

**KLASIFIKASI PENENTUAN KAMERA CCTV BERDASARKAN  
TIPE JALAN MENGGUNAKAN ALGORITMA  
*DEEP NEURAL NETWORK (DNN)***

**SKRIPSI**



Oleh :  
**FAIQ NUKHA**  
**NIM. 13650051**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

**KLASIFIKASI PENENTUAN KAMERA CCTV BERDASARKAN TIPE  
JALAN MENGGUNAKAN ALGORITMA  
DEEP NEURAL NETWORK (DNN)**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
FAIQ NUKHA  
NIM. 13650051**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

**LEMBAR PERSETUJUAN**  
**KLASIFIKASI PENENTUAN KAMERA CCTV BERDASARKAN TIPE**  
**JALAN MENGGUNAKAN ALGORITMA**  
**DEEP NEURAL NETWORK (DNN)**

**SKRIPSI**

Oleh :  
**FAIQ NUKHA**  
**NIM. 13650051**

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : Desember 2019

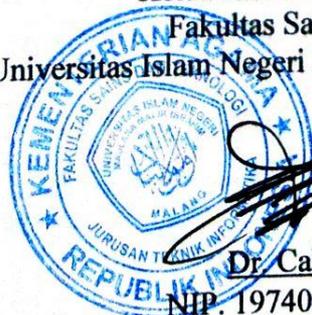
Dosen Pembimbing I

Dosen Pembimbing II

  
Fachrul Kurniawan, M.MT  
NIP. 19771020 200912 1 001

  
Dr. M. Faisal  
NIP. 19740510 200501 1 007

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

**LEMBAR PENGESAHAN**  
**KLASIFIKASI PENENTUAN KAMERA CCTV BERDASARKAN TIPE**  
**JALAN MENGGUNAKAN ALGORITMA**  
**DEEP NEURAL NETWORK (DNN)**

**SKRIPSI**

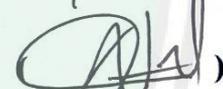
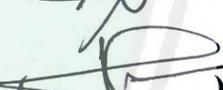
Oleh :  
**FAIQ NUKHA**  
**NIM. 13650051**

Telah Dipertahankan di Depan Dewan Penguji  
dan Diyantakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal : 10 Desember 2019

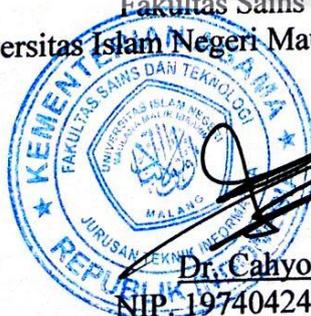
**Susunan Dewan Penguji**

- |                       |   |  |
|-----------------------|---|--|
| 1. Penguji Utama      | : | <u>Khadijah Fahmi Hayati Holle, M.Kom</u><br>NIDT. 19900626 20160801 2 077 |
| 2. Ketua Penguji      | : | <u>Fresy Nugroho, M. T</u><br>NIP. 19710722 201101 1 001                   |
| 3. Sekretaris Penguji | : | <u>Fachrul Kurniawan, M.MT</u><br>NIP. 19771020 200912 1 001               |
| 4. Anggota Penguji    | : | <u>Dr. M. Faisal</u><br>NIP. 19740510 200501 1 007                         |

**Tanda Tangan**

(  )  
(  )  
(  )  
(  )

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
**Dr. Cahyo Crysdian**  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Faiq Nukha

NIM : 13650051

Fakultas/Jurusan : Sains dan Teknologi/Teknik Infomatika

Judul Skripsi : Klasifikasi Penentuan Kamera CCTV Berdasarkan Tipe Jalan Menggunakan Algoritma *Deep Neural Network* (DNN)

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Desember 2019  
Yang membuat pernyataan,



Faiq Nukha  
NIM. 13650051

## MOTTO

“Jangan Berharap Adil, Kalau Takut Untuk Mendapatkan Hasil”



## HALAMAN PERSEMBAHAN

“Skripsi ini saya persembahkan untuk Ayah, Ibu, Mas, dan Adek Tercinta yang Selalu Memberikan Dukungan dan Nasehat Kepada Saya”



## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

Segala puji bagi Allah SWT, karena atas segala rahmat dan karunia-Nya, sehingga penulis mampu menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Skripsi dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman *jahiliyah* menuju islam yang *rahmatan lil alamin*.

Dalam proses menyelesaikan skripsi ini, penulis telah mendapatkan banyak bantuan dari berbagai pihak, baik secara moril, nasihat dan semangat maupun materiil. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada:

1. Bapak Fachrul Kurniawan, M.MT, selaku dosen pembimbing I yang telah bersedia meluangkan waktu untuk membimbing, memotivasi, dan mengarahkan selama proses pengerjaan skripsi ini.
2. Bapak Dr. M. Faisal, selaku dosen pembimbing II yang telah senantiasa memberikan masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
3. Segenap dosen dan staf Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan bimbingan dan ilmu sehingga dapat menjadi bekal untuk penulis dalam menyelesaikan studi dan skripsi selama masa studi.
4. Ayah dan Ibu tercinta serta keluarga besar penulis yang selalu memberikan doa, motivasi, nasihat, pengalaman berharga, serta membiayai penulis dalam menuntut ilmu dan menyelesaikan skripsi ini.

5. Teman-teman Jurusan Teknik Informatika, khususnya angkatan 2013, yang telah memberikan motivasi, informasi, dan masukannya pada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.

Berbagai kekurangan dan kesalahan mungkin dapat ditemukan dalam penulisan skripsi ini, oleh karena itu penulis menerima segala kritik dan saran akan penulis terima dengan senang hati dan akan menjadi bahan pertimbangan bagi penulis selanjutnya untuk menyempurnakan skripsi ini. Semoga karya ini senantiasa dapat memberi manfaat. Amin.  
*Wassalamualaikum Wr. Wb.*

Malang, Desember 2019

Penulis

## DAFTAR ISI

<b>HALAMAN PENGANTAR</b> .....	<b>i</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>ii</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>iii</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>iv</b>
<b>MOTTO</b> .....	<b>v</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>ix</b>
<b>DAFTAR GAMBAR</b> .....	<b>xi</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>ABSTRAK</b> .....	<b>xiv</b>
<b>ABSTRACT</b> .....	<b>xv</b>
<b>المخلص</b> .....	<b>xvi</b>
<b>BAB 1 PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah.....	4
1.3 Tujuan Penelitian .....	4
1.4 Batasan Masalah .....	4
1.5 Manfaat .....	5
<b>BAB 2 TINJAUAN PUSTAKA</b> .....	<b>6</b>
2.1 <i>Closed Circuit Television (CCTV)</i> .....	6
2.1.1 Pengertian CCTV .....	6
2.1.2 Sejarah CCTV .....	7
2.1.3 Jenis-Jenis CCTV.....	8
2.2 Pengertian Simpang .....	16
2.2.1 Jenis Simpang .....	16
2.2.2 Macam-Macam Simpang .....	17

2.2.3 Karakteristik Simpang.....	18
2.2.4 Pengendalian Simpang.....	18
2.2.5 Kinerja Persimpangan.....	20
2.3 <i>Deep Neural Networks</i> .....	20
2.4 Penelitian Terkait.....	25
<b>BAB 3 METODOLOGI PENELITIAN .....</b>	<b>29</b>
3.1 Prosedur Penelitian.....	29
3.2 Analisis Kebutuhan Sistem.....	30
3.2.1 Perangkat Lunak ( <i>Software</i> ).....	30
3.2.2 Perangkat Keras ( <i>Hardware</i> ).....	31
3.3 Pengumpulan Data.....	31
3.4 Normalisasi Data.....	34
3.5 Perancangan Sistem.....	35
3.5.1 Desain Sistem.....	35
3.5.2 Alur Algoritma DNN.....	56
3.5.3 Contoh Pembelajaran DNN Menggunakan <i>Backpropagation</i> .....	56
<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>	<b>66</b>
4.1 Langkah-langkah Uji Coba.....	66
4.2 Hasil Uji Coba.....	66
4.2.1 Hasil Proses <i>Training</i> .....	67
4.2.2 Hasil Proses <i>Testing</i> .....	72
4.2.3 Parameter <i>Epoch</i> dan <i>Learning Rate</i> Terhadap Akurasi.....	74
4.3 Pembahasan.....	78
<b>BAB 5 PENUTUP .....</b>	<b>83</b>
5.1 Kesimpulan.....	83
5.2 Saran.....	84
<b>DAFTAR PUSTAKA .....</b>	<b>86</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>PTZ Camera CCTV</i> .....	9
Gambar 2.2 <i>Dome Camera CCTV</i> .....	9
Gambar 2.3 <i>Bullet Camera CCTV</i> .....	10
Gambar 2.4 <i>Box Camera CCTV</i> .....	11
Gambar 2.5 <i>Board Camera CCTV</i> .....	11
Gambar 2.6 <i>Spy Camera CCTV</i> .....	12
Gambar 2.7 <i>IP Camera / Network Camera CCTV</i> .....	13
Gambar 2.8 <i>Wireless Camera Antenna</i> .....	13
Gambar 2.9 <i>HD CCTV Camera</i> .....	14
Gambar 2.10 <i>Outdoor CCTV Camera</i> .....	14
Gambar 2.11 <i>CCTV Verifocal Camera</i> .....	15
Gambar 2.12 <i>Infrared CCTV Camera</i> .....	15
Gambar 2.13 <i>Deep Neural Network</i> .....	21
Gambar 2.14 <i>Proses Latihan DNN Menggunakan Backpropagation</i> .....	22
Gambar 2.15 <i>Contoh Successive Learning</i> .....	23
Gambar 3.1 <i>Diagram Blok Prosedur Penelitian</i> .....	30
Gambar 3.2 <i>Diagram Desain Sistem</i> .....	35
Gambar 3.3 <i>Skema Jaringan</i> .....	51
Gambar 3.4 <i>Alur DNN Dengan Pembelajaran Back-Propagation</i> .....	56
Gambar 4.1 <i>Proses Training</i> .....	67
Gambar 4.2 <i>Data Training</i> .....	68
Gambar 4.3 <i>Penentuan Data pada Data Training</i> .....	69
Gambar 4.4 <i>Normalisasi min-max</i> .....	70
Gambar 4.5 <i>Pembuatan Model Jaringan</i> .....	71
Gambar 4.6 <i>Code Menampilkan Hasil Training Model Loss</i> .....	71
Gambar 4.7 <i>Hasil Proses Training Model Loss</i> .....	71
Gambar 4.8 <i>Code Menampilkan Hasil Training Model Accuracy</i> .....	71
Gambar 4.9 <i>Hasil Proses Training Model Accuracy</i> .....	72
Gambar 4.10 <i>Load Weights dan Data Testing</i> .....	72
Gambar 4.11 <i>Data Testing 5 Baris Awal dan 5 Baris Akhir</i> .....	73

Gambar 4.12 Normalisasi <i>nim-max</i> data testing .....	73
Gambar 4.13 Perintah <i>Testing</i> .....	74
Gambar 4.14 Grafik Parameter <i>Epoch</i> .....	75
Gambar 4.15 Grafik Parameter <i>Learning Rate</i> .....	77



## DAFTAR TABEL

Tabel 2.1 Hasil Pengujian Mulia Siregar (2015) .....	24
Tabel 2.2 Perbandingan <i>Epoch</i> dan <i>MSE</i> .....	24
Tabel 3.1 Data <i>Training</i> .....	36
Tabel 3.2 Data <i>Training</i> Sebelum Normalisasi .....	46
Tabel 3.3 Nilai <i>Min-Max</i> Tiap Baris dan Kolom .....	47
Tabel 3.4 Data <i>Training</i> Setelah Normalisasi .....	50
Tabel 3.5 Bobot <i>Input</i> ke <i>Hidden Layer 1</i> .....	52
Tabel 3.6 Nilai Bobot <i>Input</i> ke <i>Hidden Layer 1</i> .....	53
Tabel 3.7 <i>Bias Hidden Layer 1</i> .....	53
Tabel 3.8 Bobot <i>Hidden Layer 1</i> ke <i>Hidden Layer 2</i> .....	53
Tabel 3.9 Nilai Bobot <i>Hidden Layer 1</i> ke <i>Hidden Layer 2</i> .....	54
Tabel 3.10 <i>Bias Hidden Layer 2</i> .....	54
Tabel 3.11 Bobot dan Nilai Bobot <i>Hidden Layer 2</i> ke <i>Output</i> .....	54
Tabel 3.12 <i>Bias Output Layer</i> .....	55
Tabel 3.13 Data <i>Input</i> Pelatihan .....	58
Tabel 3.14 <i>Net Input Hidden Layer 1</i> .....	58
Tabel 3.15 Hasil Aktifasi <i>Net Input Hidden Layer 1</i> .....	59
Tabel 3.16 <i>Net Input Hidden Layer 2</i> .....	59
Tabel 3.17 Hasil Aktifasi <i>Net Input Hidden Layer 2</i> .....	60
Tabel 3.18 <i>Net Input Output Layer</i> .....	61
Tabel 3.19 Hasil Aktifasi <i>Net Input Output Layer</i> .....	61
Tabel 3.20 <i>Derrivative Cross Entropy</i> dan <i>Softmax</i> .....	63
Tabel 3.21 <i>Derrivative Hidden 2</i> ke <i>Output</i> .....	64
Tabel 3.22 <i>Derrivative Error</i> pada <i>Hidden Layer 2</i> .....	64
Tabel 3.23 Bobot Baru <i>Hidden Layer 2</i> ke <i>Output</i> .....	65
Tabel 4.1 <i>Output Data</i> .....	68
Tabel 4.2 <i>Output</i> setelah melakukan <i>excludes</i> .....	69
Tabel 4.3 <i>Epoch</i> Proses <i>Training</i> .....	74
Tabel 4.4 <i>Learning Rate</i> Proses <i>Training</i> .....	76

## ABSTRAK

Nukha, Faiq. 2019. **Klasifikasi Penentuan Kamera CCTV Berdasarkan Tipe Jalan Menggunakan Algoritma *Deep Neural Network* (DNN)**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.  
Pembimbing (I) Fachrul Kurniawan, M.MT. (II) Dr. M. Faisal

Kata Kunci : *deep neural network, preprocessing, normalize min-max, training, testing, backpropagation.*

Klasifikasi merupakan materi penelitian yang sampai saat ini terus dilakukan pengembangan. Dalam melakukan klasifikasi penentuan kamera CCTV, secara umum dilakukan oleh ahli/pakar, dimanapun lokasi yang akan dilakukan pemasangan CCTV. Algoritma *Deep Neural Network* (DNN) menjadi salah satu dari beberapa algoritma klasifikasi, baik untuk *supervised learning* maupun *unsupervised learning*. Dengan metode pembelajaran menggunakan *backpropagation* dan parameter lainnya yang digunakan untuk *learning*, seperti *epoch, activation function, dan loss*. Pada penelitian ini data yang akan dipakai merupakan data yang diperoleh dari pakar maupun dinas terkait, yang kemudian dibagi menjadi 2 bagian yaitu sebagai data *training* sebanyak 76 data dan data *testing* sebanyak 17 data. Kemudian, untuk mempermudah dalam melakukan training data, sebelumnya akan dilakukan normalisasi menggunakan normalisasi - *min-max*. Tujuan dari penelitian ini adalah untuk mengetahui cara dalam melakukan klasifikasi menggunakan algoritma DNN maupun untuk mengukur akurasi yang diperoleh dalam melakukan pengklasifikasian menggunakan algoritma DNN. Hasil yang diperoleh dan model jaringan sudah dibentuk, maka diperoleh nilai akurasi *training* sebesar 76.23% dan akurasi *testing* sebesar 88.24%.

## ABSTRACT

Nukha, Faiq. 2019. *Classification of CCTV Camera Determination Based on Road Types Using Deep Neural Network (DNN) Algorithm*. Thesis. Department of Informatics And Technology, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang.

Advisor (I) Fachrul Kurniawan, M.MT. (II) Dr. M. Faisal

**Keywords:** deep neural network, preprocessing, normalize min-max, training, testing, backpropagation.

Classification is a research material which until now continues to be developed. In carrying out the classification of the determination of CCTV camera, generally done by experts, wherever the location to be installed CCTV. Deep Neural Network (DNN) algorithm is one of several classification algorithms, both for supervised learning and unsupervised learning. With learning methods using backpropagation and other parameters used for learning, such as epoch, activation function, and loss. Research's data are data obtained from experts and related agencies, which are then divided into 2 parts, namely as much as 76 training data and as much as 17 testing data. Then, to make it easier to conduct training data, previously it will be normalized using -min-max normalization. The purpose of research is to determine how to classify using the DNN algorithm and to measure the accuracy obtained in doing classification using the DNN algorithm. The results obtained and the network model has been formed, then obtained the accuracy value of training is 76.23% and testing accuracy is 88.24%.

## الملخص

نوخا، فائق. ٢٠١٩. تصنيف تقدير كاميرات الدوائر التلفزيونية المغلقة المعتمد على أنواع الطرق باستخدام خوارزمية الشبكة العصبية العميقة (DNN). الأطروحة. قسم التقنية والمعلومات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الحكومية الإسلامية في مالانج.

المشرف (الأول) فخر كورنيوان، الماجستير والمشرف (الثاني) الدكتور فيصل

الكلمات الرئيسية: الشبكة العصبية العميقة ، معالجة مسبقة، تطبيع دقيقة، تدريب، اختبار، ارتجاع خلفي.

التصنيف هو مادة بحثية لا تزال التطوير حتى الآن. عاما ينفذ الخبراء تصنيف كاميرات الدوائر التلفزيونية المغلقة، أينما كان المكان المراد تركيبه به تعد خوارزمية الشبكة العصبية العميقة (DNN) واحدة من خوارزميات التصنيف العديدة للتعلم الخاضع للإشراف والتعلم غير الخاضع للإشراف. مع أساليب التعلم باستخدام *backpropagation* وغيرها من التعلم مثل عصر ووظيفة التنشيط والخسارة. البيانات في البحث التي حصل عليها من الخبراء والوكالات ذات الصلة، ثم قسمها إلى القسمين، أي ما يصل إلى ٧٦ من بيانات التدريب و ١٧ من بيانات الاختبار. وبعد، لتسهيل تدريب البيانات استخدم الباحث الاختبار الطبيعي باستخدام *min-max*. اهداف البحث منها تحديد كيفية التصنيف باستخدام خوارزمية الشبكة العصبية العميقة وقياس الدقة التي حصل عليها في التصنيف باستخدام خوارزمية الشبكة العصبية العميقة. النتائج هي تشكيل نموذج الشبكة، ثم حصل على قيمة دقة التدريب ٧٦,٢٣ ٪ ودقة الاختبار ٨٨,٢٤ ٪.

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Pelanggaran lalu lintas di jalan menyebabkan ketidaknyamanan para pengguna jalan. Hal ini disebabkan, karena salah satunya adalah ketidak-disiplinan para pengguna jalan dalam mentaati rambu-rambu lalu lintas. Pada 2016, jumlah pelanggaran lalu lintas di wilayah hukum Polres Malang naik 35 persen dibandingkan tahun 2015. Pelanggaran berupa tilang pada 2016 mencapai 35.492 kasus, sedangkan tahun 2015 sejumlah 26.222 kasus. Sementara, pelanggaran berupa teguran mengalami penurunan *trend* hingga 76 persen (TimesIndonesia, 2017).

Dari data yang didapat, maka dibutuhkan penggunaan CCTV dalam melakukan pemantauan kondisi lalu lintas di jalan. *Closed Circuit Television* (CCTV) adalah penggunaan kamera menangkap gambar video untuk melakukan transmisi *signal* video ke tempat spesifik, biasanya ke dalam beberapa set monitor. Berbeda dengan siaran televisi, sinyal CCTV lebih cenderung tertutup dalam melakukan transmisi.

CCTV pertama kali digunakan sebagai alat efektif untuk mencegah kejahatan. Perkembangan teknologi telah mengembangkan peran CCTV dalam hal keamanan dan investigasi perilaku kriminal dan anti-sosial. Yang meskipun CCTV sendiri tidak dapat mencegah tindak kriminal atau pelanggaran yang terjadi, namun setidaknya hanya untuk mengamati, mengontrol, dan mengidentifikasi suatu keamanan pada lokasi tertentu.

Namun, dalam penggunaan CCTV di jalan memiliki kendala dalam hal penentuan atau pemilihan CCTV apa yang cocok dengan berbagai tipe jalan, karena banyak hal yang harus diperhatikan untuk mendapatkan kamera CCTV yang sesuai dengan kondisi dan lokasi jalan yang diinginkan. Terkadang para teknisi *support* mengalami kendala dalam menjelaskan konsultasi memilih kamera yang sesuai dengan lokasi persimpangan kepada *end user*, yang disebabkan ketika teknisi *support* menjelaskan alur kinerja kamera CCTV, *end user* tidak dapat menangkap penjelasan yang hanya menjadi bayang-bayang oleh *end user*.

