

SKRIPSI

**PERBANDINGAN KINERJA KLASIFIKASI CNN
BERDASARKAN STRATEGI *SPLIT* DATA PADA
BERAGAM *DATASET* CITRA**

Disusun dan diajukan oleh :

AHMAD ALI WINANDAR KADIR

H071171001



**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
OKTOBER 2021**

**PERBANDINGAN KINERJA KLASIFIKASI CNN
BERDASARKAN STRATEGI *SPLIT* DATA PADA
BERAGAM *DATASET* CITRA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer pada Program Studi Sistem Informasi Departemen
Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas
Hasanuddin

**AHMAD ALI WINANDAR KADIR
H071171001**

PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

OKTOBER 2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Ahmad Ali Winandar Kadir
NIM : H071171001
Program Studi : Sistem Informasi
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**Perbandingan Kinerja Klasifikasi CNN Berdasarkan Strategi *Split Data*
Pada Beragam *Dataset Citra***

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 15 Oktober 2021

Yang menyatakan,




Ahmad Ali Winandar Kadir

NIM. H071171001

**PERBANDINGAN KINERJA KLASIFIKASI CNN
BERDASARKAN STRATEGI *SPLIT* DATA PADA
BERAGAM *DATASET* CITRA**

Disusun dan diajukan oleh

AHMAD ALI WINANDAR KADIR

H071171001

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal 15 Oktober 2021 dan dinyatakan telah memenuhi syarat kelulusan.

UNIVERSITAS HASANUDDIN

Menyetujui,

Pembimbing Utama



Dr. Eng. Armin Lawi, S.Si., M.Eng.
NIP. 19720423 199512 1 001

Pembimbing Pertama



A. Muh. Amil Siddik, S.Si., M.Si.
NIP. 19911003 201903 1015

Ketua Program Studi,



Dr. Muhammad Hasbi, M.Sc.
NIP. 19630720 198903 1 003



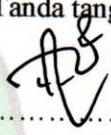
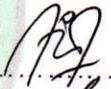
HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Ahmad Ali Winandar Kadir
NIM : H071171001
Program Studi : Sistem Informasi
Judul Skripsi : Perbandingan Kinerja Klasifikasi CNN Berdasarkan Strategi *Split Data* Pada Beragam Dataset Citra

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Ketua	: Dr. Eng. Armin Lawi, S.Si., M.Eng.	Tanda tangan  (.....)
Sekretaris	: A. Muh. Amil Siddik, S.Si. M.Si.	 (.....)
Anggota	: Edy Saputra, S.Si., M.Si.	 (.....)
Anggota	: Andi Muhammad Anwar, S.Si., M.Si.	 (.....)

Ditetapkan di : Makassar
Tanggal : 15 Oktober 2021



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan pendidikan jenjang Strata 1 pada Program Studi Sistem Informasi, Universitas Hasanuddin. Tugas akhir ini disusun guna memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer (S.Kom.).

Penulis menyadari bahwa telah mendapatkan banyak bantuan serta bimbingan dari berbagai pihak, dari masa perkuliahan hingga pada tahap akhir penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini, dengan segala kerendahan hati penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Orang tua penulis, ayahanda **Abdul Kadir** dan ibunda **Tahirah** yang tidak henti-hentinya menyempikan nama anaknya di dalam doa kebaikan, adik-adik, tekhusus kepada Kakak tercinta **Dian Fahirah Kadir** yang tidak pernah lelah membantu dan memotivasi penulis hingga berada di titik yang luar biasa ini.
2. Pemerintah melalui Direktorat Jendral Pembelajaran dan Kemahasiswaan (Ditjen Belmawa), Kementerian Riset, Teknologi dan Perguruan Tinggi (Kemenristekdikti) yang telah membantu pembiayaan selama perkuliahan melalui program **Basiswa Bidikmisi**. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubulu, MA.** beserta jajarannya, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Eng. Amiruddin, S.Si.** beserta jajarannya, dan Ketua Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Muhammad Hasbi, M.Sc.** beserta seluruh jajarannya. Tak lupa pula Bapak Wakil Dekan Bidang Kemahasiswaan **Dr. Andi Ilham Latunra, M.Si.** yang telah memberikan kesan dan banyak cerita selama penulis menjadi pengurus **BE Himatika FMIPA Unhas.**
3. Bapak **Dr. Armin Lawi, S.Si., M.Eng.** dan Bapak **A. Muh. Amil Siddik, S.Si. M.Si** atas kesediaan, kesabaran, dan waktu yang telah diluangkan untuk membimbing penulis selama proses penyusunan tugas akhir. Juga Bapak

4. **Edy Saputra, S.Si., M.Si.** dan bapak **Andi Muhammad Anwar, S.Si., M.Si.** atas waktu dan kesediannya sebagai penguji pada tahap penyusunan tugas akhir.
5. Dosen Departemen Matematika, dan terkhusus kepada Bapak dan Ibu dosen Program Studi Sistem Informasi, Fakultas MIPA, Universitas Hasanuddin atas semua ilmu yang diberikan kepada penulis selama menempuh Pendidikan.
6. Teman-teman **Gepeng+1** (**Nur Khairunisa** sebagai pengingat tugas dan teman pulang dari kampus, **Eka Kurnia** sebagai motivator hidup, **Geby Nionsi** sebagai Teman *healing*-ku, **Eka Fitriani** sebagai teman yang berbagiku, **Siti Rabiatul Adawiyah** selaku teman yang membantu dalam revisi, **Nurfadila Firdani Salam, Azzahra Mubarikah** yang selalu siap memfasilitasi rumahnya dijadikan tempat diskusi, **Fadhillah Putri Taha** yang tidak pernah berhenti memberikan arahannya, dan **Muthia Amanah Arum** serta **Mir Ataini Aprilia**), atas loyalitasnya sejak menjadi mahasiswa baru Ilmu Komputer hingga berganti nama menjadi Sistem Informasi, terima kasih sudah menjadi teman hidup penulis selama 4 tahun menjalani perkuliahan, berbagi suka dan duka selama menjadi mahasiswa hingga bukan mahasiswa lagi. Tak lupa pula teman-teman BRENG (**Arman S Maulana, Abudzar bin Mursalin, Muh. Ary Putra Ramadham, Rafly Ahmad Mubin, dan Muh. Ikhsan**), sebagai sirkel selama kuliah yang memiliki kekompakan sangat luar biasa dan dapat saling support. serta **Rigel Rivaldo Subiyakto** dan **Ilmi Kalam** yang telah banyak mengajari materi selama perkuliahan.
7. Segenap asisten lab yang menjadi pembimbing sekaligus panutan selama penulis menjadi mahasiswa. Dan teman-teman **Ilmu Komputer 2017** (berubah menjadi: Sistem Informasi awal tahun 2021) yang memberi sokongan dan kerjasamanya selama 4 tahun lebih di Universitas Hasanuddin. Terkhusus kepada **Muh. Arizky** dan **Khawaritzmi Abdallah Ahmad** yang senantiasa mewadahi kami menggunakan fasilitas-fasilitas yang menunjang dalam pembuatan tugas akhir ini dan juga **Fadil Hidayat Amin, Muhammad Fitrah, Muh. Naim, A. Amalia Dwi Ayu Sarjani M, Aris Akhyar Abdillah, Erwin Syahrul Hidayat** dan **Edo Bayu Pamungkas**, sebagai teman pengajar yang tabah dan tekun selama perkuliahan.

8. *Kaggle, Google, GitHub, Machine Learning Mastery*, dan *website-website* lainnya sebagai sumber pembelajaran yang menunjang penyelesaian tugas-tugas perkuliahan serta penyusunan tugas akhir ini. Tak lupa pula jurnal-jurnal yang berkaitan dengan penelitian ini sebagai pedoman penyusunan tugas akhir.

Makassar, 15 Oktober 2021



Ahmad Ali Winandar Kadir

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Ahmad Ali Winandar Kadir
NIM : H071171001
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

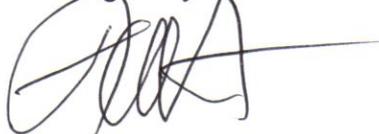
Perbandingan Kinerja Klasifikasi CNN Berdasarkan Strategi *Split Data* Pada Beragam *Dataset Citra*

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 15 Oktober 2021

Yang menyatakan



(Ahmad Ali Winandar Kadir)

ABSTRAK

Deep learning berkembang pesat beberapa tahun terakhir karena kemampuannya untuk mempelajari representasi data yang kemudian digunakan untuk deteksi atau klasifikasi. Salah satu model analisis *deep learning* dalam klasifikasi dataset citra dengan menggunakan algoritma *convolutional neural network*. pada penelitian ini, peneliti menggunakan menggunakan empat strategi split data yaitu *fixed train*, *fixed test*, *stratified* dan *random split*. Begitupun dengan karakteristik dataset digunakan yaitu *binary* dan *multiclass* serta seimbang maupun tidak seimbang. Hasil penelitian memperoleh nilai pada jenis *dataset binary* seimbang (*cat vs dog*) menghasilkan akurasi sebesar 89.20% sedangkan untuk *multiclass* seimbang (*african wildlife*) menghasilkan akurasi sebesar 77.10% dan pada *dataset multiclass* tidak seimbang (*garbage classification*) menghasilkan akurasi 80.70% sedangkan untuk *binary* tidak seimbang (*painting vs photograph*) metode *split* data yang cocok diterapkan yakni *fixed train split* menghasilkan akurasi sebesar 77.10%. menunjukkan bahwa evaluasi kinerja berdasarkan eksperimen yang telah dilakukan diperoleh bahwa metode *stratified split* sangat cocok diterapkan pada jenis data dengan karakteristik *binary* seimbang, *multiclass* seimbang dan *multiclass* tidak seimbang yang mendapatkan nilai maksimal dibanding dengan jenis *split* data lainnya dengan alasan bahwa *dataset* yang proporsional mempengaruhi hasil evaluasi kinerja yang akan diperoleh.

