

**APLIKASI PENDETEKSI KOROSI PADA PELAT KAPAL  
DENGAN METODE CONVOLUTIONAL NEURAL  
NETWORK**

**SKRIPSI**

*Ditujukan untuk memenuhi persyaratan*

*Memperoleh gelar Sarjana Teknik pada Departemen Teknik Sistem Perkapalan*

*Fakultas Teknik Universitas Hasanuddin*



**DISUSUN OLEH :**

**MIFTAHUL IKHSAN TAHIR**

**D33116307**

**DEPARTEMEN TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
2022**

**APLIKASI PENDETEKSI KOROSI PADA PELAT KAPAL  
DENGAN METODE CONVOLUTIONAL NEURAL  
NETWORK**



**DISUSUN OLEH :**

**MIFTAHUL IKHSAN TAHIR  
D33116307**

**DEPARTEMEN TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
2022**

**LEMBAR PENGESAHAN**

**"APLIKASI PENDETEKSI KOROSI PADA PELAT KAPAL DENGAN METODE  
CONVOLUTIONAL NEURAL NETWORK"**

Disusun dan diajukan oleh

**MIFTAHUL IKHSAN TAHIR**

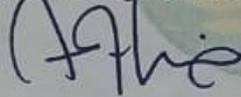
**D33116307**

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Departemen Teknik Sistem Perkapalan Fakultas Teknik Universitas Hasanuddin

Pada tanggal 28 April 2022  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

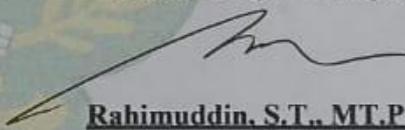
Pembimbing Utama,



**Haryanti rivai. S.T., M.T.Ph.D.**

**NIP. 19790225 200212 2 001**

Pembimbing Pendamping,



**Rahimuddin. S.T., MT.P.hD.**

**NIP. 197108325 199903 1 002**

Ketua Departemen,



**Dr. Eng. Faisal Mahmuddin. S.T., M.Inf.Tech., M.Eng.**

**NIP. 19810211 200501 1 003**

## LEMBAR PERSETUJUAN

### APLIKASI PENDETEKSI KOROSI PADA PELAT KAPAL DENGAN METODE CONVOLUTIONAL NEURAL NETWORK

**OLEH:**  
**MIFTAHUL IKHSAN TAHIR**  
**D33116307**

Skripsi ini telah direvisi dan disetujui oleh Panitia Ujian Sarjana (S1) Departemen Teknik Sistem Perkapalan, Fakultas Teknik, Universitas Hasanuddin untuk memenuhi persyaratan memperoleh gelar sarjana Teknik.

Panitia Ujian Sarjana:

Pembimbing 1 : Haryanti Rivai, S.T., M.T., Ph.D.

Pembimbing 2 : Rahimuddin, S.T., M.T., Ph.D.

Penguji 1 : Andi Haris Muhammad, S.T., M.T., Ph.D.

Penguji 2 : Dr. Ir. Ganding Sitepu, Dipl. Ing

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini;

Nama : Miftahul Ikhsan Tahir

NIM : D33116307

Departemen : Teknik Sistem Perkapalan

Jenjang : S1

dengan ini menyatakan dengan sebenar-benarnya bahwa skripsi yang berjudul :

### **APLIKASI PENDETEKSI KOROSI PADA PELAT KAPAL DENGAN METODE CONVOLUTIONAL NEURAL NETWORK**

adalah karya ilmiah saya sendiri dan sepanjang pengetahuan saya di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila di kemudian hari ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia menerima sanksi atas perbuatan tersebut dan diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Gowa, 12 Mei 2022

Yang menyatakan,



Miftahul Ikhsan Tahir

D33116307

## ABSTRAK

Kegunaan baja dalam industri perkapalan sering digunakan dalam pembuatan suatu bahan material, salah satunya adalah pelat kapal. Pelat kapal merupakan material yang digunakan sebagai kulit kapal yang berhubungan langsung dengan faktor lingkungan sekitar seperti suhu, salinitas air, udara dan lain-lain. Faktor inilah yang mempengaruhi adanya korosi pada pelat kapal. Penelitian ini bertujuan untuk merancang suatu aplikasi pendeteksi korosi pada pelat kapal dengan metode *Convolutional Neural Network* menggunakan software matlab. Metode CNN ini terdiri dari 3 tahap. Tahap pertama yaitu klasifikasi data yang dibagi menjadi tiga, *GOOD*, *FAIR*, dan *POOR* dengan total data sebanyak 1050 gambar. Tahap kedua yaitu, proses training dan testing data dengan parameter learning yang telah melalui tahap uji coba. Arsitektur CNN yang terdiri dari beberapa proses seperti konvolusi filter 3x3 dengan *padding same*, *rectified linear unit*, dan *max pooling window 2x2* dengan *padding same*. Tahap ketiga yaitu, pengaplikasian hasil dari proses training CNN. Hasil dari perancangan aplikasi ini dapat berjalan dengan baik saat melakukan prediksi terhadap gambar korosi dan mampu mendeteksi area korosi pada pelat kapal. Tingkat akurasi ketepatan pada saat *testing* data mencapai 96% dengan 25 kali percobaan pada gambar secara acak. Dalam uji validasi terhadap hasil klasifikasi data, didapatkan nilai error rata-rata pada proses komputasi yaitu 1.77% dengan simpangan rata-rata 1.26%.

Kata kunci: Korosi, Pelat, Matlab, *Deep Learning*, *Convolutional Neural Network*.