Ada banyak metode yang bisa digunakan untuk melakukan klasifikasi, salah satunya adalah menggunakan teknik *Deep Learning* (DL) yaitu metode *Deep Neural Network* (DNN). Terdapat penelitian dimana DNN menunjukkan hasil dengan akurasi yang lebih bagus daripada *neural network* biasa, yaitu penelitian yang dilakukan oleh (Mulia Siregar 2015) untuk penilaian kinerja dosen, dimana dilakukannya perbandingan dengan skema jaringan 8-2-2 dengan skema jaringan 8-3-3-3-2, menunjukkan hasil pada model 8-2-2 memiliki output yang salah/tidak sesuai dengan target sebanyak 4 item dari data sebanyak 56 item, sedangkan pada model 8-3-3-3-2 memiliki hasil yang sesuai dengan target uji keseluruhan. Dan dengan perbandingan *epoch* dan MSE untuk model 8-2-2 dengan *epoch* = 260359 dan MSE = 0.000999522, sedangkan untuk model 8-3-3-3-2 dengan *epoch* = 4898 dan MSE = 0,000996042. Penelitian tersebut menyebutkan bahwa metode *Deep Neural Network* (DNN) dapat bekerja lebih baik daripada *neural network* biasa

Sesuai dengan permasalahan tersebut, maka diperoleh solusi untuk membangun suatu cara dengan mengolah data dalam penentuan kamera CCTV berdasarkan tipe jalan dengan menerapkan metode Algoritma *Deep Neural*

*Networks* (DNN). Dengan tujuan memudahkan teknisi support dalam menentukan kamera apa yang akan digunakan dalam pemasangan CCTV di jalan. Dengan adanya solusi tersebut, diharapkan teknisi support konsisten dalam menggunakan hasil penelitian ini dan diterapkan dalam permasalahan tersebut.

Penelitian ini akan dapat bermanfaat nantinya ketika akan dikembangkan lebih lanjut, terlebih khusus jika dikembangkan dalam bidang keamanan. Seperti pada penelitian ini, dimana akan diterapkannya penentuan kamera CCTV berdasarkan tipe jalan. sabda Rasulullah SAW yang menganjurkan umatnya untuk menjadi seseorang yang bermanfaat untuk lainnya. Rasulullah SAW bersabda:

حَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

“Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia” (HR. Ahmad, ath-Thabrani, ad-Daruqutni. Hadits ini dihasankan oleh al-Albani di dalam Shahihul Jami’ no:3289).

Menjadi seseorang yang bermanfaat untuk orang lain adalah salah satu karakter yang harus dimiliki seorang muslim. Karena pada dasarnya, ketika kita memberikan atau melakukan sesuatu yang bermanfaat untuk orang lain, maka manfaatnya akan kembali untuk kebaikan diri kita sendiri. Hal ini dijelaskan dalam firman Allah SWT pada surah al-isra ayat 7 :

إِنْ أَحْسَنْتُمْ أَحْسَنْتُمْ لِأَنْفُسِكُمْ

“Jika kalian berbuat baik, sesungguhnya kalian berbuat baik bagi diri kalian sendiri” (QS. Al-Isra:7).

## 1.2 Identifikasi Masalah

1. Bagaimana pengklasifikasian penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN?
2. Bagaimana tingkat akurasi yang diperoleh dalam pengklasifikasian penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN?
3. Bagaimana hasil pengklasifikasian penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN?

## 1.3 Tujuan Penelitian

1. Mengetahui cara klasifikasi penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN.
2. Mengetahui tingkat akurasi yang diperoleh dalam pengklasifikasian penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN.
3. Mengetahui hasil pengklasifikasian penentuan kamera CCTV berdasarkan tipe jalan menggunakan metode DNN.

## 1.4 Batasan Masalah

Dari rumusan masalah yang telah di uraikan di atas, maka penulis membatasi identifikasi masalah agar pokok bahasan tidak menyimpang, yaitu:

1. CCTV yang digunakan dalam penelitian ini merupakan CCTV merk Samsung, dengan kriteria tipe yang digunakan adalah tipe *bullet*, *dome*, dan PTZ, disebabkan melihat dari kondisi di lapangan menggunakan tipe tersebut.
2. Tipe jalan yang digunakan dalam penelitian ini adalah jalan lurus, pertigaan, perempatan, dan bundaran.

### 1.5 Manfaat

Sebagai alternatif solusi pemecahan kepada teknisi support terkait penentuan kamera CCTV yang cocok untuk berbagai tipe jalan dan juga mengetahui kelebihan dan kekurangan algoritma DNN dalam perancangan sistem klasifikasi tersebut.



## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 *Closed Circuit Television (CCTV)*

##### 2.1.1 Pengertian CCTV

Menurut Herman Dwi Surjono (1996), *Closed Circuit Television (CCTV)* merupakan alat perekaman yang menggunakan satu atau lebih kamera video dan menghasilkan data video atau audio. CCTV memiliki manfaat sebagai dapat merekam segala aktifitas dari jarak jauh tanpa batasan jarak, dapat memantau dan merekam segala bentuk aktifitas yang terjadi di lokasi pengamatan dengan menggunakan laptop atau PC secara *real time* dari mana saja, dan dapat merekam seluruh kejadian secara 24 jam, atau dapat merekam ketika terjadi gerakan dari daerah yang terpantau.

Menurut Ansel Adams, CCTV merupakan sebuah media audio visual yang sangat kuat dan CCTV sangat menawarkan berbagai persepsi yang menjadi sebuah interpretasi dan eksekusi yang tak terbatas. CCTV menjadi bentuk dari kamera pengamanan dan pemantauan akan segala kejadian.

Menurut Elliot Erwint, CCTV merupakan sebuah alat untuk mengeksplorasi seni observasi yang akan menemukan sebuah hal yang luar biasa pada tempat-tempat yang memungkinkan akan segala kejadian. Dengan fotografi juga akan menjadikan sebuah tempat dengan visual yang biasa akan menjadi sebuah hal yang lebih klasik dan unik. CCTV juga bisa menjadi sebuah bentuk perbedaan antara kebanyakan orang lihat dan apa yang kamu lihat.

Menurut Amir Hamzah, CCTV merupakan sebuah bentuk alat untuk merekam dan juga memfoto akan segala kejadian yang terjadi pada sebuah tempat.

### 2.1.2 Sejarah CCTV

*Closed Circuit Television* (CCTV) adalah penggunaan kamera untuk menangkap gambar video yang kemudian mentransmisikan *signal* video ke tempat spesifik, biasanya ke dalam beberapa set monitor. Berbeda dengan siaran televisi, sinyal CCTV lebih cenderung tertutup dalam melakukan transmisi. CCTV paling banyak digunakan untuk pengawasan keamanan pada area yang memerlukan *monitoring* seperti bank, gudang, tempat umum (jalan raya, dll), dan rumah yang ditinggal oleh pemiliknya.

Sistem CCTV umumnya terdiri dari komunikasi *fixed (dedicated)* antara kamera serta monitor. Teknologi CCTV terkini terdiri berasal sistem terkoneksi dengan kamera yang bisa digerakkan (diputar, ditekuk, serta di-*zoom*) dan bisa dioperasikan dari jeda jauh lewat ruang kontrol, dan bisa dihubungkan menggunakan suatu jaringan baik LAN, *Wireless-LAN* maupun internet.

Sistem CCTV pertama dipasang oleh *Siemens AG* pada *TestStand VII* di Peenemunde, Jerman pada tahun 1942. CCTV tersebut digunakan untuk mengamati peluncuran V-2 roket, mencatat insinyur dari Jerman (*Walter Bruch*) yang bertanggung jawab untuk desain dan instalasi sistem. Sistem perekaman CCTV masih sering digunakan di tempat peluncuran modern untuk merekam penerbangan roket, untuk menemukan kemungkinan penyebab kerusakan, sementara roket yang lebih besar sering dilengkapi dengan CCTV yang memungkinkan gambar-gambar sebagai tahapan pemisahan ditransmisikan balik ke bumi dengan *link* radio.

Pada bulan September 1968, Olean, New York adalah kota pertama di Amerika Serikat yang menginstal kamera video sepanjang jalan bisnis utama dalam upaya untuk memerangi kejahatan. Penggunaan kamera televisi sirkuit tertutup untuk perpindahan gambar ke kepolisian Olean sehingga mendorong Departemen Olean ke teknologi terdepan melawan kejahatan. Penggunaan CCTV di kemudian hari menjadi sangat umum di bank dan toko digunakan untuk mencegah pencurian, dengan cara melalui merekam bukti aktivitas kriminal. Penggunaannya lebih lanjut oleh sebab itu populer dengan konsep lain. daerah pertama yang memakai CCTV pada Britania Raya adalah King's Lynn, Norfolk.

Dalam dekade belakangan ini, terutama dengan ketakutan kejahatan umum berkembang pada 1990-an/2000-an, dan penggunaan ruang publik kamera pengintai telah mati, khususnya di beberapa negara seperti Britania Raya. (Atmoko 2005)

### 2.1.3 Jenis-Jenis CCTV

#### 1. PTZ Camera

PTZ adalah singkatan berasal dari *PAN TILT ZOOM*, *PAN* kemampuan kamera agar bisa beroperasi ke kiri serta ke kanan, *TILT* kemampuan kamera bisa beroperasi ke atas dan kebawah, *ZOOM* kemampuan kamera agar bisa memperbesar gambar sampai beberapa kali lipat. Jenis kamera PTZ biasa difungsikan buat memantau wilayah yang luas hanya memakai 1 kamera, ini memudahkan pengawas CCTV dalam melakukan monitoring hanya dengan memakai 1 kamera, karena *ptz camera* bisa berputar otomatis atau secara manual digerakan melalui *controller*, dapat dilihat pada Gambar 2.1.



Gambar 2.1 PTZ Camera CCTV

## 2. *Dome Camera*

Diambil nama *Dome* sebab bentuknya yang mirip kubah (pada bahasa Inggris), tujuannya supaya arah asal kamera CCTV tak terlihat atau tersembunyi akan tetapi terlihat oleh kasat mata. *Dome* kamera yang biasa dijual merupakan tipe *fix* camera yaitu kamera yang hanya menunjuk ke 1 arah, akan tetapi jenis *dome* kamera juga biasanya ada yang bisa berputar serta cepat “*Speed Dome*“, seperti pada Gambar 2.2.



Gambar 2.2 *Dome Camera* CCTV

## 3. *Bullet Camera*

Jenis CCTV ini umumnya dipergunakan pada ruangan (*indoor cam*) dan diluar ruangan (*outdoor cam*) tentunya merupakan salah satu standar yang wajib

dipenuhi ialah tahan air. *Bullet* kamera dipasang pada dinding ataupun langit. Kamera jenis ini tidak dirancang untuk memiliki *pan / tilt / zoom control* merupakan kamera tipe *fix* dengan tujuan menangkap gambar dari area yang tetap, seperti pada Gambar 2.3.



Gambar 2.3 *Bullet Camera CCTV*

#### 4. *Box Camera*

Mempunyai kemampuan *zoom* dengan penempatan pemasangan pada bidang vertikal, kekurangan kamera jenis ini membutuhkan pencahayaan untuk dapat menangkap gambar dengan jelas. Dapat menggunakan *infrared* dengan alat tambahan serta penggunaan lensa *infrared* pada kamera ini dan akan lebih baik apabila *box* camera dilengkapi dengan *housing* kamera apabila masih dalam jangkauan tangan, dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Box Camera* CCTV

#### 5. *Board Camera*

Biasanya terhubung pada media komputer ataupun lainnya rata-rata mempunyai resolusi yang rendah, karena biasanya *board camera* digunakan untuk aplikasi *tele-conference* standar, seperti Gambar 2.5.



Gambar 2.5 *Board Camera* CCTV

#### 6. *Day/Night Camera*

Kamera tipe *day/night* merubah aneka macam kondisi cahaya agar bisa diadaptasi dengan sinar matahari secara langsung, *backlight* yang bertenaga, refleksi memiliki *dynamic range* yang luas, kegunaan *day/night* kamera umumnya dipasang di lokasi yang mempunyai pencahayaan yang berlebihan dan pada malam hari mempunyai cahaya yang cukup.

### 7. *Spy Camera*

Dinamakan *spy camera* atau *convert camera* sebab memang ditujukan buat memata-matai, bentuknya pada banyak sekali variasi seperti, jam, *smoke detector*, pulpen, serta masih banyak lagi, seperti pada Gambar 2.6.



Gambar 2.6 *Spy Camera* CCTV

### 8. *Ip Camera / Network Camera*

Kamera jenis *ip / network* baik itu dengan kabel ataupun *wireless* kabel dapat dipasang dengan mudah, rata-rata *ip camera* mempunyai tingkat resolusi gambar yang lebih tinggi dibandingkan kamera CCTV analog, namun ada beberapa syarat yang harus dipenuhi dalam penggunaan *ip camera* seperti area lokasi yang akan dipasang *ip cam* tipe *wireless* harus ter-cover jaringan *wireless* internet, dan untuk tipe *ip camera* dengan kabel jarak penarikan kabel ke *switch hub / router* hanya 80- 100 meter dengan menggunakan kabel utp, seperti pada Gambar 2.7.

Beberapa kekurangan kamera jenis *ip camera*:

- a. Karena mempunyai resolusi yang tinggi membutuhkan kapasitas *hard drive* yang lebih besar,

- b. Harga *ip camera* yang relative lebih mahal dibandingkan CCTV *camera analog*,
- c. Membutuhkan alat tambahan untuk penguat penerima sinyal (untuk lokasi yg jauh atau banyak hambatan),
- d. Membutuhkan pengertian konfigurasi internet bila memasang *ip camera* dalam jumlah banyak.



Gambar 2.7 IP Camera / Network Camera CCTV

#### 9. Wireless Camera

Tidak semua kamera *wireless* CCTV berbasis IP , beberapa dari kamera jenis *wireless* dapat menggunakan model alternatif dalam transmisi *wireless*, seperti pada Gambar 2.8.



Gambar 2.8 Wirelles Camera Antenna

#### 10. *HD (High-definition) Camera*

Kamera dengan spesifikasi HD tidak perlu dipertanyakan lagi untuk kualitas gambar, Kamera *high-definition* dengan berkemampuan digital *zoom* membantu CCTV kamera / *ip cam* memperbesar gambar dengan sangat jelas, seperti pada Gambar 2.9.



Gambar 2.9 HD CCTV Camera

#### 11. *Outdoor Camera*

Inti dari *outdoor* kamera sendiri adalah sebuah kamera yang mampu bertahan di segala kondisi cuaca, mempunyai bahan material yang baik, tahan air, kedap udara terhindar dari masuknya debu, dapat juga menggunakan *housing* kamera sebagai alat untuk melindungi kamera., seperti pada Gambar 2.10.



Gambar 2.10 Outdoor CCTV Camera

## 12. *Varifocal Camera*

Kamera *CCTV* yang mempunyai lensa varifokal yang dapat diperbesar atau disesuaikan manual tanpa mengubah titik fokus kamera, seperti pada Gambar 2.11.



Gambar 2.11 *CCTV Varifocal Camera*

## 13. *IR (Infrared Camera)*

Disebut juga *night vision camera*, kamera yang mampu melihat pada malam hari bahkan gelap gulita dengan menggunakan lampu IR LED, *infrared* kamera menghasilkan gambar hitam putih pada saat malam hari, seperti pada Gambar 2.12



Gambar 2.12 *Infrared CCTV Camera*

## 2.2 Pengertian Simpang

Menurut Kamus Besar Bahasa Indonesia (1995), simpang merupakan daerah berbelok atau bercabang berasal dari yang lurus.

Menurut (Abubakar 1995), persimpangan merupakan simpul pada jaringan jalan dimana jalan-jalan bertemu dan lintasan kendaraan berpotongan. Lalu lintas pada masing-masing kaki persimpangan menggunakan ruang bagian jalan pada persimpangan secara bersama-sama dengan lalu lintas lainnya. Persimpangan merupakan faktor yang paling penting dalam menentukan kapasitas dan waktu perjalanan pada suatu jaringan jalan, khususnya di daerah perkotaan.

Menurut (Hendarto 2001), persimpangan merupakan daerah dimana dua atau lebih dari jalan bergabung atau berpotongan/bersilangan.

### 2.2.1 Jenis Simpang

Menurut Direktorat Jendral Bina Marga dalam Manual Kapasitas Jalan Indonesia (1997), pemilihan jenis simpang untuk suatu daerah sebaiknya berdasarkan pertimbangan ekonomi, pertimbangan keselamatan lalu lintas, dan pertimbangan lingkungan.

Menurut (Morlok 1988), pengaturan persimpangan dilihat dari segi pandang untuk *control* kendaraan dapat dibedakan menjadi dua yaitu:

1. Persimpangan tanpa sinyal, dimana pengemudi kendaraan sendiri yang harus memutuskan apakah aman untuk memasuki persimpangan itu.
2. Persimpangan dengan sinyal, dimana persimpangan itu diatur sesuai sistem dengan tiga aspek lampu yaitu merah, kuning dan hijau.

Yang dijadikan kriteria bahwa suatu persimpangan sudah harus dipasang alat pemberi isyarat lalu lintas menurut Ditjen. Perhubungan Darat, 1998 adalah:

1. Arus minimal lalu lintas yang menggunakan persimpangan rata – rata diatas 750 kendaraan/jam, terjadi secara kontinu 8 jam sehari.
2. Waktu tunggu atau hambatan rata – rata kendaraan di persimpangan melampaui 30 detik.
3. Persimpangan digunakan oleh rata – rata lebih dari 175 pejalan kaki/jam terjadi secara kontinu 8 jam sehari.
4. Sering terjadi kecelakaan pada persimpangan yang bersangkutan.
5. Pada daerah yang bersangkutan dipasang suatu sistem pengendalian lalu lintas terpadu (*Area Traffic Control / ATC*), sehingga setiap persimpangan yang termasuk di dalam daerah yang bersangkutan harus dikendalikan dengan alat pemberi isyarat lalu lintas.

Syarat – syarat yang disebut di atas tidak baku dan dapat disesuaikan dengan situasi dan kondisi setempat.

### 2.2.2 Macam-Macam Simpang

Menurut (Hariyanto 2004), dilihat dari bentuknya ada 2 (dua) macam jenis persimpangan, yaitu :

1. Pertemuan atau persimpangan jalan sebidang, merupakan pertemuan dua ruas jalan atau lebih secara sebidang (tidak saling bersusun). Pertemuan jalan sebidang ada 4 (empat) macam, yaitu :
  - a. pertemuan atau persimpangan bercabang 3 (tiga),
  - b. pertemuan atau persimpangan bercabang 4 (empat),
  - c. pertemuan atau persimpangan bercabang banyak,

- d. bundaran (*rotary intersection*).
2. Pertemuan atau persimpangan jalan tidak sebidang, merupakan persimpangan dimana dua ruas jalan atau lebih saling bertemu tidak dalam satu bidang tetapi salah satu ruas berada di atas atau di bawah ruas jalan yang lain

### 2.2.3 Karakteristik Simpang

Menurut (Hariyanto 2004), dalam perencanaan suatu simpang, kekurangan dan kelebihan dari simpang bersinyal dan simpang tak bersinyal harus dijadikan suatu pertimbangan. Adapun karakteristik simpang bersinyal dibandingkan simpang tak bersinyal adalah sebagai berikut :

1. kemungkinan terjadinya kecelakaan dapat ditekan apabila tidak terjadi pelanggaran lalu lintas,
2. lampu lalu lintas lebih memberi aturan yang jelas pada saat melalui simpang,
3. simpang bersinyal dapat mengurangi konflik yang terjadi pada simpang, terutama pada jam sibuk,
4. pada saat lalu lintas sepi, simpang bersinyal menyebabkan adanya tundaan yang seharusnya tidak terjadi.

### 2.2.4 Pengendalian Simpang

Menurut (Atisusanti 2009), sesuai dengan kondisi lalu lintasnya, dimana terdapat pertemuan jalan dengan arah pergerakan yang berbeda, simpang sebidang merupakan lokasi yang potensial untuk menjadi titik pusat konflik lalu lintas yang bertemu, penyebab kemacetan, akibat perubahan kapasitas, tempat terjadinya

kecelakaan, konsentrasi para penyeberang jalan atau pedestrian. Masalah utama yang saling mengkait di persimpangan adalah :

1. *volume* dan kapasitas, yang secara langsung mempengaruhi hambatan,
2. desain geometrik, kebebasan pandangan dan jarak antar persimpangan,
3. kecelakaan dan keselamatan jalan, kecepatan, lampu jalan,
4. pejalan kaki, parkir, akses dan pembangunan yang sifatnya umum.

