

**IMPLEMENTASI *DEEP LEARNING* MENGGUNAKAN *CONVOLUTION*
NEURAL NETWORK UNTUK KLASIFIKASI GAMBAR**

(Studi Kasus: Gambar *Sport* (Bola Kaki, Bola Kok, dan Bola Basket))

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Program Studi Statistika



Boki Latupono

14 611 203

**JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2018**

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

Judul : Implementasi *Deep Learning* menggunakan *Convolution Neural Network* untuk Klasifikasi Gambar (Studi Kasus: Gambar *Sport* (Bola Kaki, Bola Kok, dan Bola Basket))

Nama Mahasiswa : Boki Latupono

Nomor Mahasiswa : 14 611 203

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DI SETUJUI UNTUK
DIUJIKAN**

Yogyakarta, 16 April 2018

Dosen Pembimbing


(Dr. R.B. Fajriya Hakim, S.Si., M.Si.)

HALAMAN PENGESAHAN

TUGAS AKHIR

IMPLEMENTASI *DEEP LEARNING* MENGGUNAKAN *CONVOLUTION NEURAL NETWORK* UNTUK KLASIFIKASI GAMBAR

(Studi Kasus: Gambar *Sport* (Bola Kaki, Bola Kok, dan Bola Basket))

Nama Mahasiswa : Boki Latupono

Nomor Mahasiswa : 14611203

TUGAS AKHIR INI TELAH DIUJIKAN

PADA TANGGAL 25 Mei 2018

Nama Penguji:

Tanda Tangan

1. Ir Ali Parkhan, M.T



2. Tuti Purwaningsih, S.Stat, M.Si



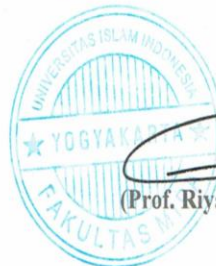
3. Dr. RB. Fajriya Hakim, M.Si



Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan

Alam



(Prof. Riyanto, S.Pd., M.Si, Ph.D)

KATA PENGANTAR



Assalamu'alaikum Warahmatullaahi Wabarakaatuh

Alhamdulillah *rabbi'l'aalamiin*, Puji Syukur senantiasa penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan nikmatnya yang tak terhingga, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Implementasi *Deep Learning* menggunakan *Convolution Neural Network* untuk Klasifikasi Gambar (Studi Kasus: Gambar Sport (Bola Kaki, Bola Kok dan Bola Basket)” sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang strata satu di Jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia. Shalawat serta salam semoga selalu tercurah kepada Nabi Muhammad SAW serta para sahabat dan pengikutnya yang senantiasa menjaga keimanan dan keislamannya hingga akhir hayatnya.

Penyelesaian tugas akhir ini tidak terlepas dari dukungan, bantuan, arahan, dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya yang sangat saya cintai, Bapak dan Ibu yang selalu memberikan semangat, do'a dan dukungan disetiap langkah saya.
2. Kakak-kakak saya, Nisa Latupono, Inya Latupono, Ricky Latupono, Rila Latupono dan Mato Salampessy, serta Keluarga Besar saya yang selalu mendo'akan, mendukung dan memberikan semangat kepada saya.
3. Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Bapak Dr. RB. Fajriya Hakim, M.Si. selaku Ketua Program Studi Statistika sekaligus sebagai pembimbing saya, yang telah banyak memberikan dukungan

dan masukan yang membangun serta selalu bersedia meluangkan waktunya untuk berkonsultasi dan memberikan arahan yang sangat inspiratif.

5. Seluruh staf pengajar Program Studi Statistika Universitas Islam Indonesia yang telah memberikan bekal ilmu kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini.
6. Sahabat sepermainan dan seperjuangan (Kontrakan Cantik): Nur Tuasikal, Khairunnisa Nur Aulia, Zia Ayu Nuansa Gemilang, dan Annisa Alwa'sia yang selalu menemani, mendukung, memberi semangat dan telah berjuang bersama-sama selama beberapa tahun ini. Semoga ukhuwah kita tetap terjaga dan selalu diridhoi Allah SWT.
7. Teman-teman bimbingan TA: Kia, Tiara, Dewi, Ridha, Furi, Asih, Mega, Salsabila, Hani, Febri, Rosita, Rizky, Hafidzan, Arfian, Sendy, dan Jimmy yang sudah sama-sama berjuang, saling mengingatkan dan memberi motivasi serta dorongan untuk menyelesaikan Tugas Akhir ini.
8. Keluarga "IPPMAP DIY-JATENG" yang selalu setia menemani, memberikan keceriaan dan hiburannya. Semoga kebersamaan kita tetap selalu terjaga.
9. Teman-teman KKN UII Unit 109 Kalitapas, Purworejo: FirmaN, Riki, Indra, Pungky, Yara, Devi, Dewi, dan Amel. Terimakasih atas pelajaran yang sangat berharga selama satu bulan bersama di lokasi KKN.
10. Teman-teman Statistika UII Angkatan 2014 yang bersama-sama menjadi pejuang gelar S.Stat dan Toga UII, terimakasih semangatnya.
11. Pihak-pihak lain yang mungkin penulis belum sebutkan, yang telah membantu dalam penyusunan tugas akhir ini.

Demikian Tugas Akhir ini, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan baik moril maupun materil sehingga tugas akhir ini dapat diselesaikan. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna dan masih banyak kekurangan. Hal tersebut dikarenakan keterbatasan ilmu dan pengetahuan yang dimiliki penulis semata. Oleh karena itu penulis mengharapkan kritik dan saran dari pembaca untuk menyempurnakan penulisan laporan ini. Semoga Tugas Akhir ini dapat memberikan manfaat bagi penulis khususnya dan umumnya bagi semua pihak yang membutuhkan. Akhir

kata, semoga Allah SWT senantiasa melimpahkan rahmat serta hidayah-Nya kepada kita semua, Amin amin ya robbal 'alamiin.

Wassalamu'alaikum Warahmatullaahi Wabarakaatuh

Yogyakarta, 16 April 2018



Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xi
PERNYATAAN	xii
INTISARI	xiii
ABSTRACT	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1. Penelitian terdahulu menggunakan metode CNN	6
2.2. Penelitian terdahulu menggunakan metode lain	8
BAB III LANDASAN TEORI	12
3.1. Olahraga	17
3.2. Citra Warna	13
3.3. Klasifikasi Citra	14
3.4. Jaringan Saraf Tiruan	14
3.4.1. Model Umum Jaringan Saraf Tiruan	15
3.4.2. Model Matematis Jaringan Saraf Tiruan	16

3.4.3. Karakteristik Jaringan Saraf Tiruan.....	17
3.4.4. Komponen Jaringan Saraf Tiruan.....	21
3.4.5. Arsitektur Jaringan Saraf Tiruan	23
3.5. <i>Machine Learning</i>	24
3.6. <i>Deep Learning</i>	26
3.7. <i>Convolution Neural Network</i>	29
BAB IV METODOLOGI PENELITIAN	49
4.1. Populasi dan Sampel.....	36
4.2. Variabel dan Definisi Operasional Variabel.....	36
4.3. Jenis dan Sumber Data	36
4.4. Metode Analisis Data	37
4.5. Tahapan Penelitian	38
BAB V ANALISIS DAN PEMBAHASAN	39
5.1. Pembuatan Dataset	39
5.1.1. <i>Fatkun Batch</i>	39
5.2. <i>Preprocessing CNN</i>	41
5.3. Pembuatan Model CNN.....	46
5.4. Evaluasi dan Prediksi Hasil <i>Train</i> dan <i>Test</i>	52
BAB VI PENUTUP	55
6.1. Kesimpulan.....	55
6.2. Saran	55
DAFTAR PUSTAKA	57
LAMPIRAN	61

DAFTAR TABEL

Tabel 2.1	Penelitian terdahulu dengan objek berbeda, metode sama ...	6
Tabel 2.2	Penelitian Terdahulu dengan objek dan metode berbeda	8
Tabel 4.1	Definisi Operasional Penelitian	36
Tabel 4.2	Diagram Alir Penelitian	38
Tabel 5.1	<i>Sintaks untuk mengaktifkan library keras dan EBImage</i>	41
Tabel 5.2	Mempersiapkan Gambar	42
Tabel 5.3	Menampilkan Dataset	43
Tabel 5.4	<i>Resize dan Combine</i>	44
Tabel 5.5	Proses Pelabelan data gambar	46
Tabel 5.6	<i>Create Model CNN</i>	47
Tabel 5.7	<i>Training Model</i>	50
Tabel 5.8	Hasil akurasi data <i>training</i> dan <i>testing</i>	50
Tabel 5.9	Hasil Klasifikasi data <i>Training</i>	52
Tabel 5.10	Hasil klasifikasi data <i>Testing</i>	52
Tabel 5.11	Hasil klasifikasi data baru	53

DAFTAR GAMBAR

Gambar 3.1	Struktur Jaringan Saraf Tiruan	15
Gambar 3.2	Model Matematis Jaringan Saraf Tiruan	16
Gambar 3.3	Jaringan Saraf Tiruan dengan lapisan tunggal.....	17
Gambar 3.4	Jaringan Saraf Tiruan dengan banyak lapisan	18
Gambar 3.5	Tampilan Struktur <i>Neural Network</i>	22
Gambar 3.6	<i>Neural Network</i> dengan tiga <i>Neuron Layers</i>	23
Gambar 3.7	Arsitektur Jaringan Saraf Tiruan	24
Gambar 3.8	Perbandingan Pemrograman dan <i>Machine Learning</i>	25
Gambar 3.9	Struktur pemodelan jaringan pada <i>Deep Learning</i>	27
Gambar 3.10	Layer-layer pada <i>Deep Learning</i>	27
Gambar 3.11	Sebuah Perceptron dengan d buah <i>input</i>	28
Gambar 3.12	Arsitektur <i>CNN</i>	30
Gambar 3.13	<i>Image RGB</i>	31
Gambar 3.14	Tampilan <i>Feature Map</i>	32
Gambar 3.15	<i>Max Pooling</i>	33
Gambar 3.16	Proses pada <i>fully connected layer</i>	35
Gambar 5.1	Tampilan <i>Google Image</i>	40
Gambar 5.2	Tampilan Lambang <i>Fatkun Batch</i>	40
Gambar 5.3	Tampilan halaman “ <i>Tab ini</i> ”	40
Gambar 5.4	Tampilan Halaman “Menyimpan Gambar”.....	40
Gambar 5.5	Tampilan data gambar yang sudah terkumpul.....	41
Gambar 5.6	Hasil <i>resize</i> dan <i>combine</i> untuk data <i>training</i>	44
Gambar 5.7	Hasil <i>resize</i> dan <i>combine</i> untuk data <i>testing</i>	45
Gambar 5.8	Model <i>CNN</i>	48
Gambar 5.9	<i>Plot Model</i>	51
Gambar 5.10	Tampilan nilai probabilitas untuk data baru.....	54

DAFTAR LAMPIRAN

Lampiran 1	<i>Script R</i> persiapan gambar	61
Lampiran 2	<i>Script R</i> pembuatan dan training model.....	64
Lampiran 3	<i>Script R Evaluate</i> dan <i>Prediction</i>	66
Lampiran 4	<i>Script R</i> Pembuatan data baru.....	67

PERNYATAAN

Dengan ini saya menyatakan bahwa Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang sepengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang diacu oleh naskah ini dan disebutkan dalam daftar pustaka.



IMPLEMENTASI *DEEP LEARNING* MENGGUNAKAN *CONVOLUTION NEURAL NETWORK* UNTUK KLASIFIKASI GAMBAR

(Studi Kasus: Gambar *Sport* (Bola Kaki, Bola Kok, dan Bola *Basket*)

Boki Latupono

Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Islam Indonesia

INTISARI

Olahraga merupakan kegiatan yang diinginkan oleh setiap individu manusia dan tidak bisa dipisahkan dalam kehidupan manusia. Dimana terdapat beberapa jenis-jenis cabang olahraga yang ada diantaranya atletik, badminton, bola basket, bola voli, dayung, sepakbola, tenis meja, tinju dll. Seiring berkembangnya teknologi terutama pada bidang kecerdasan buatan memiliki manfaat yang besar. Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas. Salah satu cabang dari AI yaitu *Machine Learning*, dimana mengajari mesin layaknya dapat berfikir seperti manusia. Salah satu pendekatan ML yaitu *deep learning*, dimana mampu melakukan pembelajaran mengenai banyaknya lapisan. *Deep learning* yang digunakan untuk pengenalan objek adalah *Convolution Neural Network* (CNN). Dengan melakukan klasifikasi objek diharapkan mampu meningkatkan kembali daya ingat untuk mengenal kembali berbagai macam cabang olahraga. Pada penelitian ini dilakukan pengenalan objek bola kaki, bola kok, dan bola *basket* menggunakan *software Rstudio* dan *package keras*. *Data sample* yang digunakan sebanyak 450 data gambar merupakan hasil *pengunduhan* dari *google image*. Untuk melakukan klasifikasi perlu dilakukan *training* data yang akan membentuk sebuah model. Model tersebut digunakan untuk klasifikasi gambar *train* dan *test* untuk 3 kategori. Akurasi yang dihasilkan model untuk data *train* adalah 0.98 atau 98% dan untuk data *test* sebesar 0.8133333 atau 81,3%. Kemudian dilakukan percobaan untuk 30 data baru, diperoleh tidak terdapat kesalahan dan masing-masing kategori berada pada tempatnya.

Kata Kunci: Klasifikasi Gambar, *Deep Learning*, *Convolutional Neural Network*, *Keras*, *Sport*

**IMPLEMENTATION DEEP LEARNING USING CONVOLUTION NEURAL
NETWORK FOR CLASSIFICATION OF IMAGES**

(Case Study: Sport figures (Football, Kok Ball, and Basket Ball))

Boki Latupono

*Department of Statitics, Faculty of Mathematics and Natural Sciences
Islamic University of Indonesia*

ABSTRACT

Sport is an activity desired by every human individual and can not be separated in human life. Where there are several types of sports that exist are athletics, badminton, basketball, volleyball, rowing, football, table tennis, boxing etc. Along with the development of technology, especially in the field of intelligence for having great benefits. Artificial intelligence (artificial intelligence) is a research area, applications and instructions related to computer programming to do something that in the human eye is intelligent. One of the branches of AI is Machine Learning, which teaches machines like to think like humans. One approach of ML is deep learning, which is able to do deeper learning. Deep learning used for object recognition is the Convolution Neural Network (CNN). By doing the object classification is expected to improve the recall to re-recognize the types of sports. In this research, the introduction of ball objects, football, and basketball using software Rstudio and hard package. The sample data used as many as 450 image data is the result of downloading from google image. To do the classification needs to be training data that will form a model. The model is used for image train and test classification for 3 categories. The accuracy of the model for data train is 0,98 or 98% and for test data of 0.8133333 or 81,3%. Then experiment for 30 new data, obtained no errors and each category is in place.