Kata Kunci—Gambar, Klasifikasi, Data Split : *Fixed Train*, *Fixed Test*, *Stratified* dan *Random Split*

ABSTRACT

Deep learning has grown rapidly in recent years because of its ability to study data representations which are then used for detection or classification. Convolutional Neural Network (CNN) is one of the algorithms from Deep Learning which is the development of Multi-Layer Perceptron (MLP) which is designed to process data in the form of a grid, one of which is a two-dimensional image, such as an image. In this study, researchers used four data split strategies, namely fixed train, fixed test, stratified and random split. as well as the characteristics of the dataset used, namely binary and multi-class as well as balanced or unbalanced. The results of the study obtained a value for the balanced binary dataset (cat vs dog) resulting in an accuracy of 89.20% while for balanced multi-class (African wildlife) an accuracy of 77.10% and an unbalanced multi-class dataset (garbage classification) an accuracy of 80.70%. binary unbalanced (painting vs photograph) a good data split method, namely the fixed train split produces an accuracy of 77.10%. It shows that the performance evaluation based on experiments that have been carried out shows that the stratified split method is very suitable to be applied to data types with balanced binary characteristics, multi class balanced and multi-class unbalanced which get the maximum value compared to other types of split data on the grounds that a proportional dataset affects the results of the performance evaluation that will be obtained.

Keywords—Image, Classification, Split Data: Fixed Train, Fixed Test, Stratified and Random Split

DAFTAR ISI

PERNYATAAN KEASLIAN.....	iii
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	ix
ABSTRAK	x
ABSTRACT	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait	6
2.2 Strategi <i>Data Splitting</i>	7
2.2.1 <i>Stratified Split</i>	7
2.2.2 <i>Random Split</i>	8
2.3 Tipe <i>Dataset</i>	8
2.4 Citra Digital.....	8
2.4.1 Citra Biner (Monokrom)	11

2.4.2	Citra Warna RGB	11
2.5	<i>Artificial Neural Network</i>	12
2.6	<i>Convolutional Neural Network</i>	13
2.6.1	<i>Convolution Layer</i>	14
2.6.2	<i>Stride</i>	19
2.6.3	<i>Padding</i>	20
2.6.4	<i>Pooling Layer</i>	21
2.6.5	Fungsi Aktivasi	22
2.6.6	<i>Flatten</i>	23
2.6.7	<i>Fully Connected Layer</i>	24
2.6.8	<i>Backpropagation</i>	24
2.7	Evaluasi Kinerja	26
2.8	<i>Categorical Crossentropy</i>	27
BAB III METODELOGI PENELITIAN		28
3.1	Waktu dan Tempat	28
3.2	<i>Dataset</i>	28
3.3	Instrumen Penelitian.....	29
3.4	Tahap Penelitian.....	29
3.5	<i>Pre-processing Data</i>	30
3.6	<i>Split Data</i>	31
3.7	<i>Training Model</i>	31
3.8	Evaluasi	34
BAB IV HASIL DAN PEMBAHASAN		35
4.1	Strategi data <i>Split</i>	35

4.1.1	Pembagian Data.....	35
4.1.2	<i>Splitting</i> Data.....	37
4.2	Implementasi Strategi <i>Split</i>	38
4.2.1	<i>Preprocessing</i> Data	38
4.2.2	Pembangunan Arsitektur	39
4.2.3	<i>Training</i> dan <i>Testing</i> Data.....	40
4.3	Evaluasi Kinerja	48
4.3.1	<i>Confusion Matrix</i>	48
4.3.2	<i>Accuracy</i>	53
4.3.3	<i>Precision</i>	53
4.3.4	<i>Recall</i>	55
4.3.5	<i>F1-Score</i>	57
BAB V KESIMPULAN DAN SARAN.....		60
5.1	Kesimpulan.....	60
5.2	Saran.....	60
Daftar Pustaka		61
LAMPIRAN		65

DAFTAR GAMBAR

Gambar 2.1 Aturan koordinat representasi citra digital	10
Gambar 2.2 Urutan pengolahan citra digital	10
Gambar 2.3 Arsitektur <i>Convolutional Neural Network</i>	13
Gambar 2.4 Operasi dot.....	15
Gambar 2.5 Proses Konvolusi 2 <i>filter</i> menghasilkan <i>Feature Map</i>	15
Gambar 2.6 Ilustrasi Operasi Konvolusi.....	17
Gambar 2.7 Ilustrasi operasi konvolusi pada citra RGB	18
Gambar 2.8 Ilustrasi operasi konvolusi pada citra <i>RGB</i>	18
Gambar 2.9 Konvolusi 1x1	19
Gambar 2.10 Ilustrasi Perhitungan Pergeseran <i>Stride</i> 1.....	19
Gambar 2.11 Ilustrasi Perhitungan dengan Pergeseran <i>Stride</i> 2	20
Gambar 2.12 Perhitungan dengan Penambahan <i>Padding</i>	21
Gambar 2.13 Ilustrasi Perhitungan pada <i>Pooling Layer</i>	21
Gambar 2.14 <i>Flatten</i>	23
Gambar 2.15 <i>Fully Connected Layer</i>	24
Gambar 3.1 Alur Penelitian.....	30
Gambar 3.2 <i>Input Layer</i>	31
Gambar 3.3 <i>Channel RGB</i> pada <i>input</i> citra.....	32
Gambar 3.4 <i>Input</i> channel merah	32
Gambar 3.5 <i>Filter</i> ukuran 5x5.....	32
Gambar 3.6 Contoh <i>fully connected layer</i> dari <i>dataset cat vs dog</i>	33

Gambar 4.1 ilustrasi dari <i>resize</i> citra <i>dataset cat</i>	38
Gambar 4.2 Ilustrasi dari <i>resize</i> citra <i>dataset Painting</i>	38
Gambar 4.3 Ilustrasi dari <i>resize</i> citra <i>dataset wildllife</i>	39
Gambar 4.4 Ilustrasi dari <i>resize</i> citra <i>dataset Garbage Classification</i>	39
Gambar 4.5 Plot <i>training accuracy</i> pada <i>dataset Dog vs Cat</i>	40
Gambar 4.6 Plot <i>loss</i> pada <i>dataset Dog vs Cat</i>	41
Gambar 4.7 Plot <i>testing performance</i> pada <i>dataset Dog vs Cat</i>	41
Gambar 4.8 Plot <i>accuracy</i> pada <i>dataset Painting vs Photograph</i>	42
Gambar 4. 9 Plot <i>loss</i> pada <i>dataset Painting vs Photograph</i>	43
Gambar 4.10 Plot <i>testing performance</i> pada <i>dataset Painting vs Photograph</i>	43
Gambar 4.11 Plot <i>training accuracy</i> pada <i>dataset African Wildlife</i>	44
Gambar 4.12 Plot <i>loss</i> pada <i>dataset African Wildlife</i>	45
Gambar 4.13 Plot <i>testing performance</i> pada <i>dataset African Wildlife</i>	45
Gambar 4.14 Plot <i>training accuracy</i> pada <i>dataset Garbage Classification</i>	46
Gambar 4.15 Plot <i>training loss</i> pada <i>dataset Garbage Classification</i>	46
Gambar 4.16 Plot <i>test performance</i> pada <i>dataset Garbage Classification</i>	47
Gambar 4.17 Plot <i>testing performance</i> setiap <i>split data</i>	48

DAFTAR TABEL

Tabel 2.1 <i>Stratified Split</i>	7
Tabel 2.2 <i>Random Split</i>	8
Tabel 2.3 <i>Intensitas Grayscale</i>	11
Tabel 2.4 <i>Kode Warna</i>	12
Tabel 4.1 <i>Pembagian Dataset 1 (Dog vs Cat)</i>	35
Tabel 4.2 <i>Pembagian Dataset 2 (Painting Vs Photograph Classification)</i>	36
Tabel 4.3 <i>Pembagian Dataset 3 (African Wildlife)</i>	36
Tabel 4.4 <i>Pembagian Dataset 4 (Garbage Images)</i>	36
Tabel 4.5 <i>Fixed train split</i>	37
Tabel 4.6 <i>Fixed test split</i>	38
Tabel 4.7 <i>Arsitektur CNN</i>	39
Tabel 4.8 <i>Confusion matrix pada dataset Cat vs Dog</i>	49
Tabel 4.9 <i>Confusion matrix pada dataset Painting vs Photograph</i>	49
Tabel 4.10 <i>Confusion matrix pada dataset African Wildlife</i>	50
Tabel 4.11 <i>Confusion matrix pada dataset Garbage Classification</i>	51
Tabel 4.12 <i>Perbandingan accuracy setiap strategi split pada data test</i>	53
Tabel 4.13 <i>Perbandingan precision pada dataset cat vs dog</i>	54
Tabel 4.14 <i>Perbandingan precision pada dataset Painting vs Photograph</i>	54
Tabel 4.15 <i>Perbandingan precision pada dataset African Wildlife</i>	54
Tabel 4.16 <i>Perbandingan precision pada dataset Garbage Classification</i>	55
Tabel 4.17 <i>Perbandingan recall pada dataset cat vs dog</i>	55
Tabel 4.18 <i>Perbandingan recall pada dataset Painting vs Photograph</i>	56

Tabel 4.19 Perbandingan <i>recall</i> pada <i>dataset African Wildlife</i>	56
Tabel 4.20 Perbandingan <i>recall</i> pada <i>dataset Garbage Classification</i>	56
Tabel 4.21 Perbandingan <i>f1-score</i> pada <i>dataset cat vs dog</i>	57
Tabel 4.22 Perbandingan <i>f1-score</i> pada <i>dataset Painting vs Photograph</i>	57
Tabel 4.23 Perbandingan <i>f1-score</i> pada <i>dataset African Wildlife</i>	58
Tabel 4.24 Perbandingan <i>f1-score</i> pada <i>dataset Garbage Classification</i>	58
Tabel 4.25 Perbandingan <i>f1-score</i> setiap strategi <i>split</i> pada data <i>test</i>	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, perkembangan yang pesat dalam sains dan teknologi telah berdampak pada pertumbuhan data mentah secara eksponensial (He & Garcia, *Learning from Imbalanced Data*, 2008). Berdasarkan *World Economic Forum* data digital di dunia akan mencapai 44 zettabyte, atau 44 triliun *gigabyte* pada tahun 2020. Jumlah ini akan terus berkembang hingga lebih dari dua kali lipat setiap dua tahunnya (Chen, Mao, & Liu, 2014).