Menurut (Abubakar 1995), sasaran yang harus dicapai pada pengendalian persimpangan antara lain adalah :

1. mengurangi atau menghindari kemungkinan terjadinya kecelakaan yang disebabkan oleh adanya titik-titik konflik seperti : berpencar (*diverging*), bergabung (*merging*), berpotongan (*crossing*), dan bersilangan (*weaving*),
2. menjaga agar kapasitas persimpangan operasinya dapat optimal sesuai dengan rencana,
3. harus memberikan petunjuk yang jelas dan pasti serta sederhana, dalam mengarahkan arus lalu lintas yang menggunakan persimpangan.

Menurut (Abubakar 1995), dalam upaya meminimalkan konflik dan melancarkan arus lalu lintas ada beberapa metode pengendalian persimpangan yang dapat dilakukan, yaitu :

- a. persimpangan prioritas

Metode pengendalian persimpangan ini adalah memberikan prioritas yang lebih tinggi kepada kendaraan yang datang dari jalan utama dari semua kendaraan yang bergerak dari jalan kecil (jalan minor),

- b. persimpangan dengan lampu pengatur lalu lintas

Metode ini mengendalikan persimpangan dengan suatu alat yang sederhana (manual, mekanis, dan elektris) dengan memberikan prioritas bagi masing-masing pergerakan lalu lintas secara berurutan untuk memerintahkan pengemudi berhenti atau berjalan,

c. persimpangan dengan bundaran lalu lintas

Metode ini mengendalikan persimpangan dengan cara membatasi alih gerak kendaraan menjadi pergerakan berpecah (*diverging*), bergabung (*merging*), berpotongan (*crossing*), dan bersilangan (*weaving*) sehingga dapat memperlambat kecepatan kendaraan,

d. persimpangan tidak sebidang

Metode ini mengendalikan konflik dan hambatan di persimpangan dengan cara menaikkan lajur lalu lintas atau di jalan di atas jalan yang lain melalui penggunaan jembatan atau terowongan.

### 2.2.5 Kinerja Persimpangan

Menurut Departemen Pendidikan dan Kebudayaan dalam Kamus Besar Bahasa Indonesia (1995), kinerja adalah suatu yang dicapai atau pergerakan sistem.

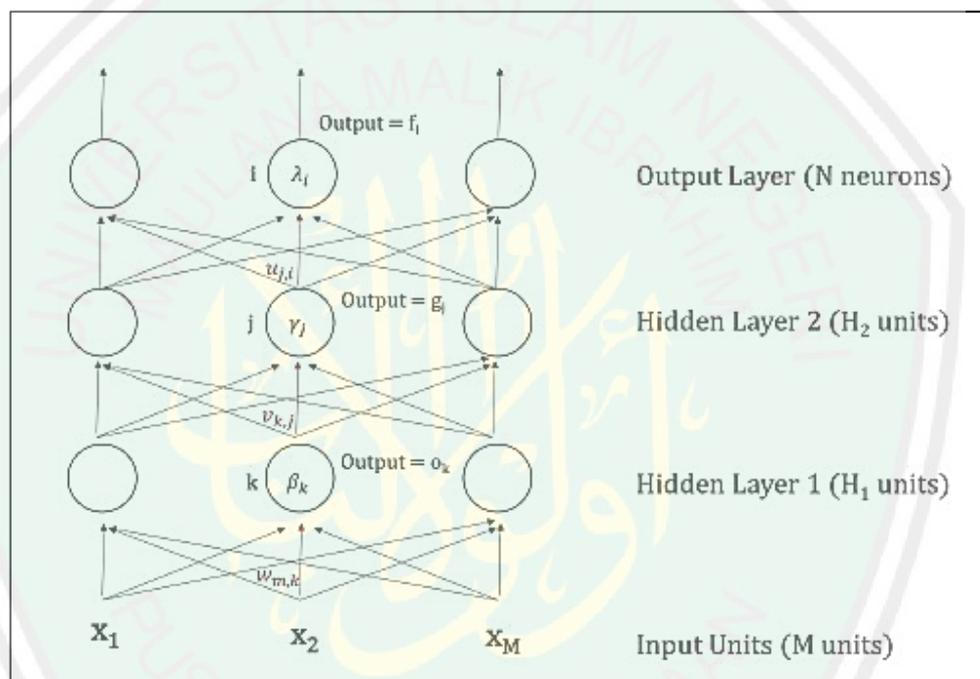
Menurut (Abubakar 1995), meningkatkan kinerja pada semua jenis persimpangan dari segi keselamatan dan efisiensi adalah dengan melakukan pelaksanaan dalam pengendalian persimpangan.

## 2.3 *Deep Neural Networks*

*Deep Neural Network* (DNN) adalah *artificial neural network* yang memiliki banyak *layer*. Pada umumnya, *deep neural network* memiliki lebih dari 3 *layers*

(*input layer, N hidden layers, output layer*), dengan kata lain adalah MLP dengan lebih banyak *layer*. Karena ada relatif banyak *layer*, disebutlah *deep*. Proses pembelajaran pada DNN disebut sebagai *deep learning* (Watanabe and Nishimori 2016). Jaringan *neural network* pada DNN disebut *deep network*.

Pada Gambar 2.13 yang memiliki 4 *layers*. Cara menghitung final *output* sama seperti MLP, diberikan pada persamaan 2.1 dibawah dimana  $\beta, \gamma, \lambda$  adalah *noise* atau bias.



Gambar 2.13 Deep Neural Network

$$f_i = \sigma \left( \sum_{j=1}^{H_2} u_{j,i} g_j, i^{\sigma} \left( \sum_{k=1}^{H_1} v_{k,j} o_k, j^{\sigma} \left( \sum_{m=1}^M x_m w_{m,k} + \beta_k \right) + \gamma_j \right) + \lambda_i \right) \quad (2.1)$$

Cara melatih *deep neural network*, salah satunya dapat menggunakan *backpropagation*. Seperti pada bagian sebelumnya, kita hanya perlu menurunkan rumusnya saja. Penurunan diserahkan pada pembaca sebagai latihan. Hasil proses penurunan dapat dilihat pada Gambar 2.14.

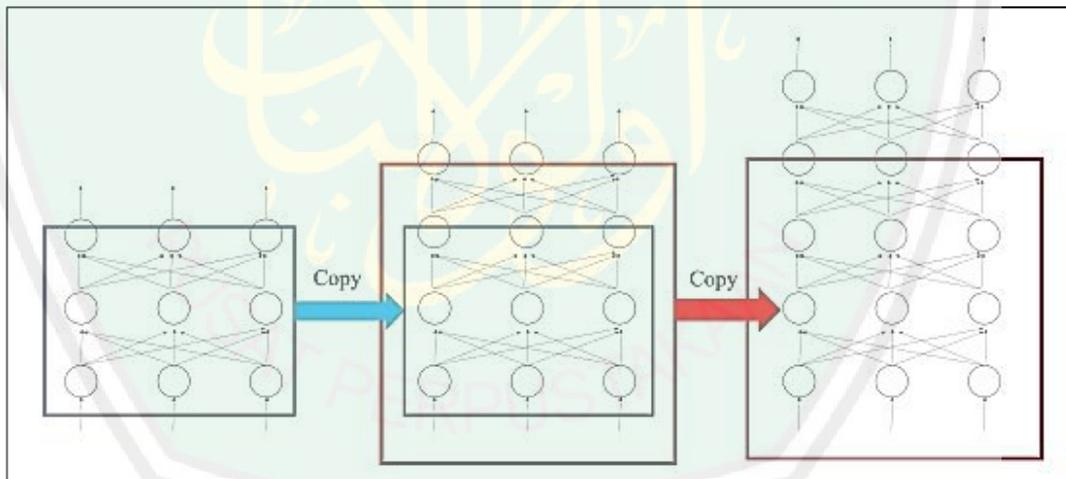
<p><b>(3) Hidden 2 to Output</b></p> $f_i = \sigma \left( \sum_{j=1}^{H_2} g_j u_{j,i} + \lambda_i \right)$	<p><b>(4) Output to Hidden 2</b></p> $\delta_i = (y_i - f_i) f_i (1 - f_i)$ $\Delta u_{j,i} = -\eta(t) \delta_i g_j$ $\Delta \lambda_i = -\eta(t) \delta_i$
<p><b>(2) Hidden 1 to Hidden 2</b></p> $g_j = \sigma \left( \sum_{k=1}^{H_1} o_k v_{k,j} + \gamma_j \right)$	<p><b>(5) Hidden 2 to Hidden 1</b></p> $\varphi_j = \sum_{i=1}^{H_2} \delta_i u_{j,i} g_j (1 - g_j)$ $\Delta v_{k,j} = -\eta(t) \varphi_j o_k$ $\Delta \gamma_j = -\eta(t) \varphi_j$
<p><b>(1) Input to Hidden Layer</b></p> $o_k = \sigma \left( \sum_{m=1}^M x_m w_{m,k} + \beta_k \right)$	<p><b>(6) Hidden 1 to Input</b></p> $\mu_k = \sum_{j=1}^{H_1} \varphi_j v_{k,j} o_k (1 - o_k)$ $\Delta w_{m,k} = -\eta(t) \mu_k x_m$ $\Delta \beta_k = -\eta(t) \beta_k$

Gambar 2.14 Proses Latihan DNN Menggunakan *Backpropagation*.

*Deep network* terdiri dari banyak layer dan *synapse weight*, karenanya estimasi parameter susah dilakukan. Arti filosofinya adalah susah/lama untuk menentukan relasi antara *input* dan *output*. Walaupun *deep learning* sepertinya kompleks, tetapi entah kenapa dapat bekerja dengan baik untuk permasalahan praktis (Watanabe and Nishimori 2016). *Deep learning* dapat menemukan relasi “tersembunyi” antara *input* dan *output*, yang tidak dapat diselesaikan menggunakan *multilayer perceptron* (3 layers). Banyak orang percaya *deep neural network* lebih baik dibanding *neural network* yang lebar tapi sedikit *layer*, karena terjadi lebih banyak transformasi. Maksud lebih banyak *transformasi* adalah kemampuan untuk merubah *input* menjadi suatu representasi (tiap *hidden layer* dapat dianggap sebagai salah satu bentuk representasi *input*) dengan langkah *hierarchical*. Seperti contoh permasalahan XOR, permasalahan *non-linearly separable* pun dapat diselesaikan apabila kita dapat mentransformasi data (representasi data) ke dalam bentuk

*linearly separable* pada ruang yang berbeda. Keuntungan utama *deep learning* adalah mampu merubah data dari *non-linearly separable* menjadi *linearly separable* melalui serangkaian transformasi (*hidden layers*).

Ada beberapa strategi untuk mempercepat pembelajaran menggunakan *deep learning*, misalnya: regularisasi, *successive learning*, dan penggunaan *autoencoder*. Arti *successive learning* adalah jaringan yang dibangun secara bertahap. Misal kita latih ANN dengan 3 *layers*, kemudian kita lanjutkan 3 *layers* tersebut menjadi 4 *layers*, lalu kita latih lagi menjadi 5 *layers*, dst. Hal ini sesuai dengan (Elman 1993), yaitu mulai dari hal kecil. Ilustrasinya dapat dilihat pada Gambar 2.15. Menggunakan *deep learning* harus hati-hati karena pembelajaran cenderung *divergen* (artinya, minimum *square error* belum tentu semakin rendah seiring berjalannya waktu - *swing* relatif sering).



Gambar 2.15 Contoh *Successive Learning*

Sebelumnya yang terdapat pada latar belakang, sudah dijelaskan alasan memilih algoritma DNN daripada NN biasa. Pada Tabel 2.1 di bawah ini adalah rincian hasil pengujian yang dilakukan oleh (Mulia Siregar 2015).

Tabel 2.1 Hasil Pengujian Mulia Siregar (2015)

No.	Target	JST 8-2-2		JST 8-3-3-3-2	
		Output		Output	
1.	10	1.0094	-0.0078	1.0282	-0.041
2.	10	1.0082	-0.0282	0.9859	0.0069
3.	10	1.0008	-0.0286	0.9878	-0.0083
4.	01	-0.018	1.0185	0.236	0.8138
5.	01	0.0295	0.9695	-0.0196	1.0376
6.	11	0.9996	0.9997	0.9969	1.0234
7.	10	1.003	0.6305	1.0299	-0.0248
8.	11	0.9994	1.0258	0.9872	1.0294
9.	10	1.0094	-0.0358	0.9904	0.0012
10.	10	1.0091	0.0384	1.056	-0.0079
11.	10	1.0002	0.932	1.0688	-0.01
12.	11	0.9994	1.02	1.0154	0.997
13.	10	1.0097	-0.0337	1.0012	-0.0054
14.	10	1.0091	0.0363	1.0543	-0.012
15.	11	0.9992	1.0391	1.0195	0.9901
16.	11	1.008	0.1436	0.9602	0.9131
17.	11	1.0016	0.7987	0.9796	1.0493
18.	11	1.0045	0.5019	0.9478	1.0006
19.	11	1.0016	0.7987	0.9796	1.0493
20.	11	1.0029	0.6632	0.9588	1.0625
21.	11	1.0071	0.2373	0.9409	1.0363
<i>Performance</i>		17/21 = 80.95%		21/21 = 100%	

Dari Tabel 2.1 di atas dapat dilihat bahwa pada model 8-2-2 memiliki output yang salah/tidak sesuai dengan *target* sebanyak 4 item, dan pada model 8-3-3-3-2 memiliki hasil yang sesuai dengan *target* data uji secara keseluruhan.

Perbandingan setiap model dalam proses pengujian dapat dilihat pada Tabel 2.2.

Tabel 2.2 Perbandingan *Epoch* dan *MSE*

	8-2-2	8-5-5-2	8-3-3-3-2	8-5-5-5-2
<b>Epoch</b>	260359	7575	4898	11490
<b>MSE</b>	0.000999522	0.000999253	0.000996042	0.00099611

Konfigurasi JST yang digunakan pada penelitian tersebut adalah dengan menggunakan data sebanyak 35 data pelatihan dan 21 data pengujian dengan jumlah *input layer* JST 8 *neuron* dan jumlah *output layer* sebanyak 2 *neuron*.

Dengan pengujian digunakan untuk penilaian kinerja dosen berdasarkan penilaian mahasiswa dengan baik.

Dari Tabel 2.2 untuk JST dengan model 8-2-2 dengan nilai MSE pada saat pelatihan sebesar 0.000999522 dengan proses perulangan *epoch* pada saat pelatihan dengan nilai *epoch* = 260359. Untuk model 8-5-5-2 dengan nilai MSE pada saat pelatihan sebesar 0.000999253 dengan proses perulangan *epoch* pada saat pelatihan dengan nilai *epoch* = 7575. Untuk model 8-3-3-3-2 dengan nilai MSE pada saat pelatihan sebesar 0.000996042 dengan proses perulangan *epoch* pada saat pelatihan dengan nilai *epoch* = 4898. Sedangkan, untuk model 8-5-5-5-2 dengan nilai MSE pada saat pelatihan sebesar 0.00099611 dengan proses perulangan *epoch* pada saat pelatihan dengan nilai *epoch* = 11490. Dari data tersebut untuk model JST paling baik adalah dengan JST model 8-3-3-3-2 dengan nilai *epoch* sebanyak 4898 dan MSE pengujian sebesar 0.000988985. Pelatihan dan pengujian dengan model 8-3-3-2 ini dapat mengenali data pelatihan maupun data pengujian sampai 100% sesuai dengan *target* yang diinginkan.

#### 2.4 Penelitian Terkait

Penelitian ini mereferensi dari beberapa penelitian sebelumnya yang memiliki latar belakang yang berkaitan. Adapun penelitian yang berhubungan adalah penelitian dari (Mohsen 2018), pada penelitian tersebut menggunakan metode *deep learning neural networks classifier* yang merupakan salah satu *architecture* DL untuk dilakukan pengklasifikasian dataset yang berisi 66 otak MRI menjadi 4 (empat) klasifikasi, yaitu normal, *glioblastoma*, *sarcoma*, dan *metastatic bronchogenic carcinoma tumors*. *Classifier* ini dikombinasikan dengan menggunakan *Discrete Wavelet Transform* (DWT) yang merupakan alat ekstraksi

yang memiliki fitur *powerful* dan *principal components analysis* (PCA) dan evaluasi kinerja yang cukup baik dari semua ukuran kinerja. Dan hasil yang diperoleh dari penelitian tersebut adalah menghasilkan arsitektur metodologi yang mirip dengan *Convolutional Neural Networks* (CNN), yang membutuhkan spesifikasi *hardware* yang rendah dan membutuhkan waktu yang tepat untuk memproses gambar berukuran besar (256x256). Selain itu, dalam penggunaan DNN *classifier* menunjukkan akurasi yang cukup tinggi dibandingkan *classifiers* yang biasa. Dalam penelitian ini mengusulkan metodologi yang efisien yang menggabungkan *Discrete Wavelet Transform* (DWT) dengan *Deep Neural Network* (DNN) untuk melakukan klasifikasi.

Penelitian terkait lainnya dari (Joko 2013), membangun sistem informasi pendukung keputusan pemilihan kamera CCTV berdasarkan kondisi dan lokasi dengan menggunakan metode *Simple Addictive Weighting* (SAW). Sistem ini dibangun dengan vb.net 2005 sebagai bahasa pemrogramannya dan MySQL sebagai *database server*-nya. SAW merupakan metode pengambilan keputusan yang diterapkan dalam pembuatan sisten informasi tersebut dengan pembobotan yang sudah ditetapkan. Dengan menggunakan metode SAW data akan diolah dan akan menghasilkan sebuah nilai keputusan berupa *report* data kamera CCTV yang sesuai dengan yang di-*input*-kan oleh konsumen. Hasil yang diperoleh berupa hasil perhitungan dari masing-masing kamera. Dari masing-masing kamera tersebut akan menghasilkan nilai dari proses normalisasi. Dari hasil ini akan menampilkan skor yang dapat ditentukan, kamera yang memiliki skor tertinggi adalah kamera yang sesuai dengan kriteria. Namun, sistem informasi ini tidak dapat dijadikan acuan yang mutlak, disebabkan semua keputusan bergantung kepada konsumen.

Penelitian terkait yang lain adalah dari (Widodo 2016), yang membahas penggunaan CCTV untuk pengawasan terhadap suatu wilayah atau biasa disebut dengan *smart monitoring* dengan menggunakan metode MCDM, yaitu metode *Simple Additive Weighting* (SAW) sebagai sebuah perhitungan dan analisa yang tepat terhadap kondisi titik yang akan dipasang. Dan hasil yang diperoleh adalah penggunaan beberapa kriteria dalam proses perhitungan menggunakan metode SAW, kemudian kriteria-kriteria tersebut diproses menggunakan preferensi bobot yang diperoleh dari pakar. Dan dari 40 titik pengamatan diperoleh 8 titik pengamatan utama yang direkomendasikan. Dan untuk pengoptimalan dari hasil penelitian yang dilakukan, terdapat beberapa saran dari penulis yaitu penggunaan metode SAW, AHP, dan WP.

Selanjutnya penelitian terkait lainnya dari (Gupta 2012), pada penelitian ini menganalisa dampak dari berbagai macam parameter pemasangan kamera CCTV pada gambar lalu lintas untuk mencakup panjang jalan yang optimal untuk kontrol sinyal lalu lintas secara *real*. Parameter pemasangan CCTV meliputi tinggi, orientasi vertikal dan horizontal, frekuensi, resolusi, dan titik fokus dari gambar yang diperoleh. Hasil yang diperoleh adalah suatu teknik untuk memutuskan parameter pemasangan kamera CCTV untuk lalu lintas secara *real time*. Analisis *viewshed* digunakan untuk menentukan bagian apa dari persimpangan yang akan terlibat. Dengan berbagai hasil dan sifat kamera CCTV, itu bisa ditentukan dengan seberapa efisien lalu lintas dapat dilakukan pemantauan. Dengan meningkatnya area cakupan, maka jumlah titik fokus berkurang tetapi faktor skala meningkat.

Penelitian terkait lainnya dari (Kim 2019), dalam penelitian ini bertujuan untuk melakukan eksploitasi berbagai macam arsitektur CNN dalam basis *Neural*

*Convolutional Learning Classifier Systems* (CN-LCS), menggabungkan CNN dan LCS untuk eksplorasi kemungkinan CN-LCS. Membandingkan tingkat akurasi klasifikasi yang diperoleh dari CN-LCS dengan *Neural Based Learning Classifier Systems* (N-LCS). Hasil yang diperoleh menunjukkan bahwa arsitektur CN-LCS jauh lebih baik daripada N-LCS dengan menunjukkan kinerja yang tinggi pada tiga (3) set data.