Keywords: *Classification image, Deep Learning, Convolutional Neural Network, Keras, Sport*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Olahraga merupakan kegiatan yang diinginkan oleh setiap individu manusia dan tidak bisa dipisahkan dalam kehidupan manusia. Dimana terdapat beberapa jenis-jenis cabang olahraga yang ada diantaranya atletik, badminton, bola basket, bola voli, dayung, sepakbola, tenis meja, tinju dll. Akan tetapi dari sekian banyaknya cabang olahraga tersebut, ada beberapa yang benar-benar memiliki popularitas tinggi dan sangat spektakuler di Indonesia. Popularitas sebuah cabang olahraga dinilai dari tingginya minat masyarakat serta banyaknya yang memainkan olahraga itu diantaranya sepakbola, badminton, dan bola basket. Indonesia ini sendiri dikenal sebagai masyarakat pencinta dan penggiat olahraga sejak era 60-an tahun. Saking tingginya minat masyarakat, baik itu untuk menonton atau melakukan kegiatan olahraga, disetiap daerah di Indonesia pasti setidaknya memiliki gelanggang olahraga yang aktif digunakan, lengkap dengan tim regional dari kota dan daerah masing-masing. Dunia olahraga saat ini dan bahkan yang akan datang, tidak dapat dipisahkan dengan kehidupan bermasyarakat, berbangsa, dan bernegara. Olahraga tidak hanya sebagai kebutuhan untuk menjaga kebugaran tubuh, akan tetapi telah merasuk dalam semua sektor kehidupan. Lebih jauh lagi, prestasi olahraga dapat mengangkat harkat dan martabat manusia baik secara individu, kelompok, masyarakat, bangsa, dan negara. Prestasi olahraga suatu negara menjadi tolok ukur kemajuan bangsa dan negara, oleh karena itu persaingan mencapai prestasi olahraga antar negara terus berjalan dengan berbagai pengembangan teknik dan teknologi bidang olahraga (olahragapendidikan.blogspot.com).

Dimana dimuat dalam Undang-undang Nomor 3 tahun 2005 tentang sistem keolahragaan nasional yang menyatakan bahwa olahraga bertujuan untuk meningkatkan dan memelihara kesehatan, kebugaran, prestasi, menanamkan nilai, moral, kualitas manusia, sportivitas, nilai moral dan akhlak mulia, disiplin, memperkuat ketahanan nasional, mempererat atau membina persatuan dan

kesatuan, serta mengangkat harkat, martabat, dan kehormatan bangsa. Adapun tujuan lain dari olahraga itu sendiri adalah bukan hanya merai piala atau medali, akan tetapi untuk membangun karakter dan mentalitas bangsa. Kemudian terkait dengan sistem olahraga nasional bahwa kondisi sekarang bisa dikatakan masih sangat minim, artinya tujuan yang diinginkan masih jauh dari yang diinginkan karena belum berjalan secara maksimal. Hal seperti ini masih terlihat dari minimnya prestasi Internasional yang berhasil diraih oleh para atletik Indonesia. Penurunan prestasi dan keikutsertakaan atlet dalam setiap penghelatan Internasional yang di alami Indonesia, salah satunya disebabkan karena kurang perhatian dari pihak pemerintah dari segi pembinaan. Sehingga menyebabkan minimnya para atlet-atlet berprestasi yang dapat mewakili perlombaan di kancah Internasional dengan berbagai cabang yang diikuti (wordpress.com).

Untuk itu model pembelajaran untuk studi kasus ini akan lebih baik dengan menerapkan konsep jaringan saraf tiruan dengan banyak lapisan, dan dengan model tersebut maka akan lebih baik lagi menggunakan performa komputasi yang cepat dimana akan menggunakan teknik *Deep Learning*. Menurut Mathworks (2017) *Deep learning* adalah cabang ilmu dari *machine learning* berbasis jaringan saraf tiruan, atau bisa dikatakan perkembangan dari jaringat saraf tiruan. Perbedaan dengan jaringan saraf tiruan sendiri adalah banyaknya *hidden layer* pada *deep learning* yang dimodelkan sedemikian rupa sehingga mampu memberikan *output* yang lebih akurat. *Deep learning* mengajari komputer melakukan sesuatu yang natural seperti manusia dan memiliki beberapa algoritma. Salah satu algoritma *deep learning* yang digunakan dalam penelitian ini adalah menggunakan *Convolution Neural Network*.

Convolution Neural Network merupakan salah satu metode yang digunakan untuk klasifikasi gambar, dimana pada metode ini terinspirasi oleh korteks mamalia visual sel sederhana dan kompleks. Model ini dapat mengurangi sejumlah parameter bebas dan dapat menangani deformasi gambar *input* seperti skala, rotasi, dan translasi. Berdasarkan penjelasan kelebihan *convolution neural network* tersebut, dapat diambil kesimpulan bahwa *convolution neural network* memiliki kemampuan klasifikasi yang diperuntukan untuk data gambar

(Mathworks, 2017). Dari permasalahan diatas, peneliti melakukan penelitian tentang “Implementasi *Deep Learning* menggunakan *Convolution Neural Network* untuk klasifikasi gambar (Studi kasus: gambar *Sport* (bola kaki, bola kok, dan bola basket)” yang bertujuan agar dengan berkembangnya waktu, masyarakat Indonesia bisa mengenal kembali berbagai macam cabang olahraga yang ada.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana mengimplementasikan konsep *Deep Learning* dengan menggunakan *Convolution Neural Network* untuk permasalahan klasifikasi gambar *Sport*?
2. Bagaimana melihat tingkat akurasi model *Convolution Neural Network* yang digunakan?
3. Bagaimana hasil klasifikasi dari data baru menggunakan model *Convolutional Neural Network*?

1.3. Batasan masalah

Adapun batasan masalah yang digunakan peneliti agar pembahasan dalam penelitian ini tidak menyimpang dari pokok pembahasan. Maka peneliti memiliki batasan masalah sebagai berikut:

1. Model *deep learning* yang diimplementasikan yaitu *Convolutional Neural Network* (CNN).
2. *Software* yang digunakan pada penelitian ini yaitu R dan *python*
3. Data yang digunakan dalam penelitian ini adalah gambar *sport* yang terdiri dari Bola Kaki, Bola Kok, dan Bola Basket.

1.4. Tujuan penelitian

Adapun tujuan penelitian yang hendak dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengimplementasikan *deep learning* dengan metode CNN pada klasifikasi *sport*.
2. Seberapa besar tingkat akurasi model CNN untuk klasifikasi gambar bola kaki, bola kok, dan bola basket.
3. Melatih komputer untuk mengklasifikasi bola kaki, bola kok, dan bola basket.
4. Mengenalkan kembali *sport* (bola kaki, bola kok, dan bola basket).

1.5. Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Mengetahui penerapan *deep learning* dengan metode CNN pada klasifikasi gambar.
2. Komputer mampu untuk mengklasifikasi bola kaki, bola kok, dan bola basket.
3. Mengingat kembali bentuk dari gambar *sport*.

1.6. Sistematis Penulisan

Adapun sistematika penulisan dari Tugas Akhir ini secara garis besar diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini akan dibahas tentang latar belakang masalah, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini memaparkan penelitian-penelitian terdahulu yang berhubungan dengan permasalahan yang diteliti dan menjadi acuan konseptual.

BAB III LANDASAN TEORI

Pada bab ini akan dibahas tentang teori-teori dan konsep yang berhubungan dengan penelitian yang dilakukan dan mendukung dalam pemecahan masalahnya. Selain itu, bab ini juga memuat teori-teori dalam pelaksanaan pengumpulan dan pengolahan data serta saat melakukan penganalisaan.

BAB IV METODE PENELITIAN

Pada bab ini memaparkan populasi dan sampel, variable penelitian, jenis dan sumber data, metode analisis data, dan tahapan penelitian.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai Analisa yang dilakukan terhadap hasil pengumpulan, pengolahan, dan analisa data yang diperoleh dari hasil penelitian.

BAB VI PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian dan analisa data yang telah dilakukan serta saran-saran yang dapat diterapkan dari hasil pengolahan data yang dapat menjadi masukan yang berguna kedepannya.

BAB II

TINJAUAN PUSTAKA

Penulis skripsi ini adalah suatu pemikiran baru yang menggunakan beberapa penelitian terdahulu sebagai acuan dan dasar penelitian. Hal ini bermanfaat untuk menunjukkan bahwa penelitian yang dilakukan mempunyai arti penting sehingga dapat diketahui kontribusi penelitian terhadap ilmu pengetahuan.

Penelitian mengenai klasifikasi gambar sebelumnya telah dilakukan oleh Laina Farsiah dkk pada tahun 2013. Penelitian ini membahas tentang algoritma penentuan kategori dari gambar berwarna menggunakan *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Pengujian dilakukan *color dataset* dan fitur terbaik dipilih menggunakan metode pemilihan fitur (*feature selection*). Hasil uji coba menunjukkan bahwa akurasi dari metode KNN sama baiknya dengan metode SVM.

Penelitian mengenai CNN sebelumnya telah dilakukan oleh Anna Gummeson pada tahun 2016. Tujuan utama dari penelitian ini adalah untuk mengotomatisasi klasifikasi dengan menggunakan *Convolutional Neural Networks* (CNN). Dengan diperkenalkannya Jaringan *Neural Convolutional* bidang pengenalan pola terputus. Cara klasik merancang fitur buatan tangan untuk klasifikasi sangat berbeda untuk membiarkan komputer menentukan sendiri fitur mana yang penting, pendekatan baru diaktifkan oleh CNN. Ini bersama dengan hasil terobosan pada patokan gambar telah membuat CNN metode yang digunakan dengan baik dalam pengenalan pola. Dalam penelitian ini, sebuah CNN dengan filter konvolusi kecil telah dilatih dari nol dengan menggunakan arus gradien stochastic dengan momentum. Tingkat kesalahan untuk CNN adalah 7,3%, yang secara signifikan lebih baik daripada pekerjaan sebelumnya yang menggunakan kumpulan data yang sama. Karena hasil yang baik diperoleh walaupun kumpulan data agak kecil, kesimpulannya adalah bahwa CNN adalah metode yang menjanjikan untuk masalah ini.

Penelitian mengenai CNN sebelumnya juga telah dilakukan oleh Daniel Gibert pada tahun 2016. Penelitian ini menyajikan dua pendekatan baru dan terukur dengan menggunakan *Convolutional Neural Networks* (CNN) untuk menetapkan perangkat lunak jahat ke keluarga yang sesuai. Di satu sisi, pendekatan pertama memanfaatkan CNN untuk mempelajari hierarki fitur untuk membedakan sampel *malware* yang digambarkan sebagai gambar skala abu-abu. Di sisi lain, pendekatan kedua menggunakan arsitektur CNN yang diperkenalkan oleh YoonKim untuk mengklasifikasikan sampel *malware* sesuai instruksi x86 mereka. Metode yang diusulkan mencapai perbaikan sebesar 93,86% dan 98,56% berkenaan dengan tolok ukur probabilitas yang sama.

Penelitian mengenai CNN sebelumnya juga telah dilakukan oleh Xavier Serra pada tahun 2017. Tujuan dari penelitian ini adalah untuk mengembangkan sistem *Face Recognition* yang lengkap untuk *GoldenSpear LLC*, sebuah perusahaan berbasis AI. Sistem yang dikembangkan menggunakan *Convolutional Neural Networks* untuk mengekstrak fitur wajah yang relevan. Fitur-fitur ini memungkinkan untuk membandingkan wajah di antara mereka dengan cara yang efisien. Sistem ini dapat dilatih untuk mengenali sekumpulan orang, dan belajar secara *on-line*, dengan mengintegrasikan orang-orang baru yang ia proses dan memperbaiki keterpaparan orang-orang di dalamnya. Keakuratan yang ada di sekitar 100 orang berhasil mencapai 95%, dan karena terlalu banyak dengan jumlah orang dalam sistem.

Penelitian mengenai CNN sebelumnya juga telah dilakukan oleh Zur Erlangung pada tahun 2007. Penelitian ini menyajikan metode berbasis CNN untuk deteksi fitur wajah otomatis. Sistem yang diusulkan menggunakan prosedur hirarkis yang pertama kali secara kasar melokalisasi mata, hidung dan mulut dan kemudian menemukan hasilnya dengan mendeteksi 10 titik fitur wajah yang berbeda. Tingkat deteksi metode ini adalah 96% untuk database AR dan 87% untuk database BioID yang menoleransi kesalahan 10% dari jarak antar-okular. Akhirnya, diusulkan pendekatan pengenalan wajah baru berdasarkan arsitektur CNN spesifik yang mempelajari pemetaan non-linear ruang gambar ke dalam ruang sub-dimensi yang lebih rendah dimana kelas yang berbeda lebih mudah

dipisahkan. Metode ini diterapkan ke beberapa database wajah publik dan memperoleh tingkat pengenalan yang lebih baik dibandingkan dengan pendekatan pengenalan wajah klasik berdasarkan PCA atau LDA. Selain itu, sistem yang diusulkan sangat kuat terhadap kebisingan dan oklusi parsial. Penelitian ini juga menyajikan metode berbasis CNN untuk masalah klasifikasi klarifikasi biner pengenalan gender dengan gambar wajah dan mencapai akurasi mutakhir. Hasil yang disajikan dalam penelitian ini menunjukkan bahwa CNN bekerja dengan sangat baik pada berbagai tugas pemrosesan gambar wajah, seperti pelurusan wajah, deteksi fitur wajah dan pengenalan wajah dan dengan jelas menunjukkan bahwa teknik CNN adalah pendekatan serbaguna, efisien dan kuat untuk analisis citra wajah.

Penelitian mengenai *crawling images* telah dilakukan oleh Purohit Shrinivasacharya dan M V Sudhamani pada tahun 2013. Penelitian ini menyajikan alat berbasis web yang mengumpulkan dan mengindeks sekelompok gambar web yang tersedia internet. Alat ini mengumpulkan kata kunci atau frase dari pengguna untuk mengambil gambar dari *web*. Kemudian kata kunci yang dikumpulkan ini diterapkan ke berbagai alat penelusuran umum seperti *Google, Yahoo, Bing* dll. Informasi halaman web yang dikumpulkan disimpan dalam file sementara sampai ukuran file 200KB dari server. Kemudian konten file ini akan dipindai dan ekstrak URL gambar dan itu dibandingkan dengan URL yang ada dalam database untuk menghindari duplikat unduhan. Gambar URL yang diekstrak diunduh dan akhirnya menyimpan gambar unik dan metadata yang sesuai seperti nama file, url, ukuran, dll di database. Akhirnya gambar yang dihasilkan ini digunakan dalam sistem Pengambilan Citra Berbasis Konten (CBIR) untuk mengekstrak gambar yang relevan yang dibutuhkan oleh klien dengan menggunakan konten gambar daripada informasi berbasis teks.

Penelitian mengenai klasifikasi telah dilakukan oleh Apriliya Fitri Cahyanti, Ristu Saptono, dan Sari Widya Sihwi pada tahun 2015. Penelitian ini menyajikan klasifikasi dalam studi kasus penilaian status gizi balita menggunakan metode *Naïve Bayes Classifier*, asumsi independensi antar parameter perlu diperhitungkan. Independensi antar parameter dilihat dari korelasi antar parameter

yang digunakan. Artikel ini membahas mengenai uji korelasi antar parameter dalam studi kasus penilaian status gizi menggunakan metode *Cosine Similarity*. Kemudian hasil uji korelasi tersebut dijadikan prosedur penentuan model dalam metode *Naive Bayes Classifier*. Sehingga dapat diketahui model yang paling baik dalam penilaian status gizi menggunakan metode *Naive Bayes Classifier*. Penentuan model terbaik dilihat dari akurasi, kesederhanaan, waktu, dan akuisisi data pada model. Pada skenario data 60%:40%, model terbaik ditunjukkan oleh model yang terdiri dari parameter berat, bmi, dan umur, dengan akurasi sebesar 94,4%. Sedangkan pada skenario data 80%:20% model terbaik ditunjukkan pada model yang terdiri dari parameter berat, bmi, tinggi, umur, dan jenis kelamin, dengan akurasi 94,8%. Penelitian ini menunjukkan bahwa korelasi parameter mempengaruhi hasil klasifikasi. Penggunaan parameter independen belum tentu menghasilkan akurasi yang maksimal. Bahkan, model terbaik yang dipilih terdiri dari parameter dependen.

