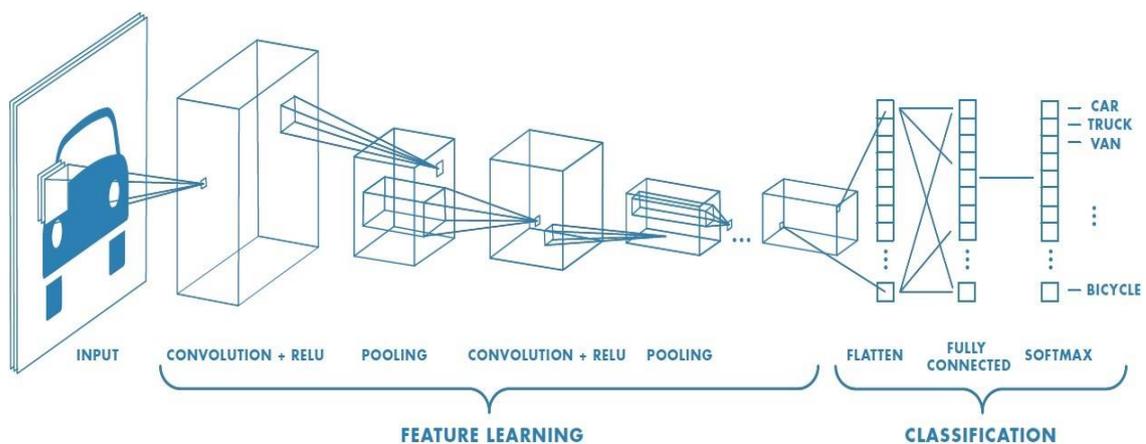
yang terdeteksi berdasarkan jenisnya. *OpenCV* adalah library dari fungsi pemrograman untuk *real time* visi komputer, uji coba dilakukan untuk mengenali jenis kendaraan melalui *input* video. Tingkat akurasi dibagi menjadi tiga kondisi sepi, normal, padat adalah 77,8 % sepi, 47,5% normal, 28,2% padat.

Penelitian selanjutnya oleh [7] berkaitan dengan deteksi jenis kendaraan berdasarkan pembelajaran lebih dalam di lalu lintas, metode yang digunakan pada penelitian ini adalah *R-CNN*. *R-CNN* digunakan karena mampu mengekstrak *filter* lebih sedikit dan lebih efisien. Hasil dari pengujian ini menunjukkan efektivitas dan efisiensi tinggi dengan metode pembelajaran yang mendalam pada deteksi jenis kendaraan.

2.2 Tinjauan Teori

2.2.1 Convolutional Neural Network

Convolutional neural network termasuk dalam jenis *Deep Neural Network* karena memiliki kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Arsitektur dalam *Convolutional neural network* dapat di training dan terdiri dari beberapa tahap. *Input* dan *output* dari masing-masing tahap adalah beberapa array yang disebut feature map atau map fitur. Terdapat empat layer utama dalam algoritma *Convolutional neural network* yaitu *convolutional layer*, *pooling layer*, *activation layer*, dan *fully connected layer*[3].



Gambar 2.1 CNN layer

2.2.2 Convolutional Layer

Convolutional layer adalah layer pertama yang akan menerima input dari gambar langsung pada arsitektur jaringan. *Convolutional layer* ini melakukan operasi konvolusi terhadap *output* dari layer sebelumnya. Tujuan utama dari konvolusi data citra yaitu untuk mengekstraksi fitur dari data citra input. Pada umumnya operasi konvolusi menggunakan persamaan sebagai berikut.

$$s(t) = (x * w)(t) \quad (2.1)$$

Keterangan:

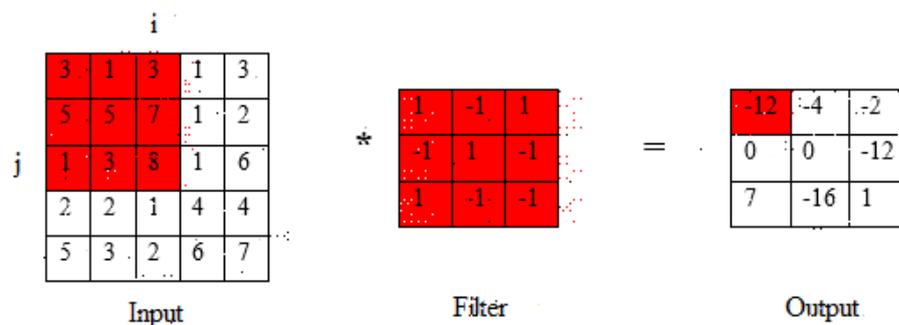
$s(t)$: Fungsi hasil proses konvolusi

x : Input

w : bobot (kernel)

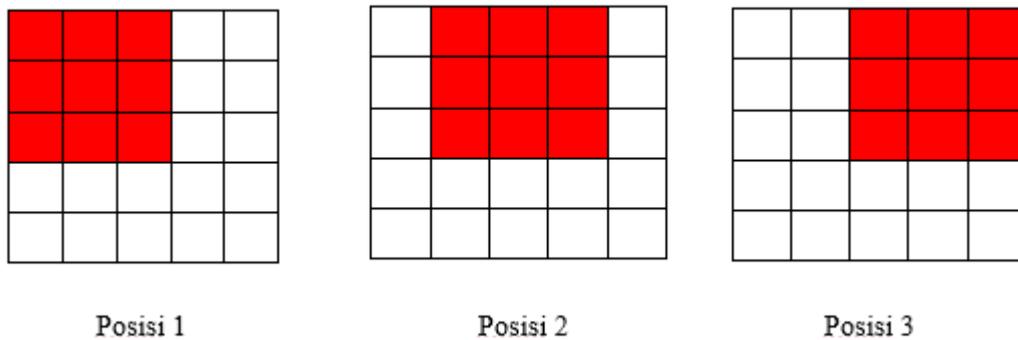
Fungsi $s(t)$ memberikan *output* tunggal berupa *feature map*. Argumen pertama berupa *input* yang merupakan (x) dan argument kedua yaitu (w) sebagai kernel atau *filter*. Jika melihat input sebagai citra dua dimensi, maka (t) diasumsikan sebagai sebuah piksel dan menggantinya dengan i dan j [8].

Agar dapat lebih memahami dari prinsip kerja konvolusi, peneliti akan menggunakan contoh dengan *input* matriks 5x5 dikarenakan keterbatasan penulisan dengan ukuran 300x300 serta menggunakan *filter* untuk operasi deteksi tepi dengan ukuran 3x3.