Terakhir penelitian terkait lainnya dari (Pribadi, Kurniawan and dkk 2017), dalam penelitian ini bertujuan untuk menemukan hubungan antara tipe CCTV dengan lokasi penempatannya. Dengan menggunakan algoritma *Decision Tree* (DT), dan memperoleh akurasi sebesar 87.96%. Dalam penelitian ini, terdapat 3 faktor penting dalam pemilihan jenis CCTV, yaitu penempatan pada gang, jalan lurus, dan daerah rawan kriminal. Jumlah variabel dalam penelitian ini masih terbatas, jadi untuk penelitian selanjutnya akan lebih baik jika variabel yang diperoleh lebih banyak. Seperti, lebar jalan, jumlah kendaraan, kepadatan penduduk, dan yang lainnya.

## BAB 3

### METODOLOGI PENELITIAN

Pada bab ini akan dilakukan pembahasan mengenai beberapa hal, yaitu tahapan penelitian yang akan dilakukan, kebutuhan sistem yang akan dibuat dan penyelesaian masalah penentuan kamera CCTV berdasarkan tipe jalan dengan batasan masalah yang sudah disebutkan di atas, menggunakan algoritma *Deep Neural Networks* (DNN) yang merupakan pengembangan dari algoritma Jaringan Saraf Tiruan (JST) dengan pembelajaran menggunakan *backpropagation*.

#### 3.1 Prosedur Penelitian

Penelitian ini diawali dengan pengambilan dan pengumpulan data dari Lembaga terkait maupun studi pustaka dari beberapa literatur atau buku-buku terkait tentang apa yang akan diteliti nanti. Kemudian langkah selanjutnya adalah melakukan normalisasi data agar data yang akan dipakai nanti dapat digunakan pada saat diterapkan dalam metode. Kemudian data yang sudah dilakukan normalisasi tadi dipisah antara *training* data dan *test* data. Langkah selanjutnya adalah implementasi *training* data yang sudah dilakukan normalisasi tadi ke dalam algoritma DNN dengan menggunakan pembelajaran *backpropagation*. Langkah berikutnya adalah melakukan pengujian sistem dengan menggunakan *test* data yang sudah disiapkan sebelumnya. Setelah dilakukan pengujian sistem, diharapkan dapat diperoleh tingkat keakuratan pengujian menggunakan algoritma DNN dalam klasifikasi penentuan kamera CCTV berdasarkan tipe jalan. Untuk diagramnya dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Diagram Blok Prosedur Penelitian

## 3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem ini memiliki tujuan untuk mengidentifikasi kebutuhan sistem yang dibuat, meliputi perangkat lunak (*software*) serta perangkat keras (*hardware*). Berikut ini merupakan penjelasan dari perangkat yang akan digunakan.

### 3.2.1 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membangun sistem yang akan dibuat dan dikembangkan antara lain:

1. Sistem Operasi Windows 10
2. Editor *visual studio code*

3. Pengujian *jupyter notebook* (Python)
4. Diagram *draw.io*

### 3.2.2 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk membangun sistem yang akan dibuat dan dikembangkan antara lain:

- Laptop Asus X555DG
- Mouse Fantech X5s Zeus

### 3.3 Pengumpulan Data

Metode Pengumpulan data yang diperlukan untuk penelitian ini diperoleh dengan cara:

#### 1. Studi Literatur

Cara pengumpulan materi studi literatur sebagian dilakukan melalui internet dan sebagian melalui *text book*. Data yang diperoleh merupakan bahan materi yang berhubungan dengan permasalahan, perancangan dan implementasi sistem, di antaranya yaitu mengenai pengertian CCTV, penjelasan mengenai CCTV dan implementasi *backpropagation* pada algoritma DNN.

#### 2. Pengambilan Data

Data yang diperoleh meliputi data spesifikasi lengkap dari CCTV yang akan digunakan dan data tipe jalan yang nantinya akan menjadi *input*-an pada sistem. Data akan didapat langsung dari dinas terkait, dalam penelitian ini dinas yang menjadi acuan dari data yang akan dipakai yaitu Dinas Perhubungan Kota Malang.

### 3. Penentuan Parameter

#### a. *Focal Length*

Panjang fokus atau biasa disebut dengan *focal length* merupakan jarak dalam satuan milimeter (mm) dari lensa ke permukaan sensor gambar. Semakin pendek jarak, maka semakin luas pandangan yang dihasilkan. Begitupun sebaliknya, semakin panjang jarak *focal length* akan semakin sempit pandangan (*telephoto*). (CCTVMAN 2015)

#### b. IR Led

IR Led merupakan lampu infra merah yang digunakan oleh kamera CCTV untuk penerangan dalam menangkap gambar video, karena memiliki kemampuan menangkap gambar video dalam kondisi rendah bahkan pada kondisi tidak ada cahaya sama sekali. (Stealth 2013)

#### c. *Face Detection*

*Face detection* merupakan salah satu dari bagian kamera CCTV, tetapi tidak semua kamera CCTV memilikinya. *Face detection* merupakan bagian kamera CCTV untuk mengenali wajah dari gambar video yang ditangkap oleh kamera. (Nurhopipah 2018)

#### d. *Defog*

*Defog* merupakan salah satu fitur dari kamera CCTV yang dapat mengurangi efek kabut apabila lokasi yang dipantau sedang dilanda kabut tebal. (VisionTeknologi)

e. Tipe CCTV

Terdapat berbagai macam tipe CCTV, tetapi pada penelitian ini akan digunakan tipe *bullet*, *dome*, dan PTZ, disebabkan melihat dari kasus yang sudah ada di jalan raya saat ini.

f. *Zoom Ratio*

*Zoom ratio* merupakan bagian dari kamera CCTV yang memiliki kemampuan untuk memperbesar atau memperkecil ukuran gambar video terutama dengan memvariasikan panjang fokus lensa. (CCTVMAN 2015)

g. Lebar Jalan

Lebar jalan merupakan parameter yang digunakan sebagai salah satu parameter tipe jalan, dan digunakan untuk menentukan kamera CCTV yang cocok, karena lebar jalan berpengaruh pada spesifikasi kamera CCTV yang akan digunakan.

h. Lokasi Jalan

Lokasi jalan pada penelitian ini dibagi menjadi empat (4), yaitu jalan lurus, jalan pertigaan, jalan perempatan, dan jalan bundaran. Seperti pada parameter lebar jalan, lokasi jalan merupakan parameter yang digunakan sebagai salah satu parameter tipe jalan.

i. Tinggi CCTV

Tinggi CCTV yang dimaksud merupakan tinggi antara kamera CCTV dengan objek yang akan ditangkap. Dengan ketentuan yang sudah dijelaskan pada “Pasal 35 ayat 1 PM Perhubungan RI Nomor PM 49 2014”, yang mengatur tentang tinggi lampu lalu lintas 3 warna.

j. Jarak

Jarak yang dimaksud merupakan jarak kamera CCTV antara kamera CCTV dengan objek yang akan ditangkap. Dengan acuan kasus yang terjadi di jalan raya saat ini, secara umum jarak kamera CCTV dengan objek adalah 5-10 meter.

k. Hasil Layar

Hasil layar yang dimaksud merupakan detail gambar yang ditangkap oleh kamera CCTV dengan melihat dari tipe jalan, dengan dua (2) hasil yaitu bagus dan tidak bagus.

Dari penentuan parameter di atas, nilai yang dihasilkan merupakan ketentuan yang diperoleh dari pakar.

### 3.4 Normalisasi Data

Menormalisasi data dengan metode agar *range* nilainya antara nol dan satu. Data ternormalisasi ini nantinya akan digunakan sebagai *input* dalam metode yang akan digunakan. Metode yang akan digunakan adalah *min-max*. Teknik ini *rescale* data dari suatu *range* ke *range* yang baru. Data tersebut dijadikan skala antara *range* nol dan satu. Diberikan nilai yang bersesuaian (dalam satu kolom)  $\{sk\}$ ,  $k=1,2,..n$ . Maka nilai normalisasinya adalah (Kusumadewi 2004), disebutkan pada persamaan 3.1.

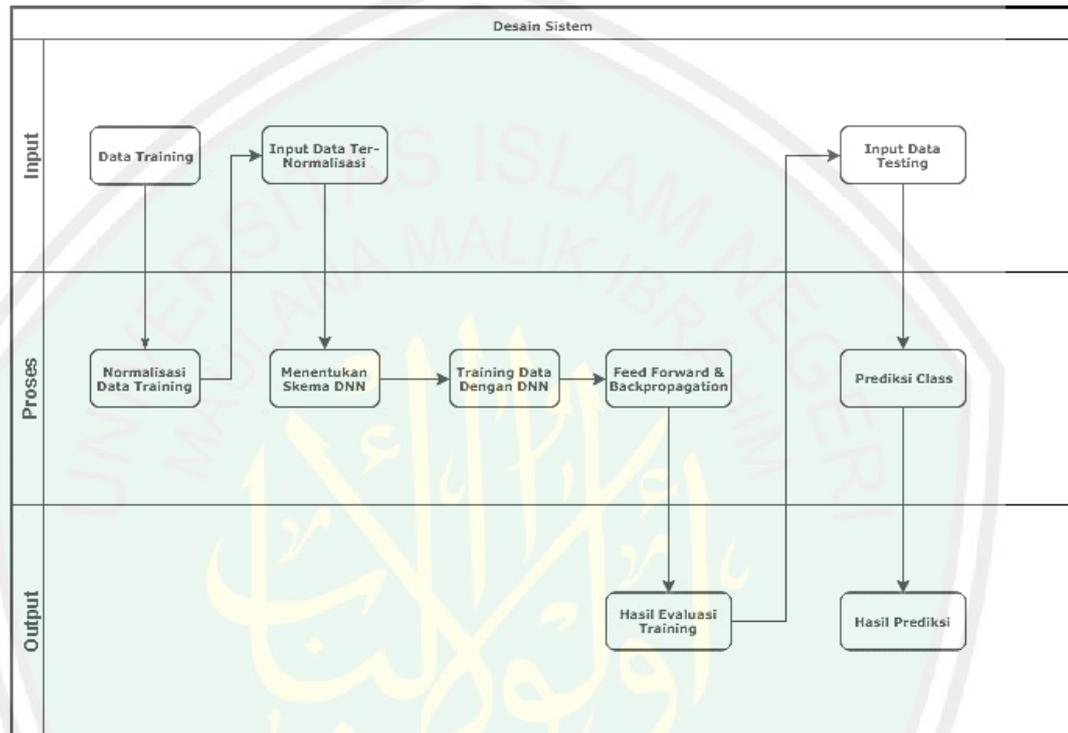
$$s' = \frac{s - \min\{sk\}}{\max\{sk\} - \min\{sk\}} \quad (3.1)$$

Alasan dari penggunaan normalisasi *min-max* disebabkan normalisasi tersebut ketika diterapkan untuk penghitungan dalam pembelajaran *backpropagation* memiliki tingkat rata-rata akurasi kurang lebih 95 persen,

daripada normalisasi yang lain dan hanya membutuhkan waktu yang relatif lebih singkat dibandingkan dengan normalisasi yang lain. (Chamidah and dkk 2012)

### 3.5 Perancangan Sistem

#### 3.5.1 Desain Sistem



Gambar 3.2 Diagram Desain Sistem

#### 1. *Input*

##### a. *Data Training*

Data *training* yang dimaksud di sini merupakan data yang sebelumnya sudah dipersiapkan terlebih dahulu sebagai data yang nantinya akan dilakukan pelatihan dengan algoritma DNN. Data ini terdiri dari data tabel dengan beberapa kolom dan baris berisi data CCTV maupun data Jalan. Berikut adalah contoh beberapa data yang nantinya akan digunakan untuk pelatihan yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Data *Training*

No.	<i>Focal Length</i>	<i>IR LED</i>	<i>Face Detection</i>	<i>Defog</i>	<i>Tipe CCTV</i>	<i>Zoom Ratio</i>	<i>Lebar Jalan</i>	<i>Lokasi Jalan</i>	<i>Tinggi CCTV</i>	<i>Jarak</i>	<i>Hasil Layar</i>	<i>Hasil</i>
1.	3.2mm	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Bullet</i>	3.1x	11	Lurus	3m	5m	Bagus	Cocok
2.	3.2mm	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Bullet</i>	3.1x	11	Lurus	3m	5m	Bagus	Cocok
3.	6mm	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Bullet</i>	No	11	Lurus	3m	5m	Bagus	Kurang Cocok
4.	4mm	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Bullet</i>	No	9	Lurus	3m	5m	Bagus	Cocok
5.	3mm	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Bullet</i>	No	14	Lurus	3m	5m	Bagus	Kurang Cocok
6.	5.2mm	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Bullet</i>	12x	14,5	Pertigaan	10m	10m	Bagus	Kurang Cocok
7.	4mm	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Bullet</i>	No	18	Pertigaan	10m	10m	Tidak Bagus	Sangat Tidak Cocok
8.	2.4mm	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Bullet</i>	No	25	Pertigaan	10m	10m	Tidak Bagus	Sangat Tidak Cocok
9.	2.8mm	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Bullet</i>	4.3x	19,5	Pertigaan	10m	10m	Tidak Bagus	Sangat Tidak Cocok

10.	2.8mm	Yes	No	No	Bullet	4.3x	17	Pertigaan	10m	10m	Bagus	Tidak Cocok
11.	3.6mm	Yes	No	No	Bullet	No	17,5	Perempatan	10m	10m	Tidak Bagus	Tidak Cocok
12.	2.8mm	Yes	No	No	Bullet	No	15	Perempatan	10m	10m	Tidak Bagus	Tidak Cocok
13.	2.8mm	Yes	No	No	Bullet	4.3x	15	Perempatan	10m	10m	Bagus	Tidak Cocok
14.	3.6mm	Yes	No	No	Bullet	No	15	Perempatan	10m	10m	Tidak Bagus	Tidak Cocok
15.	3.8mm	Yes	Yes	Yes	Bullet	No	15	Perempatan	10m	10m	Tidak Bagus	Tidak Cocok
16.	3mm	Yes	Yes	Yes	Bullet	2.8x	28	Bundaran	10m	10m	Tidak Bagus	Sangat Tidak Cocok
17.	3.2mm	Yes	No	Yes	Dome	3.1x	14	Bundaran	10m	10m	Bagus	Kurang Cocok
18.	3.2mm	Yes	No	Yes	Dome	3.1x	9	Bundaran	10m	10m	Bagus	Kurang Cocok
19.	3.2mm	Yes	No	Yes	Dome	3.1x	10,5	Bundaran	10m	10m	Bagus	Kurang Cocok

20.	3.2mm	No	No	Yes	Dome	3.1x	12	Bundaran	10m	10m	Bagus	Kurang Cocok
21.	3.2mm	Yes	No	Yes	Dome	3.1x	14	Pertigaan	10m	10m	Bagus	Cocok
22.	3.2mm	No	No	Yes	Dome	3.1x	9,5	Perempatan	10m	10m	Bagus	Cocok
23.	3.2mm	Yes	No	Yes	Dome	3.1x	35	Bundaran	10m	10m	Tidak Bagus	Tidak Cocok
24.	3.2mm	Yes	No	No	Dome	3.1x	15	Lurus	3m	5m	Bagus	Tidak Cocok
25.	6mm	Yes	No	No	Dome	No	12	Pertigaan	10m	10m	Bagus	Cocok
26.	4mm	Yes	No	No	Dome	No	13	Perempatan	10m	10m	Tidak Bagus	Kurang Cocok
27.	3mm	Yes	No	No	Dome	No	13	Bundaran	10m	10m	Tidak Bagus	Tidak Cocok
28.	3.2mm	Yes	No	No	Dome	3.1x	14,5	Lurus	3m	5m	Bagus	Tidak Cocok
29.	6mm	Yes	No	No	Dome	No	11	Lurus	3m	5m	Bagus	Kurang Cocok

30.	4mm	Yes	No	No	Dome	No	8	Lurus	3m	5m	Bagus	Kurang Cocok
31.	3mm	Yes	No	No	Dome	No	7	Lurus	3m	5m	Bagus	Kurang Cocok
32.	5.2mm	No	Yes	Yes	Dome	12x	7	Lurus	3m	5m	Bagus	Kurang Cocok
33.	4mm	Yes	Yes	Yes	Dome	No	7	Lurus	3m	5m	Bagus	Kurang Cocok
34.	4mm	Yes	Yes	Yes	Dome	No	7	Lurus	3m	5m	Bagus	Kurang Cocok
35.	2.4mm	No	Yes	Yes	Dome	No	7	Bundaran	10m	10m	Tidak Bagus	Sangat Tidak Cocok
36.	2.4mm	No	Yes	Yes	Dome	No	7	Bundaran	10m	10m	Tidak Bagus	Sangat Tidak Cocok
37.	2.8mm	Yes	Yes	Yes	Dome	4.3x	9	Bundaran	10m	10m	Bagus	Kurang Cocok
38.	2.8mm	No	Yes	Yes	Dome	4.3x	9	Bundaran	10m	10m	Bagus	Kurang Cocok
39.	2.8mm	Yes	Yes	Yes	Dome	4.3x	9	Bundaran	10m	10m	Bagus	Kurang Cocok

40.	2.8mm	No	Yes	Yes	Dome	4.3x	9,5	Bundaran	10m	10m	Bagus	Kurang Cocok
41.	5.2mm	Yes	Yes	Yes	Dome	12x	18	Bundaran	10m	10m	Bagus	Tidak Cocok
42.	2.8mm	No	Yes	Yes	Dome	4.3x	9	Pertigaan	10m	10m	Bagus	Cocok
43.	2.8mm	Yes	Yes	Yes	Dome	4.3x	13	Pertigaan	10m	10m	Bagus	Cocok
44.	2.8mm	Yes	No	No	Dome	4.3x	9,5	Pertigaan	10m	10m	Bagus	Cocok
45.	2.8mm	Yes	No	No	Dome	4.3x	18	Pertigaan	10m	10m	Bagus	Kurang Cocok
46.	6mm	Yes	No	No	Dome	No	9	Pertigaan	10m	10m	Bagus	Cocok
47.	3.6mm	Yes	No	No	Dome	No	10	Pertigaan	10m	10m	Tidak Bagus	Kurang Cocok
48.	2.8mm	Yes	No	No	Dome	No	9	Perempatan	10m	10m	Tidak Bagus	Kurang Cocok
49.	6mm	Yes	No	No	Dome	No	11	Perempatan	10m	10m	Bagus	Cocok

50.	3.6mm	Yes	No	No	Dome	No	9	Perempatan	10m	10m	Tidak Bagus	Kurang Cocok
51.	2.8mm	Yes	No	No	Dome	No	10,5	Perempatan	10m	10m	Tidak Bagus	Kurang Cocok
52.	2.8mm	Yes	No	No	Dome	4.3x	8	Perempatan	10m	10m	Bagus	Cocok
53.	10mm	Yes	Yes	Yes	Dome	2.3x	23	Perempatan	10m	10m	Bagus	Kurang Cocok
54.	2.8mm	Yes	No	No	Dome	4.3x	10.5	Lurus	3m	5m	Bagus	Kurang Cocok
55.	3.6mm	Yes	No	No	Dome	No	10,5	Lurus	3m	5m	Bagus	Kurang Cocok
56.	3mm	No	Yes	Yes	Dome	2.8x	21.5	Lurus	3m	5m	Bagus	Tidak Cocok
57.	3mm	Yes	Yes	Yes	Dome	2.8x	8	Lurus	3m	5m	Bagus	Kurang Cocok
58.	3.8mm	Yes	Yes	Yes	Dome	No	8,5	Lurus	3m	5m	Bagus	Kurang Cocok
59.	3mm	No	Yes	Yes	Dome	2.8x	9	Lurus	3m	5m	Bagus	Kurang Cocok

60.	3mm	Yes	Yes	Yes	Dome	2.8x	15	Lurus	3m	5m	Bagus	Tidak Cocok
61.	4.4mm	No	Yes	Yes	PTZ	23x	11,5	Bundaran	10m	10m	Bagus	Cocok
62.	4.4mm	No	Yes	Yes	PTZ	23x	9	Bundaran	10m	10m	Bagus	Cocok
63.	4.4mm	No	Yes	Yes	PTZ	32x	7	Bundaran	10m	10m	Bagus	Cocok
64.	4.4mm	No	Yes	Yes	PTZ	32x	7	Bundaran	10m	10m	Bagus	Cocok
65.	4.4mm	No	Yes	Yes	PTZ	32x	9	Bundaran	10m	10m	Bagus	Cocok
66.	2.8mm	No	Yes	Yes	PTZ	4.3x	9	Perempatan	10m	10m	Bagus	Cocok
67.	5.2mm	No	Yes	Yes	PTZ	12x	9	Perempatan	10m	10m	Bagus	Cocok
68.	4.4mm	Yes	Yes	Yes	PTZ	23x	9	Perempatan	10m	10m	Bagus	Cocok
69.	4.4mm	No	Yes	Yes	PTZ	23x	9	Perempatan	10m	10m	Bagus	Cocok

70.	4.4mm	No	Yes	Yes	PTZ	23x	9	Perempatan	10m	10m	Bagus	Cocok
71.	4.4mm	Yes	Yes	Yes	PTZ	23x	19	Perempatan	10m	10m	Bagus	Kurang Cocok
72.	4.4mm	No	Yes	Yes	PTZ	32x	9	Pertigaan	10m	10m	Bagus	Kurang Cocok
73.	4.4mm	No	Yes	Yes	PTZ	32x	9	Pertigaan	10m	10m	Bagus	Kurang Cocok
74.	4.4mm	No	Yes	Yes	PTZ	32x	8	Pertigaan	10m	10m	Bagus	Kurang Cocok
75.	4.4mm	No	Yes	Yes	PTZ	32x	7	Pertigaan	10m	10m	Bagus	Kurang Cocok
76.	4.4mm	Yes	Yes	Yes	PTZ	32x	7	Pertigaan	10m	10m	Bagus	Kurang Cocok

### b. *Input Data Ter-Normalisasi*

*Input data ter-normalisasi* yang dimaksud di sini merupakan data *training* yang sebelumnya dilakukan normalisasi dengan menggunakan *min-max* yang sudah dijelaskan sebelumnya.