Penelitian mengenai klasifikasi menggunakan CNN telah dilakukan juga oleh Rismiyati pada tahun 2016. Proses yang dilakukan pada penelitian ini adalah pengumpulan data, proses *preprocessing*, klasifikasi dan pengujian. *Preprocessing* dilakukan dengan memotong *region of interest* (ROI) yang berisi salak saja. Klasifikasi dilakukan dengan CNN, dimana untuk mendapatkan akurasi yang terbaik parameter-parameter yang ada harus diujikan. Terdapat dua model yang digunakan yaitu model dua kelas dan empat kelas. Untuk masing-masing model, nilai parameter optimal perlu dicari untuk mendapat akurasi terbaik. Selain itu percobaan juga dilakukan dengan menggunakan ukuran data latih yang beragam untuk melihat pengaruh jumlah data latih terhadap akurasi. Pengujian dilakukan dengan metode *stratified cross validation* untuk mengukur akurasi berdasar confusion matrix. Hasil penelitian menunjukkan bahwa akurasi terbaik untuk model dua kelas didapatkan dengan metode CNN dengan menggunakan learning rate 0.0001, satu lapisan konvolusi dengan jumlah filter lima belas dengan ukuran 3x3x3, dan jumlah neuron pada lapisan tersembunyi 100. Akurasi yang didapatkan adalah 81,5%. Model empat kelas mendapat akurasi 70,7% dengan dua lapisan konvolusi.

Beberapa tinjauan pustaka diatas merupakan referensi untuk mengembangkan penelitian ini. Dari hasil analisa berbagai penelitian yang sudah dilakukan terdahulu akan dilakukan riset tentang penerapan pengolahan citra untuk klasifikasi gambar menggunakan metode *Deep Learning* yaitu *Convolution Neural Network*. Metode *deep learning* ini digunakan untuk memecahkan masalah klasifikasi gambar karena penggunaannya yang memiliki proses komputasi yang relatif cepat, sehingga penelitian ini dapat menghasilkan klasifikasi gambar yang akurat.

Tabel 2.1 merupakan tabel rangkuman perbandingan dengan penelitian sebelumnya yang berkaitan klasifikasi gambar khususnya metode *convolution neural network*.

Tabel 2.1 Perbandingan dengan penelitian terdahulu

No.	Penulis	Judul	Metode	Persamaan
1.	Anna Gummesson (2016)	<i>Prostate Cancer Classification using Convolutional Neural Networks</i>	<i>Convolution Neural Network</i>	Sama-sama menggunakan <i>Convolution Neural Network</i>
2.	Daniel Gibert (2016)	<i>Convolutional Neural Networks for Malware Classification</i>	<i>Convolution Neural Network</i>	Sama-sama menggunakan <i>Convolution Neural Network</i>
3.	Laina Farsiah et al. (2013)	Klasifikasi gambar berwarna menggunakan <i>K-Nearest Neighbor</i> dan <i>Support Vector Machine</i>	<i>K-Nearest Neighbor</i> dan <i>Support Vector Machine</i>	Sama-sama melakukan klasifikasi gambar
4.	Xavier Serra (2017)	<i>Face Recognition using Deep Learning</i>	<i>Deep learning Convolution Neural Network</i>	Sama-sama menggunakan <i>Deep Learning Convolution Neural Network</i>
5.	Zur Erlangung. (2007)	<i>Face Image Analysis With Convolutional Neural Networks</i>	<i>Convolution Neural Network</i>	Sama-sama menggunakan <i>Convolution Neural Network</i>
6.	<i>Purohit Shrinivasacharya dan M V Sudhamani</i>	<i>An Image Crawler For Content Based Image Retrieval System</i>	<i>Crawling Image</i>	Sama – sama <i>Crawling images</i>
7.	<i>Apriliya Fitri Cahyanti, Ristu Saptono, dan Sari Widya Sihwi</i>	<i>Penentuan Model Terbaik pada Metode Naive Bayes Classifier dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter</i>	<i>Naive Bayes</i>	Sama – sama klasifikasi

8.	<i>Rismiyati</i>	<i>Implementasi Convolutional Neural Network Untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital</i>	<i>Convolutional Neural Network</i>	Sama-sama menggunakan <i>Convolution Neural Network</i>
----	------------------	---	-------------------------------------	---

BAB III

LANDASAN TEORI

3.1. Olahraga

Olahraga adalah aktivitas gerak manusia menurut teknik tertentu dalam pelaksanaannya ada unsur bermain, ada rasa senang, dilakukan waktu luang, aktivitas dipilih (sukarela), kepuasan dalam proses, Jika tidak dilaksanakan ada sanksi dan nilai positif. Definisi Olahraga Secara umum, Olahraga adalah suatu bentuk aktivitas fisik yang terencana dan terstruktur yang melibatkan gerakan tubuh berulang-ulang dan ditujukan untuk meningkatkan kebugaran jasmani.

Olahraga adalah proses sistematis yang berupa segala kegiatan atau usaha yang dapat mendorong mengembangkan, dan membina potensi-potensi jasmaniah dan rohaniah seseorang sebagai perorangan atau anggota masyarakat berupa permainan, pertandingan, dan prestasi puncak dalam pembentukan manusia yang memiliki Ideologi yang seutuhnya dan berkualitas berdasarkan Dasar Negara atau Pancasila. (Cholik, 2004).

1) Olahraga Sepak Bola

Sepakbola merupakan salah satu olahraga yang sangat popularitas, dimana olahraga ini memiliki peringkat pertama dari semua cabang olahraga, hal ini dikarenakan sangat disukai, digemari, dan pula diminati oleh masyarakat baik di Indonesia bahkan dunia. Dalam pertandingan, olahraga ini dimainkan oleh dua kelompok berlawanan, dimana masing-masing berjuang untuk memasukkan bola ke gawang kelompok lawan (Wikipedia, 2017).

2) Olahraga Bulutangkis

Bulutangkis atau yang dikenal dengan istilah badminton merupakan salah satu olahraga yang sangat popularitas di Indonesia bahkan di dunia,

dimana olahraga ini memiliki peringkat kedua setelah olahraga sepakbola, hal ini dikarenakan sangat diminati dan digemari oleh para atlet-atlet yang memiliki prestasi dibidangnya. Olahraga ini dimainkan oleh dua orang atau dua pasangan yang saling berlawanan. Bulutangkis bertujuan untuk memukul bola permainan (“kok” atau “*shuttlecock*”) melewati net agar jatuh di bidang permainan lawan yang sudah ditentukan dan berusaha mencegah lawan melakukan hal yang sama (Wikipedia, 2017).

3) Olahraga Bola Basket

Olahraga ini merupakan salah satu olahraga yang sangat popularitas di Indonesia bahkan di dunia, dimana olahraga ini memiliki peringkat ketiga setelah olahraga sepakbola dan badminton, hal ini dikarenakan sangat disukai dan diminati oleh para atlet-atlet yang memiliki prestasi dibidangnya. Olahraga ini terdiri atas dua tim beranggotakan masing-masing lima orang yang saling bertanding mencetak poin dengan memasukkan bola ke dalam keranjang lawan. Bola basket sangat cocok untuk ditonton karena biasa dimainkan di ruang olahraga tertutup dan hanya memerlukan lapangan yang relatif kecil. Selain itu, bola basket mudah dipelajari karena bentuk bolanya yang besar, sehingga tidak menyulitkan pemain ketika memantulkan atau melempar bola tersebut (Wikipedia, 2017).

3.2. Citra Warna

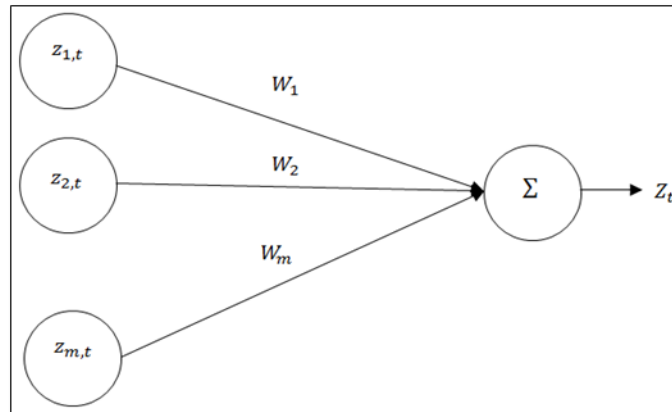
Citra warna atau biasa dikenal dengan citra RGB merupakan suatu model warna aditif yang terdiri atas tiga buah warna dasar yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Sedangkan warna lain merupakan hasil kombinasi dari ketiga warna tersebut. Model warna RGB merupakan warna dasar yang dapat dibedakan oleh sel kerucut mata manusia berdasarkan teori trikromatik. Citra warna merupakan penumpukan dari tiga matriks, dengan masing-masing matriks merepresentasikan nilai masing-masing ketiga warna dasar merah, hijau, dan biru dalam setiap piksel sehingga setiap piksel berkaitan dengan tiga nilai. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB dengan mengubah nilai HSV (*Hue, Saturation, Values*) agar dapat menghasilkan bermacam-macam warna (Sianipar, 2013).

3.3. Klasifikasi Citra

Klasifikasi citra merupakan sebuah pekerjaan untuk memasukkan sebuah citra dan menetapkannya ke sebuah kategori. Ini adalah salah satu permasalahan dalam *Computer Vision* yang dapat disederhanakan dan memiliki berbagai macam aplikasinya. Salah satu aplikasi dalam klasifikasi citra adalah pengklasifikasian nama tempat pada suatu citra. Dalam klasifikasi yang terawasi, setiap masukkan citra pada *training set* diberi label. Saat klasifikasi, label tersebut akan menjadi perbandingan dengan hasil hipotesis yang diberikan oleh model pembelajaran dan akan menghasilkan nilai *error*. Klasifikasi yang terawasi bisa sangat efektif dan akurat dalam mengklasifikasikan citra tempat maupun objek lainnya. Banyak metode dan algoritma yang dapat mendukung proses klasifikasi yang terawasi terutama dengan teknik *Deep Learning* (A'la, 2016).

3.4. Jaringan Saraf Tiruan

Cabang ilmu kecerdasan buatan cukup luas kaitannya dengan disiplin ilmu yang lainnya. Hal ini bisa dilihat dari berbagai aplikasi yang merupakan hasil kombinasi dari berbagai ilmu pengetahuan. Salah satunya adalah *artificial neural network* (ANN). Jaringan saraf tiruan adalah bagian dari sistem kecerdasan buatan (*artificial intelligence*) yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah tiruan atau buatan dimaksudkan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Fausett, 1994).



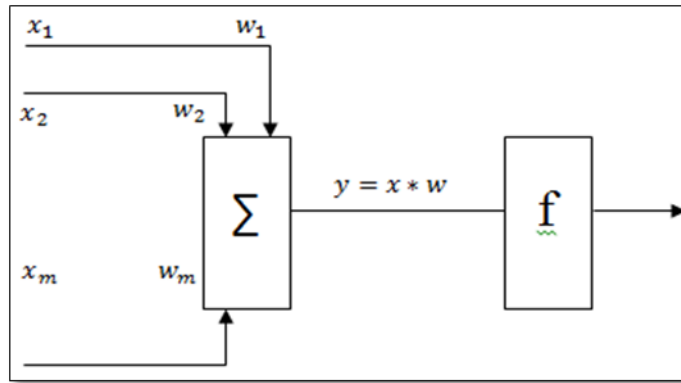
Gambar 3.1 Struktur Jaringan Saraf Tiruan (Fausett,1994)

3.4.1. Model Umum Jaringan Saraf Tiruan

Komponen terkecil dari jaringan saraf tiruan adalah unit atau *neuron*, dimana akan mentransformasikan informasi yang diterima menuju neuron lainnya. hubungan ini dikenal dengan nama bobot yang akan menyimpan informasi pada suatu nilai tertentu pada bobot tersebut. Informasi (*input*) akan dikirim ke neuron dengan bobot kedatangan tertentu, dan akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang akan datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron. Biasanya neuron pada suatu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan *output*). Informasi yang diberikan akan dirambatkan dari lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan lainnya atau dikenal dengan nama lapisan tersembunyi (Erna, 2009).

3.4.2. Model Matematis Jaringan Saraf Tiruan

Masukan dan bobot merupakan model matematis dari dendrit, penjumlahan serta fungsi pengaktif merupakan model matematis dari soma (badan sel), serta keluaran merupakan model matematis dari akson. Dilihat dari gambar 3.2 ini menunjukkan model yang disederhanakan dari sebuah sel saraf tiruan yang merupakan dasar dari jaringan saraf tiruan (Erna, 2009).



Gambar 3.2 Model Matematis Jaringan Saraf Tiruan (Erna, 2009)

Pada gambar 3.2 diatas menyatakan perkalian antara *input* dan nilai bobot koneksi dan hasil perkalian tersebut dijumlahkan serta disimpan dalam neuron. Setelah ini baru dimasukan dalam fungsi *nonlinier* f . Pada aras ambang (*threshold*) tertentu, fungsi f tidak akan menghasilkan *output*. Apabila dari gambar tersebut dinyatakan dalam notasi matematika, maka persamaannya adalah sebagai berikut:

$$y = f(x_1 * w_1 + x_2 * w_2 + \dots + x_m * w_m) \quad 3.1$$

Atau dalam notasi vektor:

$$y = f(x * w) \quad 3.2$$

dimana:

- x : vektor baris yang terdiri dari m anggota
- w : vektor kolom yang terdiri dari m anggota
- y : besaran skalar
- f : fungsi *nonlinear*

3.4.3. Karakteristik *Artificial Neural Network*

Menurut Erna (2009), Ada tiga karakteristik yang ada dalam *Artificial Neural Network* atau jaringan syaraf tiruan adalah sebagai berikut:

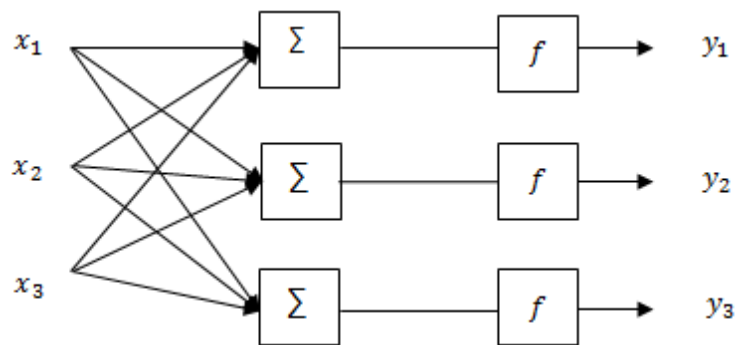
1) Arsitektur Jaringan

Arsitektur ini menggambarkan bagaimana cara menghubungkan neuron tersebut dengan bobot, serta mengatur neuron-neuron yang ada dalam lapisan.