Dengan jumlah data yang sangat banyak, tidak mungkin oleh manusia untuk menganalisis dan membuat perhitungan mengenai data secara manual. Maka dari itu diperlukan bidang khusus untuk mengolah dan menganalisis data. *Data Science* adalah bidang yang mempelajari bagaimana meng-ekstraksi *raw* data menjadi *meaningful information* atau informasi yang berguna (Berman, et al., 2018). *Data Science* melibatkan prinsip, proses, dan teknik untuk memahami fenomena melalui data (Provost & Fawcett, 2013). *Data Science* merupakan bidang yang sangat luas dan sedang dikembangkan (Provost & Fawcett, 2013), namun salah satu bidang khusus dari *data science* adalah *machine learning*. *Machine learning* membahas mengenai bagaimana membangun sistem komputer yang dapat belajar melalui pengalaman tanpa harus diprogram secara spesifik dan manual.

Untuk menyelesaikan suatu masalah pada komputer, dibutuhkan algoritma. Namun untuk beberapa masalah, tidak ada algoritma yang memadai. Contohnya adalah bagaimana komputer mengenali angka dalam bentuk tulisan tangan (Bishop, 1995) dan mengklasifikasi suatu email menjadi spam atau bukan spam (Ethem, 2009). Dalam masalah seperti ini, seluruh objek dengan label atau kelas berbeda dikumpulkan dan diidentifikasi ciri-ciri yang membedakan mereka. Di dalam hal ini lah *machine learning* bekerja. *Artificial Neural Network* (ANN) merupakan algoritma klasifikasi yang terinspirasi dari sistem jaringan saraf makhluk hidup. Jaringan tersebut terdiri dari *neuron* yang sangat banyak dan saling terhubung

(Umam, 2018). Perkembangan teknologi *Artificial Intelligence* atau kecerdasan buatan sudah mengalami kemajuan yang sangat pesat. Kemajuan teknologi ini didorong melalui penerapannya dalam kehidupan sehari-hari. Konsep perkembangan *Artificial Intelligence* sendiri ada dua yaitu *machine learning* dan algoritma *deep learning*.

Kemampuan algoritma CNN diklaim sebagai model terbaik untuk memecahkan permasalahan *object recognition* dan *object detection*, hal ini dapat dibuktikan berdasarkan hasil penelitian terdahulu. Penelitian tersebut antara lain (Wirtjes, PENGENALAN EKSPRESI WAJAH MENGGUNAKAN CONVOLUTIONAL , 2019) Pengenalan Ekspresi Wajah Menggunakan *Convolutional Neural Network* (CNN) dengan akurasi sebesar 80% dari 100 gambar dan (Nurhikmat, Implementasi Deep Learning Untuk Image Classification menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek, 2018) Implementasi Deep Learning Untuk Image Classification menggunakan Algoritma *Convolutional Neural Network* (CNN) Pada Citra Wayang Golek sebesar 93 %.

Namun dalam CNN, seperti model *Deep Learning* lainnya memiliki kelemahan yaitu proses pelatihan model cukup lama. Selain itu, masalah yang paling sering muncul adalah kurangnya jumlah data pelatihan yang cukup atau keseimbangan kelas data yang tidak merata dalam *dataset*, hal ini dapat disebabkan dari beberapa faktor yaitu sulitnya untuk memperoleh data yang akan dijadikan sebagai *dataset*. Sebagaimana yang telah diketahui bahwa lebih banyak data yang diakses akan lebih efektif hasil akurasi. Menurut (Sun, 2011) dalam penelitiannya menyebutkan bahwa kinerja CNN secara logaritma sebanding dengan jumlah data pelatihan.

Untuk distribusi data pada suatu dataset, terdapat istilah kelas yang terdistribusi secara seimbang (*balanced*) dan secara tak seimbang (*imbalanced*). Dataset dengan kelas yang seimbang berarti jumlah observasi untuk setiap kelas tidak jauh dari kelas-kelas yang lain (Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2011).

Selain itu, ada pula penelitian yang dilakukan untuk mengatasi masalah data kelas tidak seimbang, seperti yang dilakukan oleh Zhou dan Liu menunjukkan bahwa menyelesaikan masalah *multiclass* lebih sulit dari dua *class*. Penelitian lain juga menunjukkan bahwa algoritma standar tidak bekerja pada data kelas tidak seimbang atau masalah data *multiclass*, sehingga menggunakan metode tradisional kurang tepat menangani masalah ini, apabila dipaksakan dapat menimbulkan prediksi bias dan hasil akurasi yang menyesatkan.

(Ilham, Komparasi Algoritma Klasifikasi dengan Pendekatan Level Data Untuk Menangani Data Kelas Yang Tidak Seimbang, 2017) dengan penelitiannya berjudul Komparasi Algoritma Klasifikasi Dengan Pendekatan Level Data Untuk Menangani Data Kelas Tidak Seimbang. Penelitian ini menggunakan teknik *under-sampling* dan *over-sampling* untuk menangani data kelas tidak seimbang,

Pemisahan data merupakan pertimbangan penting selama pengembangan jaringan saraf tiruan (*ANN*) di mana validasi biasanya digunakan untuk memastikan generalisasi. Bahkan untuk ukuran sampel sedang, metodologi pengambilan sampel yang digunakan untuk pemisahan data dapat memiliki pengaruh yang signifikan terhadap kualitas himpunan bagian yang digunakan untuk pelatihan dan pengujian *ANN*. Pemisahan data yang buruk dapat menghasilkan kinerja model yang tidak akurat dan sangat bervariasi; namun, pilihan metodologi pengambilan sampel jarang dipertimbangkan oleh pemodel *ANN*. Peningkatan kepercayaan dalam pengambilan sampel sangat penting, karena pengambilan sampel yang ditunda umumnya dilakukan hanya sekali selama pengembangan *ANN* (R J May, 2009) sehingga latar belakang pada penelitian ini yaitu pada strategi data splitting, strateginya hanya membagi data 80% 20% dan tidak ditetapkan apakah *fixed train*, *fixed test*, *stratified* maupun *random split* sehingga ini menjadi suatu pertimbangan dalam penelitian yang saya lakukan.

Berdasarkan latar belakang diatas, peneliti mengajukan penelitian berjudul **“Perbandingan Kinerja Klasifikasi CNN Berdasarkan Metode *Split Data* pada Beragam *Dataset Citra*”**, dimana CNN digunakan untuk melatih data untuk

mengenali beragam jenis *dataset* yang akan dilatih. Kemudian model dievaluasi dengan menghitung *accuracy*, *presisi*, *recall*, *f1-score*.

1.2 Rumusan Masalah

Rumusan masalah yang akan dipecahkan melalui penelitian ini pada dasarnya tidak lepas dari ruang lingkup latar belakang yang telah dijelaskan, yaitu:

1. Bagaimana mengimplementasikan dan merancang strategi *split* data pada klasifikasi CNN?
2. Bagaimana mengimplementasi metode klasifikasi CNN pada beberapa *dataset* berdasarkan strategi data *split*?
3. Bagaimana evaluasi kinerja metode klasifikasi CNN pada beberapa *dataset* berdasarkan metode data *split*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Menyajikan metode strategi *split* data pada klasifikasi CNN.
2. Memberikan rekomendasi *strategi split data* yang tepat berdasarkan evaluasi kinerja hasil percobaan klasifikasi CNN dari beberapa *dataset*.
3. Menyajikan pengolahan metode klasifikasi CNN beserta hasilnya berdasarkan implementasi strategi *data split*.

1.4 Batasan Masalah

Mengingat ruang lingkup permasalahan begitu luas, maka pada penelitian ini terdapat batasan masalah agar pembahasan yang dilakukan dapat terfokus. Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan oleh peneliti merupakan data dari *Kaggle*.
2. *Dataset* yang diteliti merupakan *dataset* citra yaitu *Cat-vs-dog*, *Painting-vs-photograph*, *African Wildlife*, dan *Garbage Images* dengan format.jpg.
3. Model pembelajaran yang digunakan untuk identifikasi ragam *dataset* adalah algoritma *Convolutional Neural Network*.
4. Penelitian ini fokus pada strategi *split data* yang memberikan rekomendasi suatu jenis strategi *split data* cocok pada karakteristik suatu dataset.