Gambar 2.2 Konvolusi Neural Network

Dapat dilihat pada gambar 2.2 adalah proses perhitungan konvolusi dengan menggunakan ukuran *filter* 3x3 dengan i dan j sebagai piksel dari sebuah citra dan menggunakan pergeseran *filter* dengan matriks *input* berjumlah satu. Jika divisualisasikan sebagai berikut:



Gambar 2.3 Proses Konvolusi Neural Network

Gambar 2.3 merupakan perhitungan yang terjadi pada saat proses konvolusi sebuah *filter* dengan ukuran 3x3 yang diawali pada sisi bagian kiri proses ini disebut dengan *sliding window*. Berikut adalah contoh proses perhitungan proses konvolusi dari posisi 1 hingga ke posisi 3 :

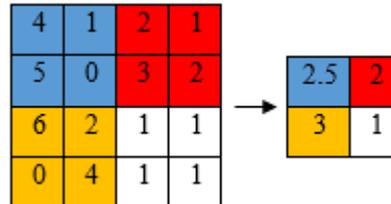
- a. Posisi 1 = $(3 \times 1) + (5 \times (-1)) + (1 \times 1) + (1 \times (-1)) + (5 \times 1) + (3 \times (-1)) + (3 \times 1) + (7 \times (-1)) + (8 \times (-1)) = -12$
- b. Posisi 2 = $(1 \times 1) + (5 \times (-1)) + (3 \times 1) + (3 \times (-1)) + (7 \times 1) + (8 \times (-1)) + (1 \times 1) + (1 \times (-1)) + (1 \times 1) = -4$
- c. Posisi 3 = $(3 \times 1) + (7 \times (-1)) + (8 \times 1) + (1 \times (-1)) + (1 \times 1) + (1 \times (-1)) + (3 \times 1) + (2 \times (-1)) + (6 \times (-1)) = -2$

Terdapat dua parameter untuk memodifikasi *layer* yaitu.

1. *Stride* adalah parameter yang akan menentukan berapa jumlah pergeseran *filter*, jika nilai *stride* satu maka *feature map* akan bergeser sebanyak satu piksel secara horizontal dan vertikal. Semakin kecil nilai *stride* maka didapatkan hasil semakin detail serta membutuhkan komputasi lebih jika dibandingkan dengan nilai *stride* yang besar.
2. *Padding* adalah parameter yang akan menentukan jumlah piksel (bernilai nol) yang ditambahkan pada tiap sisi dari *input*. Tujuan dilakukan *padding* adalah untuk mengatur nilai *output* agar sama dengan *input* atau tidak terlalu berkurang drastis sehingga dapat dilakukan ekstrasi *feature* lebih mendalam.

2.2.3 Pooling Layer

Pooling atau *subsampling* adalah pengurangan ukuran matriks yang biasanya dilakukan setelah operasi *convolutional layer*. Terdapat dua tipe *pooling* yang paling sering digunakan yaitu *average pooling* dan *max pooling*. Dalam *average pooling* nilai yang diambil adalah nilai rata-rata dari *input* awal matriks.



Gambar 2.4 Average Pooling

Output dari *pooling layer* yaitu berupa matriks yang lebih kecil dibandingkan dengan matriks awal. Proses konvolusi dan *pooling* dilakukan beberapa kali sehingga akan mendapatkan hasil yang diinginkan.

2.2.4 Activation Function

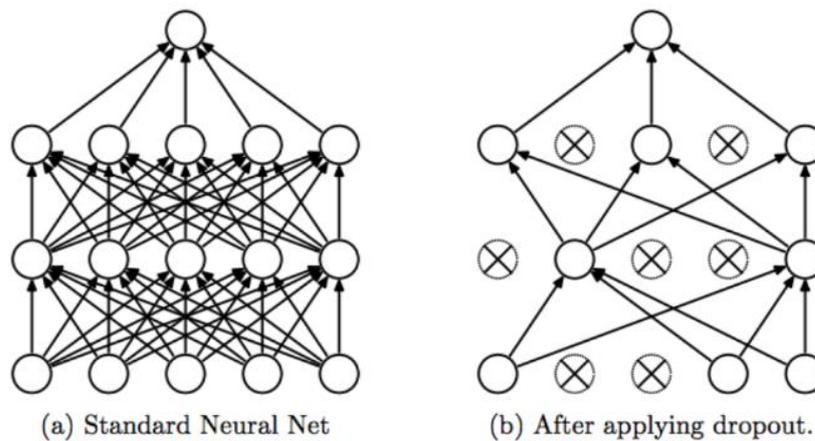
Activation function adalah sebuah *node* yang ditambahkan di akhir keluaran dari setiap jaringan syaraf. Pada arsitektur CNN, fungsi aktivasi dapat digunakan pada perhitungan akhir keluaran *feature map* atau setelah proses konvolusi maupun *pooling layer*. Terdapat banyak fungsi aktivasi yang sering digunakan dalam *neural network* yaitu *sigmoid*, *tanh*, *ReLU* (*Rectified Linear*), parameter *ReLU*, dan *leaky ReLU*.

2.2.5 Fully Connected Layer

Sebelum masuk ke *fully connected layer*, keluaran dari *feature map* masih berbentuk multidimensional *array*. Sehingga perlu melakukan *flatten* atau *reshape feature map* menjadi sebuah *vector* agar dapat digunakan sebagai *input fully connected layer*. Pada lapisan *fully connected layer*, semua *neuron* aktivasi dari lapisan sebelumnya terhubung pada *neuron* dilapisan selanjutnya sehingga seperti jaringan syaraf tiruan biasa. Pada lapisan ini pula menghasilkan *output* berupa klasifikasi citra yang diinginkan.

2.2.6 Dropout Regularization

Regularization adalah teknik yang digunakan untuk mengurangi *overfitting* atau *noise*. adalah kondisi di mana sistem mampu belajar dengan baik dengan data pelatihan, namun tidak dapat menggeneralisasi dengan data uji. *Dropout* adalah teknik yang dapat digunakan untuk mencegah *overfitting* serta mempercepat proses pelatihan. *Dropout* menghilangkan *neuron* yang berupa *hidden* maupun *layer* yang terlihat pada jaringan. *Neuron* yang hilang akan dipilih secara acak oleh sistem dengan probabilitas dari nol hingga satu. Berikut gambaran dari proses saat adanya *dropout*[9].