Pada setiap neuron yang berada dalam satu lapisan yang sama mempunyai fungsi aktivasi yang sama, dimana setiap neuron dalam satu lapisan harus dihubungkan dengan setiap neuron pada lapisan berikutnya. Adapun arsitektur jaringan syaraf tiruan yang sering digunakan jaringan syaraf tiruan adalah sebagai berikut.

a) Jaringan dengan lapisan tunggal (*Single Layer Net*)

Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi, dimana hanya memiliki satu lapisan dengan bobot-bobot terhubung. berdasarkan gambar 3.3 dibawah ini, lapisan *input* memiliki tiga neuron diantaranya x_1 , x_2 , dan x_3 . Sedangkan pada lapisan *output* memiliki dua neuron diantaranya y_1 dan y_2 . Kedua neuron pada lapisan ini saling berhubungan, dimana seberapa besar hubungan antara kedua neuron ditentukan oleh bobot (*weight*) yang bersesuaian. Semua unit *input* akan dihubungkan dengan setiap unit *output*.



Gambar 3.3 Jaringan dengan memiliki lapisan tunggal (Erna, 2009)

Dari gambar 3.3 diatas, maka dapat dibuat persamaan model matematikanya sebagai berikut:

$$y_1 = f(x_1 * w_{11} + x_2 * w_{21} + \dots + x_m * w_{m1}) \quad 3.3$$

$$y_2 = f(x_1 * w_{12} + x_2 * w_{22} + \dots + x_m * w_{m2}) \quad 3.4$$

$$y_3 = f(x_1 * w_{13} + x_2 * w_{23} + \dots + x_m * w_{m2}) \quad 3.5$$

⋮

$$y_n = f(x_1 * w_{1n} + x_2 * w_{2n} + \dots + x_m * w_{mn}) \quad 3.6$$

Atau dengan notasi vektor:

$$y = f(x * w)$$

3.7

dimana:

x : vektor baris yang terdiri dari m anggota

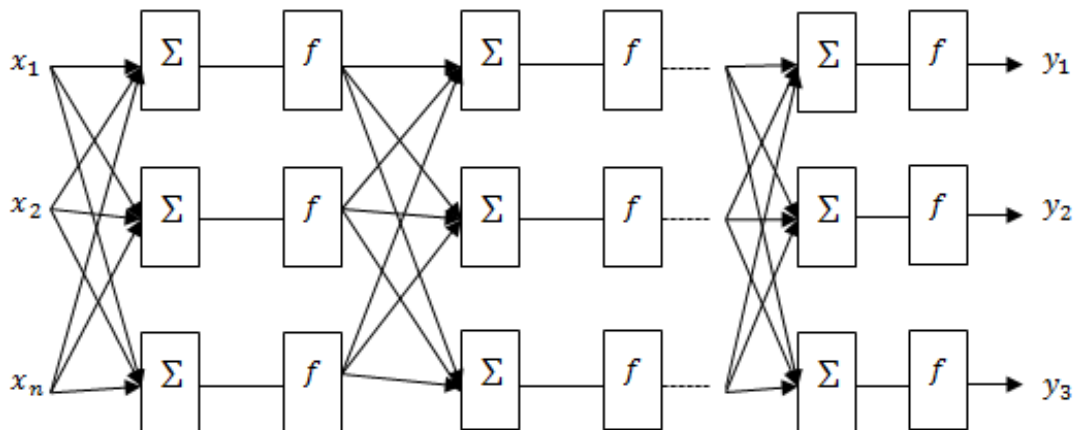
w : vektor kolom yang terdiri dari m anggota

y : vektor kolom yang terdiri dari n elemen

f : fungsi nonlinear

b) Jaringan dengan banyak lapisan (*Multilayer Net*)

Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih rumit daripada lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit pula. Akan tetapi jaringan dengan banyak lapisan ini lebih mampu menyelesaikan masalah. Jaringan dengan banyak lapis memiliki satu atau lebih lapisan, dimana lapisan ini terletak diantara lapisan *input* dan *output*. Berdasarkan gambar 3.4 dibawa ini, biasanya ada lapisan bobot yang terletak antara kedua lapisan yang bersebelahan.



Gambar 3.4 Jaringan saraf tiruan dengan banyak lapisan (Erna, 2009)

2) Algoritma Jaringan

Adapun beberapa algoritma jaringan yang ada dalam jaringan saraf tiruan, diantaranya adalah sebagai berikut:

a) Algoritma Jaringan Pembelajaran

Algoritma pembelajaran ini digunakan untuk menemukan nilai-nilai bobot yang tepat dalam mengirimkan suatu informasi. Secara umum, algoritma pembelajaran terbagi menjadi dua bagian, yaitu:

- i. Pembelajaran Terawasi (*Supervised Learning*), pelajaran ini merupakan algoritma pembelajaran yang memerlukan target *output* yang diharapkan untuk diketahui sebelumnya dalam proses pembelajarannya. Setiap pola *input* beserta target *output* yang ditentukan disebut sebagai pasangan pembelajaran. Algoritma pembelajaran *neural network* yang termasuk dalam kelompok ini antara lain *hebb*, *perceptron*, dan *backpropagation*
- ii. Pembelajaran Tak Terawasi (*Unsupervised Learning*), pembelajaran ini tidak memerlukan target *output* dan tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Yang masuk dalam *unsupervised learning* adalah *Kohonen Self Organizing Maps*, *Learning Vector Quantization*, dan *Counterpropagation*.

b) Algoritma Pengenalan

Setelah menemukan nilai bobot antar neuron yang bersesuaian dengan nilai *output*, maka nilai bobot tersebut digunakan untuk menguji *neural network*. Jika dimasukan nilai suatu *input*, maka akan menghasilkan nilai *ouput*. Proses ini dinamakan proses pengenalan (pengujian). Algoritma pengenalan yang digunakan tergantung pada algoritma pembelajaran yang digunakan, biasanya merupakan bagian dari algoritma pembelajarannya.

3) Fungsi Aktivasi

Menurut Erna (2009), Fungsi aktivasi adalah fungsi yang mentransformasikan nilai penjumlahan menjadi sebuah nilai yang dapat diproses lebih lanjut. Terdapat beberapa fungsi aktivasi, yaitu:

a) Fungsi Undak Biner (*Hardlim*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak untuk mengkonversi input dari suatu variabel yang bernilai kontinu ke suatu output biner. Fungsi hard limit dirumuskan:

$$y = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases}$$

b) Fungsi Bipolar (*Hardlimits*)

Hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0 atau -1. Fungsi Symetric Hard Limit dirumuskan sebagai:

$$y = \begin{cases} 1, & \text{jika } x \geq 0 \\ -1, & \text{jika } x < 0 \end{cases}$$

c) Fungsi Identitas atau Linear (*Purelin*)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *input*. Fungsi ini memiliki rumus yaitu:

$$y = x$$

d) Fungsi *Saturating Linear*

Fungsi ini akan memiliki *output* 0, jika *input*-nya kurang dari $-\frac{1}{2}$, dan akan bernilai 1 jika *input*-nya lebih dari $\frac{1}{2}$. Sedangkan jika nilai *input* terletak antara $-\frac{1}{2}$ dan $\frac{1}{2}$, maka *output*-nya akan bernilai sama dengan nilai *input* ditambah $\frac{1}{2}$. Fungsi *saturating linear* dirumuskan:

$$= \begin{cases} 1; & \text{jika } x \geq 0.5 \\ x + 0.5; & \text{jika } -0.5 \leq x \leq 0.5 \\ 0; & \text{jika } x \leq -0.5 \end{cases}$$

e) Fungsi *Symmetric Saturating Linear*

Fungsi *symetric saturating linear* akan menghasilkan *output* -1 , jika *input*-nya kurang dari -1 . Sedangkan jika nilai *input* terletak antara -1 dan 1 , maka *output*-nya akan bernilai sama dengan nilai *input*-nya. Fungsi *Symmetric Saturating Linear* dirumuskan:

$$= \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ -1; & \text{jika } x \leq -1 \end{cases}$$

f) Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi ini memiliki nilai pada *range* 0 sampai 1. Fungsi *sigmoid biner* dirumuskan:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

dengan $f'(x) = \sigma f(x)[1 - f(x)]$

g) Fungsi *Sigmoid Bipolar*

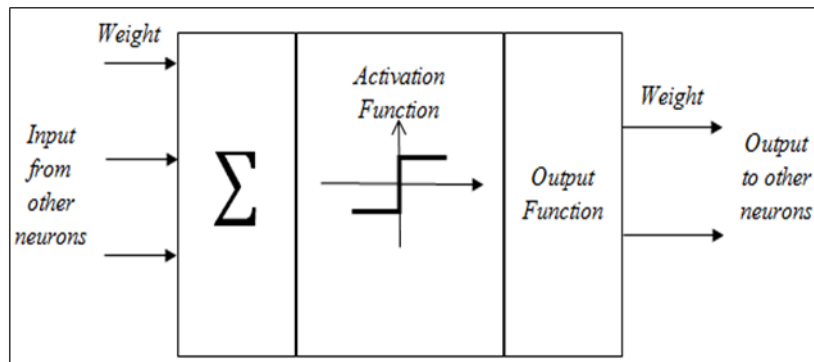
Output dari fungsi ini memiliki *range* antara 1 sampai -1 . Fungsinya dirumuskan:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

dengan $f'(x) = \frac{\sigma}{2}[1 + f(x)][1 - f(x)]$

3.4.4. Komponen Jaringan Jaraf Tiruan

Didalam *neural network* terdapat banyak struktur, tetapi semuanya mempunyai komponen yang hampir sama. Dimana dilihat pada gambar 3.5 berikut memperlihatkan struktur ideal *neural network*.

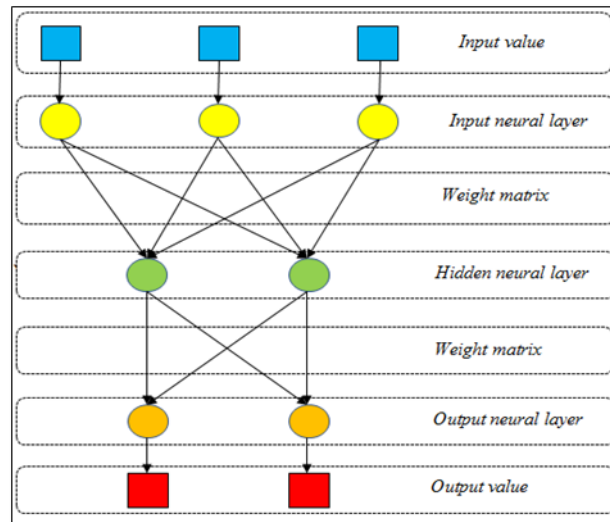


Gambar 3.5 Tampilan struktur NN (Erna, 2009)

Dari struktur neuron pada otak manusia dan proses kerja yang dijelaskan dari gambar diatas, maka dapat dibentuk konsep dasar pembangunan *neural network* buatan. Dimana jaringan ini memiliki ide yang mendasar adalah sebuah aplikasi yang menyerupai otak manusia, baik untuk *parallel processing*, elemen yang diterima, pemrosesan berbagai sinyal, dan juga toleransi terhadap kesalahan. Dilihat dari pola hubungan antar neuron, *artificial neural network* ini memiliki karakteristik, metode penentuan bobot dari tiap koneksi, dan fungsi aktivasinya. Dilihat dari gambar 3.5 diatas merupakan struktur *artificial neural network*, dimana memuat *input* yang berfungsi seperti *dendrite*, *output* yang berfungsi seperti *akson*, dan fungsi aktivasi yang berfungsi seperti sinapsis.

Neural network dibangun dari banyak *node* yang dihubungkan oleh *link* secara langsung. *Link* dari unit yang satu ke unit yang lainnya digunakan untuk melakukan propagasi aktivasi dari unit pertama ke unit selanjutnya. Setiap *link* memiliki bobot numerik. Bobot ini menentukan kekuatan serta penanda dari sebuah konektivitas. Proses pada *artificial neural network* dimulai dari *input* yang diterima oleh neuron beserta dengan nilai bobot dari tiap-tiap *input* yang ada. Setelah masuk ke dalam neuron, nilai *input* yang ada akan dijumlahkan oleh suatu fungsi perambatan (*summing function*), yang bisa dilihat seperti pada gambar dengan lambang sigma (Σ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap neuron, disini akan dibandingkan hasil penjumlahan dengan *threshold* tertentu. Jika nilai melebihi *threshold*, maka aktivasi neuron akan dibatalkan. Sebaliknya jika masih dibawah nilai *threshold*, maka neuron akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai *output* melalui bobot-bobot *output*-

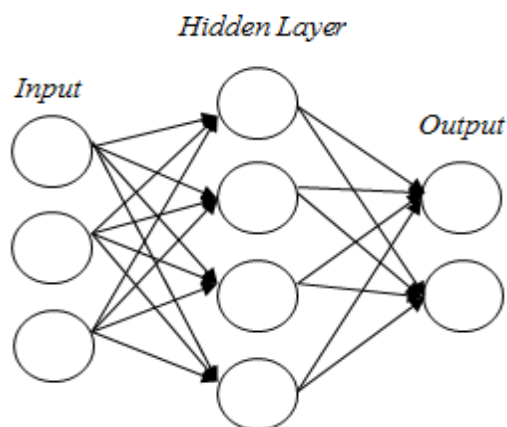
nya ke semua neuron yang berhubungan dengannya. Proses ini akan terus berulang pada *input-input* selanjutnya. Seperti yang ada pada gambar 3.6 menunjukkan *neural network* yang memiliki tiga neuron layer (Erna, 2009).



Gambar 3.6 Tampilan neural network yang memiliki tiga neural layer

3.4.5. Arsitektur Jaringan Saraf Tiruan

Secara umum, Arsitektur ini memiliki beberapa lapisan diantaranya adalah lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Masing-masing lapisan ini memiliki jumlah neuron yang berbeda-beda. Berikut ini adalah gambar arsitektur jaringan saraf tiruan.



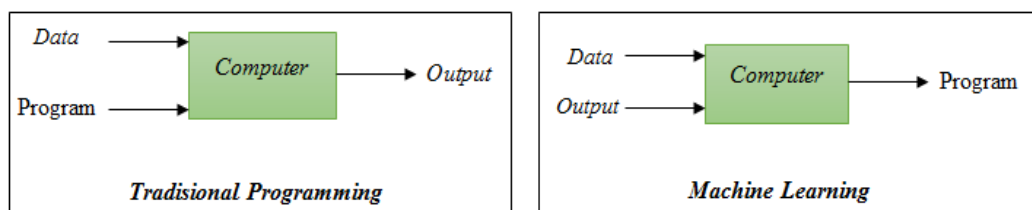
Gambar 3.7 Tampilan arsitektur jaringan saraf tiruan (Erna, 2009)

Dimana dari gambar 3.7 diatas memiliki lapisan masukan (*input layer*), lapisan yang terdiri dari beberapa neuron yang akan menerima sinyal dari luar dan

kemudian meneruskan ke neuron-neuron lain dalam jaringan. Kemudian terdapat lapisan tersembunyi (*hidden layer*), lapisan ini merupakan tiruan dari sel-sel syaraf konektor pada jaringan saraf biologis. Dimana lapisan tersembunyi berfungsi meningkatkan kemampuan jaringan dalam memecahkan masalah. Selanjutnya lapisan keluaran (*output layer*), lapisan ini berfungsi menyalurkan sinyal-sinyal keluaran hasil pemrosesan jaringan dan juga terdiri dari sejumlah neuron.