### c. *Input Data Testing*

*Input data testing* yang dimaksud di sini merupakan data *testing* yang akan digunakan untuk mengukur akurasi maupun prediksi kelas (klasifikasi).

## 2. Proses

### a. *Normalisasi Data Training*

Normalisasi data *training* yang dimaksud di sini merupakan proses dilakukannya normalisasi data *training* menggunakan rumus dari *min-max* yang sudah dijelaskan sebelumnya. Dengan ketentuan sebagai berikut untuk penggunaan normalisasi *min-max*. Untuk nilai *output* sendiri tidak dilakukan normalisasi, tetapi dilakukan *encode* nilai seperti pada aturan di bawah.

#### 1. *Build IR LED*

- a. *Yes* = 1
- b. *No* = -1

#### 2. *Face Detection*

- a. *Yes* = 1
- b. *No* = -1

#### 3. *Defog*

- a. *Yes* = 1
- b. *No* = -1

#### 4. *Tipe CCTV*

- a. *Bullet* = 1
- b. *Dome* = 2
- c. *PTZ* = 3

#### 5. *Zoom Ratio*

- a. *No* = -1

#### 6. *Lokasi Jalan*

- a. *Lurus* = 1
- b. *Pertigaan* = 2
- c. *Perempatan* = 3

d. Bundaran = 4

### 7. Hasil Layar

a. Bagus = 1

b. Tidak Bagus = -1

### 8. Hasil

a. Cocok (1, 0, 0, 0)

b. Kurang Cocok (0, 1, 0, 1)

c. Tidak Cocok (0, 0, 1, 0)

d. Sangat Tidak Cocok (0, 0, 0, 1)

Dengan aturan yang disebutkan di atas, maka dari data pada Tabel 3.1 akan diambil sepuluh (10) data awal untuk dilakukan normalisasi dan akan mendapatkan hasil yang ada pada Tabel 3.2.



Tabel 3.2 Data *Training* Sebelum Normalisasi

No.	<i>Focal Length</i>	<i>IR LED</i>	<i>Face Detection</i>	<i>Defog</i>	<i>Tipe CCTV</i>	<i>Zoom Ratio</i>	<i>Lebar Jalan</i>	<i>Lokasi Jalan</i>	<i>Tinggi CCTV</i>	<i>Jarak</i>	<i>Hasil Layar</i>	<i>Hasil</i>
1.	3.2	1	-1	1	1	3.1	11	1	3	5	1	0,0,0
2.	3.2	1	-1	-1	1	3.1	11	1	3	5	1	0,0,0
3.	6	1	-1	-1	1	-1	11	1	3	5	1	0,1,0,0
4.	4	1	-1	-1	1	-1	9	1	3	5	1	0,0,0,0
5.	3	1	-1	-1	1	-1	14	1	3	5	1	0,1,0,0
6.	5.2	1	1	1	1	12	14.5	2	10	10	1	0,1,0,0
7.	4	1	1	1	1	-1	18	2	10	10	-1	0,0,0,1
8.	2.4	1	1	1	1	-1	25	2	10	10	-1	0,0,0,1
9.	2.8	1	1	1	1	4.3	19.5	2	10	10	-1	0,0,0,1
10.	2.8	1	-1	-1	1	4.3	17	2	10	10	-1	0,0,1,0

Untuk melakukan normalisasi *min-max*, akan ditentukan terlebih dahulu nilai *min* maupun nilai *max* untuk tiap baris dan juga kolom.

Tabel 3.3 Nilai *Min-Max* Tiap Baris dan Kolom

Nilai <i>Min</i>	<i>Focal Length</i>	2.4	Nilai <i>Max</i>	<i>Focal Length</i>	6
	<i>IR LED</i>	1		<i>IR LED</i>	1
	<i>Face Detection</i>	-1		<i>Face Detection</i>	1
	<i>Defog</i>	-1		<i>Defog</i>	1
	Tipe CCTV	1		Tipe CCTV	1
	<i>Zoom Ratio</i>	-1		<i>Zoom Ratio</i>	12
	Lebar Jalan	9		Lebar Jalan	25
	Lokasi Jalan	1		Lokasi Jalan	2
	Tinggi CCTV	3		Tinggi CCTV	10
	Jarak	5		Jarak	10
	Hasil Layar	-1		Hasil Layar	1

Berdasarkan Tabel 3.3, maka selanjutnya untuk normalisasi akan menggunakan rumus:

$$\text{Rumus new data} = (\text{data} - \text{min}) / (\text{max} - \text{min}) \quad (3.2)$$

**Catatan :** Untuk Nilai *min-max* = 0 dan Nilai *min-max* = 1, maka tidak perlu dilakukan normalisasi *min-max*.

1. *Focal Length* (3.2)

$$\begin{aligned} \text{New data} &= (3.2 - 2.4) / (6 - 2.4) \\ &= 0.8 / 3.6 \\ &= 0.22 \end{aligned}$$

2. *Built-in IR LED* (1)

Disebabkan pada data di atas nilai *min* = 1, dan nilai *max* = 1, maka *new data* normalisasi *min-max* akan mendapatkan hasil = 1.

3. *Face Detection* (-1)

$$\begin{aligned} \text{New data} &= (-1 - (-1)) / (1 - (-1)) \\ &= 0 \end{aligned}$$

4. *Defog* (1)

$$\begin{aligned} \text{New data} &= (1 - (-1)) / (1 - (-1)) \\ &= 2/2 \\ &= 1 \end{aligned}$$

## 5. Tipe CCTV (1)

Disebabkan pada data di atas nilai *min* = 1, dan nilai *max* = 1, maka *new* data normalisasi *min-max* akan mendapatkan hasil = 1.

6. *Zoom Ratio* (3.1)

$$\begin{aligned} \text{New data} &= (3.1 - (-1)) / (12 - (-1)) \\ &= 4.1/13 \\ &= 0.32 \end{aligned}$$

## 7. Lebar Jalan (11)

$$\begin{aligned} \text{New data} &= (11 - 9) / (25 - 9) \\ &= 2/16 \\ &= 0.13 \end{aligned}$$

## 8. Lokasi (1)

$$\begin{aligned} \text{New data} &= (1-1)/(2-1) \\ &= 0 \end{aligned}$$

## 9. Tinggi CCTV (3)

$$\begin{aligned} \text{New data} &= (3-3)/(10-3) \\ &= 0 \end{aligned}$$

## 10. Jarak (5)

$$\begin{aligned} \text{New data} &= (5-5)/(10-5) \\ &= 0 \end{aligned}$$

## 11. Hasil Layar (1)

$$\begin{aligned} \text{New data} &= (1-(-1))/(1-(-1)) \\ &= 1 \end{aligned}$$

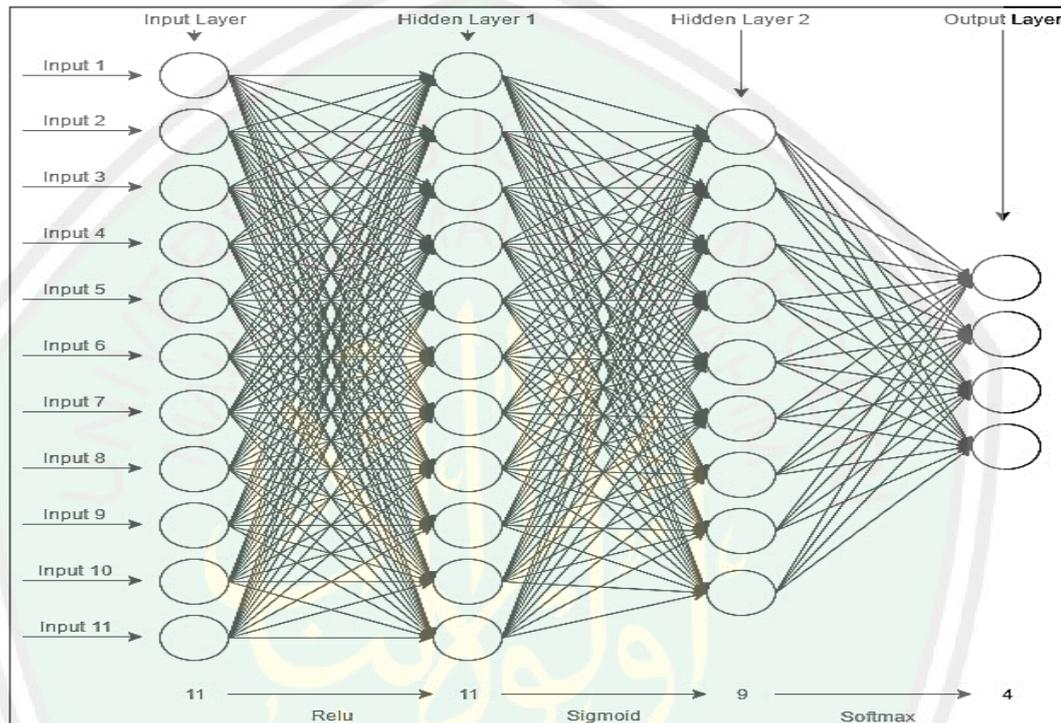
Dari perhitungan normalisasi data dengan mengambil satu data untuk dijadikan contoh bagaimana melakukan perhitungan normalisasi *min-max* secara manual, maka akan diperoleh dari Tabel 3.2, menjadi seperti pada Tabel 3.4.

Tabel 3.4 Data *Training* Setelah Normalisasi

No.	<i>Focal Length</i>	<i>IR LED</i>	<i>Face Detection</i>	<i>Defog</i>	<i>Type CCTV</i>	<i>Zoom Ratio</i>	<i>Lebar Jalan</i>	<i>Lokasi Jalan</i>	<i>Tinggi CCTV</i>	<i>Jarak</i>	<i>Hasil Layar</i>	<i>Hasil</i>
1.	0.22	1.00	0.00	1.00	1.00	0.32	0.13	0.00	0.00	0.00	1.00	1, 0, 0, 0
2.	0.22	1.00	0.00	0.00	1.00	0.32	0.13	0.00	0.00	0.00	1.00	1, 0, 0, 0
3.	1.00	1.00	0.00	0.00	1.00	0.00	0.13	0.00	0.00	0.00	1.00	0, 1, 0, 0
4.	0.44	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	1, 0, 0, 0
5.	0.17	1.00	0.00	0.00	1.00	0.00	0.31	0.00	0.00	0.00	1.00	0, 1, 0, 0
6.	0.78	1.00	1.00	1.00	1.00	1.00	0.34	1.00	1.00	1.00	1.00	0, 1, 0, 0
7.	0.44	1.00	1.00	1.00	1.00	0.00	0.56	1.00	1.00	1.00	0.00	0, 0, 0, 1
8.	0.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	0.00	0, 0, 0, 1
9.	0.11	1.00	1.00	1.00	1.00	0.41	0.66	1.00	1.00	1.00	0.00	0, 0, 0, 1
10.	0.11	1.00	1.00	0.00	1.00	0.41	0.50	1.00	1.00	1.00	1.00	0, 0, 1, 0

### b. Menentukan Skema DNN

Menentukan skema DNN yang dimaksud di sini merupakan proses dilakukannya penentuan skema DNN. Skema DNN yang di maksud termasuk penentuan *activation function*, *learning rate*, maupun *epoch*.



Gambar 3.3 Skema Jaringan

Dari skema jaringan pada Gambar 3.3 di atas dengan 11 *input layer*, 11 *hidden layer 1*, 9 *hidden layer 2*, dan 4 *output layer*, atau 11-11-9-4. Sedangkan untuk *activation function*, menggunakan *relu*, *sigmoid*, dan *softmax*, tapi untuk skema jaringan di atas dari *input layer* menuju *hidden layer 1*, menggunakan *relu*, *hidden layer 1* menuju *hidden layer 2* menggunakan *sigmoid*, dan *hidden layer 2* menuju *output layer* menggunakan *softmax*. Dan untuk *learning rate*, menggunakan nilai sebesar 0.01.

### c. *Training Data Dengan DNN*

*Training data* dengan DNN yang dimaksud di sini merupakan proses dilakukannya *training data* yang sebelumnya sudah dilakukan normalisasi maupun DNN itu sendiri sudah ditentukan skemanya.

Setiap neuron pada skema jaringan pada Gambar 3.3 saling berhubungan yang diberi tanda dengan tanda panah. Tiap koneksi memiliki bobot (*weights*) dengan nilai yang berbeda-beda

*Hidden layer* dan *output layer* memiliki tambahan “*input*” yang biasa disebut dengan *bias*. (Pada Gambar 3.3 tidak disebutkan). Nilai dari *bias* membantu dalam mengendalikan nilai, dimana fungsi aktivasi berada. (GeeksforGeeks 2018)

Berikut data *inputs*, *weights*, *bias*, dan *output*:

1. *Inputs* = [0.22, 1.00, 0.00, 1.00, 1.00, 0.32, 0.13, 0.00, 0.00, 0.00, 1.00]
2. *Weights Input ke Hidden 1*

*Weight* diperoleh dengan menggunakan rumus `RAND()` pada excell, dengan skala 0 sampai 1.

Tabel 3.5 Bobot *Input ke Hidden Layer 1*

w11	w12	w13	w14	w15	w16	w17	w18	w19	w20	w21
w21	w22	w23	w24	w25	w26	w27	w28	w29	w30	w31
w31	w32	w33	w34	w35	w36	w37	w38	w39	w40	w41
w41	w42	w43	w44	w45	w46	w47	w48	w49	w50	w51
w51	w52	w53	w54	w55	w56	w57	w58	w59	w60	w61
w61	w62	w63	w64	w65	w66	w67	w68	w69	w70	w71
w71	w72	w73	w74	w75	w76	w77	w78	w79	w80	w81
w81	w82	w83	w84	w85	w86	w87	w88	w89	w90	w91
w91	w92	w93	w94	w95	w96	w97	w98	w99	w100	w101
w101	w102	w103	w104	w105	w106	w107	w108	w109	w110	w111
w111	w112	w113	w114	w115	w116	w117	w118	w119	w120	w121

Tabel 3.6 Nilai Bobot *Input* ke *Hidden Layer 1*

0.73	0.43	0.01	0.01	0.31	0.22	0.30	0.46	0.89	0.16	0.58
0.90	0.54	0.23	0.11	0.21	0.63	0.24	0.22	0.41	0.11	0.80
0.49	0.76	0.49	0.20	0.00	0.91	0.40	0.84	0.90	0.91	0.76
0.53	0.00	0.69	0.13	0.66	0.04	0.33	0.41	0.82	0.84	0.86
0.48	0.87	0.71	0.41	0.02	0.64	0.23	0.44	0.79	0.82	0.35
0.15	0.44	0.18	0.37	0.42	0.50	0.44	0.44	0.42	0.62	0.36
0.54	0.33	0.48	0.23	0.33	0.28	0.09	0.19	0.68	0.71	0.58
0.55	0.14	0.65	0.93	0.65	0.05	0.81	0.65	0.39	0.90	0.17
0.23	0.22	0.23	0.16	0.37	0.37	0.31	0.90	0.38	0.22	0.27
0.22	0.96	0.47	0.49	0.63	0.60	0.83	0.77	0.35	0.28	0.17
0.00	0.41	0.94	0.75	0.47	0.52	0.48	0.17	0.08	0.33	0.63

3. *Bias Hidden Layer 1*Tabel 3.7 *Bias Hidden Layer 1*

<i>Bias Hidden 1</i>	
bh1	0.87
bh2	0.96
bh3	0.54
bh4	0.74
bh5	0.30
bh6	0.13
bh7	0.02
bh8	0.79
bh9	0.66
bh10	0.14
bh11	0.24

4. *Weights Hidden Layer 1 ke Hidden Layer 2*Tabel 3.8 Bobot *Hidden Layer 1* ke *Hidden Layer 2*

wh11	wh12	wh13	wh14	wh15	wh16	wh17	wh18	wh19
wh21	wh22	wh23	wh24	wh25	wh26	wh27	wh28	wh29
wh31	wh32	wh33	wh34	wh35	wh36	wh37	wh38	wh39
wh41	wh42	wh43	wh44	wh45	wh46	wh47	wh48	wh49
wh51	wh52	wh53	wh54	wh55	wh56	wh57	wh58	wh59
wh61	wh62	wh63	wh64	wh65	wh66	wh67	wh68	wh69
wh71	wh72	wh73	wh74	wh75	wh76	wh77	wh78	wh79
wh81	wh82	wh83	wh84	wh85	wh86	wh87	wh88	wh89

wh91	wh92	wh93	wh94	wh95	wh96	wh97	wh98	wh99
wh101	wh102	wh103	wh104	wh105	wh106	wh107	wh108	wh109
wh111	wh112	wh113	wh114	wh115	wh116	wh117	wh118	wh119

Tabel 3.9 Nilai Bobot *Hidden Layer 1* ke *Hidden Layer 2*

0.72	0.59	0.13	0.37	0.55	0.06	0.25	0.01	0.42
0.96	0.92	0.65	0.67	0.02	0.59	0.74	0.07	0.17
0.06	0.60	0.70	0.73	0.85	0.77	0.06	0.89	0.35
0.20	0.69	0.21	0.13	0.13	0.54	0.38	0.58	0.12
0.14	0.20	0.42	0.92	0.98	0.57	0.12	0.75	0.69
0.70	0.49	0.99	0.27	0.78	0.10	0.86	0.37	0.86
0.33	0.02	0.80	0.23	0.65	0.87	0.83	0.07	0.18
0.84	0.05	0.37	0.11	0.71	0.84	0.41	0.73	0.58
0.16	0.64	0.49	0.17	0.06	0.58	0.69	0.06	0.39
0.86	0.03	0.42	0.77	0.58	0.64	0.38	0.23	0.51
0.38	0.31	0.80	0.05	0.67	1.00	0.66	0.13	0.79

5. *Bias Hidden Layer 2*Tabel 3.10 *Bias Hidden Layer 2*

bh21	0.58
bh22	0.31
bh23	0.28
bh24	0.99
bh25	0.57
bh26	0.97
bh27	0.39
bh28	0.01
bh29	0.68

6. *Weights Hidden Layer 2* ke *Output Layer*Tabel 3.11 Bobot dan Nilai Bobot *Hidden Layer 2* ke *Output*

wo11	wo12	wo13	wo14	0.51	0.62	0.01	0.92
wo21	wo22	wo23	wo24	0.97	0.68	0.23	0.02
wo31	wo32	wo33	wo34	0.43	0.28	0.71	0.69
wo41	wo42	wo43	wo44	0.82	0.36	0.82	0.13
wo51	wo52	wo53	wo54	0.95	0.09	0.56	0.14
wo61	wo62	wo63	wo64	0.14	0.70	0.50	0.84

wo71	wo72	wo73	wo74	0.36	0.35	0.55	0.10
wo81	wo82	wo83	wo84	0.40	0.24	0.49	0.56
wo91	wo92	wo93	wo94	0.71	0.43	0.07	0.12

### 7. Bias Output Layer

Tabel 3.12 Bias Output Layer

<i>Bias Output</i>	
BO1	0.39
BO2	0.35
BO3	0.72
BO4	0.76

#### d. Feed Forward & Backpropagation

*Training* data itu sendiri dilakukan dengan proses *feed forward* dan juga *backpropagation*. Hal ini akan dijelaskan lebih lanjut di bawah.