Gambar 2.5 Perbandingan Dengan Dropout

2.2.7 Rumus perhitungan objek terprediksi

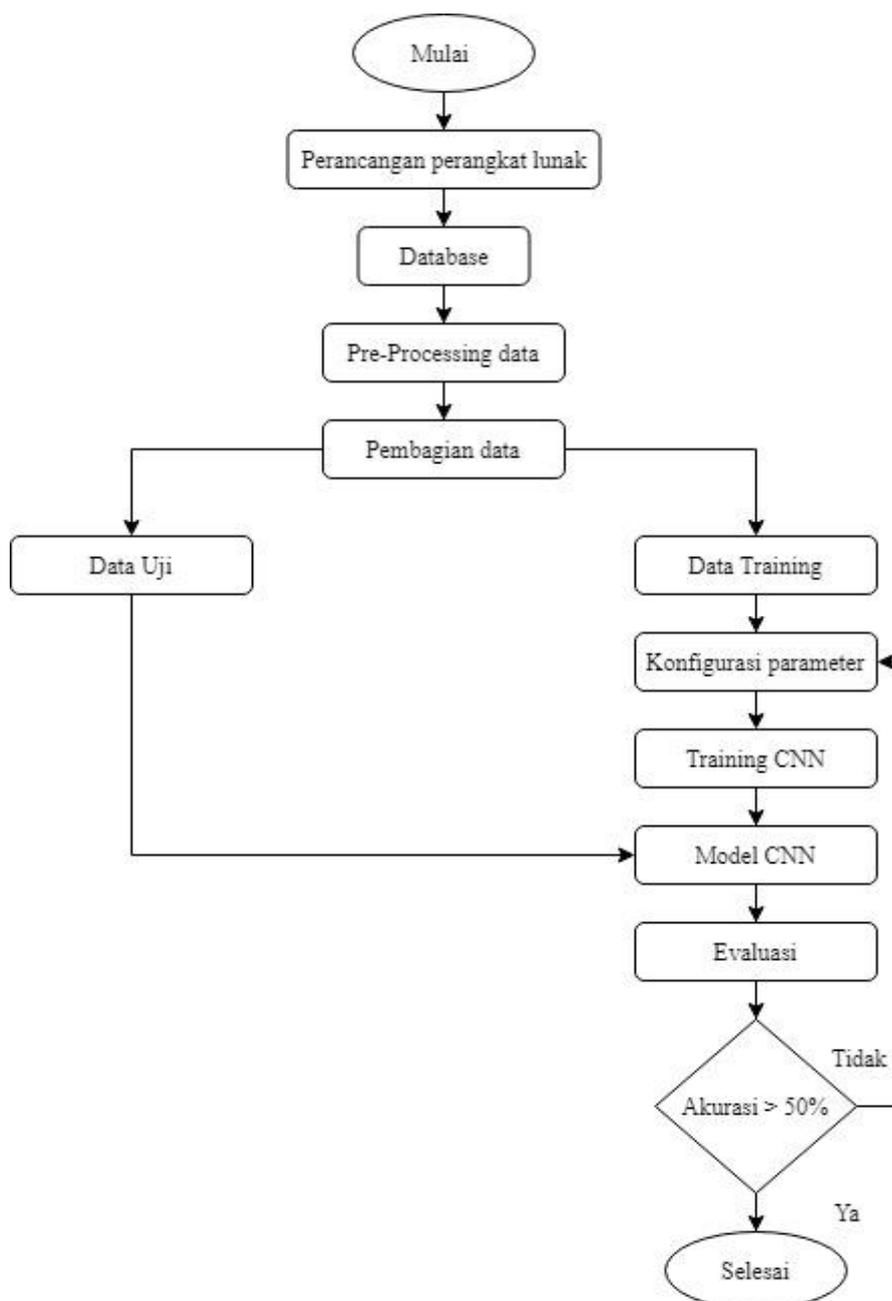
Untuk menentukan hasil persentase sebuah objek yang telah di prediksi, maka dibutuhkan suatu rumus perhitungan untuk mendapatkan nilai akurasi persentase. Rumus perhitungan objek dapat dihitung dengan rumus berikut :

$$\text{Akurasi} = \frac{\text{Jumlah objek terprediksi}}{\text{total objek pengujian}} \times 100\% \quad (2.2)$$

BAB 3

METODOLOGI

Pada Bab ini peneliti akan menjelaskan metode yang digunakan untuk menyelesaikan rumusan masalah yang diteliti. Metode yang digunakan dalam mendeteksi dan klasifikasi jenis kendaraan menggunakan algoritma *Convolutional Neural Network* dapat dilihat pada bagan alir 3.1.



Gambar 3.1 Bagan alir metode penelitian

3.1 Perancangan sistem penelitian

Sistem deteksi dan klasifikasi jenis kendaraan ini menggunakan perangkat keras *Personal Computer* (PC) untuk menjalankan sistem konfigurasi *Convolutional Neural Network* (CNN). *Personal Computer* (PC) digunakan untuk memonitor serta sebagai pengendali sistem. Perangkat lunak yang digunakan dalam penelitian ini yaitu *Python-3.6*, *Visual Studio 2015*, *Cuda-9 Nvidia*. Bahasa program yang digunakan pada *Python* adalah bahasa C, *Visual studio* berguna sebagai editor program, dan *Cuda* berfungsi untuk menjalankan program menggunakan *Graphics Processing Unit* (GPU) agar kemampuan komputasi menjadi lebih cepat dibandingkan *Central Processing Unit* (CPU). Pada penelitian ini menggunakan *framework* dari Google yaitu *TensorFlow*, di mana untuk keperluan *deep learning*, *machine learning*, maupun *artificial intelligence* digunakan *framework* ini.

3.2 Database dan Klasifikasi

Tabel 3.1 Konfigurasi data Uji dan *Training*

Pengujian	Total data <i>training</i> 200 gambar							
1	25	25	25	25	25	25	25	25
2	25	25	25	25	25	25	25	25
3	25	25	25	25	25	25	25	25
4	25	25	25	25	25	25	25	25
5	25	25	25	25	25	25	25	25
6	25	25	25	25	25	25	25	25
7	25	25	25	25	25	25	25	25
8	25	25	25	25	25	25	25	25

Ket:

	Data uji yang di gunakan pada proses pengujian
	<i>Database</i> pada data <i>training</i>

Langkah pertama dalam mendeteksi objek adalah mempersiapkan data berupa citra kendaraan dalam format JPEG. Citra yang dijadikan *database* pada penelitian ini diambil secara langsung di jalan raya dengan menggunakan kamera digital. *Input* dari rancangan sistem yang digunakan peneliti berupa gambar dengan resolusi 1280x851 piksel.

Pada tabel 3.1 jumlah total data yang digunakan peneliti sebanyak 200 gambar dari jenis kendaraan, data tersebut merupakan data *training* dan data uji, di mana data *training*

menggunakan 200 gambar yang terdiri dari 40 gambar (Minibus), 40 gambar (Bus), 40 gambar (Mobil MVP), 40 gambar (Pick Up), dan 40 gambar campuran dari (Minibus, Bus, Mobil MVP, dan Pick Up) dan data uji 25 gambar (Minibus, Bus, Mobil MVP, Pick Up, dan gambar campuran). Dari tabel 3.1 tahapan pada pengerjaannya yaitu pengujian dilakukan 8 kali percobaan dengan menggunakan 200 gambar sebagai *input database*. Dengan 8 kali percobaan tersebut setiap percobaan nya menggunakan 25 gambar yang berbeda sebagai data pengujian dan dengan *Iterasi* sebesar 10.000. Sebelum melakukan training model seluruh data *training* diklasifikasikan sesuai dengan objek yang ingin di deteksi berdasarkan jenis kendaraan yang bertujuan untuk mempermudah komputer mempelajari objek deteksi.