3.5. *Machine Learning*

Machine Learning adalah serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. *Machine Learning* dapat membuat komputer memprogram diri mereka sendiri. Jika pemrograman adalah pekerjaan untuk membuat otomasi, maka *Machine Learning* mengotomatiskan proses otomasi. Pada dasarnya *Machine Learning* membiarkan data melakukan pekerjaan. Berikut gambaran umum *Machine Learning* dibandingkan dengan pemrograman secara tradisional (A'la, 2016).



Gambar 3.8 Perbandingan Pemrograman Tradisional dengan *Machine Learning* (A'la, 2016)

Dari gambar 3.8 diatas dapat dilihat bahwa pemrograman secara tradisional data dan program dijalankan di komputer untuk menghasilkan *output*. Sedangkan pada *Machine Learning*, data dan *output* dijalankan di komputer untuk membuat sebuah program. Ada banyak algoritma *Machine Learning* yang dikembangkan setiap tahunnya. Setiap algoritma pembelajaran mesin memiliki tiga komponen penting, antara lain:

- a) Representasi: bagaimana merepresentasikan pengetahuan. Contohnya termasuk *Decision tree*, *Neural Network*, *Support Vector Machine* dan lain-lain.

- b) Evaluasi: cara mengevaluasi prediksi dan hipotesis. Contohnya meliputi *Mean Squared Error*, *Cost function* dan lain-lain.
- c) Optimasi: cara program dari model dihasilkan dan proses pencarian parameter terbaik. Misalnya *Convex Optimization* dan *Gradient Descent*.

Selain dari algoritma pembelajaran ada empat jenis cara pembelajaran pada *Machine Learning*, yakni:

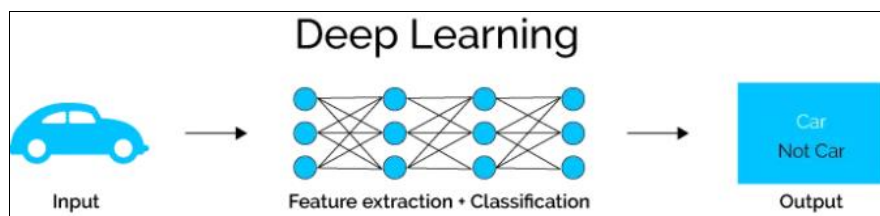
- a) *Supervised Learning*: data pembelajaran mencakup keluaran yang sudah ditentukan.
- b) *Unsupervised Learning*: data pembelajaran tidak mencakup keluaran yang ditentukan
- c) *Semi-supervised Learning*: data pembelajaran mencakup beberapa keluaran yang ditentukan.
- d) *Reinforcement Learning*: pemberian hadiah dari setiap serangkaian tindakan yang dilakukan.

Supervised Learning adalah cara pembelajaran yang banyak dipelajari dan memiliki banyak algoritma. Cara pembelajaran ini sangatlah efektif untuk permasalahan klasifikasi dan regresi. Ada 5 langkah dasar yang digunakan untuk melakukan tugas *Machine Learning* dan tugas ini sangatlah penting dalam mempersiapkan solusi untuk segala bentuk permasalahan dalam *Machine Learning* maupun *Deep Learning* adalah:

- 1) Mengumpulkan data: dibentuk dari beberapa *file* yang berisi deretan data yang dapat dipelajari dan setelah itu dipisah antara fitur masukkan dan keluaran.
- 2) Mempersiapkan data: penentuan kualitas data dan *preprocess* data sehingga hasil yang didapat juga baik.
- 3) Melatih sebuah model: langkah ini melibatkan pemilihan algoritma dan representasi data yang tepat dalam bentuk model.
- 4) Mengevaluasi model: untuk menguji keakuratan berdasarkan bagian test set pada dataset.
- 5) Meningkatkan kinerja: langkah ini melibatkan pemilihan *hyperparameter* untuk meningkatkan efisiensi dan biasanya menggunakan *cross-validation*.

3.6. *Deep Learning*

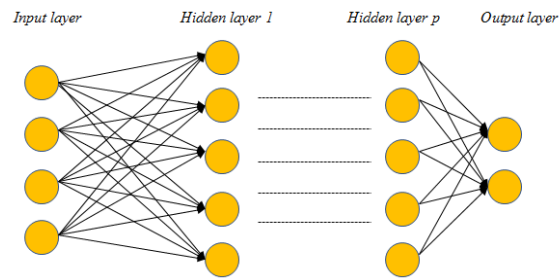
Deep Learning merupakan salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input dan mengolahnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi nonlinier dari data masukan untuk menghitung nilai *output*. Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan, algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. Dalam jaringan saraf tiruan tipe *Deep Learning* setiap lapisan tersembunyi bertanggung jawab untuk melatih serangkaian fitur unik berdasarkan *output* dari jaringan sebelumnya. Algoritma ini akan menjadi semakin kompleks dan bersifat abstrak ketika jumlah lapisan tersembunyi (*hidden layer*) semakin bertambah banyak. Jaringan saraf yang dimiliki oleh *Deep Learning* terbentuk dari hirarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi layer*). Berdasarkan hal itulah *Deep Learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi nonlinier (LeCun et al, 2015).



Gambar 3.9 Struktur pemodelan jaringan pada *Deep Learning* ((LeCun et al, 2015).

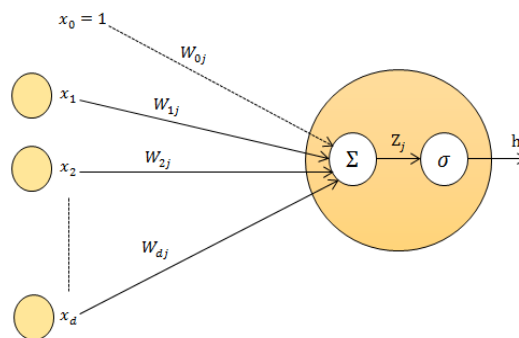
Metode *deep learning* merupakan metode pembelajaran representasi dengan beberapa tingkat representasi, dimana representasi membentuk medan

arsitektur jaringan syaraf yang berisi banyak layer (lapisan). Lapisan pada *deep learning* terdiri atas tiga bagian, yaitu *input layer*, *hidden layer*, dan *output layer*. Pada *hidden layer* dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar meminimalisir *error* pada *output*.



Gambar 3.10 Layer-layer pada *deep learning* (Rio, Akira, Takashi, & Kuniaki, 2016)

Pada gambar 3.10 diatas dapat mengilustrasikan layer-layer pada *deep learning* yang memiliki $p + 2$ layer (p *hidden layer*, 1 *input* dan 1 *output layer*). Bulatan berwarna kuning menggambarkan neuron. Di setiap lapis *hidden layer* terdapat satu atau lebih neuron. Neuron-neuron tersebut akan terhubung langsung dengan neuron lain pada layer selanjutnya. Koneksi antar neuron hanya terjadi di antara 2 buah layer (*input* dan *output*) tidak ada koneksi pada layer yang sama walaupun secara teknis bisa saja dibuat dan juga *fully connected*.



Gambar 3.11 Sebuah perceptron dengan d buah input (LeCun et al, 2015)

Sebuah sistem yang terdiri dari sebuah neuron beserta *input* dan *output*-nya disebut sebagai perceptron. Seperti yang dilihat pada gambar 3.11 sebuah neuron diperbesar disimbolkan sebagai h_j yang menerima d buah *input* $x_1, \dots, x_d \in R$ dapat berasal dari data ataupun *output* dari lapisan sebelumnya. Sedangkan x_0

tidak dianggap sebagai *input* (atau sebagai *dummy*) dan selalu bernilai 1. Variabel-variabel $W_{ij}, \dots, W_{dj} \in R$ merupakan bobot/*weights* dari koneksi *input* ke neuron h_j dan W_{0j} dinamakan sebagai bias. Bobot adalah koneksi antar lapisan yang berupa nilai yang menentukan fungsi *input-output* dari mesin. Bobot adalah parameter penyesuaian yang diatur oleh mesin untuk mengukur kesalahan antara nilai *output* dan pola nilai yang diinginkan pada pembelajaran. Sehingga bobot inilah yang diatur oleh mesin untuk mengurangi kesalahan yang terjadi. Dalam sistem *deep learning* kemungkinan terdapat ratusan juta bobot yang dapat diatur. Untuk menyesuaikan vektor bobot dengan benar, algoritma menghitung vektor gradien untuk setiap bobot berdasarkan jumlah kesalahan yang meningkat atau menurun jika bobot meningkat dalam jumlah kecil (LeCun et al, 2015).

Operasi pada sebuah perceptron merupakan dua buah operasi terpisah yaitu operasi linear Σ dan operasi non-linear/aktivasi σ . Operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi dari operasi linear akan ditransformasi menggunakan operasi non-linear yang disebut sebagai fungsi aktivasi.

$$Z_j = \sum_{i=0}^d W_{ij} X_i \quad 3.8$$

$$h_j = \sigma(Z_j) \quad 3.9$$

Pada operasi aktivasi/non-linear $\sigma : R \rightarrow R$ terdapat berbagai macam fungsi aktivasi yang dapat diimplementasikan pada *hidden neuron*. Di dalam penelitian ini digunakan salah satu fungsi aktivasi yaitu *softmax layer*. *Softmax layer* digunakan apabila pada permasalahan *multiclass classification*, *output layer* biasanya memiliki lebih dari satu neuron. Misalkan $\mathbf{a} = [a_1, \dots, a_m]^T$ merupakan sebuah vektor dengan m buah elemen, softmax didefinisikan sebagai berikut:

$$\sigma(a_j) = \frac{\exp(a_j)}{\sum_{k=1}^m \exp(a_k)} \quad 3.10$$

Dapat dicek bahwa $\sum_{j=1}^m \sigma(a_j) = 1$ untuk *softmax*.

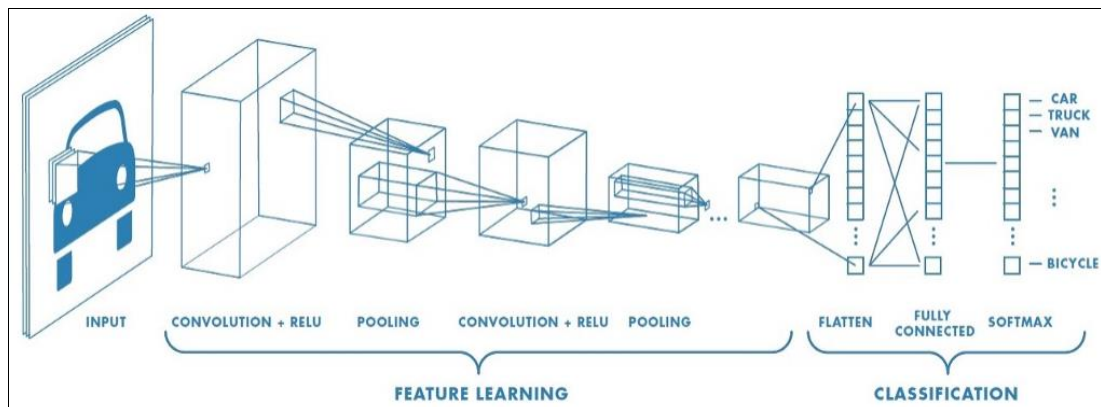
Beberapa algoritma yang menerapkan konsep *deep learning* antara lain *Deep Convolutional Neural Networks* (DCNN) untuk klasifikasi gambar, *Deep Belief Network-Deep Neural Network* (DBN-DNN) untuk pengenalan suara,

Recurrent Neural Network (RNN) untuk penerjemahan bahasa, *Query-Oriented Deep Extraction* (QODE) yang berbasis *Restricted Boltzmann Machine* (RBM) untuk peringkasan multi dokumen, *Conditional Restricted Boltzmann Machine* (RBM) untuk memprediksi *Drug-Target Interaction* (DTI), dan *Deep Belief Network* (DBN) untuk prediksi data sesuai waktu (LeCun, Bengio, & Hinton, 2015).

3.7. Convolution Neural Network

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*. Secara garis besar *Convolutional Neural Network* (CNN) tidak jauh beda dengan *neural network* biasanya. CNN terdiri dari neuron yang memiliki *weight*, bias dan *activation function*. *Convolutional layer* juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*).

Arsitektur dari *Convolutional Neural Network* (CNN) dibagi menjadi 2 bagian yaitu *Feature Extraction Layer* dan *Fully Connected Layer* (MLP)



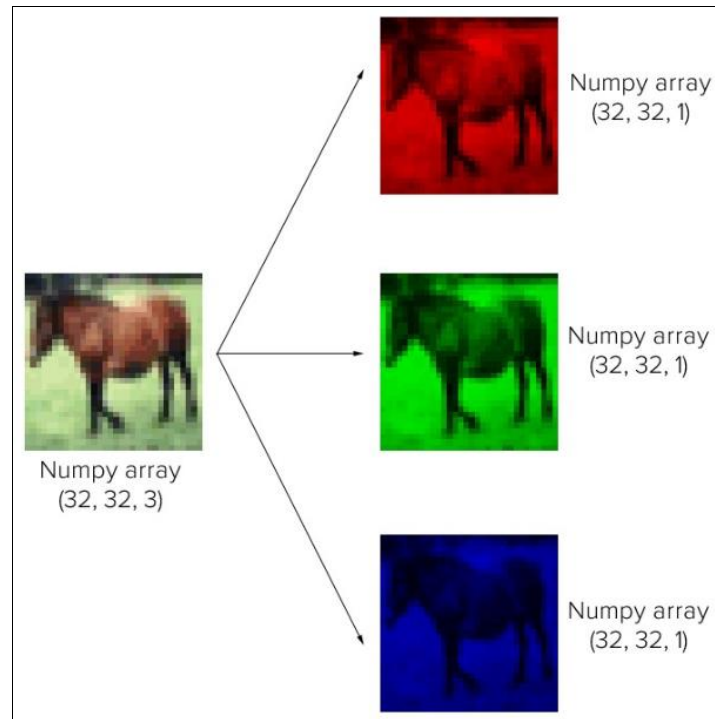
Gambar 3.12 Tampilan Arsitektur Convolution Neural Network (CNN)

1) Feature Extraction Layer

Dikatakan *Feature Extraction Layer*, karena proses yang terjadi pada bagian ini adalah melakukan “*encoding*” dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut (*Feature Extraction*). *Feature extraction layer* terdiri dari dua bagian yaitu:

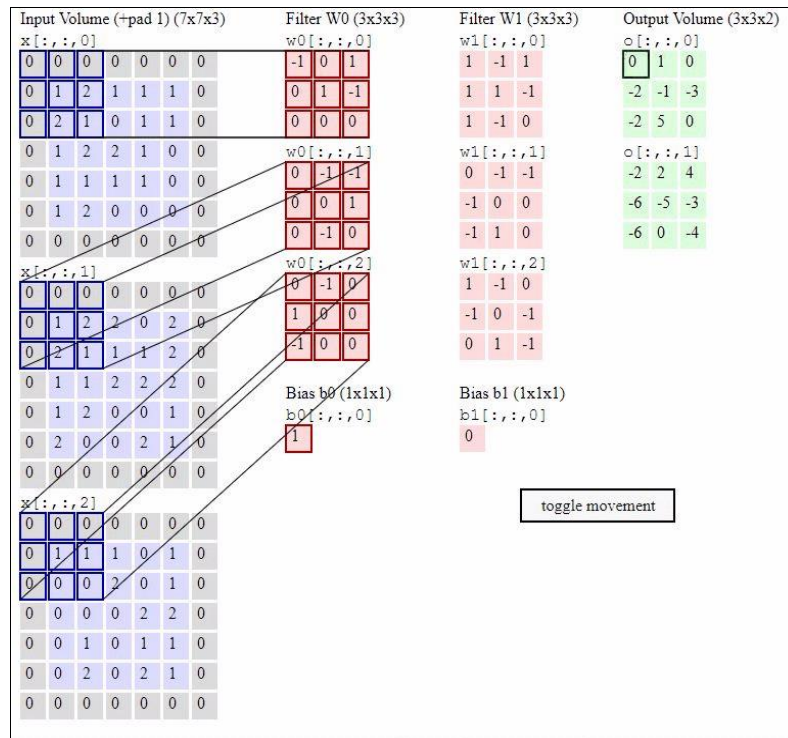
a) *Convolutional Layer*

Convolutional layer terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*).