1.5 Manfaat Penelitian

Mampu menemukan rekomendasi metode *data split* sehingga dapat digunakan untuk mendapatkan kinerja model yang tepat dan menerapkan sebuah sistem yang dapat mengklasifikasi dari beberapa *dataset* yang diuji yang bertujuan mengukur kemampuan sebuah komputer dalam membedakan sebuah objek. Penelitian ini diharapkan juga dapat menjadi sebuah referensi yang dapat digunakan untuk penelitian yang lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Melalui hasil dari studi literatur maka telah ditemukan beberapa rujukan utama dari penelitian ini. Dalam rujukan terdapat beberapa penelitian yang memiliki studi kasus yang sama yaitu identifikasi jenis hewan dengan menggunakan algoritma *Deep Learning*.

Joceline S Wirtjes dalam penelitiannya melakukan pengenalan ekspresi wajah menggunakan metode *Convolutional Neural Network* (CNN). Tahapan keseluruhan metode yang digunakan adalah tahap *pre-processing*, dan tahap klasifikasi. Tahap *pre-processing* yang dilakukan pada penelitian ini adalah *resize* dan *grayscale*. Untuk melakukan pengenalan ekspresi wajah, jaringan CNN dilatih dengan data citra ekspresi. Proses *training* dilakukan dengan menggunakan *optimizer*, *batch size*, dan *epoch* yang berbeda-beda untuk mendapatkan model terbaik. Untuk mengatasi *overfitting*, dilakukan data augmentation pada data latih dan data validasi. Hasil percobaan pada penelitian ini menunjukkan bahwa metode *Convolutional Neural Network* dapat mengenali ekspresi wajah dengan tingkat akurasi dari pengujian sebesar 80% dari 100 gambar (Wirtjes, 2019).

Nurhikmat pada tahun 2018 melakukan melakukan pendekatan dalam pengenalan suatu gambar menggunakan metode *Convolutional Neural Network*. Metode ini salah satu metode *Deep Learning* yang dapat digunakan untuk mengenali dan mengklasifikasi sebuah objek pada sebuah citra digital. Berdasarkan hasil pembahasan didapatkan tingkat akurasi sebesar 95% pada proses *training* dan 90 % pada proses *testing*. Kemudian penelitian ini menggunakan data baru untuk menguji model yang telah dibuat. Tingkat akurasi yang dihasilkan menggunakan data baru sebesar 93% dalam mengklasifikasikan gambar wayang golek (Nurhikmat, 2018) .

Ilham 2017 pada penelitiannya penulis menangani masalah data kelas tidak seimbang dapat menimbulkan efek buruk pada metode klasifikasi untuk ketepatan

prediksi pada data. Untuk menangani masalah ini, telah banyak penelitian sebelumnya menggunakan algoritma klasifikasi menangani masalah data kelas tidak seimbang. Pada penelitian ini akan menyajikan teknik *under-sampling* dan *over-sampling* untuk menangani data kelas tidak seimbang. Teknik ini akan digunakan pada tingkat preprocessing untuk menyeimbangkan kondisi kelas pada data. Hasil eksperimen menunjukkan *neural network* (NN) lebih unggul dari *decision tree* (DT), *linear regression* (LR), *naïve bayes* (NB) dan *support vector machine* (SVM) (Ilham, 2017).

2.2 Strategi Data Splitting

Data Splitting adalah suatu aksi untuk mempartisi data menjadi dua bagian atau *subset*. Bagian data pertama digunakan untuk mengembangkan model, biasa disebut *Data Train*, bagian kedua digunakan untuk mengevaluasi performa model, atau biasa disebut *Data Test*. (Picard & Berk, 1990). Dalam *Machine Learning* terdapat banyak strategi untuk membagi data, strategi-strategi ini berdasarkan bagaimana membagi *Data Train* dan *Data Test* terhadap kelas *dataset*. Berbagai strategi memerhatikan pembagian setiap kelas pada *data train* dan memberikan sisanya kepada *data test*, dan hal yang sama juga berlaku untuk kebalikannya.

2.2.1 Stratified Split

Pada strategi *Stratified Split*, setiap *subset* data dibagi berdasarkan rasio r untuk *data train* dan $1 - r$ untuk *data test* pada setiap kelas di *dataset D*. Hal ini disajikan dalam tabel 2.1

Tabel 2.1 Stratified Split

Subset	C1	C2	C3	Total
All	100	200	300	600
Train	80	160	240	480
Test	20	40	60	120

2.2.2 *Random Split*

Pada strategi *Random Split*, setiap *subset* data dibagi berdasarkan rasio r untuk *data train* dan $1 - r$ untuk *data test* di *dataset D*, tanpa memerhatikan jumlah setiap kelas. Hal ini disajikan dalam tabel 2.2

Tabel 2.2 *Random Split*

Subset	C1	C2	C3	Total
All	100	200	300	600
Train	83	145	252	480
Test	17	55	48	120

2.3 *Tipe Dataset*

Pada penelitian ini terdapat beberapa tipe *dataset* yang digunakan, yaitu terdiri atas:

1. *Dataset binary* memiliki satu atribut kelas yang terdiri dari dua nilai. Nilai dari suatu kelas dapat direpresentasikan dengan positif atau negatif, 0 atau 1, *true* atau *false*, dan lain-lain. *Dataset binary* terbagi menjadi dua yakni; *dataset binary* seimbang. *Dataset* yang digunakan dalam penelitian ini adalah kumpulan data berisi dua kelas gambar *cat vs dog* sedangkan, *dataset binary* tidak seimbang. *Dataset* yang digunakan dalam penelitian untuk mengklasifikasikan yakni *painting vs photograph*.
2. *Dataset multiclass* merupakan *dataset* yang memiliki lebih dari dua kelas. *Dataset binary* terbagi menjadi dua yakni *dataset multiclass* seimbang yang digunakan dalam penelitian ini yakni *African Wildlife* dan *dataset multiclass* tidak seimbang yang digunakan yakni *Garbage Classification Dataset*.

2.4 *Citra Digital*

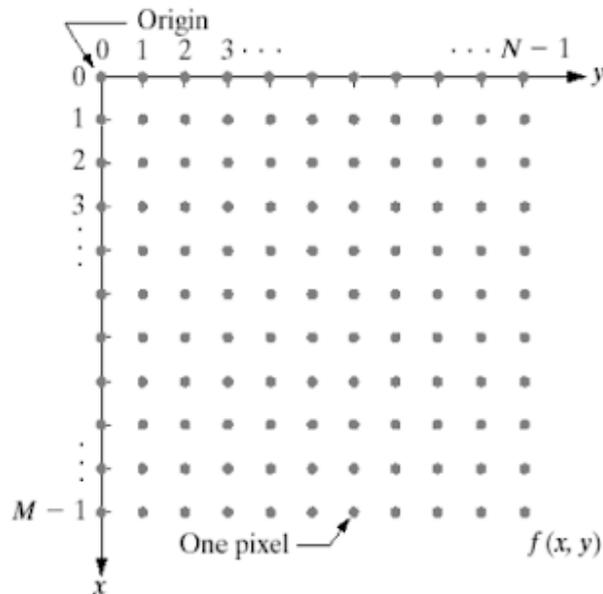
Citra (*image*) adalah kombinasi antara titik, garis, bidang dan warna untuk menciptakan suatu imitasi dari suatu objek, biasanya objek fisik atau manusia. Citra bisa berwujud gambar (*picture*) dua dimensi, seperti lukisan, foto dan berwujud tiga dimensi, seperti patung. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada

citra yang bersifat digital. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Citra analog harus dikonversi menjadi citra digital terlebih dahulu agar dapat diproses di komputer (Sutojo, 2017).

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil *CT-scan*, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan di dalam komputer, oleh karena itu citra analog dapat dikonversi ke dalam citra digital agar nantinya dapat dibaca dan diproses oleh komputer.

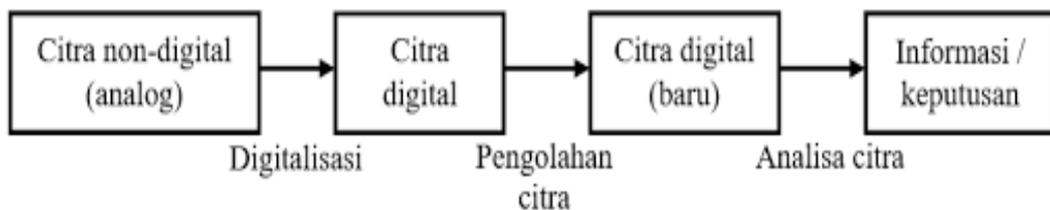
Tujuan pengolahan citra antara lain dapat memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik). Selain itu pengolahan citra juga dilakukan untuk proses penarikan informasi atau deskripsi objek atau juga untuk pengenalan objek yang terkandung pada citra tersebut (Hermawati, 2013).

Sebuah citra dapat didefinisikan sebagai fungsi $f(x, y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x , y dan nilai amplitudo f secara keseluruhan berhingga dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital (Putra, 2010). Gambar 2.1 menunjukkan posisi koordinat citra digital.



Gambar 2.1 Aturan koordinat representasi citra digital

Gambar 2.1 merupakan gambaran aturan koordinat citra digital. Nilai suatu irisan antara baris dan kolom (pada posisi x, y) disebut dengan *picture elements, image elements, pels* atau *pixel*. Istilah yang paling banyak digunakan pada citra digital adalah piksel. Sebuah citra digital adalah sebuah matriks (*array* dua dimensi) dari kumpulan piksel sebagai balok-balok bangunan dasar. Nilai setiap piksel sebanding dengan kecerahan titik yang sesuai dengan lokasinya. Matriks piksel dari sebuah gambar biasanya berbentuk persegi dan digambarkan sebagai piksel $N \times M$ dimana terdiri atas N kolom dan M baris.