#### e. Prediksi Class

Prediksi *class* yang dimaksud di sini merupakan prediksi dari *input* data *testing* dibagi menjadi beberapa *class* dengan acuan data *training* yang sebelumnya sudah dilakukan pelatihan menggunakan algoritma.

### 3. Output

#### a. Hasil Evaluasi Training

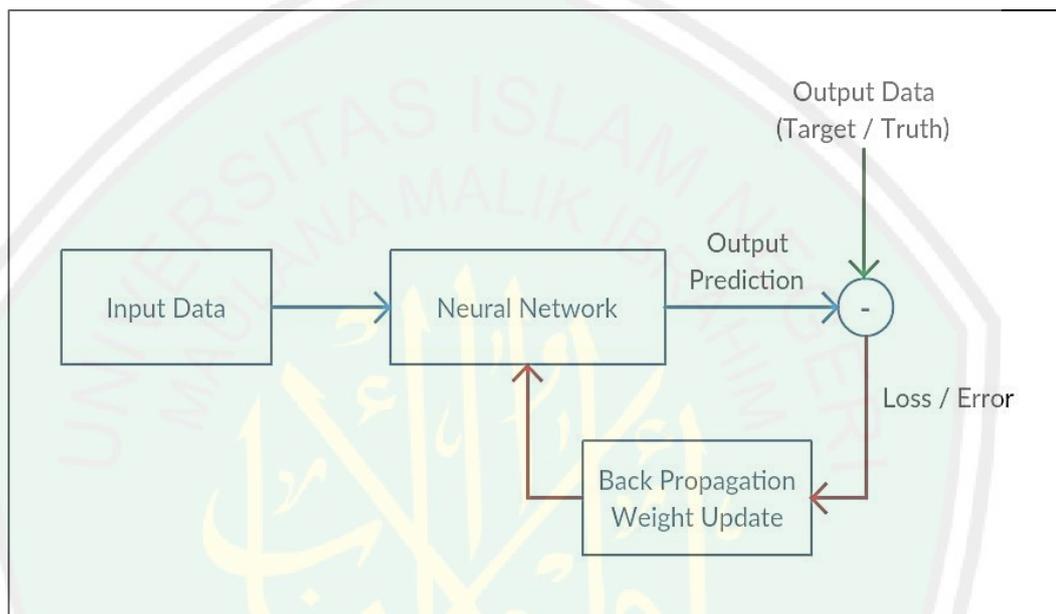
Hasil evaluasi *training* yang dimaksud di sini merupakan hasil dari *training* data dengan menunjukkan tingkat akurasi dari *training* data.

#### b. Hasil Prediksi

Hasil prediksi yang dimaksud di sini merupakan hasil dari proses prediksi *class* yang sudah membagi tiap data *testing* ke dalam *class*-nya masing-masing.

### 3.5.2 Alur Algoritma DNN

Berikut adalah alur metode *Deep Neural Network* dengan pembelajaran *backpropagation* yang akan dipakai dalam penelitian ini dengan panah biru menunjukkan *forward pass* dan panah merah adalah *backward pass*, seperti Gambar 3.4.



Gambar 3.4 Alur DNN Dengan Pembelajaran *Backpropagation*

### 3.5.3 Contoh Pembelajaran DNN Menggunakan *Backpropagation*

Pada bagian ini, hanya akan dibahas mengenai pelatihan data *training* dengan menggunakan *Deep Neural Network* (DNN) dan pembelajaran menggunakan *backpropagation*.

1. Inisialisasi Skema Jaringan

Skema jaringan yang akan digunakan seperti yang disebutkan di atas dengan skema 11-11-9-4.

2. Analisis Pelatihan

Tahap selanjutnya yang harus dikerjakan dalam pelatihan *backpropagation* adalah:

1. Inisialisasi bobot

Inisialisasi bobot dan bias dilakukan dengan menggunakan fungsi `rand()` pada excell, dengan skala 0 dan 1. Sedangkan untuk *input* sendiri berdasarkan data mentah dimana sebelumnya dilakukan normalisasi terlebih dahulu dengan normalisasi *min-max*. Pada tahap ini juga sudah dijelaskan pada desain sistem di atas.

m. Tahap perambatan maju (*Feed forward*)

Tiap-tiap unit tersembunyi menjumlahkan sinyal-sinyal *input* terbobot. Kemudian dari hasil penjumlahan tersebut untuk tiap *layer* akan digunakan fungsi aktivasi berbeda. Untuk *layer* 1 menggunakan fungsi aktivasi *relu*, *layer* 2 menggunakan fungsi aktivasi *sigmoid* dan *layer* 3 menggunakan fungsi aktivasi *softmax*. Hasil yang diperoleh adalah sebagai berikut:

1. Relu

$$Relu = \max(0, x) \quad (3.3)$$

2. Sigmoid

$$Sigmoid = 1/(1 + e^{-x}) \quad (3.4)$$

3. Softmax

$$Softmax = e^{I_{ina}} / (\sum e^{O_{ina}}) \quad (3.5)$$

Hasil yang diperoleh adalah :

1. Hasil *net input* + aktivasi dari *input layer* ke *hidden layer 1* adalah (Pada *input layer* mengambil satu baris data yang digunakan untuk pelatihan saat ini) :

Tabel 3.13 Data *Input* Pelatihan

<i>Input</i>	
i1	0.22
i2	1.00
i3	0.00
i4	1.00
i5	1.00
i6	0.32
i7	0.13
i8	0.00
i9	0.00
i10	0.00
i11	1.00

Dengan menggunakan rumus operasi matriks :

$$[i_1 i_2 i_3 \dots \dots i_{11}] \times \begin{bmatrix} w_{11} & \dots & w_{18} \\ \dots & \dots & \dots \\ w_{111} & \dots & w_{118} \end{bmatrix} + [bh_{11} \dots bh_{18}] = h_{1in1} \dots h_{1in8} \quad (3.6)$$

Maka akan menghasilkan *net input* :

Tabel 3.14 *Net Input* Hidden Layer 1

<i>Net Input</i>
3.06
3.05
3.23
2.29
1.91
2.21
1.50
2.30
3.17
2.56
3.20

Selanjutnya *net input* akan diubah menggunakan fungsi aktivasi *relu*:

Contoh perhitungan :

$$\begin{aligned} Relu &= \max(0,x) \\ &= \max(0, 3.06) \\ &= 3.06 \end{aligned}$$

Tabel 3.15 Hasil Aktivasi *Net Input Hidden Layer 1*

<i>Net Input</i>		$f(\text{Net Input})$		<i>Output Hidden Layer 1</i>		
3.06		$f(3.06)$		3.06	=	$h1_{out1}$
3.05		$f(3.05)$		3.05	=	$h1_{out2}$
3.23		$f(3.23)$		3.23	=	$h1_{out3}$
2.29	→	$f(2.29)$	=	2.29	=	$h1_{out4}$
1.91		$f(1.91)$		1.91	=	$h1_{out5}$
2.21		$f(2.21)$		2.21	=	$h1_{out6}$
1.50		$f(1.50)$		1.50	=	$h1_{out7}$
2.30		$f(2.30)$		2.30	=	$h1_{out8}$
3.17		$f(3.17)$		3.17	=	$h1_{out9}$
2.56		$f(2.56)$		2.56	=	$h1_{out10}$
3.20		$f(3.20)$		3.20	=	$h1_{out11}$

2. Hasil *net input* + aktivasi dari *hidden layer1* ke *hidden layer2* adalah

:

Dengan rumus operasi matriks :

$$[h1_{out1} \dots h1_{out8}] \times \begin{bmatrix} wh_{11} & \dots & wh_{14} \\ \dots & \dots & \dots \\ wh_{81} & \dots & wh_{84} \end{bmatrix} + [bh_{21} \dots bh_{24}] =$$

$$h2_{in1} \dots h2_{in4} \quad (3.7)$$

Dengan operasi matriks tersebut maka akan menghasilkan *net input* sebagai berikut :

Tabel 3.16 *Net Input Hidden Layer 2*

<i>Net Input at Hidden Layer 2</i>
------------------------------------

14.52
13.19
15.64
12.42
15.38
17.91
14.10
9.70
13.80

Selanjutnya *net input* akan diubah menggunakan fungsi aktivasi *sigmoid* :

Contoh perhitungan :

$$\begin{aligned}
 \text{Sigmoid} &= 1/(1 + e^{-x}) \\
 &= 1/(1 + e^{-14.52}) \\
 &= 0.999999507
 \end{aligned}$$

Tabel 3.17 Hasil Aktivasi *Net Input Hidden Layer 2*

<i>Net Input</i>		<i>f(Net Input)</i>		<i>Output Hidden Layer 2</i>	
14.52		$f(14.52)$		0.999999507	= $h_{2out1}$
13.19		$f(13.19)$		0.999998119	= $h_{2out2}$
15.64		$f(15.64)$		0.999999838	= $h_{2out3}$
12.42	→	$f(12.42)$	=	0.999995967	= $h_{2out4}$
15.38		$f(15.38)$		0.99999979	= $h_{2out5}$
17.91		$f(17.91)$		0.999999983	= $h_{2out6}$
14.10		$f(14.10)$		0.999999244	= $h_{2out7}$
9.70		$f(9.70)$		0.999938944	= $h_{2out8}$
13.80		$f(13.80)$		0.999998978	= $h_{2out9}$

3. Hasil *net input* + aktivasi dari *hidden layer 2* ke *output layer* adalah

:

Dengan rumus operasi matriks :

$$[h_{2_{out1}} \dots h_{2_{out4}}] x \begin{bmatrix} w_{011} & \dots & w_{012} \\ \dots & \dots & \dots \\ w_{041} & \dots & w_{042} \end{bmatrix} + [b_{01} \quad b_{02}] =$$

$$O_{in1} \quad O_{in2} \quad (3.8)$$

Dengan operasi matriks tersebut maka akan menghasilkan *net input* sebagai berikut :

Tabel 3.18 *Net Input Output Layer*

<i>Net Input at Output Layer</i>
5.68
4.09
4.66
4.25

Selanjutnya *net input* akan diubah menggunakan fungsi aktifasi

*softmax* :

Contoh perhitungan :

$$\begin{aligned} \text{Softmax} &= e^{\text{Ina}} / (\sum e^{\text{Oina}}) \\ &= e^{5.68} / (e^{5.68} + e^{4.09} + e^{4.66} + e^{4.25}) \\ &= 0.555205128 \end{aligned}$$

Tabel 3.19 Hasil Aktifasi *Net Input Output Layer*

<i>Net Input</i>		<i>f(Net Input)</i>		<i>Output</i>		
5.68	→	$f(5.68)$	=	0.56	=	$O_{out1}$
4.09		$f(4.09)$		0.11	=	$O_{out2}$
4.66		$f(4.66)$		0.20	=	$O_{out3}$
4.25		$f(4.25)$		0.13		$O_{out4}$

Cek *error* dengan menggunakan *cross-entropy*

$$\text{Error} = -\left(\frac{1}{n}\right) (\sum_{i=1}^3 (y_i x \log(O_{outi})) + ((1 - y_i) x \log((1 - O_{outi})))) \quad (3.9)$$

$$\begin{aligned} \text{Error} &= -\log(0.56) - \log(1-0.11) - \log(1-0.20) - \log(1-0.13) \\ &= 0.465870708 \end{aligned}$$

Menghitung matriks penurunan dari *cross entropy*

$$\frac{\partial E_n}{\partial O_{outn}} = \frac{\partial(-1 \times ((y1 \times \log(O_{outn})) + (1-y1) \times \log((1-O_{outn})))}{\partial O_{outn}} \quad (3.10)$$

$$\frac{\partial E_1}{\partial O_{out1}} = -1 * \left( \left( 1 * \left( \frac{1}{0.56} \right) + 0 * \left( \frac{1}{1-0.56} \right) \right) \right) = -1.80$$

$$\frac{\partial E_2}{\partial O_{out2}} = -1 * \left( \left( 1 * \left( \frac{1}{0.11} \right) + 0 * \left( \frac{1}{1-0.11} \right) \right) \right) = -8.87$$

$$\frac{\partial E_2}{\partial O_{out2}} = -1 * \left( \left( 1 * \left( \frac{1}{0.20} \right) + 0 * \left( \frac{1}{1-0.20} \right) \right) \right) = -5.03$$

$$\frac{\partial E_2}{\partial O_{out2}} = -1 * \left( \left( 1 * \left( \frac{1}{0.13} \right) + 0 * \left( \frac{1}{1-0.13} \right) \right) \right) = -7.51$$

Menghitung matriks penurunan fungsi aktivasi *softmax*

$$\frac{\partial O_{out1}}{\partial O_{in1}} = \frac{(e^l + e^k + e^j)e^x}{(e^x + e^l + e^k + e^j)^2} \quad (3.11)$$

$$\frac{\partial O_{out1}}{\partial O_{in1}} = 0.2222810$$

$$\frac{\partial O_{out2}}{\partial O_{in2}} = 0.1683552$$

$$\frac{\partial O_{out3}}{\partial O_{in3}} = 0.1790109$$

$$\frac{\partial O_{out4}}{\partial O_{in4}} = 0.1708499$$

n. Tahap perambatan balik (*Backpropagation*)

Berikut rumus turunan untuk fungsi aktivasi yang sudah digunakan tadi:

## 1. Relu

$$relu = (0, x) \quad (3.12)$$

$$\text{if } x > 0, \frac{\partial(relu)}{\partial x} = 1 \quad (3.13)$$

$$\text{Otherwise, } \frac{\partial(relu)}{\partial x} = 0 \quad (3.14)$$

## 2. Sigmoid

$$Sigmoid = 1/(1 + e^{-x}) \quad (3.15)$$

$$\frac{\partial\left(\frac{1}{1+e^{-x}}\right)}{\partial x} = \frac{1}{1+e^{-x}} x \left(1 - \frac{1}{1+e^{-x}}\right) \quad (3.16)$$

$$\frac{\partial Sigmoid}{\partial x} = Sigmoid x (1 - Sigmoid) \quad (3.17)$$

## 3. Softmax

$$Softmax = e^{xa} / (\sum_{a=1}^n e^{xa}) = e^{x1} / (e^{x1} + e^{x2} + e^{xn}) \quad (3.18)$$

$$\frac{\partial(Softmax)}{\partial x1} = (e^{x1} x (e^{x2} + e^{xn})) / (e^{x1} + e^{x2} + e^{xn})^2 \quad (3.19)$$

Kemudian untuk melakukan perubahan bobot pada *layer 2*, sebelumnya dengan menentukan turunan dari *cross entropy* dan turunan dari *softmax*.

Tabel 3.20 Derrivative Cross Entropy dan Softmax

Matriks Cross Entropy Derrivative		Matriks Softmax Derrivative	
-1.80	$dEn/dOoutn$	0.2222810	$dOoutn/dOinn$
-8.87		0.1683552	
-5.03		0.1790109	
-7.51		0.1708499	

Menghitung perubahan bobot dengan laju pemahaman = 0.01 pada *hidden layer 2* menuju *output*. Dengan melihat skema jaringan, untuk melakukan perubahan bobot pada *hidden layer 2* langkah pertama

adalah mencari nilai penurunan *hidden 2* ke *output* dan penurunan *error* pada *hidden layer 2*.

- *Derrivative hidden 2 ke output*

$$\frac{\partial O_{in1}}{\partial W_{o11}} = h2_{out1} \quad (3.20)$$

Tabel 3.21 *Derrivative Hidden 2 ke Output*

$h2_{out1}$	0.999999507
$h2_{out2}$	0.999998119
$h2_{out3}$	0.999999838
$h2_{out4}$	0.999995967
$h2_{out5}$	0.99999979
$h2_{out6}$	0.999999983
$h2_{out7}$	0.999999244
$h2_{out8}$	0.999938944
$h2_{out9}$	0.999998978

- *Derrivative error pada hidden layer 2*

$$\frac{\partial E_1}{\partial W_{o11}} = \frac{\partial E_1}{\partial O_{out1}} \times \frac{\partial O_{out1}}{\partial O_{in1}} \times \frac{\partial O_{in1}}{\partial W_{o11}} \quad (3.21)$$

Contoh perhitungan :

$$\begin{aligned} &= -1.80 * 0.2222810 * 0.999999507 \\ &= -0.400358046 \end{aligned}$$

Tabel 3.22 *Derrivative Error pada Hidden Layer 2*

$\frac{\partial E_1}{\partial W_{o11}}$	$\frac{\partial E_2}{\partial W_{o12}}$	$\frac{\partial E_3}{\partial W_{o13}}$	$\frac{\partial E_4}{\partial W_{o14}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{o21}}$	$\frac{\partial E_2}{\partial W_{o22}}$	$\frac{\partial E_3}{\partial W_{o23}}$	$\frac{\partial E_4}{\partial W_{o24}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{o31}}$	$\frac{\partial E_2}{\partial W_{o32}}$	$\frac{\partial E_3}{\partial W_{o33}}$	$\frac{\partial E_4}{\partial W_{o34}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{o41}}$	$\frac{\partial E_2}{\partial W_{o42}}$	$\frac{\partial E_3}{\partial W_{o43}}$	$\frac{\partial E_4}{\partial W_{o44}}$	-0.400	-1.493	-0.900	-1.284

$\frac{\partial E_1}{\partial W_{051}}$	$\frac{\partial E_2}{\partial W_{052}}$	$\frac{\partial E_3}{\partial W_{053}}$	$\frac{\partial E_4}{\partial W_{054}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{061}}$	$\frac{\partial E_2}{\partial W_{062}}$	$\frac{\partial E_3}{\partial W_{063}}$	$\frac{\partial E_4}{\partial W_{064}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{071}}$	$\frac{\partial E_2}{\partial W_{072}}$	$\frac{\partial E_3}{\partial W_{073}}$	$\frac{\partial E_4}{\partial W_{074}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{081}}$	$\frac{\partial E_2}{\partial W_{082}}$	$\frac{\partial E_3}{\partial W_{083}}$	$\frac{\partial E_4}{\partial W_{084}}$	-0.400	-1.493	-0.900	-1.284
$\frac{\partial E_1}{\partial W_{091}}$	$\frac{\partial E_2}{\partial W_{092}}$	$\frac{\partial E_3}{\partial W_{093}}$	$\frac{\partial E_4}{\partial W_{094}}$	-0.400	-1.493	-0.900	-1.284

Perubahan bobot dilakukan dengan rumus :

$$\dot{W}_o = \text{Bobot lama} - (\text{learning rate} \times \text{Bobot baru}) \quad (3.22)$$

Tabel 3.23 Bobot Baru *Hidden Layer 2* ke *Output*

0.51	0.62	0.01	0.92	=>	0.52	0.63	0.02	0.93
0.97	0.68	0.23	0.02	=>	0.97	0.69	0.24	0.03
0.43	0.28	0.71	0.69	=>	0.43	0.29	0.72	0.70
0.82	0.36	0.82	0.13	=>	0.83	0.37	0.83	0.14
0.95	0.09	0.56	0.14	=>	0.95	0.10	0.57	0.15
0.14	0.70	0.50	0.84	=>	0.14	0.72	0.51	0.85
0.36	0.35	0.55	0.10	=>	0.37	0.36	0.56	0.11
0.40	0.24	0.49	0.56	=>	0.40	0.25	0.49	0.58
0.71	0.43	0.07	0.12	=>	0.72	0.45	0.08	0.13

Dan itu akan dilakukan sampai ke perubahan bobot dari *hidden layer* 1 ke *input*, dan akan dilakukan secara terus-menerus sampai iterasi yang ditentukan di awal atau sampai *minimum error* pada waktu pelatihan tercapai.

Untuk pelatihan menggunakan data *testing* akan lebih dijelaskan pada bab selanjutnya.

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Langkah-langkah Uji Coba

Langkah-langkah yang akan dilakukan untuk melakukan pengujian klasifikasi CCTV berdasarkan tipe jalan menggunakan algoritma *Deep Neural Network* (DNN) dapat dijabarkan sebagai berikut:

- a. Proses *training*, pada tahap proses ini akan dilakukan pelatihan data yang sudah disiapkan sebelumnya menggunakan model algoritma DNN. Model ini yang sudah dilatih akan digunakan untuk mengukur seberapa bagus algoritma dalam melakukan klasifikasi.
- b. Proses *testing*, pada tahap proses ini merupakan tahap yang bisa dibilang cukup penting karena pada tahap kali ini, program yang sudah dibangun dan dilakukan *training* sebelumnya akan dilakukan pengujian. Data yang sudah disiapkan untuk dilakukannya pengujian menggunakan model yang sudah dilakukan pelatihan sebelumnya.
- c. Pengembangan parameter *learning*, pada tahap proses ini akan ditampilkannya perbandingan akurasi yang diperoleh dengan merubah beberapa parameter *learning* seperti *epoch* dan *activation function*.