Gambar 3.13 *Image RGB*

Gambar 3.13 diatas adalah RGB (*Red, Green, Blue*) *image* berukuran 32x32 *pixels* yang sebenarnya adalah *multidimensional array* dengan ukuran 32x32x3 (3 adalah jumlah *channel*). Sebagai contoh, layer pertama pada *feature extraction layer* biasanya adalah *conv. layer* dengan ukuran 5x5x3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal atau jumlah 3 buah sesuai dengan *channel* dari *image* tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “*dot*” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*.



Gambar 3.14 Tampilan Feature Map

Untuk menghitung dimensi dari *feature map*, bisa gunakan rumus seperti dibawah ini:

$$output = \frac{W - N + 2P}{S} + 1 \quad 3.11$$

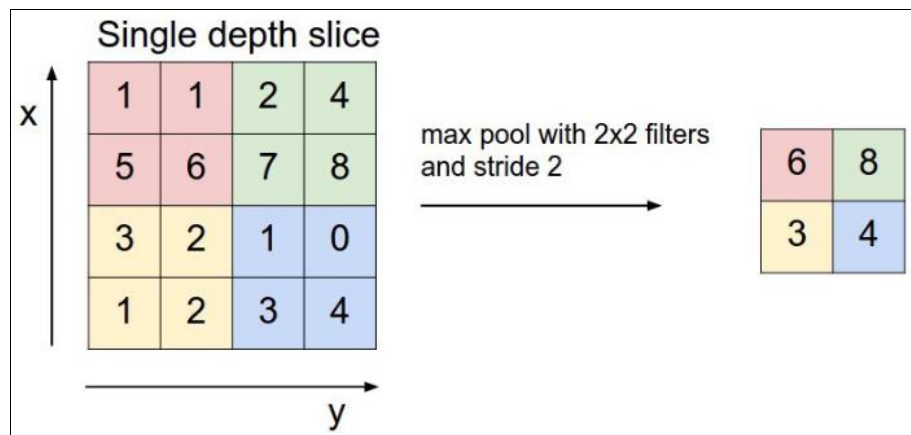
Dimana:

- W = Panjang atau Tinggi *input*
- N = Panjang atau Tinggi *filter*
- P = *Zero Padding*
- S = *Stride*

b) *Pooling Layer*

Pooling layer biasanya berada setelah *conv. layer*. Pada prinsipnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh, jika menggunakan *Max Pooling* 2x2 dengan *stride* 2, maka pada setiap pergeseran filter, nilai *maximum* pada area 2x2 *pixel* tersebut yang akan dipilih, sedangkan *Average*

Pooling akan memilih nilai rata-ratanya. Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus di *update* semakin sedikit dan mengatasi *overfitting*.



Gambar 3.15 *Max Pooling*

c) *Stride*

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *conv. filter* akan bergeser sebanyak 1 *pixels* secara horizontal lalu vertikal. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Namun perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil, maka tidak selalu akan mendapatkan performa yang bagus.

d) *Padding*

Sedangkan *Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *conv. layer* (*Feature Map*). Ada beberapa tujuan dari penggunaan *padding* adalah sebagai berikut:

- i. Dimensi *output* dari *conv. layer* selalu lebih kecil dari *input*-nya (kecuali penggunaan 1x1 *filter* dengan *stride* 1). *Output* ini akan digunakan kembali sebagai *input* dari *conv. layer* selanjutnya, sehingga makin banyak

informasi yang terbuang. Kemudian dengan menggunakan *padding*, maka dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga bisa menggunakan *conv. layer* yang lebih dalam sehingga lebih banyak *features* yang berhasil di-*extract*.

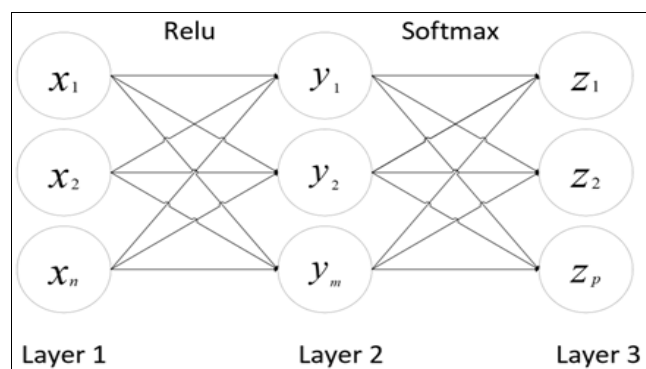
- ii. Meningkatkan performa dari model karena *conv. filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut.

e) *Dropout*

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden* maupun layer yang *visible* di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas p yang bernilai antara 0 dan 1.

2) *Fully-Connected Layer* (MLP).

Proses pada *Fully Connected Layer* berfungsi untuk melakukan proses klasifikasi yaitu dengan menggunakan *softmax* yang sesuai dengan proses yang digambarkan pada gambar 3.14.



Gambar 3.16 Proses pada fully connected layer (Yudha, 2014)

Dari gambar 3.16 diatas, dimana *layer 1* akan dilakukan *feedforwarding* menuju *layer 2* dengan menggunakan fungsi aktivasi Rel U. Pada *layer 2* akan dilakukan klasifikasi dengan menggunakan *softmax*.

BAB IV

METODE PENELITIAN

4.1. Populasi dan Sampel

Populasi dalam penelitian ini adalah gambar olahraga yang terdiri dari gambar bola kaki, bola kok, dan bola basket yang diambil dari situs pencarian *google*. Sedangkan sampel yang digunakan dalam penelitian ini adalah 150 data gambar dari setiap kategori yang hanya memuat satu *object* dalam satu gambar.

4.2. Variabel Penelitian dan Definisi Operasi Variabel

Variabel yang digunakan dalam penelitian ini adalah Bola Kaki, Bola Kok, dan Bola Basket. Ketiga variabel tersebut memiliki warna *background* adalah putih. Dimana data gambar ini dibagi kedalam 2 kelompok yaitu data *train* dan data *test*. Dimana data *train* berisi 300 data gambar, dimana berisi 100 data gambar dari masing-masing variabel. Kemudian untuk data *test* terdapat 150 data gambar, dimana berisi 50 gambar dari masing-masing variabel.

4.3. Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data *sekunder*. Data tersebut diperoleh dengan menggunakan teknik unduh gambar menggunakan *fatkun batch* dari situs *google images*.

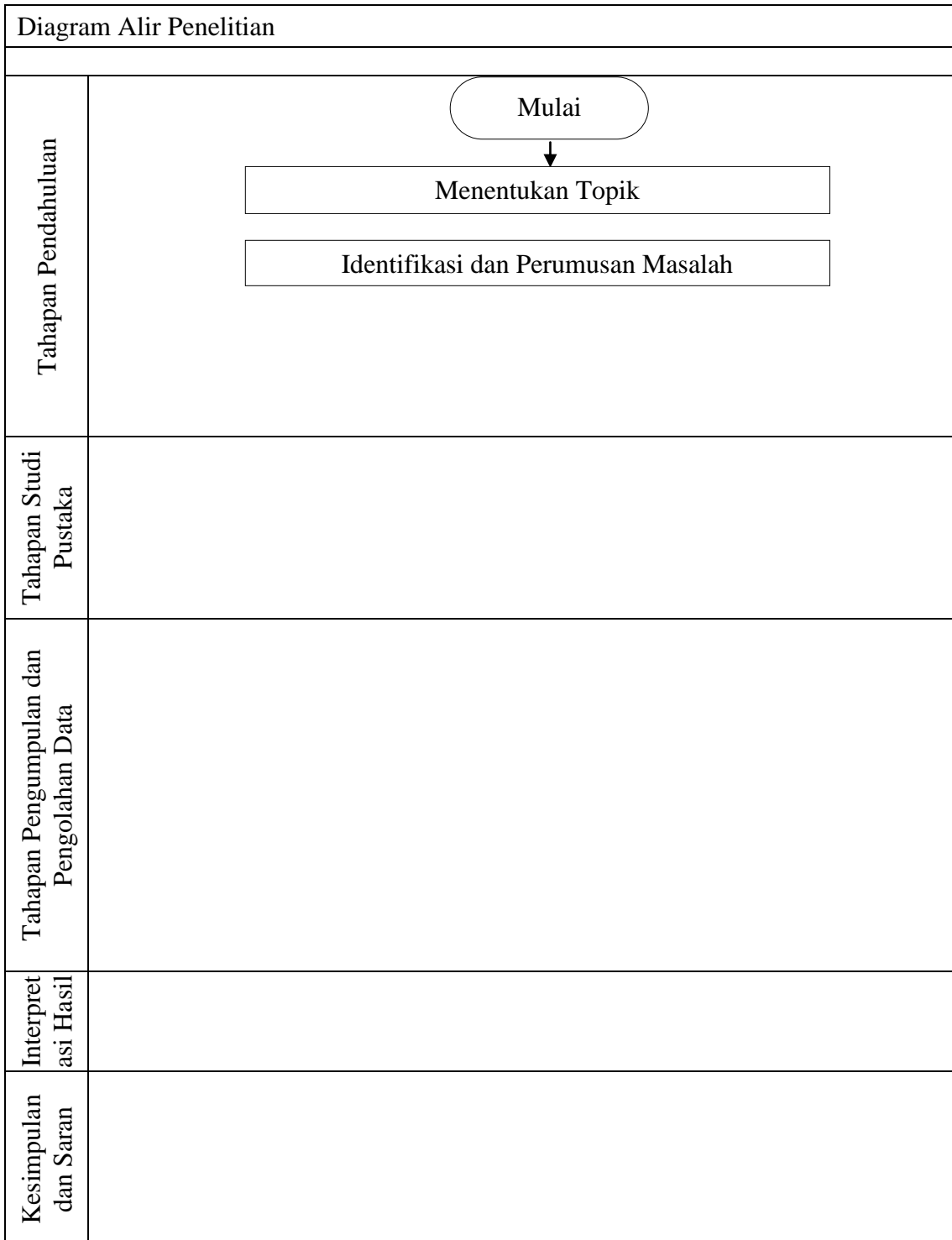
4.4. Metode Analisis

Software yang digunakan dalam penelitian ini adalah *google chrome*, *python 3.5.4*, dan *R Studio*. Metode analisis data yang digunakan dalam penelitian ini adalah metode *Convolution Neural Network*, digunakan untuk mengklasifikasikan gambar bola kaki, bola kok, dan bola basket.

4.5. Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini digambarkan melalui gambar 4.1 dibawah ini:

Tabel 4.1 *Diagram Alir Penelitian*



BAB V

HASIL DAN PEMBASAN

Pada penelitian ini, peneliti melakukan klasifikasi gambar bola kaki, bola kok, dan bola basket dengan menggunakan *Convolutional Neural Network* (CNN). Proses utama dalam pembuatan model klasifikasi metode CNN adalah perancangan klasifikasi. Prinsip dasar dari proses ini adalah dengan melakukan pelatihan pada CNN untuk menghasilkan model yang paling bagus dalam hal akurasi yang baik. Dalam pelatihan CNN terdapat data *train* dan data *test*. Data *train* digunakan untuk melakukan *training* pada model dan data *test* digunakan untuk validasi dari data *training*. Akurasi model tersebut dapat ditentukan dengan melakukan validasi dengan menggunakan data uji.

5.1. Pembuatan Dataset

Peneliti mengumpulkan dan membuat dataset berupa gambar yang di dapat dari google image. *Deep learning* terutama CNN biasanya membutuhkan banyak data gambar, untuk mendownload gambar pada *google image* secara manual tentu membutuhkan banyak waktu dan membosankan. Sehingga perlu ditemukan solusi untuk memanfaatkan *google image* sebagai saran membuat dataset yang baik dan cepat. Peneliti menggunakan *fatkun batch* untuk mengumpulkan gambar dari *google image*.

5.1.1. Fatkun Batch

Fatkun Batch ini digunakan untuk mengambil gambar secara keseluruhan pada *google image*. Berikut ini langkah-langkah untuk mengunduh gambar keseluruhan dengan menggunakan *Fatkun Batch* adalah sebagai berikut:

1. Peneliti mengetik gambar “bola kok”. Kemudian klik pada “gambar” maka akan memunculkan semua gambar bola kok pada *google image*.

Gambar 5.1 *Tampilan google image*

2. Kemudian peneliti mengklik pada lambang *fatkun* batch atau bisa langsung klik “alt+z” secara bersamaan untuk mengambil gambar secara keseluruhan.

Gambar 5.2 *Tampilan lambang fatkun batch*

3. Lalu peneliti mengklik “Tab ini”

Gambar 5.3 *Tampilan halaman “Tab ini”*

4. Kemudian klik “menyimpan gambar”, lalu semua gambar terdownload secara otomatis.

Gambar 5.4 *Tampilan halaman “menyimpan gambar*

Setelah semua data terkumpul, lalu peneliti melakukan seleksi gambar yang akan digunakan. Ciri-ciri gambar yang baik yaitu terdapat gambar yang memiliki *background* warna putih, dimana peneliti mendapatkan 450 gambar untuk dijadikan data *train* dan data *test*. Kemudian akan dibuat dan disimpan dalam satu *folder* yaitu “data150”. Berikut ini terdapat daftar gambar yang sudah peneliti kumpulkan.

Gambar 5.5 *Tampilan data gambar yang sudah terkumpul*

5.2. *Preprocessing Convolution Neural Network (CNN)*

Setelah membuat dataset, selanjutnya akan mengaktifkan *packages* “Keras” dan “*EImage*”. Dimana *packages* keras digunakan untuk menyelesaikan masalah *Neural Network* (NN), *Convolution Neural Network* (CNN), dan *Recurrent Neural Network* (RNN). Dan *package EImage* digunakan untuk membaca dan menampilkan data berupa gambar atau citra. Kemudian pada table 5.1 dibawa ini, terdapat perintah untuk mengaktifkan *packages* tersebut maka dipanggil menggunakan perintah *library* seperti yang ditunjukkan pada kode baris 3-4. Selanjutnya pada kode baris 6, menentukan letak atau lokasi dari gambar yang telah diseleksi yaitu pada folder “data150”.

Tabel 5.1 *Tampilan sintaks untuk mengaktifkan library keras dan EImage*

1	# Convolutional Neural Networks
2	#Packages
3	library(keras)
4	library(EBImage)
5	
6	setwd('D://data150')

Selanjutnya untuk tabel 5.2 dibawah ini digunakan untuk melakukan *list files* untuk membuat semua daftar gambar yang ada dalam folder tersebut, dan melihat deskripsi dari *file* yang ditunjukkan pada kode baris 8-10. *Output* dari *summary* ini adalah banyaknya gambar, *class* dan mode. dimana terdapat 450 character. Kemudian pada baris 11 digunakan untuk membaca gambar menggunakan *function* “*readImage*” dari data gambar *image* menggunakan “*lapply*”. Pada kode baris 13-14 digunakan untuk menampilkan gambar ke-10, dan melihat *properties* dari gambar ke-10 pada data *list_of_image*.