Gambar 2.2 Urutan pengolahan citra digital

Gambar 2.2 merupakan urutan pengolahan citra digital. Dalam bidang *computer vision*, secara umum proses yang terjadi yaitu: citra non-digital (analog)

digitalisasi menjadi citra digital. Kemudian dilakukan pengolahan citra menjadi citra digital baru yang selanjutnya dilakukan analisa citra informasi/keputusan.

2.4.1 Citra Biner (Monokrom)

Citra monokrom adalah citra dengan suatu sistem grafik yang tidak memiliki kemampuan warna selain warna hitam atau warna putih. Perbedaan hanya diperoleh dengan menentukan tingkat intensitas *grayscale*. Nilai numerik yang digunakan biasanya adalah range 0 – 1. Citra monokrom yang diwakili dengan beberapa nilai kekuatan cahaya bernilai dari hitam sampai putih sebagai *grayscale image*.

Tabel 2.3 Intensitas Grayscale

Kode Intensitas Nilai	Intensitas	Biner	Tingkat Intensitas yang ditampilkan
0	0	0.0	<i>Black</i>
0.33	1	0.1	<i>Darkgray</i>
0.67	2	1.0	<i>Lightgray</i>
1	3	1.1	<i>White</i>

Pada Tabel 2.1 terdapat empat tingkat intensitas yang dapat ditampilkan. Nilai intensitas mendekati 0.33 akan disimpan dengan nilai biner 0 1 dalam memori *layer* dan menghasilkan titik dengan tingkat intensitas *darkgray* atau abu-abu kehitam-hitaman. Sedangkan tingkat intensitas itu sendiri ditentukan oleh program aplikasi kemudian diubah menjadi nilai biner yang sesuai.

2.4.2 Citra Warna RGB

Citra warna RGB biasanya juga disebut sebagai Citra *True Color*. Pada citra dengan tipe RGB, setiap piksel memiliki 3 komponen warna, yaitu merah (R), hijau (G) dan biru (B). Setiap komponen warna memiliki jangkauan nilai antara 0 sampai 255 (8 bit). Warna pada piksel ditentukan dari kombinasi merah, hijau dan biru. Hal ini akan memberikan kemungkinan total warna sebanyak $255^3 = 16.581.375$. Jadi total ukuran bit untuk setiap piksel adalah 24 bit (8 bit R, 8 bit G dan 8 bit B). Berikut adalah contoh dari citra RGB dapat dilihat pada tabel 2.4.

Tabel 2.4 Kode Warna

<i>Colour</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
<i>Black</i>	0	0	0
<i>Blue</i>	0	0	255
<i>Green</i>	0	255	0
<i>Cyan (Blue+Green)</i>	0	255	255
<i>Red</i>	255	0	0
<i>Magenta (Red+Green)</i>	255	0	255
<i>Yellow (Red+Green)</i>	255	255	0
<i>White (Red+Green+Blue)</i>	255	255	255
<i>Gray</i>	128	128	128

Pada tabel 2.2 diperlihatkan kode warna hasil gabungan warna RGB.

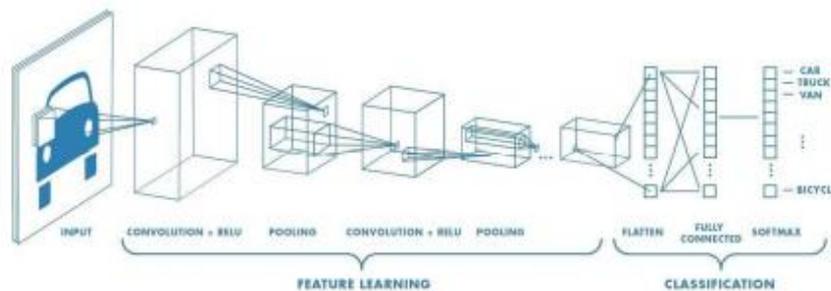
2.5 *Artificial Neural Network*

Artificial Neural Network atau Jaringan Saraf Tiruan (JST) adalah salah satu cabang dari *Artificial Intelligence*. JST merupakan suatu sistem pemrosesan informasi yang memiliki karakteristik-karakteristik menyerupai jaringan saraf biologi (Fausset, 1994). Ide dasar *Neural Network* dimulai dari otak manusia yang disusun atas miliaran *neuron* dimana masing-masing *neuron* akan terhubung pada puluhan ribu *neuron* lain. Masing-masing sel itu saling berinteraksi satu sama lain yang menghasilkan kemampuan tertentu pada kerja otak manusia *Loss Function*.

Dalam konteks algoritma optimisasi, fungsi yang digunakan untuk mengevaluasi solusi kandidat disebut sebagai fungsi objektif. Fungsi yang ingin diperkecil atau maksimalkan disebut pula fungsi objektif. Ketika meminimalkan fungsi objektif tersebut, secara tidak langsung dapat menyebutnya fungsi biaya (*cost function*), fungsi kerugian (*loss function*), atau fungsi kesalahan (*error function*) (Goodfellow IJ, 2014).

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu algoritma dari *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk *grid*, salah satunya citra dua dimensi, misalnya gambar atau suara. *Convolutional Neural Network* digunakan untuk mengklasifikasikan data yang terlabel dengan menggunakan metode *supervised learning* yang mana cara kerja dari *supervised learning* adalah terdapat data yang dilatih dan terdapat variabel yang ditargetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atau pemandangan, dan melakukan deteksi dan segmentasi objek.



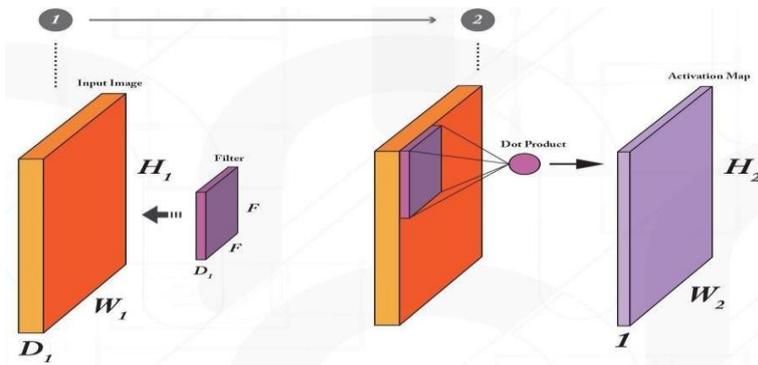
Gambar 2.3 Arsitektur *Convolutional Neural Network*

Gambar 2.3 menggambarkan *Convolutional Neural Network* menggabungkan tiga pokok arsitektur, yaitu *local receptive fields*, *Shared weight* yang berupa *filter*, dan *Spatial subsampling* yang berupa *Pooling*. Konvolusi atau yang biasa disebut dengan *Convolution* merupakan matriks yang berfungsi untuk melakukan *filter*.

Convolutional Neural Network termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Secara teknis, *convolutional network* adalah arsitektur yang bisa di *training* dan terdiri dari beberapa tahap. Arsitektur yang dimiliki oleh *Convolutional Neural Network* sebagai berikut.

2.6.1 *Convolution Layer*

Convolutional Layer merupakan *layer* yang pertama kali menerima gambar yang diinputkan. Layer ini melakukan proses konvolusi menggunakan sebuah *filter*. *Filter* ini diinisialisasi dengan nilai tertentu (*random* atau menggunakan teknik tertentu seperti Glorot), dan nilai dari *filter* inilah yang menjadi parameter yang akan di perbaharui dalam proses *learning*. *Filter* ini akan bergeser ke seluruh bagian gambar. Pergeseran tersebut akan menghasilkan *dot product* antara *input* dan nilai dari *filter* tersebut seperti pada gambar 2.4.



Gambar 2.4 Operasi Dot

Dengan menggeser *filter* keseluruhan bagian gambar, dihasilkan sebuah *output* yang disebut sebagai *activation map* atau *feature map* (Nurfita, 2018).