3.3 Training Data

Pada saat proses training data, algoritma *Convolutional Neural Network* membutuhkan beberapa tahapan dalam pengerjaannya. Tahapan yang terjadi pada algoritma ini yaitu konvolusi *layer*, *activation (Relu)*, *dropout*, *pooling layer*, *fully connected layer*, klasifikasi dan deteksi objek.

3.3.1 Konversi File XML ke CSV dan Generate TFRecord

File XML ini didapatkan setelah melakukan klasifikasi data, maka perlu di konversi ke file berekstensi CSV sehingga dapat di *Generate TFRecord*. *Generate TFRecord* digunakan untuk merekam proses pelatihan data.

3.3.2 Label Map

Proses pelabelan data dilakukan untuk memberikan keterangan pada objek yang ditargetkan sebelum memulai proses *training* data yang bertujuan untuk mempermudah pembacaan data pada saat hasil keluaran objek serta untuk pembagian kelas berdasarkan jenisnya, sehingga sistem lebih mudah untuk mengenali objek berdasarkan kelasnya pada *database*. Pada file Label Map jumlah kelas akan sama dengan database dan *generate TFRecord*.

3.3.3 Konfigurasi Pipeline

Tahapan selanjutnya adalah konfigurasi *pipeline* yang berguna untuk mengatur algoritma dari *Convolutional Neural Network* (CNN), dan mengatur berapa banyak klasifikasi yang digunakan serta mengatur direktori *file training* dan pelabelan map. Pada konfigurasi *pipeline* ini juga dapat mengatur berapa banyak data yang akan di *training* maupun yang akan di evaluasi dengan menggunakan *protobuf*.

3.3.4 Training Database

Setelah semua proses dilakukan maka tahap selanjutnya adalah pelatihan data dengan menggunakan algoritma *Convolutional Neural Network*. Pada proses ini model akan menghasilkan *Checkpoint* yang secara otomatis dibuat oleh *framework tensorflow* berbentuk *graph tensor* gunanya untuk menyimpan informasi pada saat proses *training* model.

3.3.5 Export Graph

Setelah melakukan *training database* maka didapatkan output data yang dapat digunakan untuk mendeteksi objek yang diinginkan. Setelah proses *training* selesai hasil dari *checkpoint* terakhir di *export* sehingga menjadi model data yang siap untuk dilakukan pengujian.

3.4 Evaluasi

Tahap terakhir adalah pengujian model data terakhir yang telah di *export graph* dengan data uji. Pada tahap ini peneliti menggunakan *jupyter notebook* pada *command window* untuk melakukan pengujian data. Penelitian dapat dikatakan sukses jika hasil dari akurasi mendapat nilai tinggi, jika nilai akurasinya rendah dari yang diinginkan maka dilakukan *training ulang database* dan merubah konfigurasi maksimum pelatihan apabila melebihi nilai yang telah ditentukan pada konfigurasinya. Sehingga diharapkan nilai kemiripan akurasi lebih dari 50% pada saat proses *training* yang dilakukan untuk pengujian data pada setiap jenis kendaraan yaitu mobil MVP, Bus, Pick Up, dan Minibus.

3.4.1 Praproses Sebelum Evaluasi

Pada tahapan ini untuk mencapai tahap evaluasi dilakukan dengan cara mengatur konfigurasi pada *Convolution Neural Network* (CNN) untuk mengubah nilai klasifikasi yang digunakan dan seberapa banyak jumlah iterasi digunakan pada konfigurasi serta *batch size* yang berpengaruh terhadap kecepatan pembacaan sistem terhadap *Graphics Processing Unit* (GPU) komputer. Didalam konfigurasi peneliti menggunakan banyaknya 4 klasifikasi objek dan dapat dilihat pada Gambar 3.2.

```
model {
  ssd {
    num_classes: 4
    box_coder {
      faster_rcnn_box_coder {
        y_scale: 10.0
        x_scale: 10.0
        height_scale: 5.0
        width_scale: 5.0
      }
    }
  }
}
```

Gambar 3.2 Tampilan Konfigurasi Klasifikasi Objek

Dapat dilihat bahwa pada Gambar 3.2 *num_classes* 4 merupakan konfigurasi untuk menentukan banyaknya jumlah kasifikasi yang digunakan oleh peneliti sebanyak 4 klasifikasi objek. Kemudian tahap selanjutnya menentukan jumlah *iterasi* untuk *training database* dapat dilihat pada Gambar 3.3.

```
hard_example_miner {
  num_hard_examples: 100000
}
```

Gambar 3.3 Tampilan Konfigurasi Untuk Jumlah *Iterasi Training Database*

Pada Gambar 3.3 dapat dilihat bahwa nilai *num_hard_examples*: 100.000 adalah nilai konfigurasi untuk *training database* sebesar 100.000 *iterasi*. Tahap selanjutnya yaitu menentukan nilai *batch size* yang berada pada konfigurasi *Convolution Neural Network* (CNN) dapat dilihat pada Gambar 3.4 sebagai berikut:

```
train_config: {
  batch_size: 10
  optimizer {
    rms_prop_optimizer {
      learning_rate: {
        exponential_decay_learning_rate {
          initial_learning_rate: 0.004
          decay_steps: 800720
          decay_factor: 0.95
        }
      }
    }
  }
}
```

Gambar 3.4 Tampilan Konfigurasi yang digunakan untuk GPU Komputer

Dapat dilihat bahwa pada Gambar 3.4 *batch size* yang digunakan pada penelitian sebesar 10. Pada dasarnya nilai *batch size* ini mempengaruhi pada saat proses *training* data apabila semakin besar maka semakin cepat pembacaannya GPU dalam memproses *training database*.