Tabel 5.2 Mempersiapkan Gambar

7	#Read Images
8	images <- list.files()
9	Images
10	summary(images)
11	list_of_images = lapply(images, readImage)
12	list_of_images
13	display(list_of_images[[10]])
14	print(list_of_images[[10]])

Selanjutnya untuk tabel 5.3 dibawah ini digunakan untuk membuat data *train* dan data *test*. Dimana ada 450 data gambar yang akan dibagi menjadi dua bagian yaitu 300 data *training*, dimana setiap kategori memuat 100 data. Dan 150 untuk data *testing*, dimana setiap kategori memuat 50 data gambar. Selain menentukan data *training* dan data *testing*, data tersebut pun bisa ditampilkan untuk menentukan apakah data yang akan digunakan tersedia semua atau tidak. Perintah yang bisa digunakan yaitu menggunakan “*str*” seperti pada kode baris 17

dan 23. Dataset juga bisa ditampilkan menggunakan perintah *display*. Berdasarkan skrip pada kode baris 19 dan 25, dilakukan perintah untuk menampilkan data gambar ke-10 untuk data *training* dan data *testing*, dimana hasilnya merupakan gambar “bola basket”.

Tabel 5.3 Menampilkan dataset

```
15 #create train
16 train <- list_of_images[c(1:100,151:250,301:400)]
17 str(train)
18 Train
19 display(train[[10]])
20
21 #create test
22 test <-list_of_images[c(101:150,251:300,401:450)]
23 str(test)
24 Test
25 display(test[[10]])
```

Tahapan selanjutnya dilihat pada tabel 5.4 dibawah ini yaitu mengubah ukuran gambar menjadi 32x32 piksel. Ukuran gambar yang dianjurkan adalah nilai pangkat dua. Nilai yang lebih kecil dari 32 adalah 16, sedangkan yang lebih besar adalah 64. Penggunaan ukuran 16x16 menjadikan informasi piksel banyak yang hilang, sedangkan penggunaan ukuran 64x64 menjadikan pemrosesan menjadi lebih lambat. Sehingga pada akhirnya ukuran yang digunakan adalah sebesar 32x32 piksel. Perintah yang digunakan yaitu *resize* seperti yang ada pada kode baris 27 dan 30. Proses pengubahan ukuran dari data *training* dan *testing* dilakukan satu per satu atau menggunakan perulangan hingga selesai.

Tabel 5.4 *Resize dan Combine*

```
26 #Resize
27 for (i in 1:300) {train[[i]] <- resize(train[[i]], 32, 32)}
28 str(train)
29
30 for (i in 1:150) {test[[i]] <- resize(test[[i]], 32, 32)}
31 str(train)
32
33 #Combine
34 train <- combine(train)
35 str(train)
36 x <- tile(train, 30)
37 display(x, title = 'Pictures')
38
39 test <- combine(test)
40 y <- tile(test, 15)
41 display(y, title = 'Pics')
42 str(train)
```

Berdasarkan sintaks yang sudah peneliti buat diatas tentang *resize* dan *combine* untuk data *training* dan *testing*, maka akan menghasilkan *output* dari hasil *resize* dan *combine* untuk data *training* dan *testing*. Dimana hasil *resize* untuk data *training* menggunakan ukuran 32x32 *pixel*, sehingga dapat dilihat bentuk dari gambar *train* adalah hasil penggabungan dari bola kaki, bola kok, dan bola basket yang dimuat dalam satu *frame* dengan menggunakan 10 kolom sehingga baris yang tercipta yaitu 30. Seperti yang dilihat pada gambar 5.6 dibawah ini.

Gambar 5.6 *Hasil resize dan combine untuk data training*

Sedangkan untuk data *testing* terdapat hasil penggabungan dari bola kaki, bola kok, dan bola basket yang dimuat dalam satu *frame* dengan menggunakan 10 kolom sehingga baris yang tercipta yaitu 15 baris. Hasilnya bisa dilihat pada gambar 5.7 dibawah ini.

Gambar 5.7 Hasil *resize* dan *combine* untuk data *testing*

Kemudian setelah proses perubahan ukuran gambar menandakan bahwa data gambar siap untuk diproses lebih lanjut menggunakan model *Convolution neural network*. Akan tetapi terlebih dahulu peneliti memberikan label untuk data *training* dan *testing*. Dimana digunakan perintah pada kode baris 44 dan 46 untuk melabelkan setiap data kategori, pelabelan ini menggunakan angka 0 (nol) untuk gambar bola kaki, 1 (satu) untuk gambar bola kok, dan 2 (dua) untuk gambar bola basket. Pelabelan ini digunakan untuk memudahkan pembacaan data antara gambar bola, kaki, bola kok, dan bola basket. Hasil sintaks untuk proses pelabelan yang sudah peneliti buat seperti pada tabel 5.5 dibawah ini.

Tabel 5.5 Proses Pelabelan data gambar

43	# Response
44	trainy <- c(rep(0,100), rep(1,100), rep(2,100))
45	trainy
46	testy <- c(rep(0,50), rep(1,50), rep(2,50))
47	testy

5.3. Pembuatan Model *Convolution Neural Network* (CNN)

Setelah data siap untuk di latih, selanjutnya penentuan model yang digunakan, dimana disini menggunakan CNN menggunakan API keras pada library *Tensorflow*. Dalam keras terdapat 2 cara dalam membangun model yaitu model *sequential* dan *function* API. *Function* API digunakan untuk menentukan model kompleks, seperti model *multi-output*, grafik asiklik terarah, atau model dengan lapisan bersama. Sedangkan *sequential* model lebih seperti tumpukan linier dari layer. Model CNN mempunyai beberapa layer diantaranya *convolution*,

pooling, dan *dropout*. Proses konvolusi dilakukan sebanyak 4 kali ditunjukkan oleh banyaknya *convolution layer* yang digunakan.

Tabel 5.6 *Create Model CNN*

```
48 # Model
49 model <- keras_model_sequential()
50
51 model %>%
52   layer_conv_2d(filters = 32,
53                 kernel_size = c(3,3),
54                 activation = 'relu',
55                 input_shape = c(32, 32, 3)) %>%
56   layer_conv_2d(filters = 32,
57                 kernel_size = c(3,3),
58                 activation = 'relu') %>%
59   layer_max_pooling_2d(pool_size = c(2,2)) %>%
60   layer_dropout(rate = 0.01) %>%
61   layer_conv_2d(filters = 64,
62                 kernel_size = c(3,3),
63                 activation = 'relu') %>%
64   layer_conv_2d(filters = 64,
65                 kernel_size = c(3,3),
66                 activation = 'relu') %>%
67   layer_max_pooling_2d(pool_size = c(2,2)) %>%
68   layer_dropout(rate = 0.01) %>%
69   layer_flatten() %>%
70   layer_dense(units = 256, activation = 'relu') %>%
71   layer_dropout(rate=0.01) %>%
72   layer_dense(units = 3, activation = 'softmax') %>%
73
74   compile(loss = 'categorical_crossentropy',
75           optimizer = optimizer_sgd(lr = 0.01,
```

76	<code>decay = 1e-6,</code>
77	<code>momentum = 0.9,</code>
78	<code>nesterov = T),</code>
79	<code>metrics = c('accuracy'))</code>
80	<code>summary(model)</code>

Dari table 5.6 diatas, Kode baris 49 menunjukkan pengaktifan *sequential* model yaitu model sederhana yang hanya perlu penambahan layer terhadap model yang ada. Pada kode baris 51-73 menggunakan beberapa layer diantaranya *convolution layer*, *pooling layer*, *dropout layer*, *flatten layer* dan *dense layer*. Pada model diatas, untuk *convolution* pertama dan kedua (kode baris 52-58) menggunakan jumlah filter 32, *kenel size* 3x3 dan *activation relu* dengan *input shape* 32x32x3. Kemudian untuk *convolution* yang ketiga dan keempat (kode baris 61-68) menggunakan filter 64, dengan *pool size* 2x2 dan *learning rate* 0.01. Kode baris 68-72 merupakan *fully-connected layer*, pada *layer dense*. baris 72 menunjukkan *output* yang dihasilkan yaitu 3 kategori. Kemudian didapatkan model seperti gambar 5.8 dibawah ini.

Gambar 5.8 Model CNN

Dari gambar 5.8 diatas merupakan model CNN yang sudah dibuat. Pada proses konvolusi pertama, gambar sebagai *input* berukuran 32x32 piksel sebenarnya adalah *multidimensional array* dengan ukuran 32x32x3 (3 adalah jumlah *channel* RGB). gambar inilah yang akan dikenakan beberapa proses seperti yang disebutkan pada model. *Filter* akan digerakkan dengan *stride* 1 ke seluruh bagian dari *input* gambar, mulai dari sudut kiri atas sampai kanan bawah. Setiap pergeseran filter pada *input* gambar dilakukan operasi “dot” atau perhitungan matematis. Pada penelitian ini tidak ditambahkan *zero padding*, sehingga ukuran *input* yang digunakan sama dengan 32x32 piksel. Dengan kata lain, *output* yang dihasilkan dari proses konvolusi memiliki ukuran yang lebih kecil. Besarnya ukuran gambar yang dihasilkan dari proses konvolusi pertama ini menjadi 30x30

piksel. Hasil ukuran ini juga bisa didapatkan menggunakan perhitungan dibawah ini.

$$output = \frac{W - N + 2P}{S} + 1 = \frac{32 - 3 + 2(0)}{1} + 1 = 30$$

Begitu juga seterusnya untuk melakukan perhitungan ukuran gambar yang terbentuk. Hasil operasi ini kemudian dikenakan dengan fungsi aktivasi yaitu *ReLU* dan *pooling*. Pada prinsipnya, *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu akan bergeser ke seluruh area *feature map*. *Pooling* yang digunakan yaitu *Max Pooling*. Tujuan dari penggunaan *pooling layer* yaitu untuk mengurangi dimensi dari *feature map* (*downsampling*). Dengan menggunakan filter ukuran 2x2 dan *stride* 1 pada operasi *max pooling*, didapatkan ukuran gambar yang terbentuk setelah dilakukan proses konvolusi pertama. Dikenakan fungsi aktivasi, kemudian operasi *max pooling* adalah sebesar 14x14 *pixel*. Hasil dari proses konvolusi berupa *feature map* yang digunakan kembali sebagai *input* untuk proses konvolusi berikutnya. Proses tersebut berjalan hingga berakhirnya proses konvolusi. Selain itu, metode regularisasi yang digunakan *dropout*, dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Hasil dari proses konvolusi berupa *feature map* masih berbentuk *multidimensional array*, sehingga dilakukan *flatten feature map* menjadi sebuah vektor agar bisa digunakan sebagai *input* dari *fully-connected layer*, hingga pada akhirnya dilakukan klasifikasi gambar. Pada *fully-connected layer* digunakan neuron sebanyak 256, hingga pada akhirnya model terbentuk dan berhasil melakukan klasifikasi antara gambar bola kaki, bola kok, dan bola basket. Kemudian untuk Jumlah parameter yang dilatih dalam model ini sebanyak 476.195 parameter. Ukuran gambar pada setiap layer konvolusi semakin berkurang. Ukuran gambar pada proses konvolusi terakhir sebelum masuk ke *fully connected layer* adalah 5x5 piksel. Sedangkan jumlah filter yang digunakan yaitu sebanyak 64. Kemudian dilakukan *reshape* sehingga ada 1600 neuron yang akan masuk pada *fully connected layer*. Jumlah neuron pada *hidden layer* yang digunakan yaitu sebanyak 256 neuron. Pada *fully connected layer* juga

ditambahkan proses *dropout* saat melakukan pelatihan, hingga pada akhirnya dilakukan klasifikasi sebanyak 3 kategori.

Selanjutnya berdasarkan model yang terbentuk, dapat diketahui bagaimana hasil yang ditunjukkan terhadap data *training* dan *testing* yang digunakan. Dimana pada *fit* model disini menggunakan *epoch* sebesar 10 dan *batch size* sebesar 50.

Tabel 5.7 *Training Model*

```

81 # Fit model
82 history <- model %>%
83   fit(train,
84     trainLabels,
85     epochs = 10,
86     batch_size = 50,
87     validation_split = 0.2,
88     validation_data = list(test, testLabels))
89 plot(history)

```

Kemudian hasil dari *training* akan ditampilkan dari hasil *loss* dan *accuracy* kedua data tersebut ditunjukkan pada tabel 5.8 berikut.

Table 5.8 *Hasil akurasi data training dan testing*

Data	Jumlah Data	Nilai <i>Loss</i>	Nilai <i>Accuracy</i>
<i>Training</i>	100	0.06877642	0.98
Testing	50	0.746397	0.8133333

Berdasarkan tabel 5.8 di atas, nilai *loss* yang dihasilkan untuk data *training* yaitu sebesar 0.06877642. Nilai *loss* pada data *training* lebih rendah dibandingkan dengan data *testing*. Kedua nilai ini bisa dikatakan cukup rendah dan ini bagus dari model yang telah didapat. Hal ini juga didukung oleh tingginya nilai *accuracy*, yaitu mencapai 98% untuk data *training* dan 81,3% untuk data *testing*.

Ini menunjukkan bahwa model yang diperoleh mampu melakukan klasifikasi dengan baik. Pergerakan *loss* yang mendekati nilai nol atau rendah dan pergerakan *accuracy* yang terus meningkat menunjukkan hasil yang baik seiring dengan berjalannya *epoch*. Kondisi berhentinya proses pelatihan ditentukan dari banyaknya *epoch* yang digunakan, dalam hal ini yaitu sebanyak 10 epoch. Proses pembelajaran akan dihentikan jika sudah memenuhi kondisi tersebut.

Kemudian Kinerja dari *loss* dan *accuracy* model yang terbentuk, ditunjukkan melalui *plot* berikut ini.

Gambar 5.9 *Plot Model*

Dari plot pada gambar 5.9 diatas, dapat dilihat akurasi yang didapatkan, untuk titik berwarna merah menunjukkan akurasi dan *loss* data *training*, sedangkan warna biru menunjukkan *accuracy* data *validation*. Dari plot dapat dilihat bahwa akurasi dari kedua data cenderung konstan. Akurasi akhir yang didapat untuk data *train* sebesar 0,98 dan data *test* sebesar 0,8133333. Ini merupakan hasil yang cukup baik apabila dilihat dari jumlah data yang digunakan yaitu 450 data gambar. Posisi gambar juga sangat mempengaruhi akurasi apabila data sedikit.

5.4. Evaluasi dan Prediksi Hasil *Train* dan *Test*

Kemudian untuk melihat hasil prediksi dari data *training* dan data *testing* akan ditunjukkan melalui tabel *confusion matrix* seperti dibawah ini.