Input Volume (+pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[:, :, 0]$	$w0[:, :, 0]$	$w1[:, :, 0]$	$o[:, :, 0]$
0 0 0 0 0 0 0	1 -1 -1	0 0 1	-6 -7 -5
0 1 2 0 0 1 0	-1 0 0	1 1 0	-9 -6 -9
0 2 2 0 1 0 0	-1 -1 0	0 0 0	3 -5 -8
0 2 2 1 2 1 0	$w0[:, :, 1]$	$w1[:, :, 1]$	$o[:, :, 1]$
0 0 1 1 0 2 0	-1 0 1	1 0 0	2 3 -2
0 2 1 0 2 1 0	-1 0 0	1 -1 1	7 4 1
0 0 0 0 0 0 0	-1 -1 -1	-1 0 0	5 5 7
$x[:, :, 1]$	$w0[:, :, 2]$	$w1[:, :, 2]$	
0 0 0 0 0 0 0	-1 -1 1	-1 1 0	
0 0 2 0 0 1 0	0 0 0	0 -1 1	
0 1 1 0 2 2 0	1 -1 -1	1 0 -1	
0 0 1 1 0 2 0			
0 1 2 0 2 0 0			
0 0 2 0 1 0 0			
0 0 0 0 0 0 0			
$x[:, :, 2]$	Bias $b0$ (1x1x1)	Bias $b1$ (1x1x1)	
0 0 0 0 0 0 0	$b0[:, :, 0]$	$b1[:, :, 0]$	
0 2 2 1 0 0 0	1	0	
0 2 1 0 0 1 0			
0 0 2 2 2 1 0			
0 1 2 1 0 2 0			
0 2 1 1 1 1 0			
0 0 0 0 0 0 0			

Gambar 2.5 Proses Konvolusi 2 filter menghasilkan Feature Map

Dalam *machine learning*, *input* citra merupakan data berbentuk *array* multidimensi dan kernel merupakan parameter berbentuk *array* multidimensi yang

disesuaikan dengan model algoritma. Konvolusi dapat digunakan pada lebih dari satu dimensi. Sebagai contoh, jika menggunakan gambar dua dimensi I sebagai *input*, maka kernel K juga berbentuk dua dimensi:

$$S(i,j) = (I * K)(i,j) = \sum_a \sum_b I(m,n)K(i-m,j-n) \quad (3)$$

Keterangan :

$S(i,j)$ = Fungsi hasil konvolusi

I = *Input*

K = *Kernel* atau *Filter*

(I,j) = *Index Pixel Output*

(m,n) = *Index Pixel Kernel*

konvolusi bersifat komutatif, yang berarti persamaan (3) dapat ditulis:

$$S(i,j) = (K * I)(i,j) = \sum_a \sum_b I(i-m,j-n)K(m,n) \quad (4)$$

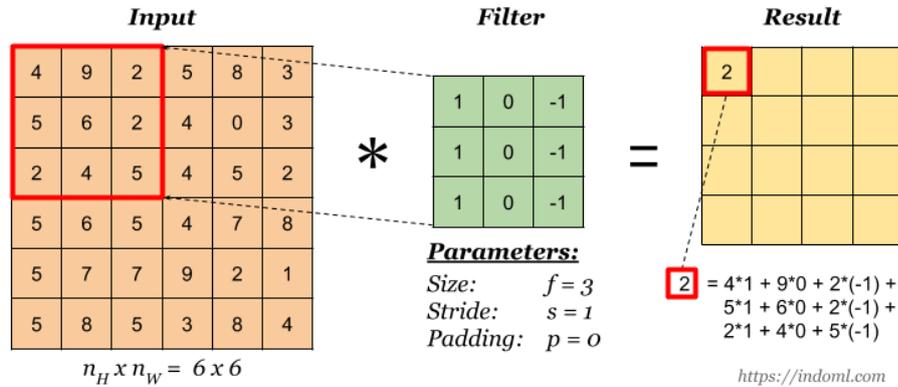
Sifat komutatif konvolusi digunakan karena kernel telah dibalik relatif terhadap *input*. Sifat komutatif ini berguna untuk menulis bukti (*proof*), tapi sifat ini bukan merupakan sifat yang essential pada implementasi jaringan saraf tiruan. Terdapat fungsi serupa yang disebut *cross-correlation* yang sama seperti konvolusi, namun tanpa membalik kernel:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m,j+n)K(m,n) \quad (5)$$

Banyak *library machine learning* yang menerapkan *cross-correlation*, namun menyebutnya konvolusi. Pada penelitian ini fungsi itu juga disebut konvolusi (Goodfellow dkk, 2016).

Tujuan konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi linear dari data *input* sesuai informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan *kernel*

konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input* pada CNN.



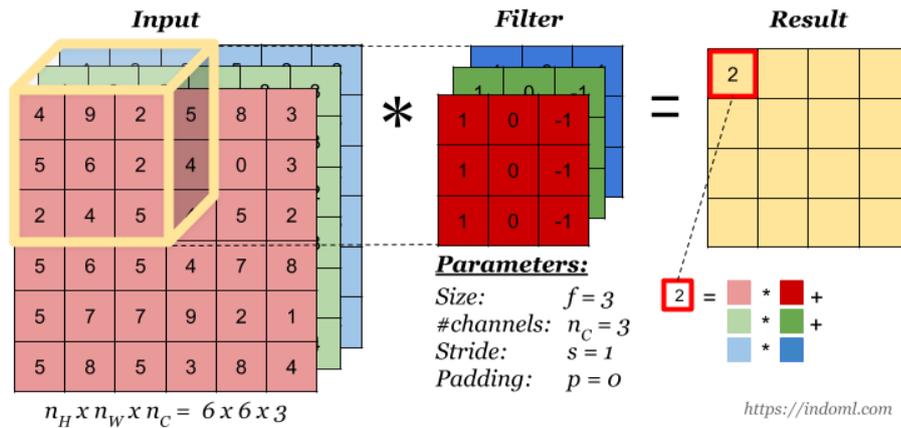
Gambar 2.6 Ilustrasi Operasi Konvolusi

Operasi konvolusi pada gambar 2.6 hanya berlaku pada citra biner yang hasil representasi gambarnya hanya 1 lapisan. Untuk citra RGB, representasi gambarnya akan menghasilkan 1 lapis matriks dari operasi konvolusi. Masing-masing lapisan matriks akan dioperasikan dengan *filter*. Jika *input* citra berukuran $n_h \times n_w \times n_c$ dengan n_c adalah jumlah *channel* atau *layer* pada *input* citra, maka persamaan (5) menjadi:

$$S(i, j, k) = (I * K)(i, j) = \sum_c^{n_c-1} \sum_m^{f-1} \sum_n^{f-1} I(i + m, j + n, c) K(m, n, c) \quad (6)$$

dengan $c = 1 \dots n_c$

Hasil operasi konvolusi dari masing-masing lapisan akan saling dijumlahkan untuk menghasilkan matriks hasil konvolusinya seperti pada gambar 2.7

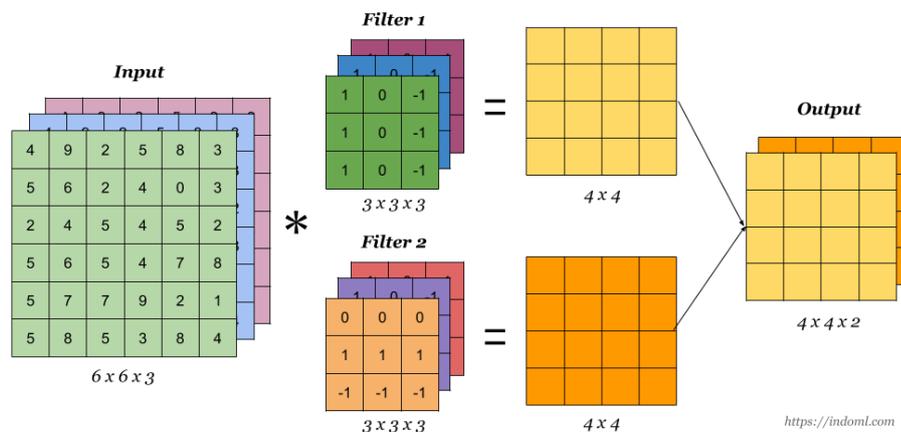


Gambar 2.7 Ilustrasi operasi konvolusi pada citra RGB

Adapun operasi konvolusi pada CNN, biasanya tidak hanya menggunakan 1 lapisan *filter*. Pada gambar 2.7 diilustrasikan bagaimana operasi konvolusi pada citra RGB dengan 2 *filter*. Jika n_f adalah jumlah *filter* yang digunakan dan $k = 1 \dots n_f$ maka persamaan (6) menjadi:

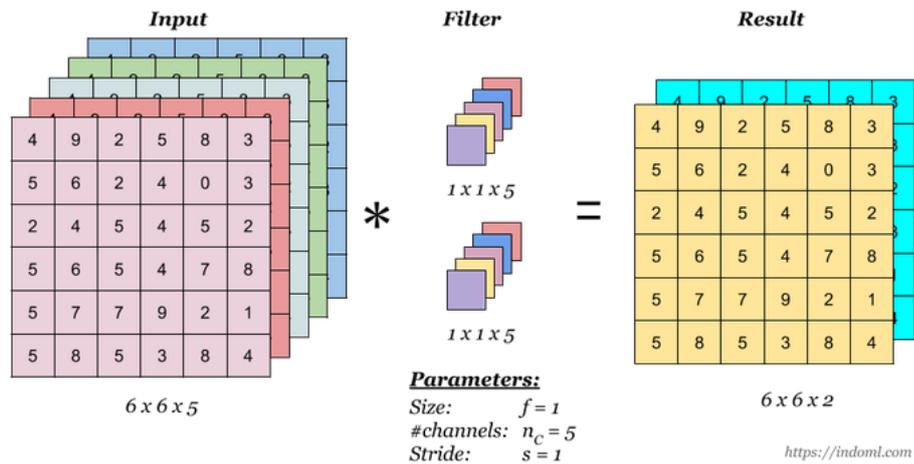
$$S(i, j, k) = (I * K_k)(i, j) = \sum_c \sum_m \sum_n I(i + m, j + n, c) K_k(m, n, c) \quad (7)$$

Operasi untuk masing-masing *filter* sama dengan ilustrasi pada gambar 2.8. Jumlah lapisan *output* dari hasil operasi konvolusi dengan banyak lapisan sama dengan banyaknya filter yang digunakan (Priyono, 2018).