BAB 4

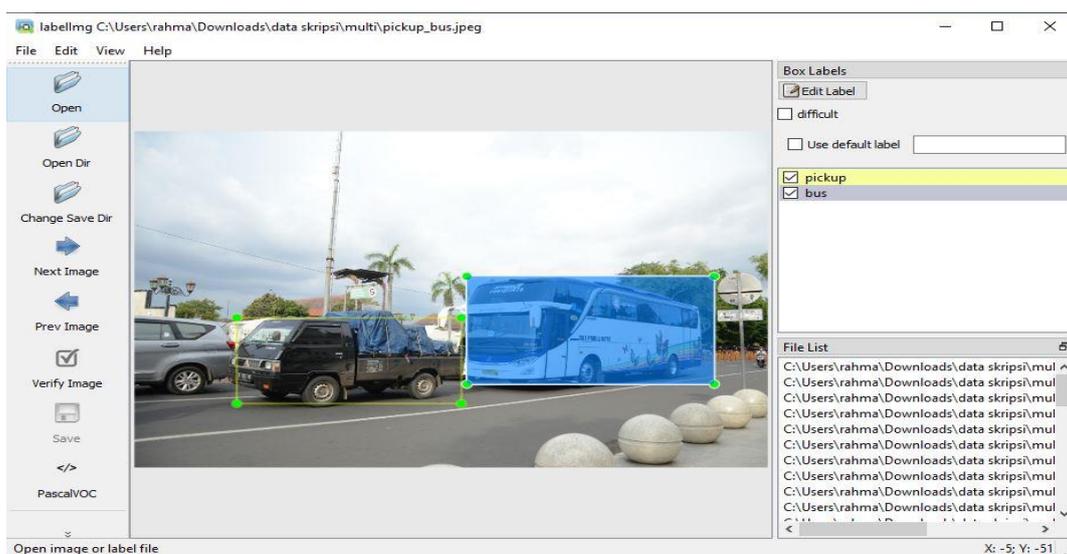
HASIL DAN PEMBAHASAN

Pada penelitian ini, langkah awal yang dilakukan adalah klasifikasi objek menjadi empat kelas jenis kendaraan yaitu Bus, Mobil MVP, Pickup, dan Minibus dengan menggunakan algoritma *Convolutional Neural Network* (CNN). Proses utama yang dilakukan dalam pembuatan *database* yaitu dengan melakukan *training* data, hal ini bertujuan untuk membuat *database* mendapatkan nilai akurasi tinggi pada saat proses deteksi objek yang diinginkan. Parameter pengujian untuk mengukur tingkat keberhasilan objek deteksi adalah dengan mendapatkan nilai dengan tingkat akurasi sebesar lebih dari 50%.

Untuk mempermudah peneliti dalam *deep learning*, *machine learning* maupun *artificial intelligence* maka digunakan *Tensorflow* yang merupakan penelitian dari *Google*. Sistem rancangan yang digunakan pada saat penelitian dilakukan adalah masukan (*input*), pengolahan data, dan keluaran (*output*).

4.1 Pelabelan Data Bus, Mobil MVP, Pick Up dan Minibus

Pelabelan data yang dilakukan menggunakan program *labelImg.py* yang berguna untuk menentukan titik objek yang ingin dideteksi sebagai Bus, Mobil MVP, Pick Up, dan Minibus. Pelabelan data dilakukan sebanyak 200 gambar yang terdiri dari 40 gambar (Minibus), 40 gambar (Bus), 40 gambar (Mobil MVP), 40 gambar (Pick Up), dan 40 gambar campuran dari (Minibus, Bus, Mobil MVP, dan Pick Up). Berikut ini tampilan dari program *labelImg.py*.



Gambar 4.1 Pelabelan Database

Gambar 4.1 menunjukkan proses pelabelan database sesuai dengan objek yang ingin dideteksi. Setelah proses pelabelan dilakukan, hasilnya akan tersimpan dalam *format XML*.

4.2 Database Hasil *Training*

Bagian ini merupakan pembahasan hasil dari proses *implementasi* pelatihan *database*. Berikut adalah tahapan-tahapan dari proses *database* hasil dari *training* yang telah dilakukan.

4.2.1 *Training Step*

Training Step merupakan proses *training* yang gunanya untuk melatih sistem agar mengerti pola suatu objek yang diinginkan. Berdasarkan hasil penelitian dengan jumlah *iterasi* 10.000 proses *training database* memerlukan waktu rata-rata 2 detik – 2.5 detik untuk sistem dapat membaca objek sebagai Bus, Mobil MVP, Pick Up, dan Minibus.

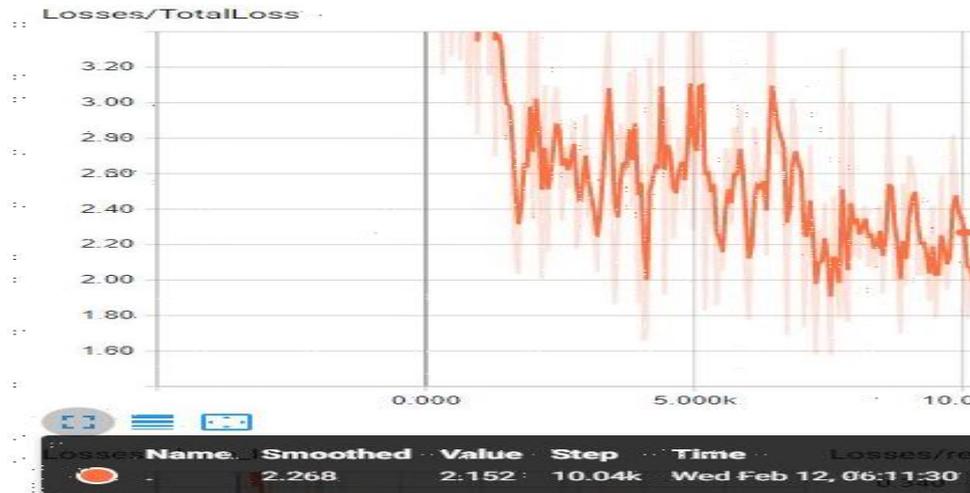
```
INFO:tensorflow:global step 30367: loss = 2.2920 (2.906 sec/step)
INFO:tensorflow:global step 30368: loss = 1.8138 (2.203 sec/step)
INFO:tensorflow:global step 30369: loss = 2.1548 (2.187 sec/step)
INFO:tensorflow:global step 30370: loss = 1.7385 (2.187 sec/step)
INFO:tensorflow:global step 30371: loss = 1.8368 (2.187 sec/step)
INFO:tensorflow:global step 30372: loss = 1.8433 (2.203 sec/step)
INFO:tensorflow:global step 30373: loss = 1.7945 (2.281 sec/step)
INFO:tensorflow:global step 30374: loss = 1.6919 (2.234 sec/step)
INFO:tensorflow:global step 30375: loss = 2.0828 (2.187 sec/step)
INFO:tensorflow:global step 30376: loss = 2.6828 (2.187 sec/step)
INFO:tensorflow:global step 30377: loss = 1.8046 (2.218 sec/step)
INFO:tensorflow:global step 30378: loss = 1.3901 (2.171 sec/step)
INFO:tensorflow:global step 30379: loss = 1.4923 (2.218 sec/step)
INFO:tensorflow:global step 30380: loss = 1.5557 (2.203 sec/step)
INFO:tensorflow:global step 30381: loss = 1.3710 (2.218 sec/step)
INFO:tensorflow:global step 30382: loss = 2.3800 (2.203 sec/step)
INFO:tensorflow:global step 30383: loss = 1.6659 (2.218 sec/step)
INFO:tensorflow:global step 30384: loss = 2.3528 (2.203 sec/step)
INFO:tensorflow:global step 30385: loss = 2.3044 (2.218 sec/step)
INFO:tensorflow:global step 30386: loss = 1.6531 (2.203 sec/step)
```

Gambar 4.2 *Training Step* Pada *Command Window*

Gambar 4.2 merupakan proses *training* yang terdapat pada *command window*. Lamanya proses *training database* dapat dipengaruhi oleh nilai *batch size*. Agar proses *training* tidak terlalu lama maka peneliti menggunakan *batch size* 10 dalam proses *training database*. Jika nilai *resolusi* gambar pada *database* besar, maka diperlukan nilai *batch size* yang besar dan memerlukan memori *personal computer* yang tinggi.