#### 4.2 Hasil Uji Coba

Berikut merupakan hasil dari uji coba yang telah dilakukan pada klasifikasi CCTV berdasarkan tipe jalan menggunakan algoritma DNN.

#### 4.2.1 Hasil Proses *Training*

Hal yang menjadi bagian terpenting dari berhasilnya proses penentuan CCTV berdasarkan tipe jalan adalah bagaimana mendapatkan hasil yang optimal dalam melakukan proses *training*. Dengan mendapatkan hasil yang maksimal dalam proses ini akan memberikan dampak yang sangat berpengaruh terhadap hasil yang diperoleh pada proses *testing* nantinya.

Salah satu hal yang membuat proses ini kemudian susah untuk mendapatkan hasil yang optimal, adalah dalam menentukan skema jaringan yang akan digunakan dalam proses *training*. Oleh sebab itu, dalam proses kali ini akan dilakukan dengan banyak bentuk dari skema jaringan itu sendiri, agar mendapatkan hasil yang optimal. Data yang akan digunakan untuk proses *training* adalah sebesar 100% dari total data yang dimiliki.

Proses *training* akan dilakukan dengan iterasi sebanyak 1000 *epoch*, yang artinya pada proses kali ini akan dilakukan perulangan sebanyak 1000 kali. Selanjutnya untuk nilai *learning rate* sendiri akan digunakan *default* dari *optimizer sgd* dengan nilai 0.01. Nilai ini akan digunakan dalam melakukan perubahan bobot setiap kali algoritma melakukan proses *backward-pass*.

Langkah yang pertama dalam melakukan proses *training* tentunya adalah membaca data yang akan dimasukkan dalam proses ini.

```
df = pd.read_csv('data/data.csv')
df.isnull().values.any()
df
```

Gambar 4.1 Proses *Training*

*Code* tersebut digunakan untuk membaca *file* berbentuk *.csv*, dimana *file* tersebut berisi data yang akan digunakan dalam proses *training*. Baris kedua,

merupakan perintah untuk melakukan pengecekan apakah di data yang akan dilakukan proses *training* memiliki data dengan nilai sama dengan *null*, baris ketiga akan menampilkan total dari jumlah data *training*, dan hasilnya pada data yang akan dilakukan *training* tidak memiliki data bernilai *null*.

	Focal Length	IR LED	Face Detection	Defog	Type CCTV	Zoom Ratio	Lebar Jalan	Lokasi Jalan	Tinggi CCTV	Jarak	Hasil Layar	Hasil
0	3.2	1	-1	1	1	3.1	11.0	1	3	5	1	1
1	3.2	1	-1	-1	1	3.1	11.0	1	3	5	1	1
2	6.0	1	-1	-1	1	-1.0	11.0	1	3	5	1	2
3	4.0	1	-1	-1	1	-1.0	9.0	1	3	5	1	1
4	3.0	1	-1	-1	1	-1.0	14.0	1	3	5	1	2
...	...	...	...	...	...	...	...	...	...	...	...	...
71	4.4	-1	1	1	3	32.0	9.0	2	10	10	1	2
72	4.4	-1	1	1	3	32.0	9.0	2	10	10	1	2
73	4.4	-1	1	1	3	32.0	8.0	2	10	10	1	2
74	4.4	-1	1	1	3	32.0	7.0	2	10	10	1	2
75	4.4	1	1	1	3	32.0	7.0	2	10	10	1	2

76 rows × 12 columns

Gambar 4.2 Data Training

Selanjutnya akan dilakukan proses penentuan *output* data dengan beberapa *class*, sebagai berikut:

Tabel 4.1 Output Data

Class	Hasil
1	Cocok
2	Kurang Cocok
3	Tidak Cocok
4	Sangat Tidak Cocok

Untuk mengetahui berapa data yang dimiliki pada tiap *class* adalah dengan cara menulis *code*, sebagai berikut:

```
cocok = len(df.loc[df['Hasil'] == 1])
kurang_cocok = len(df.loc[df['Hasil'] == 2])
tidak_cocok = len(df.loc[df['Hasil'] == 3])
sangat_tidak_cocok = len(df.loc[df['Hasil'] == 4])

print("Cocok = {}".format(cocok))
print("Kurang Cocok = {}".format(kurang_cocok))
print("Tidak Cocok = {}".format(tidak_cocok))
```

```
print("Sangat Tidak Cocok = {}".format(sangat_tidak_cocok))
print("Total hasil di dataset = {}".format(len(df)))
```

Gambar 4.3 Penentuan Data pada Data *Training*

Hasil yang didapat dalam menjalankan perintah di atas, sebagai berikut:

Cocok = 22

Kurang Cocok = 35

Tidak Cocok = 13

Sangat Tidak Cocok = 6

Total hasil di dataset = 76

Selanjutnya adalah melakukan *excludes* kolom hasil pada *input*, memilih hanya kolom hasil pada *output*, merubah nilai pada kolom hasil dengan ketentuan, sebagai berikut:

Tabel 4.2 *Output* setelah melakukan *excludes*

Class	Hasil
1	1, 0, 0, 0
2	0, 1, 0, 0
3	0, 0, 1, 0
4	0, 0, 0, 1

Kemudian melakukan normalisasi *min-max* pada data yang akan dilakukan proses *training*. *Code* perintah yang akan digunakan adalah sebagai berikut:

```
# excludes kolom hasil
x = np.array(df.drop(['Hasil'], 1))

# hanya kolom hasil
y = np.array(df['Hasil'], dtype = 'float')

# merubah kolom hasil ke kolom bentuk vektor
y.shape = (len(y), 1)

# encode class values as integers
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
```

```

encoder = LabelEncoder()
encoder.fit(y)
encoded_y = encoder.transform(y)

# convert integers to dummy variables
dummy_y = np_utils.to_categorical(encoded_y)
# normalize data
min_max_scaler = preprocessing.MinMaxScaler()

x_f_train = min_max_scaler.fit_transform(x_train)

y_f_train = dummy_y

```

Gambar 4.4 Normalisasi *min-max*

Pada tahap selanjutnya adalah membuat model jaringan, dengan skema 11-11-9-4, dimana angka pertama merupakan *input layer*, angka kedua merupakan *hidden layer 1*, angka ketiga merupakan *hidden layer 2* dan angka terakhir merupakan *output layer*, dimana 4 merupakan jumlah dari *class* yang terdapat pada kolom hasil pada data. Alasan mengapa menggunakan skema jaringan tersebut adalah disebabkan dengan dilakukannya percobaan dengan sebanyak mungkin dan hasil yang terbaik adalah menggunakan skema jaringan tersebut. Kemudian setelah menentukan skema jaringan maka, proses *training* dimulai.

```

# define model
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(11, activation =
=tf.nn.relu))
model.add(tf.keras.layers.Dense(9, activation =
tf.nn.relu))
model.add(tf.keras.layers.Dense(4, activation =
tf.nn.softmax))

model.compile(optimizer='sgd',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.optimizer.get_config()

start_time = time.time()

hist = model.fit(x_f_train, y_f_train, epochs=1000)

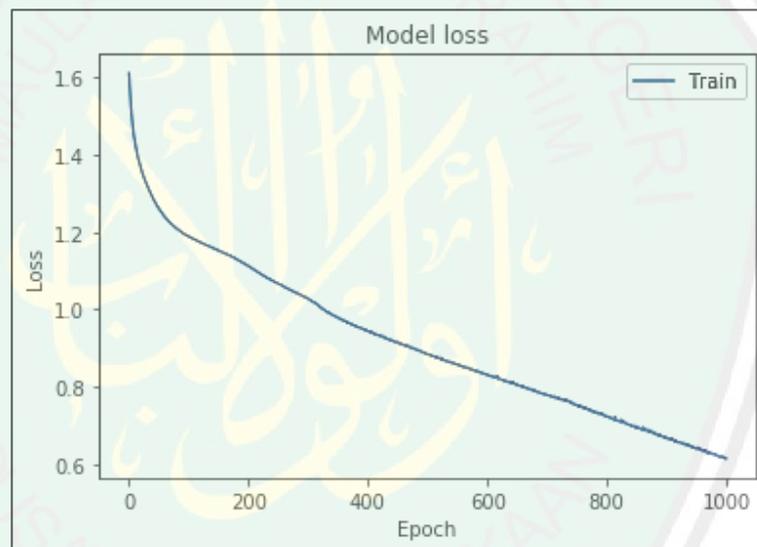
```

```
end_time = time.time()
print_time_taken(start_time, end_time)
```

Gambar 4.5 Pembuatan Model Jaringan

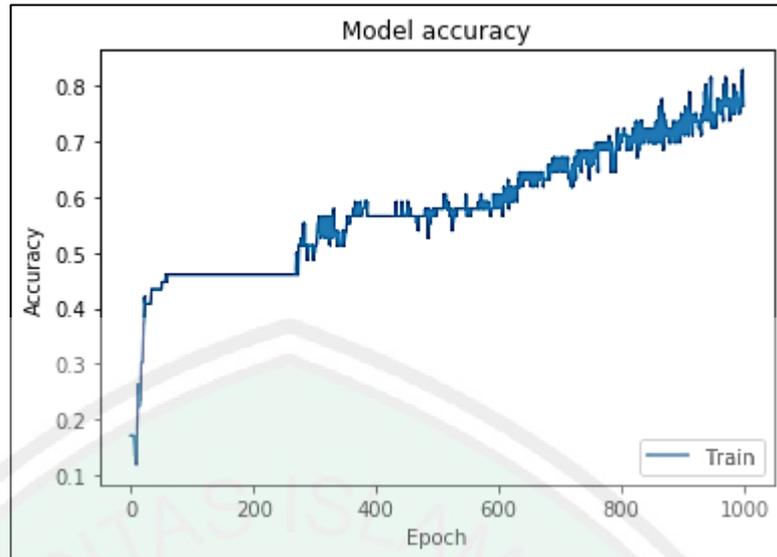
Dengan *code* di atas maka akan diperoleh hasil dari proses *training* yang akan ditunjukkan dengan perintah sebagai berikut:

```
plt.plot(hist.history['loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train'], loc='upper right')
plt.show()
```

Gambar 4.6 Code Menampilkan Hasil *Training* Model *Loss*Gambar 4.7 Hasil Proses *Training* Model *Loss*

```
plt.plot(hist.history['accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train'], loc='lower right')
plt.show()
```

Gambar 4.8 Code Menampilkan Hasil *Training* Model *Accuracy*



Gambar 4.9 Hasil Proses *Training Model Accuracy*

#### 4.2.2 Hasil Proses *Testing*

Pada tahap proses *testing* akan digunakan data uji sebanyak 17 baris dengan berbagai macam *class*. Proses untuk melakukan *testing* akan dijelaskan sebagai berikut:

Langkah awal adalah memanggil bobot yang sudah pernah dilakukan *training* sebelumnya. Kemudian membaca data *testing* yang berbentuk file *.csv*.

```
# load weights into new model
model.load_weights("weights new3 11 9 4 relu.h5")
print("Loaded model from disk")

# load data testing
df1 = pd.read_csv('data/test.csv')
df1.head()
df1.tail()
```

Gambar 4.10 *Load Weights dan Data Testing*

	Focal Length (mm)	Built-in IR LED	Face Detection	Defog	Comparison	Zoom Ratio (x)	Lebar Jalan	Lokasi	Tinggi CCTV (m)	Jarak (m)	Hasil Layar	Hasil
0	3.2	1	-1	1	2	3.1	14.0	2	10	10	1	1
1	3.2	-1	-1	1	2	3.1	9.5	3	10	10	1	1
2	3.2	1	-1	1	2	3.1	35.0	4	10	10	-1	3
3	3.2	1	-1	-1	2	3.1	15.0	1	3	5	1	3
4	6.0	1	-1	-1	2	-1.0	12.0	2	10	10	1	1

	Focal Length (mm)	Built-in IR LED	Face Detection	Defog	Comparison	Zoom Ratio (x)	Lebar Jalan	Lokasi	Tinggi CCTV (m)	Jarak (m)	Hasil Layar	Hasil
12	4.0	1	1	1	2	-1.0	7.0	1	3	5	1	2
13	4.0	1	1	1	2	-1.0	7.0	1	3	5	1	2
14	2.4	-1	1	1	2	-1.0	7.0	4	10	10	-1	4
15	2.4	-1	1	1	2	-1.0	7.0	4	10	10	-1	4
16	2.8	1	1	1	2	4.3	9.0	4	10	10	1	2

Gambar 4.11 Data *Testing* 5 Baris Awal dan 5 Baris Akhir

Kemudian seperti pada proses *training*, data yang akan dilakukan *testing* akan dilakukan normalisasi terlebih dahulu dengan menggunakan normalisasi *min-max*.

```
# excludes kolom hasil
x1 = np.array(df1.drop(['Hasil'], 1))

# hanya kolom hasil
y1 = np.array(df1['Hasil'], dtype = 'float')

# merubah kolom hasil ke kolom bentuk vektor
y1.shape = (len(y1), 1)

# encode class values as integers
encoder = LabelEncoder()
encoder.fit(y1)
encoded_y1 = encoder.transform(y1)

# convert integers to dummy
dummy_y1 = np_utils.to_categorical(encoded_y1)

min_max_scaler = preprocessing.MinMaxScaler()

x_f_test1 = min_max_scaler.fit_transform(x1)
y_f_test1 = dummy_y1
```

Gambar 4.12 Normalisasi *min-max* data *testing*

Baris perintah *code* untuk melakukan *testing* adalah:

```
hasil1 = model.evaluate(x f test1, y f test1)
```

Gambar 4.13 Perintah *Testing*

Hasil yang diperoleh dari proses perhitungan akurasi di atas adalah sebesar “*accuracy: 88.24%*” dengan nilai *loss* sendiri sebesar “*loss: 0.5045*”, maka hasil yang diperoleh dalam proses perhitungan ini bisa diketahui bahwa lebih dari 80% data yang diuji bernilai benar.

#### 4.2.3 Parameter *Epoch* dan *Learning Rate* Terhadap Akurasi

Salah satu hal yang dapat mempengaruhi tinggi rendahnya suatu akurasi dalam proses *learning* adalah nilai *epoch* dan juga *activation function*. Tujuan dari penentuan ini adalah digunakan untuk melakukan pembandingan pengaruh parameter yang disebutkan di atas pada algoritma *Deep Neural Network* (DNN).

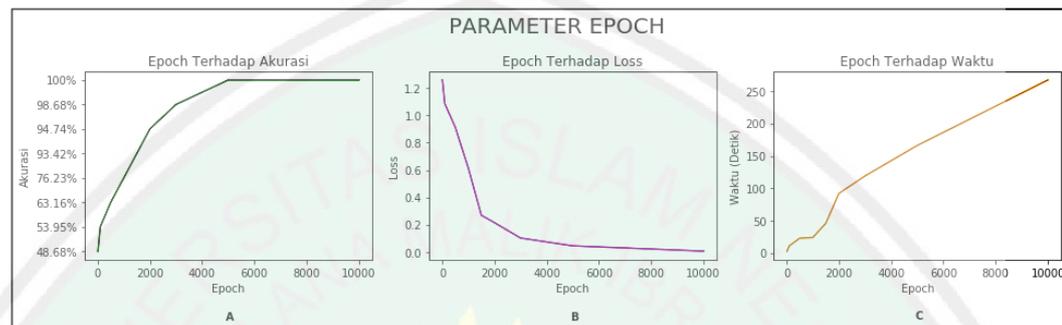
##### 1. Pengaruh Jumlah Epoch

Dalam melakukan proses *learning*, terdapat sebuah parameter yang dinamakan *epoch*, yang merupakan iterasi atau putaran perhitungan dari awal proses *training* pada *neural network* yang dihitung dari *input layer* kembali lagi ke *input layer* itu merupakan satu iterasi. Hingga sekarang belum terdapat penelitian yang menunjukkan atau meng-klain berapa jumlah *epoch* yang paling bagus digunakan dalam proses *training*. Namun jika melihat penelitian yang dilakukan oleh (Andre Lopes, 2017), mereka melakukan percobaan dengan variasi jumlah *epoch* sebanyak 1000 hingga 2000 iterasi. Dan pada penelitian saat ini akan dilakukan percobaan jumlah *epoch* sebanyak nilai *min epoch* = 10 dan nilai *max epoch* = 10000.

Tabel 4.3 *Epoch* Proses *Training*

<i>Epoch</i>	<i>Akurasi</i>	<i>Loss</i>	<i>Time</i>
--------------	----------------	-------------	-------------

10	48.68%	1.2581	3 detik
100	53.95%	1.0853	11 detik
500	63.16%	0.9136	23 detik
1000	76.23%	0.6152	24 detik
1500	93.42%	0.2695	46 detik
2000	94.74%	0.2155	1 menit 32 detik
3000	98.68%	0.1030	1 menit 59 detik
5000	100%	0.0458	2 menit 46 detik
10000	100%	0.0071	4 menit 27 detik



Gambar 4.14 Grafik Parameter *Epoch*

Berdasarkan Tabel 4.3 di atas dengan menggunakan *learning rate* 0.01 dan skema jaringan yang sama seperti disebutkan di atas, untuk jumlah *min epoch* 10 mendapatkan akurasi 48.68%, dengan nilai *loss* 1.2581, dan dengan waktu 3 detik. Sedangkan, pada jumlah *epoch* 500 ke atas terjadi peningkatan akurasi yang signifikan, untuk *epoch* 500 mendapatkan akurasi sebesar 63.16%, dengan nilai *loss* 0.9136, dan dengan waktu 23 detik. Sedangkan, pada jumlah *epoch* 3000 terjadi peningkatan akurasi dengan nilai akurasi hampir sama dengan 100%, untuk *epoch* 3000 mendapatkan nilai akurasi sebesar 98.68%, dengan nilai *loss* 0.1030, dan dengan waktu 1 menit 59 detik. Maka dapat diambil kesimpulan bahwa jumlah *epoch* dapat mempengaruhi dari proses *training* yang dilakukan, tetapi ketika jumlah *epoch* semakin besar akan mempengaruhi waktu dari proses *training* itu sendiri.

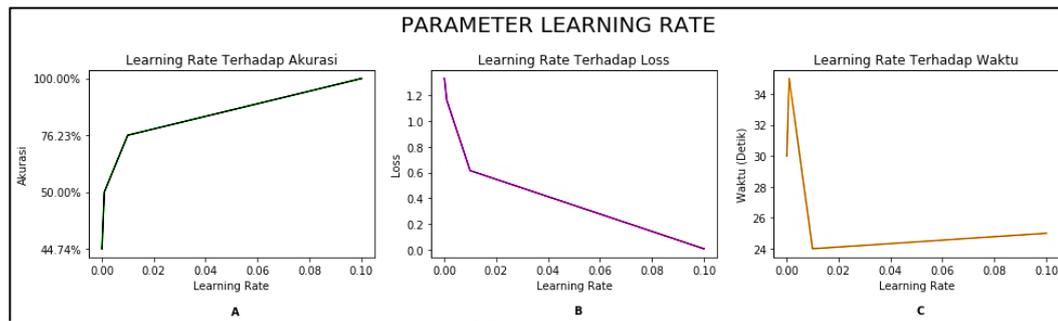
Berdasarkan Gambar 4.14 di atas yang merupakan hasil visualisasi dari Tabel 4.3 memperoleh tiga (3) grafik, yang pertama grafik untuk huruf “A” yaitu grafik yang menunjukkan pengaruh *epoch* terhadap akurasi, dimana untuk *epoch* mulai dari 5000 ke atas, hasil akurasi yang diperoleh menunjukkan nilai yang stabil, dengan nilai akurasi menuju 100%. Grafik kedua untuk huruf “B” yaitu grafik yang menunjukkan pengaruh *epoch* terhadap *loss*, dimana apabila jumlah *epoch* semakin besar, maka nilai untuk *loss* akan menuju nilai paling minimum dalam proses *training* yang dilakukan. Grafik ketiga untuk huruf “C” yaitu grafik yang menunjukkan pengaruh *epoch* terhadap *time* atau waktu, dimana apabila semakin besar jumlah *epoch*, maka untuk waktu yang digunakan dalam proses *training* sendiri akan semakin lama.