Gambar 2.8 Ilustrasi operasi konvolusi pada citra RGB

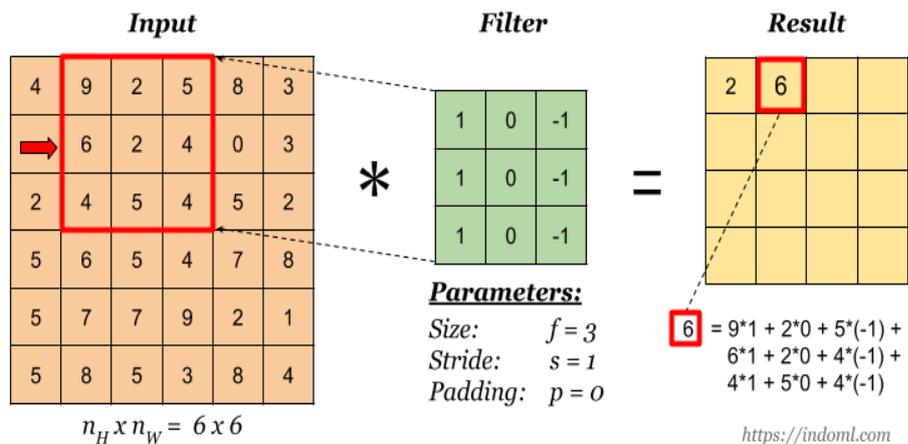
Pada gambar 2.9 diilustrasikan, konvolusi 1x1 yang mereduksi *input* dengan n_c *channel* menjadi satu *channel*, hal ini dikarenakan setiap *channel* dikalikan dengan suatu bilangan skalar pada *kernel* kemudian dijumlahkan.



Gambar 2.9 Konvolusi 1x1

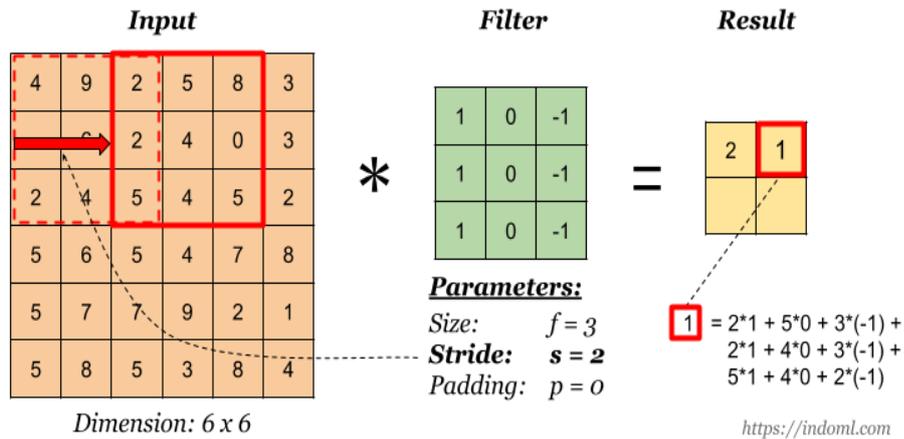
2.6.2 Stride

Stride adalah parameter yang mengontrol berapa jumlah pergeseran *filter*. jika nilai *stride* adalah 1, maka *convolutional filter* akan bergeser sebanyak 1 piksel secara horizontal. Semakin kecil *stride* maka informasi yang didapatkan semakin detail (Arrofiqoh, 2018).



Gambar 2.10 Ilustrasi Perhitungan Pergeseran *Stride* 1

Pada gambar 2.10 diilustrasikan sebuah *kernel* yang diaplikasikan pada *input*. Pergeseran *kernel* ini ditentukan oleh besarnya ukuran *stride*. Nilai *stride* pada gambar 2.10 adalah 1, maka pergeseran *kernel* pada *input* hanya sebanyak 1 piksel saja.

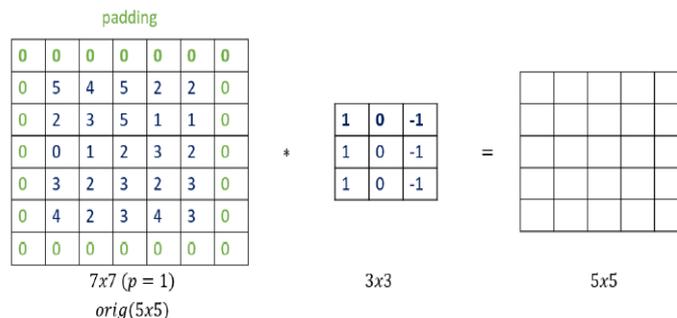


Gambar 2.11 Ilustrasi Perhitungan dengan Pergeseran *Stride* 2

Dan pada gambar 2.11 diilustrasikan sebuah *kernel* yang diaplikasikan pada *input*. Pergeseran *kernel* ini ditentukan oleh besarnya ukuran *stride*. Nilai *stride* pada gambar 2.11 adalah 2, maka pergeseran *kernel* pada *input* hanya sebanyak 2 piksel saja.

2.6.3 Padding

Padding adalah penambahan ukuran piksel dengan nilai tertentu disekitar data *input* agar hasil dari bidang *receptive* tidak terlalu kecil sehingga tidak banyak informasi yang hilang. Nilai ini biasanya nol sehingga disebut dengan *zero padding* (Arrofiqoh, 2018).

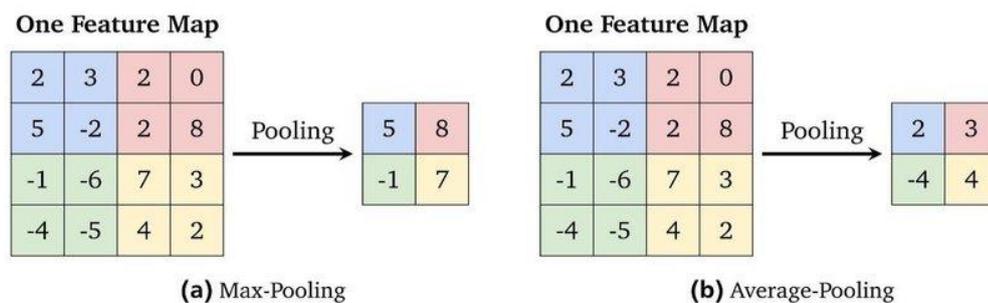


Gambar 2.12 Perhitungan dengan Penambahan *Padding*

Pada gambar 2.12 diilustrasikan sebuah matriks *input* dengan ukuran asli sebesar 5x5 dengan *filter* sebesar 3x3. *Input* ini kemudian diberi *padding* sebanyak satu sehingga membuat ukuran matriks *input* menjadi 7x7. Matriks yang telah diberi *padding* kemudian dikonvolusikan dengan *kernel* sehingga menghasilkan *feature map* dengan ukuran yang sama dengan ukuran *input* aslinya yaitu 5x5.

2.6.4 *Pooling Layer*

Pooling Layer atau *subsampling* adalah pengurangan ukuran matriks. Terdapat dua macam *pooling* yang sering digunakan yaitu *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata sedangkan pada *max pooling* adalah nilai maksimal (Arrofiqoh, 2018). Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*down sampling*), sehingga mempercepat komputasi karena parameter yang harus di *update* semakin sedikit dan mengatasi *overfitting* (Sena, 2017).

Gambar 2.13 Ilustrasi Perhitungan pada *Pooling Layer*

Pada gambar 2.13 langkah melakukan *pooling* adalah dengan membagi hasil keluaran *convolutional layer* menjadi beberapa grid berdasarkan penentuan jumlah *stride* dan jenis *pooling* yang digunakan.

2.6.5 Fungsi Aktivasi

Activation Function merupakan sebuah *node* yang ditambahkan ke akhir *output* dari setiap jaringan syaraf. Hal ini juga dikenal sebagai *Transfer Function*. *Activation function* digunakan untuk menentukan *output* jaringan syaraf tiruan. *Activation function* memetakan nilai yang dihasilkan di antara 0 sampai 1 atau -1 sampai 1 dan lain-lain. *Activation function* pada dasarnya dapat dibagi menjadi 2 tipe yaitu fungsi aktivasi linier dan fungsinya aktivasi non linear (Sharma, 2021).

Fungsi aktivasi biasa disebut sebagai lapisan pemetaan non-linear. Fungsi aktivasi biasa digunakan untuk meningkatkan kemampuan klasifikasi *network* (Chen dkk, 2018). Salah satu peranan dari fungsi aktivasi adalah untuk memberikan kemampuan *network* agar dapat melakukan tugas non-linear. Tanpa fungsi aktivasi, *neural network* hanyalah kombinasi operasi linear yang hanya melakukan tugas-tugas yang *linear* pula. Padahal kebanyakan kasus nyata di lapangan merupakan kasus *non-linear* (Santoso, 2018). Berikut beberapa fungsi aktivasi yang digunakan:

a. *Rectified Linear Unit* (ReLU)

Fungsi aktivasi ini merupakan yang paling umum digunakan pada *CNN*, untuk menghindari hilangnya gradien dan meningkatkan kecepatan pelatihan. Fungsi *ReLU* didefinisikan sebagai:

$$f(x) = \max(0, x) \quad (8)$$

Menurut persamaan di atas, *output ReLU* adalah nilai maksimum antara nol dan nilai *input*. *Output* sama dengan nol ketika nilai *input* negatif dan x jika nilai *input* positif. Dengan demikian, dapat diubah menjadi persamaan sebagai berikut:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (9)$$

b. *Softmax*

Fungsi aktivasi softmax digunakan untuk mendapatkan hasil klasifikasi. Fungsi aktivasi menghasilkan nilai yang diinterpretasi sebagai probabilitas

yang belum dinormalisasi untuk tiap kelas. Nilai kelas dihitung dengan menggunakan fungsi softmax (Arrofiqoh, 2018).