4.2.2 Total Loss

Saat proses *training database* berjalan, maka data akan diproses kemudian akan terekam dalam sebuah *file*. *File* ini dapat dilihat dengan menggunakan *TensorBoard* yang berbentuk seperti *local host*. Semua informasi yang ada seperti *learning rate*, *num negative*, *num positives*, *target assignment*, dan juga *total loss* tersimpan didalam *file* tersebut. Berikut hasil dari *total loss* pada proses *training database*.



Gambar 4.3 Grafik Total Loss

Gambar 4.3 adalah hasil grafik total loss yang didapat pada proses *training* yang dilakukan sampai dengan selesai. Nilai dari *total loss* dikatakan baik yaitu pada saat *steady state* dan dapat dilihat juga bahwa hasil yang didapatkan berisolasi yaitu dari 1.80-3.10. Untuk hasil akhir dari proses training pada langkah 10.040 didapatkan nilai losses sebesar 2.268.

4.2.3 Model

Setelah dari tahap *training* selesai maka akan menghasilkan *checkpoint folder*, *checkpoint* ini nantinya akan dapat digunakan untuk mendapatkan model sistem, *checkpoint folder* ini fungsinya sebagai *output* untuk model *database* yang digunakan. Berikut adalah isi didalam *folder* model pada Gambar 4.4:

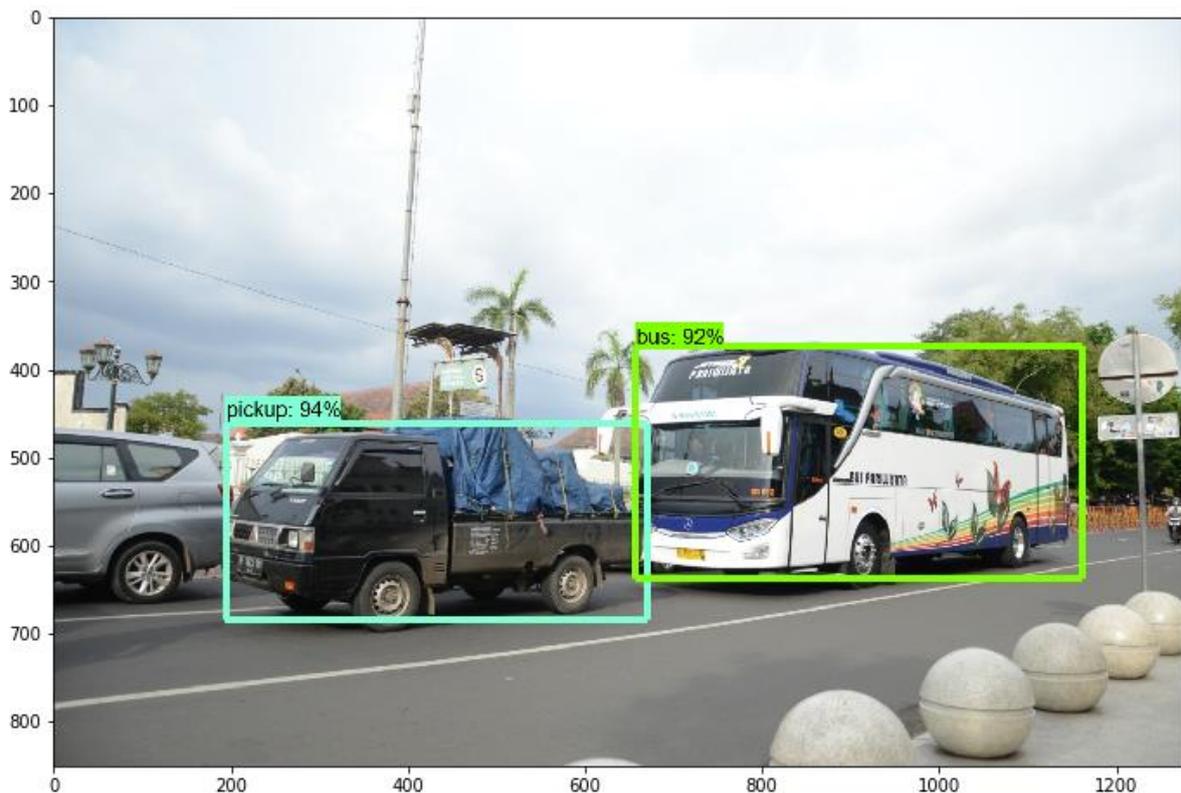
saved_model	12/02/2020 6:24	File folder	
checkpoint	12/02/2020 6:24	File	1 KB
frozen_inference_graph.pb	12/02/2020 6:24	PB File	22,319 KB
model.ckpt.data-00000-of-00001	12/02/2020 6:24	DATA-00000-OF-0...	21,826 KB
model.ckpt.index	12/02/2020 6:24	INDEX File	9 KB
model.ckpt.meta	12/02/2020 6:24	META File	1,067 KB
pipeline.config	12/02/2020 6:24	XML Configuratio...	4 KB

Gambar 4.4 Hasil Model Training

Pada Gambar 4.4 merupakan tampilan dari hasil model *training* yang akan digunakan untuk memasuki tahap selanjutnya yaitu menampilkan hasil model.

4.3 Hasil deteksi Mobil MVP, Mini Bus, BUS, Pick Up

Hasil deteksi seperti gambar 4.5 didapat setelah proses *training* data dilakukan dan telah juga dilakukan proses pengujian data.



Gambar 4.5 Gambar Hasil *Training Database*

Pengujian model menggunakan jumlah total data sebanyak 200 gambar. Data *training* sebanyak 200 gambar terdiri dari gambar Mobil MVP , Pick Up, Minibus, Bus, dan gambar campuran yang masing-masing terdiri dari 40 gambar setiap kategorinya. Gambar yang digunakan sebagai pengujian terdiri dari 25 gambar yang terdiri 5 gambar Mobil MVP, 5 gambar Mini Bus, 5 gambar Bus, 5 gambar Pick Up dan 5 gambar campuran. Pada gambar 4.5 dapat dilihat hasil dari pengujian *database* mendapatkan hasil Pick Up 94% dan Bus 92%.

4.3.1 Hasil Pengujian CNN

Pada pengujian *database* dilakukan dengan delapan kali percobaan dengan iterasi 10.000. Pengujian ini dilakukan 1-8 kali tahap percobaan dengan menggunakan gambar uji

database yang berbeda-beda setiap percobaannya, untuk melihat perbedaan hasil akurasi di setiap tahap pengujiannya.

4.3.1.1 Pengujian 1

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 1.