Tabel 5.9 *Hasil Klasifikasi data Training*

		Actual		
		Bola Kaki	Bola Kok	Bola Basket
Prediksi	Bola Kaki	98	3	0
	Bola Kok	1	96	0
	Bola Basket	1	1	100

Tabel 5.10 *Hasil Klasifikasi data testing*

		Actual		
		Bola Kaki	Bola Kok	Bola Basket

Prediksi	Bola Kaki	35	9	0
	Bola Kok	11	37	0
	Bola Basket	4	4	50

Dari tabel 5.9 diatas, didapatkan hasil prediksi untuk data *training*, dimana gambar bola kaki yang berhasil diklasifikasi adalah sebanyak 98 dari 100 gambar yang digunakan, artinya dari 100 data gambar yang dimasukan hanya terdapat 98 data gambar yang berhasil diklasifikasi sedangkan yang sisanya tidak dapat diklasifikasi. Kemudian untuk gambar bola kok yang berhasil diklasifikasi adalah sebanyak 96 dari 100 gambar yang digunakan, dan gambar bola basket semuanya berhasil diklasifikasi. Kemudian untuk tabel 5.10 diatas, didapatkan hasil prediksi untuk data *testing*, dimana gambar bola kaki yang berhasil diklasifikasi adalah sebanyak 35 dari 50 gambar yang digunakan, gambar bola kok yang berhasil diklasifikasi adalah sebanyak 37 dari 50 gambar yang digunakan, dan gambar bola basket semuanya berhasil diklasifikasi

Dari hasil model diatas dapat dilihat jika penerapan metode ini untuk klasifikasi cukup baik sehingga penggunaan untuk mengenal gambar *sport* sebagai alat membantu promosi dan agar selalu mengingat alat tersebut cukup baik. Sehingga penelitian ini bisa digunakan untuk membantu mengingat berbagai jenis-jenis cabang olahraga yang sangat populer di Indonesia bahkan dunia khususnya bola kaki, bola kok, dan bola basket.

Kemudian setelah didapatkan model, peneliti ingin menguji model menggunakan data baru. Gambar yang digunakan berjumlah 30 gambar dengan 10 gambar dari masing-masing kategori (bola kaki, bola kok, dan bola basket). Gambar baru yang digunakan harus melalui proses *preprocessing* terlebih dahulu diantaranya *meresize*. Kemudian memasukkan gambar kedalam model yang telah dibuat, sehingga didapatkan hasil dibawah ini.

Tabel 5.11 Hasil Klasifikasi data baru

		Actual		
		Bola Kaki	Bola Kok	Bola Basket
Prediksi	Bola Kaki	9	0	0
	Bola Kok	1	9	0

Bola Basket	0	1	10
-------------	---	---	----

Berdasarkan tabel 5.11 diatas merupakan hasil prediksi data baru, dimana dapat dilihat prediksi yang didapat bahwa dari 30 data gambar yang dimasukan computer terdapat 2 data yang tidak berhasil diklasifikasi dan yang lain telah berhasil diklasifikasi.

Dari penelitian yang dilakukan, didapatkan bahwa hasil akurasi untuk data *training* dan *testing*, maupun uji coba dengan data baru menghasilkan akurasi yang tinggi, maka dapat dikatakan bahwa metode ini baik untuk melakukan klasifikasi gambar olahraga. Sehingga, dengan adanya penelitian ini sehingga bisa menjadi alternatif lain untuk melakukan klasifikasi gambar olahraga selain menggunakan cara manual.

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Untuk melakukan proses pengunduhan gambar pada *google image*, dibutuhkan bantuan beberapa *tools* seperti *fatkun batch*. Sehingga didapatkan gambar bola kaki, bola kok, dan bola basket dengan total 450 gambar yang telah dipilih oleh peneliti.
2. Penerapan *Convolution Neural Network* (CNN) pada klasifikasi gambar *sport* dimana kasus ini menggunakan gambar bola kaki, bola kok, dan bola basket. Dari hasil penelitian didapatkan bahwa arsitektur CNN yang digunakan menggunakan empat *layer convolution*, dengan beberapa *activation function*. *Software* yang digunakan Rstudio menggunakan *package keras*.
3. Tingkat akurasi model *Convolutional Neural Network* yang digunakan pada klasifikasi gambar bola kaki, bola kok, dan bola basket adalah 98% untuk data *training*, sedangkan 81,3% untuk data *testing*. Hasil ini terbilang cukup baik mengingat data yang digunakan terbatas yaitu total 450 gambar.
4. Hasil klasifikasi dari data baru menggunakan model *Convolutional Neural Network* dengan 30 gambar, 10 gambar masing-masing kategori diperoleh hasil adalah tidak terdapat kesalahan dan masing-masing kategori berada pada tempatnya.
5. Dengan tingkat akurasi yang cukup baik maka bisa dikatakan program ini bisa membantu dalam proses pengenalan dan mengingat kembali jenis-jenis gambar *sport*.

6.2. Saran

Berdasarkan kesimpulan di atas, peneliti dapat diberikan beberapa saran sebagai berikut:

1. Diharapkan gambar yang digunakan hanya memuat objek yang diteliti sehingga membuat akurasi semakin tinggi.
2. Diharapkan data yang digunakan pada penelitian selanjutnya lebih banyak dengan kategori yang lebih banyak juga agar akurasi lebih tinggi.
3. Spesifikasi komputer juga diharapkan lebih tinggi agar mampu melatih model dengan gambar yang lebih banyak.
4. Untuk penelitian selanjutnya diharapkan menggunakan perbandingan dari beberapa percobaan dengan melakukan perubahan pada model.

DAFTAR PUSTAKA

- A'la, F. Y. 2016. *Deteksi Retak Permukaan Jalan Raya Berbasis Pengolahan Citra Menggunakan Metode Ekstraksi Ciri Wavelet*. Yogyakarta: Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Yogyakarta.
- Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. 2012. *Imagenet Classification with Deep Convolutional Neural Networks*. International Journal of Engineering and Technology (IJET).
- Alpaydin, E., 2009. *Introduction to Machine Learning*, Second Edition. London: MIT Press.
- Danukusumo, Kevin Pudi. 2017. *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU*. Tugas Akhir. Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.
- Cholik, Mutoir. 2004. *Dasar-Dasar Olahraga*. Diakses pada tanggal 07 Februari 2018 pukul 22.30 WIB dari <http://olahragapendidikan.blogspot.com/2010/11/pembinaan-dan-pengembangan-olahraga.html>
- Erlangung, Zur. 2017. *Face Image Analysis with Convolutional Neural Networks*. Disertasi, Breisgan, Universitas Freiburg.
- Erna, Dwi E. 2009. *Pengantar Jaringan Saraf Tiruan*. Jawa Tengah: Star Publishing.
- Fikrie, abdillah. 2016. "Penggunaan Deep Learning Untuk Prediksi Churn Pada Jaringan Telekomunikasi Mobile". Ilmu Komputer, Universitas Telkom, Bandung.
- Grupe & Kruger. 1994. *Automatic Fruit Grading and Classification System Using Computer Vision: A Review*, International Conference on Advances in Computing and Communication Engineering, (I), pp.598–603.

- Gummeson, Anna. 2016. *Prostate Cancer Clasification Using Convolutional Neural*. Thesis, Lund, Swedia, Universitas Lund.
- Han, J. and Kamber, M. 2006. *Data Mining: Concepts and Techniques Second Edition*. San Francisco: Morgan Kauffman.
- He, Zhang, Ren, & Sun. *Flexible High Performance Convolutional Neural Networks for Image Classification*. 2012.
- Hermawati, F. A. 2013. *Pengolahan Citra Digital: Konsep dan Teori*. Yogyakarta: Andi.
- Hidayatullah, Aisyah. 2013. *Identifikasi Tingkat Kematangan Buah Tomat (Lycopersicon Esculentum Mill) Menggunakan Metode Pengolahan Citra dan Jaringan Saraf Tiruan*. Tugas Akhir. Jurusan Teknologi Industri Pertanian Fakultas Teknologi Pertanian Universitas Gadjah Mada.
- Jessica, D. 2011. *Aplikasi dunia olahraga*. Diakses pada tanggal 07 Februari 2018 pukul 22.30 WIB dari <https://jalantikus.com/tips/aplikasi-berita-olahraga/>
- Kanta, Imam Anggara. 2013. *Pengenalan Pola Huruf Hijaiyah Tulisan Tangan Menggunakan Logika Fuzzy Dengan JST Backpropagation*. Universitas Muhammadiyah. Surakarta
- Khoje, S. A., Bodhe, S. K., & Adsul, A. 2013. *Automated Skin Defect Identification System for Fruit Grading Based on Discrete Curvelet Transform*. International Journal of Engineering and Technology (IJET).
- LeCun, Y., Bengio, Y., & Hinton, G. 2015. *Deep Learning*. Nature, 521 (7533), 436-444.
- Mathworks. 2017. *Convolutional Neural Network*. Diakses pada tanggal 07 Februari 2018 pukul 22.30 WIB dari <https://www.mathworks.com/discovery/convolutional-neural-network.html>
- Matthew D. Zeiler & Rob Fergus. 2014. "Education Aspects: Handwriting Recognition-Neural Networks-Fuzzy Logic". *Proceedings of the IAPR International Conference on Pattern Recognition and Information Processing-PRIP'97*, vol. 1, 1997, pp.39-47.

- Ocher, R., Huval, B., Bath, B., Manning, C. D., & Ng, A. Y. 2012. *Convolutional-Recursive Deep Learning for 3D Object Classification*. In *Advances in Neural Information Processing Systems* (pp. 656-664).
- Putra, W. S. E. 2016. *Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101*. *Jurnal Teknik ITS*, 5(1).
- Ranjit, K. N., Chethan, H. K., & Naveena, C. 2016. *Identification and Classification of Fruit Diseases*. *International Journal of Engineering Research and Application (IJERA)*, pp.11 – 14).
- Rismiyati. 2016. *Implementasi Convolution Neural Network Untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital*. Thesis, Yogyakarta, Indonesia, Universitas Gajah Mada.
- Ryo Akita, Akira Yoshihara, Takashi Mutsubara, Kuniaki Uehara. 2016. “*Deep Learning for Stock Prediction Using Numrical And Textual Information*”, IEEE.
- Seema, Kumar, A. dan Gill, G.S. 2015. *Automatic Fruit Grading and Classification System Using Computer Vision: A Review, International Conference on Advances in Computing and Communication Engineering*, (I), pp.598–603.
- Sianipar, R. 2013. *Pemograman MATLAB dalam Contoh dan Terapan*. Bandung: Penerbit Informatika.
- Szegedy, Christian. 2014. “*Pengenalan huruf tulisan tangan Menggunakan Fuzzy Feature Extraction dengan Pendekatan Radial Basis Function Neural Network*”. Bengkulu.
- Tan, et al. 2006. *Introduction to Data Mining*. USA: Addison-Wesley.
- Tsai, C. 2016. *Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks*. Stanford University, Department of Chemical Engineering. California: Stanford University.
- Turban, E., et.al. 2005. *Decision Support System and Intelligent System*. Upper Saddle River, New Jersey USA: Prentice Hall
- Wikipedia. 2017. *Sport*. Diakses pada tanggal 07 Februari 2018 pukul 22.30 WIB dari <https://id.wikipedia.org/wiki/Olahraga>

- Yudha, Ancemona. 2014. "*Pengenalan huruf tulisan tangan Menggunakan Fuzzy Feature Extraction dengan Pendekatan Radial Basis Function Neural Network*". Bengkulu.
- Zufar, Muhammad., et.al. 2016. *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time*. Jurnal Sains dan Seni ITS Vol. 5 No. 2 (2016) 2337-3520 (2301-928X Print).

LAMPIRAN

Lampiran 1 Script R Persiapan Gambar

```
#Convolutional Neural Networks

#Packages

library(EBImage)

library(keras)

setwd('D://data150')

# Read Images

images <- list.files()

images

summary(images)

list_of_images = lapply(images, readImage )

list_of_images

display(list_of_images[[10]])

print(list_of_images[[10]])

#create train

train <- list_of_images[c(1:100,151:250,301:400)]

str(train)

train

display(train[[10]])

#create test

test <- list_of_images[c(101:150,251:300,401:450)]

test

display(test[[10]])

#Plot

par(mfrow = c(30, 10))

for (i in 1:300) plot(train[[i]])
```



```

# Resize & combine
#Resize
for (i in 1:300) {train[[i]] <- resize(train[[i]], 32, 32)}
str(train)

for (i in 1:150) {test[[i]] <- resize(test[[i]], 32, 32)}
str(train)

#COMBINE
train <- combine(train)
str(train)
x <- tile(train, 30)
display(x, title = 'Pictures')

test <- combine(test)
y <- tile(test, 15)
display(y, title = 'Pics')

# Reorder dimension
train <- aperm(train, c(4, 1, 2, 3))
str(train)
test <- aperm(test, c(4, 1, 2, 3))
str(test)

# Response
trainy <- c(rep(0,100), rep(1,100), rep(2,100))
testy <- c(rep(0,50), rep(1,50), rep(2,50))

# One hot encoding
trainLabels <- to_categorical(trainy)
testLabels <- to_categorical(testy)
str(train)
dim(train)

```

Lampiran 2 Script R Pembuatan dan Training Model

```
# Model
model <- keras_model_sequential()

model %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu',
                input_shape = c(32, 32, 3)) %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.01) %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.01) %>%
  layer_flatten() %>%
  layer_dense(units = 256, activation = 'relu') %>%
  layer_dropout(rate=0.01) %>%
  layer_dense(units = 3, activation = 'softmax') %>%

compile(loss = 'categorical_crossentropy',
        optimizer = optimizer_sgd(lr = 0.01,
                                   decay = 1e-6,
```

```
momentum = 0.9,  
nesterov = T),  
    metrics = c('accuracy'))  
summary(model)  
# Fit model  
history <- model %>%  
  fit(train,  
      trainLabels,  
      epochs = 10,  
      batch_size = 50,  
      validation_split = 0.2,  
      validation_data = list(test, testLabels))  
plot(history)
```

Lampiran 3 Script R Evaluate and Prediction

```
# Evaluation & Prediction - train data
model %>% evaluate(train, trainLabels)
pred <- model %>% predict_classes(train)
table(Predicted = pred, Actual = trainy)

prob <- model %>% predict_proba(train)
cbind(prob, Predicted_class = pred, Actual = trainy)

# Evaluation & Prediction - test data
model %>% evaluate(test, testLabels)
pred <- model %>% predict_classes(test)
table(Predicted = pred, Actual = testy)

prob <- model %>% predict_proba(test)
cbind(prob, Predicted_class = pred, Actual = testy)

#save model

save_model_weights_hdf5(model,filepath='D://hasildata.hdf5',overwrite=TRUE)

model=load_model_weights_hdf5(model,filepath="D://hasildata.hdf5",
by_name=FALSE)

summary(model)

#####
#####very fixxx
```

Lampiran 4 Script R pembuatan data Baru

```
setwd('D://databaru150')
images <- list.files()
images
summary(images)
img = lapply(images, readImage )
img

# Get the image as a matrix
for (i in 1:30) {img[[i]] <- resize(img[[i]], 32, 32)}
for (i in 1:30) {img[[i]] <- toRGB(img[[i]])}
fixx <- combine(img)
y <- tile(fixx, 10)
display(y, title = 'Pics')
str(fixx)
got <- aperm(fixx, c(4, 1, 2, 3))
str(got)
testy <- c(rep(0,10),rep(1,10),rep(2,10))

# One hot encoding
testLabels <- to_categorical(testy)
pred <- model %>% predict_classes(got)
table(Predicted = pred, Actual = testy)
prob <- model %>% predict_proba(got)
cbind(prob, Predicted_class = pred, Actual = testy)
```