$$\text{Softmax}(Z_j) = \frac{e^{z_j}}{\sum_{i=0}^{K-1} e^{z_i}} \quad (10)$$

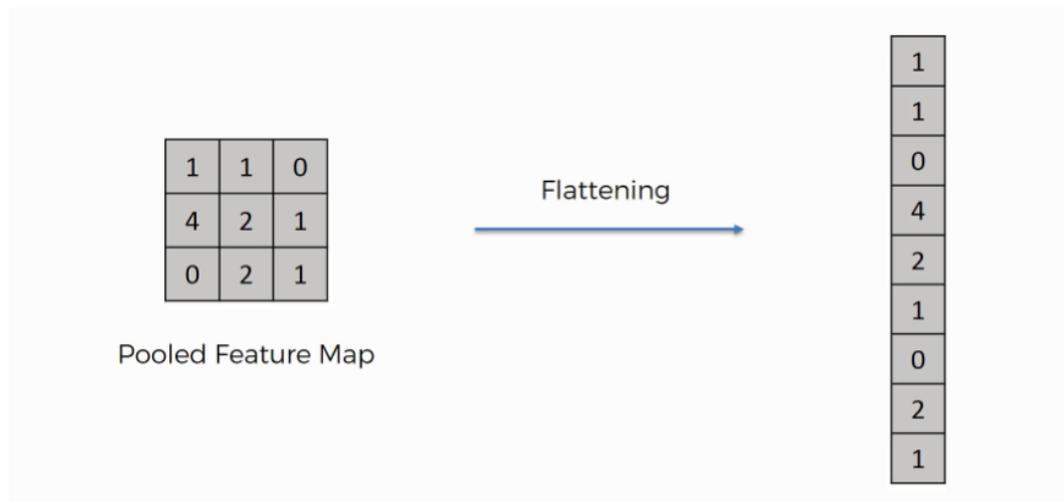
Keterangan:

K = Jumlah kelas

z_j = nilai *vector* ke- j

2.6.6 Flatten

Setelah proses akhir dari *feature extraction layer* telah selesai, maka dilakukan metode *Flatten* yang mengubah matriks n dimensi menjadi 1 dimensi yang nantinya akan digunakan sebagai *input* untuk proses klasifikasi dari *fully-connected layer*.

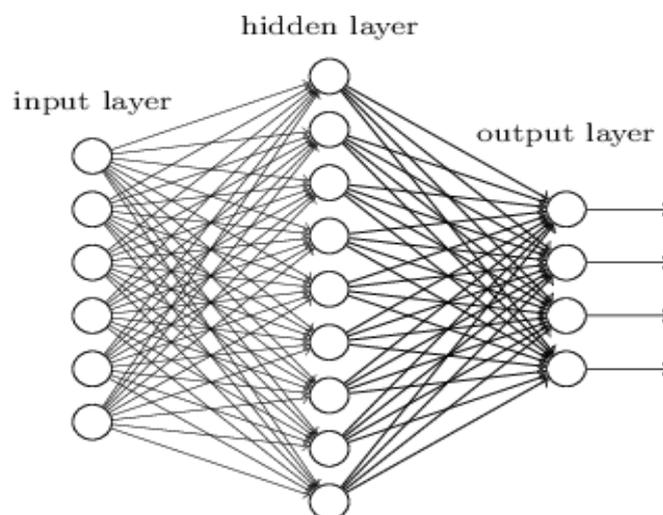


Gambar 2.14 Flatten

Contoh penerapan *flatten* dapat diilustrasikan pada gambar 2.14 Terdapat sebuah *feature map* berukuran 3x3 yang telah diberi *pooling*. *Feature map* ini kemudian di*flatten* agar dapat dijadikan sebuah *input* pada *fully connected layer*. *Feature map* yang awalnya berukuran 3x3 kini berubah menjadi matriks 1 dimensi.

2.6.7 Fully Connected Layer

Fully connected layer adalah proses akhir dari *Convolutional Neural Network* yang mirip dengan *neural network* biasa. *Fully connected layer* mengambil *input* dari hasil output *pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk *multidimensional array*. Oleh karena itu, lapisan ini akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak n - dimensi dimana n adalah jumlah kelas *output* yang harus dipilih program. Misalnya, jika lapisan terdiri dari 500 *neuron*, maka akan diterapkan fungsi aktivasi *softmax* yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan (Dutt, 2017).



Gambar 2.15 *Fully Connected Layer*

Pada gambar 2.15 setiap node pada *fully connected layer* saling terhubung dengan *node* yang berada di *layer* sebelum dan setelahnya. *Fully connected layer* merupakan *layer* yang memiliki parameter paling banyak, serta memiliki waktu komputasi yang tinggi saat proses *training*.

2.6.8 Backpropagation

Backpropagation adalah sebuah metode sistematis jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi (*Supervised Learning*) dan biasanya digunakan oleh perceptron dengan banyak layer lapisan untuk mengubah

bobot-bobot yang ada pada lapisan tersembunyinya. Algoritma pelatihan backpropagation pertama kali dirumuskan oleh Paul Werbos pada tahun 1974 dan dipopulerkan oleh Rumelhart bersama McClelland untuk dipakai pada *neural network*. ini berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*) (Purnomo, 2006). *Backpropagation* memiliki dua tahap, yaitu perambatan maju dan perambatan mundur. Kedua tahap tersebut dilakukan untuk setiap pola yang diberikan selama jaringan dalam pelatihan.

Pada dasarnya, pelatihan dengan metode *backpropagation* terdiri atas tiga langkah, yaitu sebagai berikut:

- a. Data dimasukkan ke *input* jaringan (*feedforward*)
- b. Perhitungan dan propagasi balik dari *error* yang bersangkutan
- c. Pembaharuan (*adjustment*) bobot.

Jika nilai *error* yang dihasilkan lebih besar dari batas *error* yang digunakan dalam sistem, maka akan dilakukan koreksi bobot. Koreksi bobot dapat dilakukan dengan menambah atau menurunkan nilai bobot.

Adapun cara kerja dari *Backpropagation*:

1. Tiap-tiap *output* menerima target pola yang berhubungan dengan *input* pembelajaran hitung informasi errornya.

$$\delta_k = \check{y} - y \quad (11)$$

2. Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki nilai bobot w_{jk} dengan laju pembelajaran α).

$$\Delta w_{jk} = \alpha \delta_k Z_j, (K = 1, 2, \dots, m \text{ dan } j = 0, 1, 2, \dots, p) \quad (12)$$

3. Tiap-tiap unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) menjumlahkan delta *inputnya* (dari uni-unit yang ada pada lapisan dibawahnya).

$$\delta_{-in j} = \sum_{k=1}^m \delta_k w_{jk} \quad (13)$$

Hitung informasi *error* nya dimana $\theta^{(l)T}$ adalah matriks bobot yang ada di *hidden* ke *layer* yang ada dibelakangnya $\theta^{(l+1)}$ adalah *hidden* ke *layer* ke $i+1$ dari arah belakang kemudian dikalikan dengan fungsi aktivasi.

$$\delta_j = \theta^{(l)T} \theta^{(l+1)} * (z_{in_j}) \quad (14)$$

Kemudian hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai v_{ij} dengan laju pembelajaran α .

$$\Delta v_{ij} = \alpha \vartheta_j x_i, (j = 1, 2, \dots, p; i = 0, 1, 2, \dots, n) \quad (15)$$

4. Hitung semua perubahan bobot
 - a. Perubahan bobot garis yang menuju ke unit keluaran

$$w_{jk}(\text{baru}) = v_{jk}(\text{lama}) + \Delta w_{jk} \quad (16)$$

- b. Perubahan bobot garis yang menuju ke unit tersembunyi

$$w_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta w_{ij} \quad (17)$$

- c. Proses iterasi selesai

Keterangan symbol:

ϑ_k = Informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tersembunyi.

ϑ_j = Informasi tentang kesalahan dari lapisan output ke unit tersembunyi.

α = Laju Pembelajaran (*Learning Rate*).

2.7 Evaluasi Kinerja

1. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* merupakan rasio prediksi benar dengan keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai *actual* (nilai sebenarnya) Persamaan (18).

$$Accuracy = \frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}} \quad (18)$$

2. *Precision* menggambarkan tingkat kebenaran suatu pengklasifikasi dari seluruh kelas positif yang diprediksi. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif, Persamaan (19).

$$Precision_i = \frac{M_{ii}}{\sum_i M_{ji}} \quad (19)$$

3. *Recall (Sensitivity)* menggambarkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan kelas positif. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif, Persamaan (20).

$$Recall_i = \frac{M_{ii}}{\sum_j M_{ij}} \quad (20)$$

4. *F1-score* merupakan gabungan antara *precision* dan *recall* yang bertujuan untuk melihat seberapa baik performa model dalam memprediksi suatu kelas, Persamaan (21).

$$f1\text{-score}_i = \frac{2(\text{Precision}_i \times \text{Recall}_i)}{\text{Precision}_i + \text{Recall}_i} \quad (20)$$

Keterangan:

M : *Confusion Matrix*

i,j : $i,j \in \{A, B, C, D, E, F, G, \dots, Z\}$

2.8 Categorical Crossentropy

Categorical crossentropy adalah fungsi kerugian yang digunakan dalam tugas klasifikasi multi-kelas. Klasifikasi multi-kelas adalah tugas-tugas di mana contoh hanya dapat dimiliki salah satu dari banyak kategori yang mungkin, dan model harus memutuskan yang mana. Dengan kata lain data yang digunakan