Tabel 4.1 Pengujian 1

Prediksi Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (10)	7	0	0	0
Mini Bus (5)	0	5	0	0
Bus (10)	0	0	10	0
Pick Up (5)	0	0	0	4

Pada tabel 4.1 dapat dilihat dari citra mobil MVP mengalami 3 kesalahan dengan 7 yang di prediksi benar dengan persentase 70% dan pada citra mini bus di prediksi benar semua dengan persentase 100% sesuai yang diinginkan. Sedangkan pada klasifikasi citra bus sama seperti mini bus semua di prediksi dengan tingkat akurasi 100% dan pada citra pick up mengalami 1 kesalahan dengan 4 citra yang di prediksi dengan persentase 80% sesuai yang diinginkan dan rata-rata akurasi sebesar 81,5%.

4.3.1.2 Pengujian 2

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 2.

Tabel 4.2 Pengujian 2

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (10)	7	0	0	0
Mini Bus (5)	0	5	0	0
Bus (10)	0	0	10	0
Pick Up (5)	0	0	0	5

Pada tabel 4.2 dapat dilihat pada citra mobil MVP mengalami 3 kesalahan dengan 7 yang di prediksi benar dengan tingkat akurasi sebesar 70% dan pada citra mini bus, bus, dan pick up tidak mengalami kesalahan prediksi, semua citra di prediksi benar dengan tingkat akurasi 100% sesuai dengan yang diinginkan dan mendapatkan rata-rata akurasi 84,566%.

4.3.1.3 Pengujian 3

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 3.

Tabel 4.3 Pengujian 3

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (10)	10	0	0	0
Mini Bus (5)	0	5	0	0
Bus (10)	0	0	7	0
Pick Up (5)	0	0	0	4

Pada tabel 4.3 prediksi terhadap citra mobil MVP dan mini bus benar semua tanpa ada kesalahan sesuai dengan yang diinginkan dengan tingkat akurasi sebesar 100%, sedangkan pada citra bus mengalami 3 kesalahan dengan 7 citra yang terprediksi dengan akursi sebesar 70% dan pada pick up mengalami 1 kesalahan dengan 5 citra yang terprediksi benar dengan akurasi pengujian sebesar 80% dan rata-rata akurasi sebesar 76,766%.

4.3.1.4 Pengujian 4

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 4.

Tabel 4.4 Pengujian 4

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (7)	5	0	0	0
Mini Bus (7)	0	7	0	0
Bus (10)	0	0	9	0
Pick Up (8)	0	0	0	7

Pada tabel 4.4 menunjukkan citra mobil MVP mengalami 2 kesalahan dengan 5 di prediksi benar dengan akurasi sebesar 71,42% dan pada citra minibus tidak mengalami kesalahan prediksi dan benar semua dengan akurasi 100% sesuai keinginan. Sedangkan pada citra bus mengalami 1 kesalahan dengan 9 citra di prediksi benar dengan akurasi 90% dan citra pick up mengalami 1 kesalahan dengan 7 di prediksi benar dengan tingkat akurasi 87,5% sesuai dengan yang diinginkan dan rata-rata akurasi sebesar 80,562%.

4.3.1.5 Pengujian 5

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 5.

Tabel 4.5 Pengujian 5

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (8)	5	0	0	0
Mini Bus (8)	0	8	0	0
Bus (7)	0	0	7	0
Pick Up (7)	0	0	0	7

Pada tabel 4.5 dapat diketahui objek yang di prediksi sebagai mobil MVP ada 2 kesalahan dengan 5 yang di prediksi benar dengan akurasi persentase sebesar 62,5% dan pada citra mini bus, bus, dan pick up tidak mengalami kesalahan prediksi dan benar semua dengan tingkat akurasi persentase sebesar 100% dan rata-rata akurasi sebesar 84,133% sesuai dengan yang diinginkan.

4.3.1.6 Pengujian 6

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 6.

Tabel 4. 6 Pengujian 6

Prediksi Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (10)	9	0	0	0
Mini Bus (5)	0	5	0	0
Bus (5)	0	0	5	0
Pick Up (10)	0	0	0	6

Pada tabel 4.6 dapat diketahui objek yang terdeteksi sebagai mobil MVP ada 9 citra dengan dengan 1 kesalahan prediksi dengan akurasi persentase 90% dan pada citra mini bus dan bus tidak mengalami kesalahan prediksi semua citra terdeteksi benar dengan tingkat akurasi 100%. Sedangkan pada citra pick up mengalami 4 kesalahan dengan 6 di prediksi benar dengan akurasi persentase sebesar 60 % dan rata-rata akursi sebesar 77,166%.

4.3.1.7 Pengujian 7

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 7.

Tabel 4.7 Pengujian 7

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (10)	8	0	0	0
Mini Bus (5)	0	5	0	0
Bus (5)	0	0	5	0
Pick Up (10)	0	0	0	9

Pada tabel 4.7 dapat diketahui objek yang terdeteksi sebagai mobil MVP ada 2 kesalahan dengan yang di prediksi benar ada 8 citra dengan persentase sebesar 80% dan pada mini bus dan bus semua citra yang di prediksi benar tanpa ada kesalahan dengan akurasi persentase 100%. Sedangkan pada citra pick up mengalami 1 kesalahan prediksi dengan 9 total citra yang di prediksi benar dan akuarasi persentase sebesar 90% sesuai dengan yang diinginkan dan mendapatkan rata-rata akurasi pengujian sebesar 81,033%.

4.3.1.8 Pengujian 8

Berikut adalah hasil akurasi terprediksi dari jenis kendaraan Mobil MVP, Mini Bus, Bus, dan Pick Up pada pengujian 8.

Tabel 4.8 Pengujian 8

Prediksi \ Aktual	Mobil MVP	Mini Bus	Bus	Pick Up
Mobil MVP (5)	5	0	0	0
Mini Bus (5)	0	5	0	0
Bus (10)	0	0	10	0
Pick Up (10)	0	0	0	10

Pada tabel 4.8 dapat diketahui pada semua objek dapat di prediksi semua tanpa ada ada kelahan sedikitpun. Dari percobaan terakhir ini semua objek mendapatkan nilai maksimal dengan tingkat akurasi persentase yaitu 100% dan rata-rata akurasi sebesar 92,233%. Hal ini berarti model *CNN* telah sangat baik dalam melakukan klasifikasi citra berdasarkan kelasnya.