## 2. Pengaruh Learning Rate

Dalam melakukan proses *training* pastinya membutuhkan *learning rate*, yang dapat mempengaruhi hasil dari akurasi maupun *loss* itu sendiri. Peran dari *learning rate* adalah digunakan untuk melakukan update bobot pada saat proses *training* berlangsung. Berikut hasil uji dari *learning rate* terhadap akurasi dan *loss* dengan jumlah *epoch* sebanyak 1000 kali pada Tabel 4.4.

Tabel 4.4 *Learning Rate* Proses *Training*

<i>Learning Rate</i>	Akurasi	<i>Loss</i>	<i>Time</i>
0.1	100%	0.0068	25 detik
0.01	76.23%	0.6152	24 detik
0.001	50%	1.1631	35 detik
0.0001	44.74%	1.3318	30 detik



Gambar 4.15 Grafik Parameter *Learning Rate*

Jika dilihat dari Tabel 4.4 di atas, dengan jumlah *epoch* 1000 perubahan dari nilai *learning rate* juga cukup untuk dapat mempengaruhi nilai dari akurasi dan *loss*. Sedangkan, untuk waktu dengan jumlah *epoch* yang sama, perubahan yang terjadi tidak terlalu signifikan. Untuk nilai  $lr = 0.1$ , mendapatkan akurasi sebesar 100% dengan nilai *loss* sebesar 0.0068 dan jumlah waktu yang ditempuh 25 detik. Untuk nilai  $lr = 0.01$ , mendapatkan akurasi sebesar 76.23% dengan nilai *loss* sebesar 0.6152 dan jumlah waktu yang ditempuh 24 detik. Untuk nilai  $lr = 0.001$ , mendapatkan akurasi sebesar 50% dengan nilai *loss* sebesar 1.1631 dan jumlah waktu yang ditempuh 35 detik. Sedangkan, untuk nilai  $lr = 0.0001$ , mendapatkan akurasi sebesar 44.74% dengan nilai *loss* sebesar 1.3318 dan jumlah waktu yang ditempuh 30 detik.

Pada Gambar 4.15 yang merupakan hasil visualisasi dari Tabel 4.4 di atas, menunjukkan tiga (3) grafik pengaruh parameter *learning rate* dalam proses *training* dengan *epoch* sebanyak 1000. Grafik pertama dengan huruf “A”, menunjukkan hasil apabila *learning rate* semakin tinggi maka akurasi yang diperoleh akan semakin tinggi juga, begitu juga dengan sebaliknya apabila *learning rate* semakin rendah maka akurasi yang diperoleh juga semakin rendah. Grafik kedua dengan huruf “B”, menunjukkan hasil *learning rate*

terhadap *loss* yang sama seperti pada grafik sebelumnya pada huruf “A”, dimana apabila *learning rate* semakin tinggi maka akurasi yang diperoleh akan semakin tinggi juga, begitu juga dengan sebaliknya. Grafik ketiga dengan huruf “C”, menunjukkan hasil *learning rate* terhadap *time* atau waktu yang dari grafik memperoleh hasil bahwa pengaruh *learning rate* terhadap waktu menunjukkan hasil perubahan yang terjadi tidak terlalu signifikan.

### 4.3 Pembahasan

Algoritma *Deep Neural Network* (DNN) merupakan salah satu algoritma dari *deep learning* yang berbasis jaringan saraf yang dapat digunakan untuk melakukan pengambilan keputusan. Inti dalam perhitungan pada algoritma berbasis jaringan saraf adalah dengan mencari bobot terbaik dari contoh/sampel data yang sudah ada, kemudian nilai bobot terbaik akan dapat digunakan kembali untuk menghitung data lain yang tidak diketahui hasilnya.

Pada penelitian ini, algoritma DNN dapat melakukan klasifikasi penentuan CCTV berdasarkan tipe jalan dengan akurasi 88.24 untuk uji coba data *testing* yang telah dilakukan. Jika melihat dari penelitian-penelitian yang lain di luar sana dengan akurasi tersebut dapat dikatakan masih kurang optimal dan seharusnya dapat ditingkatkan lagi. Peneliti melakukan analisa sebab kurang optimalnya akurasi adalah disebabkan jumlah data yang terlalu sedikit dan kualitas data yang kurang baik.

Apabila jumlah dan kualitas data sudah baik, untuk mendapatkan akurasi yang optimal, dapat dilakukan dengan melakukan percobaan sebanyak mungkin dengan merubah skema jaringan, parameter baik itu *epoch*, *activation function*,

*learning rate*, *optimizer* dan lain sebagainya yang dirasa perlu untuk mendapatkan hasil akurasi optimal dalam *learning*.

Selain masalah di atas, masalah yang membuat hasil dari klasifikasi pada penelitian ini kurang maksimal adalah keterbatasan kemampuan perangkat yang digunakan dan juga keterbatasan ilmu yang dimiliki. Oleh sebab itu, hal itu membuat penelitian yang dilakukan tidak mampu melakukan eksperimen lebih untuk meningkatkan hasil yang diperoleh dari penelitian ini.

Melihat dari pembahasan di atas bisa dilihat bahwa pentingnya untuk belajar dan mempelajari sesuatu secara mendalam. Ketika algoritma tidak mampu mempelajari suatu objek secara lebih dalam, maka hasil yang diperoleh-pun akan berkurang. Hal ini tidak beda dengan manusia, ketika kita tidak mau untuk belajar dan mempelajari sesuatu maka *output* dari diri kita pasti kurang baik.

Proses untuk selalu belajar dan mempelajari sesuatu ini selaras dengan wahyu pertama yang Allah turunkan kepada nabi Muhammad SAW yaitu QS al-‘Alaq/96: 1-5

أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (١) خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ (٢) أَلْفَرَأْ وَرَبُّكَ الْأَكْرَمُ (٣) الَّذِي عَلَّمَ بِالْقَلَمِ  
(٤) عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ (٥)

Artinya : *Bacalah dengan (menyebut) nama Tuhan Yang menciptakan, Dia telah menciptakan manusia dari segumpal darah. Bacalah, dan Tuhanmulah Yang Maha Pemurah, Yang mengajar (manusia) dengan perantaraan kalam. Dan mengajarkan kepada manusia apa yang tidak diketahuinya.*

Ayat di atas, mengandung pesan bahwa Allah SWT mengajarkan pada manusia Al Qur'an dan mengajarkan padanya hikmah, yaitu ilmu. Allah

mengajarkannya dengan *qalam* (pena) yang bisa membuat ilmunya semakin lekat. Allah pun mengutus Rasul supaya bisa menjelaskan pada mereka. Kata Syaikh As Sa'di *rahimahullah*, “Manusia dikeluarkan dari perut ibunya ketika lahir tidak mengetahui apa-apa. Lalu Allah menjadikan baginya penglihatan dan pendengaran serta hati sebagai jalan untuk mendapatkan ilmu.” (Tafsir Al Karimir Rahman fii Tafsiril Kalamil Mannan 1423 H, 930)

Timbul pertanyaan, mengapa kata *iqra'* atau perintah membaca sederetan ayat di atas terulang dua kali yakni pada ayat 1 dan 3. Jika merujuk pada kitab tafsir Ibnu Katsir *rahimahullah*, beliau berkata “Seseorang itu akan semakin mulia dengan ilmu *diin* yang ia miliki. Ilmu itulah yang membedakan bapak manusia, yaitu Adam dengan para malaikat. Ilmu ini terkadang di pikiran. Ilmu juga kadang di lisan. Ilmu juga terkadang di dalam tulisan tangan untuk menyalurkan apa yang dalam pikiran, lisan, maupun yang tergambar di pikiran.” (Tafsir Al Qur'an Al 'Azhim 1431 H)

Bisa disimpulkan bahwa kemampuan manusia untuk belajar dan mempelajari inilah yang membuat manusia semakin mulia disisi Allah SWT. Selain itu Allah juga mengisyaratkan bahwa Allah menciptakan tubuh kita ini agar dimanfaatkan untuk belajar. Hal ini juga diisyaratkan Allah SWT pada QS An-Nahl (16): 78 berbicara tentang bagian pada diri manusia yang harus digunakan dalam kegiatan belajar dan pembelajaran:

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ وَالْأَفْئِدَةَ لَعَلَّكُمْ تَشْكُرُونَ

(٧٨)

Artinya: *Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatupun, dan Dia memberi kamu pendengaran, penglihatan dan hati, agar kamu bersyukur.*

Berdasarkan (Tafsir al-Jalalin) “(Dan Allah mengeluarkan kalian dari perut ibu kalian dalam keadaan tidak mengetahui sesuatu pun) jumlah kalimat *laa ta’lamuuna syaian* berkedudukan menjadi hal atau kalimat keterangan (dan Dia memberi kalian pendengaran) lafal *as-sam’u* bermakna jamak sekali pun lafalnya *mufrad* (penglihatan dan hati) kalbu (agar kalian bersyukur) kepada-Nya atas hal-hal tersebut, oleh karenanya kalian beriman kepada-Nya”. Semua itu merupakan rahmat dan anugerah Allah SWT kepada manusia yang tidak terhingga.

Menurut hadis nabi Muhammad SAW, dari Abu Hurairah r.a., ia berkata, Rasulullah bersabda, Sesungguhnya Allah SWT berfirman, “Siapa yang memusuhi kekasih-Ku, maka Aku menyatakan perang kepadanya. Dan tiada mendekat kepada-Ku seorang hamba-Ku dengan sesuatu yang lebih Aku sukai daripada menjalankan pekerjaan yang Aku wajibkan kepadanya. Hamba-Ku selalu mendekatkan dirinya kepada-Ku dengan menjalankan ibadah-ibadah sunah sehingga Aku menyukainya. Apabila Aku telah menyukainya, maka Aku menjadi pendengarannya yang ia pakai mendengar, penglihatannya yang ia pakai melihat, tangannya yang ia pakai memukul, dan kakinya yang ia pakai berjalan. Apabila ia memohon kepada-Ku, pasti akan Kukabulkan permohonannya, dan apabila ia minta perlindungan kepada-Ku, pasti Aku lindungi dia”. (Riwayat Bukhari, No. 6021)

Kemudian dilanjutkan lagi pada QS An-Nahl (16): 79 yang menunjukkan tanda kebijaksanaan Allah SWT, ilmu-Nya yang luas, dan perhatian-Nya kepada semua makhluk serta sempurnanya kekuasaan-Nya:

أَلَمْ يَرَوْا إِلَى الطَّيْرِ مُسَخَّرَاتٍ فِي جَوِّ السَّمَاءِ مَا يُمْسِكُهُنَّ إِلَّا اللَّهُ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّقَوْمٍ يُؤْمِنُونَ  
(٧٩)

*Artinya: Tidakkah mereka memperhatikan burung-burung yang dimudahkan terbang di angkasa bebas. Tidak ada yang menahannya selain daripada Allah. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kebesaran Rabb) bagi orang-orang yang beriman.*

Setelah Allah mengisyaratkan bahwa Allah telah memberi pendengaran, penglihatan dan hati dengan suatu tujuan tertentu. Allah memerintahkan kita untuk memperhatikan berbagai hal disekitar kita. Dicontohkan dalam ayat tersebut berdasarkan (Tafsir al-Jalalin) “(Tidakkah mereka memperhatikan burung-burung yang dimudahkan) terbang (di angkasa bebas) di udara antara langit dan bumi. (Tidak ada yang menahannya) sewaktu ia melipat sayap atau mengembangkannya sehingga ia tidak jatuh ke bawah (selain daripada Allah) yakni dengan kekuasaan-Nya. (Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda kekuasaan Allah bagi orang-orang yang beriman) yaitu penciptaan burung itu sehingga dapat terbang dan penciptaan udara sehingga dapat memungkinkan bagi burung untuk terbang mengarunginya dan menahan burung untuk tidak jatuh ke tanah” merupakan perintah untuk mengamati burung-burung yang terbang di angkasa. Secara eksplisit Allah memerintahkan kita untuk berfikir bahwa segala sesuatu yang terjadi disekitar kita, bisa terjadi karena kekuasaan Allah SWT.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil penelitian yang dilakukan untuk klasifikasi penentuan kamera CCTV berdasarkan tipe jalan menggunakan algoritma *Deep Neural Network* (DNN) ini, dapat ditarik kesimpulan yang dijelaskan sebagai berikut:

1. Klasifikasi dilakukan dengan menentukan terlebih dahulu variabel atau parameter yang akan digunakan untuk proses *training* dan proses *testing*. Parameter tersebut yang digunakan dalam penelitian ini adalah *focal length*, *ir led*, *face detection*, *defog*, tipe CCTV, *zoom ratio*, lebar jalan, lokasi jalan, tinggi CCTV, jarak, dan hasil layar, yang kesemua parameter yang disebutkan merupakan *input* dari data yang akan digunakan untuk proses *training* maupun proses *testing*. Kemudian, data parameter yang sudah diperoleh dilakukan normalisasi terlebih dahulu, pada penelitian ini menggunakan normalisasi *min-max*. Langkah selanjutnya adalah melakukan proses *training* dan proses *testing* dan pada penelitian ini menggunakan metode pembelajaran *backpropagation*, dengan skema jaringan yang dipakai adalah 11-11-9-4, dimana 11 awal merupakan *input* data, 11 kedua merupakan *hidden layer 1*, 9 merupakan *hidden layer 2*, dan 4 merupakan *output* data.
2. Hasil tingkat akurasi yang diperoleh dari proses *training* dengan jumlah data sebanyak 76 dan proses *testing* dengan jumlah data sebanyak 17 menggunakan algoritma DNN sudah cukup baik dalam melakukan

klasifikasi penentuan kamera CCTV berdasarkan tipe jalan. Dengan melakukan *training* pada 1000 *epoch* dan nilai *learning rate* = 0.01 diperoleh nilai akurasi dari *training* sebesar 76.23% dan akurasi dari *testing* sendiri adalah sebesar 88.24%. Hasil yang diperoleh sudah cukup baik melihat kualitas dan kuantitas data yang dimiliki tidak begitu bagus dan jumlah data yang bisa dikatakan kurang banyak.

3. Hasil dari klasifikasi penentuan kamera CCTV berdasarkan tipe jalan pada proses *testing* dari 17 data uji, yang menunjukkan data salah atau tidak sesuai adalah sebanyak 2 data uji, yaitu data *testing* dengan *actual output* cocok (1, 0, 0, 0) atau *class* 1, pada waktu prediksi salah satu nilai mendapatkan hasil prediksi kurang cocok (0, 1, 0, 0) atau *class* 2 dan data *testing* dengan *actual output* kurang cocok (0, 1, 0, 0) atau *class* 2, pada waktu prediksi salah satu nilai mendapatkan hasil prediksi tidak cocok (0, 0, 1, 0) atau *class* 3. Sedangkan, yang menunjukkan data benar atau sesuai sebanyak 15 data uji.

## 5.2 Saran

Untuk pengembangan dari penelitian dengan judul “Klasifikasi Penentuan Kamera CCTV Berdasarkan Tipe Jalan Menggunakan Algoritma *Deep Neural Network* (DNN)”, diperlukannya beberapa perbaikan untuk mencapai hasil yang lebih optimal, maka perlu dilakukan perbaikan untuk penelitian selanjutnya, diantaranya:

1. Memperbanyak jumlah data dan memperbaiki kualitas dari data, mungkin dengan dilakukan seleksi fitur atau perubahan fitur, agar algoritma mampu bekerja lebih optimal, atau mampu mendapatkan hasil yang lebih baik.

2. Melakukan perubahan pada parameter *learning* seperti merubah skema jaringan, *learning rate*, *epoch*, *activation function*, *optimizer*, dan yang lainnya yang memiliki pengaruh dalam melakukan proses *training* maupun proses *testing* pada algoritma *Deep Neural Network* (DNN).



## DAFTAR PUSTAKA

- Abubakar, I., Yani, A., Sutiono, E. 1995. *Menuju Lalu Lintas dan Angkutan Jalan yang Tertib*. Jakarta: Direktorat Jendral Perhubungan Darat.
- al-Asqalani, Ibnu Hajar. 1997. *Fathul Bari Syarah Shahih al-Bukhari*. Riyadh: Maktabah Darussalam.
- Andre Lopes, E. D. 2017. "Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order." *Science Direct* 610-628.
- Anonim. 1995. *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- . 1997. *Manual Kapasitas Jalan Indonesia*. Jakarta: Direktorat Jendra Bina Marga.
- as-Suyuthi, Jalaluddin al-Mahalli & Jalaluddin. n.d. *Tafsir al-Jalalin*.
- Atisusanti, D. 2009. "Evaluasi Kinerja Kapasitas Simpang Bersinyal (Studi Kasus Persimpangan Empat Palembang Gamping, Yogyakarta." *Universitas Atma Jaya Yogyakarta*.
- Atmoko, Eko Hari. 2005. *Membuat Sendiri CCTV berkelas Enterprise dengan Biaya Murah*. Yogyakarta: Andi.
- CCTVMAN. 2015. *CCTVMAN - Professional CCTV Service*. 11 13. Accessed 11 17, 2019. <http://cctvman.co.id/lensa-cctv/>.
- Chamidah, Nurul, and dkk. 2012. "Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropogasi Gradient Descent Adaptove Gain (BPGDAG) untuk Klarifikasi." *ITSMART* 2301-7201.
- Elman, Jeffrey L. 1993. "Learning and development in neural network: The impotance of starting small." *Journal of Cognition* 71-99.
- GeeksforGeeks. 2018. *GeeksforGeeks - A computer science portal for geeks*. 09 25. Accessed 11 17, 2019. <https://www.geeksforgeeks.org/effect-of-bias-in-neural-network/>.
- Gupta, Pratishta, dkk. 2012. "Computing Installation Parameters of CCTV Cameras for Traffic Surveillance." *International Journal of Computer Science & Information Technology (IJCSIT)* 105-124.
- Hariyanto, J. 2004. *Perencanaan Persimpangan Tidak Sebidang Pada Jalan Raya*. Medan: KMTS FT USU.
- Hendarto, Sri, dkk. 2001. *Dasar-dasar Transportasi*. Bandung: Institut Teknologi Bandung.
- Joko, Teguh. 2013. *Sistem Pendukung Keputusan Pemilihan Kamera CCTV Berdasarkan Kondisi dan Lokasi dengan Menggunakan Metode SAW*. Surabaya: Universitas Pembangunan Nasional "Veteran".

- Katsir, Ibnu. 1431 H. *Tafsir Al Qur'an Al 'Azhim*. Dar Ibnul Jauzi.
- Kim, Ji-Yoon, Cho, Sung-Bae. 2019. "Exploiting Deep Convolutional Neural Networks for a Neural-Based Learning Classifier System." *Department of Computer Science, Yonsei University, Seoul, Republic of Korea* 61-70.
- Kusumadewi, Sri. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab & Exel Link*. Yogyakarta: Graha Ilmu.
- Mohsen, Heba, dkk. 2018. "Classification Using Deep Learning Neural Network for Brain Tumors." *Future Computing and Informatic Journal* 68-71.
- Morlok, E.K. 1988. *Pengantar Teknik dan Perencanaan Transportasi*. Jakarta: Erlangga.
- Mulia Siregar, Victor Marudut. 2015. "Penggunaan Algoritma Nguyen Widrow Dalam Jaringan Syaraf Tiruan Backpropagation Untuk Penilaian Kinerja Dosen." *Proceedings Seminar Nasional Ilmu Komputer (SNIKOM)* 257-261.
- Nurhopipah, Ade & Harjoko, Agus. 2018. "Motion Detection and Face Recognition For CCTV Surveillance System." *Indonesian Journal of COmputing and Cybernetics Systems (IJCCS)* 107-118.
- Prasetyo, E. 2012. *Data Mining: Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: Andi Offset.
- Pribadi, Arif, Fachrul Kurniawan, and dkk. 2017. "Urban Distribution CCTV for Smart City Using Decision Tree Methods." *IEEE* 21-24.
- Sa'di, Syaikh 'Abdurrahman bin Nashir As. 1423 H. *Tafsir Al Karimir Rahman fii Tafsiril Kalamil Mannan*. Muassasah Ar Risalah.
- Stealth. 2013. *Stealth - Advanced CCTV Technology*. 05 02. Accessed 11 17, 2019. <http://www.stealth.co.id/kamera-cctv/analog-camera/dome-camera/infrared-series/>.
- VisionTeknologi. n.d. *Vision Teknologi*. Accessed 11 17, 2019. <http://visionteknologi.com/cctv-system/ip-cctv/>.
- Watanabe, Sumio, and Hidetoshi Nishimori. 2016. "Fall Lecture note on statistical learning theory." *Lecture notw for tokyo Institute of Teknologi*.
- Widodo, Gianto, dkk. 2016. "Analisa Penempatan Kamera CCTV Menggunakan Metode Simple Weighting (SAW) Untuk Smart Monitoring." *Jurnal Ilmu Komputer dan Teknologi Informasi* 44-47.