4.3.1.9 Tabel Rata-rata Perbandingan Jumlah Data Pengujian

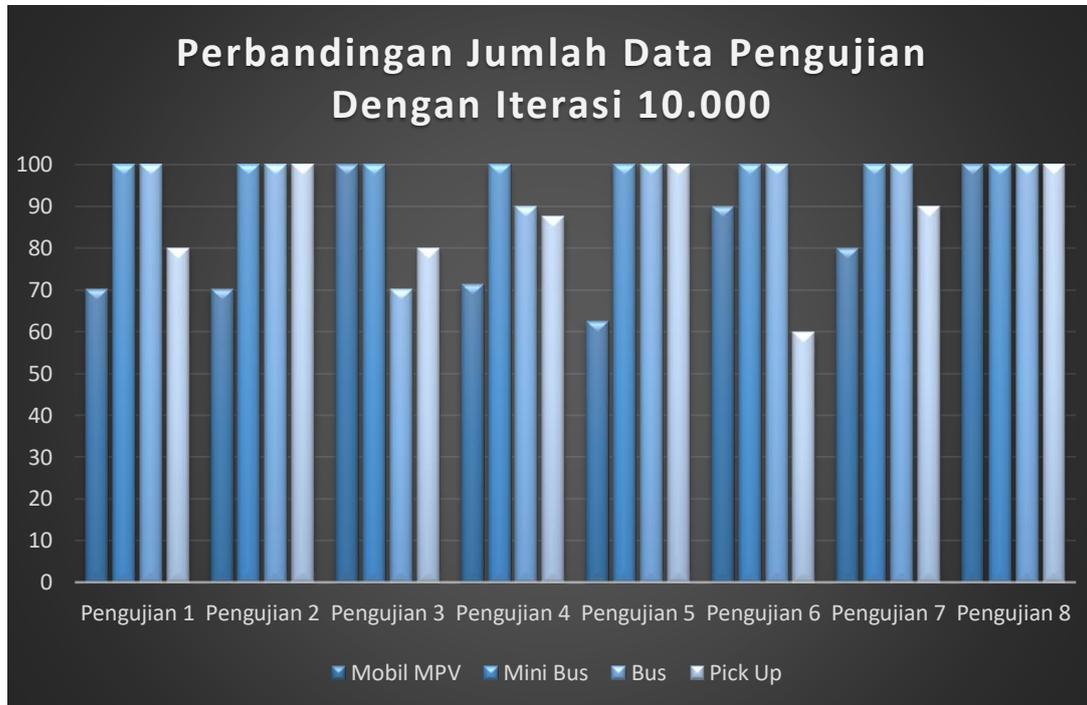
Tabel 4.9 Perbandingan Rata-rata Akurasi Terprediksi dan Persentase tingkat akurasi pengujian

Pengujian	Akurasi terprediksi (%)	Persentase tingkat akurasi pengujian (%)
1	86,666	81,5
2	90	84,566
3	86,666	76,766
4	87,5	80,562
5	90	84,133
6	83,333	77,166
7	90	81,033
8	100	92,233
	Rata-rata	81,948

Pada tabel 4.9 merupakan tabel perbandingan akurasi data uji dari penelitian yang di mana nilai akurasi yang didapatkan pada penelitian ini mendapatkan pesentase diatas 50% setiap objeknya sesuai dengan yang diinginkan penulis, hal ini berarti model *CNN* telah sangat baik dalam melakukan klasifikasi citra berdasarkan kelasnya dan dapat diimplementasikan terhadap citra jenis-jenis kendaraan.

4.3.1.10 Grafik Perbandingan Jumlah Data Pengujian

Berikut adalah grafik perbandingan jumlah data pengujian dengan *Iterasi* 10.000



Gambar 4.6 Perbandingan Jumlah Data Pengujian

Pada gambar 4.7 dapat dilihat bahwa dengan *iterasi* 10.000 telah didapatkan hasil yang sangat baik terhadap objek yang telah dideteksi. Ini menandakan bahwa sistem *CNN* yang digunakan dapat bekerja sangat baik sehingga mampu melakukan klasifikasi berdasarkan kelasnya dengan hasil yang sangat baik. Pada percobaan data pengujian 8 dapat dilihat bahwa objek dapat terdeteksi semua dengan nilai persentase sebesar 100%. Pada penelitian ini tidak diperlukan jumlah *Iterasi* yang cukup banyak karena dengan *Iterasi* 10.000 objek sudah dapat terdeteksi dengan sangat baik sesuai dengan yang ditargetkan.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil penelitian yang telah lakukan, didapatkan kesimpulan sebagai berikut :

1. Penelitian menggunakan 200 gambar dari jenis kendaraan, data tersebut merupakan data training dan data uji, di mana data training menggunakan 200 gambar yang terdiri dari 40 gambar (Minibus), 40 gambar (Bus), 40 gambar (Mobil MVP), 40 gambar (Pick Up), dan 40 gambar campuran dari (Minibus, Bus, Mobil MVP, dan Pick Up) dan data uji 25 gambar (Minibus, Bus, Mobil MVP, Pickup, dan gambar campuran) dengan *iterasi* sebesar 10.000
2. Arsitektur jaringan yang digunakan untuk mendeteksi jenis kendaraan terbagi menjadi beberapa layer yaitu *layer input*, *layer convolutional*, *layer activation*, *layer pooling* dan *fully connected layer*.
3. Hasil dari pendeteksian jenis-jenis kendaraan dengan menggunakan algoritma *Convolutional Neural Network* dapat nilai bekerja dengan sangat baik.
4. Hasil terbaik penelitian ini yaitu pada data pengujian 8 dengan mendapatkan hasil persentase akurasi sebesar 100% pada setiap objeknya.

5.2 Saran

Saran yang dapat dilakukan untuk penelitian selanjutnya agar lebih baik lagi kedepannya :

1. Untuk pengambilan gambar yang akan digunakan untuk *database* penelitian, ada baiknya menggunakan *camera* yang bagus agar hasil yang didapatkan maksimal serta dapat dibaca sistem dengan baik.
2. Memperbanyak *database* dan *iterasi* serta menggunakan *droupout regularization* untuk membantu peningkatan akurasi.
3. Gunakan *personal computer* dengan jumlah memori yang besar untuk menggunakan nilai *batch size* yang tinggi pada sistem *CNN* agar mempercepat proses dari *training* data.
4. Mencoba untuk melakukan deteksi jenis kendaraan dengan menggunakan data *real time*.

DAFTAR PUSTAKA

- [1] “Badan pusat statistik,” 2020. <https://www.bps.go.id/linkTableDinamis/view/id/1133>.
- [2] “No Title,” 2007. <http://bpjt.pu.go.id/konten/golongan-kendaraan>.
- [3] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [4] M. E. Muzakki *et al.*, “Klasifikasi Jenis Kendaraan dengan Gabor Filtering dan Naive Bayes,” *e-Proceeding Eng.*, vol. 4, no. 1, pp. 1083–1092, 2017.
- [5] M. Irfan, B. A. Ardi Sumbodo, and I. Candradewi, “Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 7, no. 2, p. 139, 2017, doi: 10.22146/ijeis.18260.
- [6] A. Andrew, J. L. Buliali, and A. Y. Wijaya, “Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV,” *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23489.
- [7] S. Li, J. Lin, G. Li, T. Bai, H. Wang, and Y. Pang, “Vehicle type detection based on deep learning in traffic scene,” *Procedia Comput. Sci.*, vol. 131, pp. 564–572, 2018, doi: 10.1016/j.procs.2018.04.281.
- [8] S. J. Lee, T. Chen, L. Yu, and C. H. Lai, “Image Classification Based on the Boost Convolutional Neural Network,” *IEEE Access*, vol. 6, pp. 12755–12768, 2018, doi: 10.1109/ACCESS.2018.2796722.
- [9] G. Hinton, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” vol. 15, pp. 1–30, 2014.