## **ABSTRACT**

*The use of steel in the shipping industry is often used in the manufacture of a material, one of which is the ship plate. The ship plate is a material used as a ship's shell which is directly related to the surrounding environmental factors such as temperature, water salinity, air and others. This factor affects the corrosion on the ship plate. This study aims to design a corrosion detection application on ship plates using the Convolutional Neural Network method using Matlab software. This CNN method consists of 3 stages. The first stage is data classification which is divided into three, GOOD, FAIR, and POOR with a total data of 1050 images. The second stage is the process of training and testing data with learning parameters that have gone through the trial stage. CNN architecture which consists of several processes such as convolution filter 3x3 with the same padding, rectified linear unit, and max pooling window 2x2 with the same padding. The third stage is the application of the results of the CNN training process. The results of this application design can run well when predicting corrosion images and are able to detect corrosion areas on the ship's plate. The level of accuracy when testing data reached 96% with 25 trials on random images. In the validation test on the results of data classification, the average error value in the computational process was 1.77% with an average deviation of 1.26%.*

*Keywords: Corrosion, Plate, Matlab, Deep Learning, Convolutional Neural Network.*

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas skripsi ini. Tidak sedikit kendala yang menghadang penyusun dalam menyelesaikan proposal ini, namun berkat rahmat dan hidayah-Nya telah membimbing penyusun untuk terus berusaha menyelesaikan salah satu mata kuliah di departemen Teknik Sistem Perkapalan, Universitas Hasanuddin.

Ini merupakan persyaratan untuk menyelesaikan studi di departemen Teknik Sistem Perkapalan – Universitas Hasanuddin. Penyusun harus mengakui, skripsi ini masih sangat jauh dari sempurna, semua karena keterbatasan waktu dan pengetahuan serta kemampuan penyusun sebagai manusia biasa. Untuk itu penyusun mohon maaf atas semua kekurangan dan kesalahan yang terjadi di dalam penyusunan Proposal “Aplikasi Pendeteksi Korosi Pada Pelat Kapal Dengan Metode *Convolutional Neural Network*” ini, serta penyusun berharap masukan dan saran agar ke depannya penyusun dapat lebih baik lagi dalam menyusun skripsi.

Akhirnya penyusun berharap semoga skripsi ini dapat bermanfaat bagi penyusun secara pribadi serta pada pembaca yang menjadikan skripsi ini sebagai acuan atau pedoman dalam pembelajaran ataupun dalam menyusun makalah. Semoga Allah SWT senantiasa memberikan rahmat dan hidayah-Nya pada kita semua. Aamiin.

Gowa, Februari 2022

Penulis

## DAFTAR ISI

HALAMAN SAMPUL .....	i
HALAMAN SAMPUL .....	ii
PERNYATAAN KEASLIAN .....	iii
LEMBAR PERSETUJUAN .....	iv
ABSTRAK .....	vi
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	1
DAFTAR TABEL.....	2
BAB 1. PENDAHULUAN .....	3
1.1. Latar Belakang.....	3
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah .....	4
1.4. Tujuan penelitian .....	5
1.5. Manfaat Penelitian .....	5
1.6. Sistematika Penulisan .....	5
BAB 2. TINJAUAN PUSTAKA .....	7
2.1. Pelat Kapal.....	7
2.2. Korosi .....	7
2.2.1.Korosi Seragam ( <i>Uniform Attack</i> ).....	8
2.3. Regulasi IMO.....	9
2.3.1. <i>Assessment of coating condition (GOOD, FAIR, POOR)</i> .....	9
2.4. Spektrum Cahaya.....	10
2.5. <i>Color Threshold</i> .....	12
2.5.1.HSV (Hue Saturation Value).....	12
2.6. Metode <i>Classification</i> .....	13
2.7. <i>Machine Learning</i> .....	14
2.7.1. <i>Artificial Neural Network</i> .....	16
2.7.2.Karakteristik <i>Neural Network</i> .....	18
2.7.3.Arsitektur <i>Neural Network</i> .....	20
2.7.4. <i>Backpropagation</i> .....	22
2.8. <i>Deep Learning</i> .....	24
2.9. <i>Convolutional Neural Network</i> .....	26
2.10. <i>Convolutional Layer</i> .....	27
2.11. <i>Pooling Layer</i> .....	31
BAB 3. METODE PENELITIAN .....	33
3.1. Tempat Dan Waktu Penelitian.....	33

3.2. Studi Literatur .....	33
3.3. Pengumpulan Data.....	33
3.4. Diagram alir penelitian .....	34
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>36</b>
4.1. Klasifikasi Data .....	36
4.1.1. Diagram Alir Klasifikasi Data.....	36
4.1.2. Langkah Klasifikasi Data .....	37
4.1.3. Uji Validasi Toolbox Color Threshold.....	39
4.1.4. Hasil Klasifikasi Data.....	53
4.2. <i>Training</i> dan <i>Testing</i> Data .....	57
4.2.1. Diagram Proses <i>Training</i> dan <i>Testing</i> Data.....	57
4.2.2. Langkah Proses <i>Training</i> dan <i>Testing</i> Data.....	58
4.2.3. Hasil Proses <i>Training</i> dan <i>Testing</i> Data .....	59
4.3. Desain GUI .....	61
4.3.1. Diagram Alir GUI.....	61
4.3.2. Tampilan dan Fitur Aplikasi.....	62
4.3.3. Cara Kerja Aplikasi .....	63
4.4. Pengaruh Parameter <i>Learning</i> Terhadap Akurasi.....	65
4.4.1. Pengaruh Jumlah <i>Epoch</i> .....	66
4.4.2. Pengaruh Nilai <i>Learning Rate</i> .....	66
4.4.3. Pengaruh Ukuran Citra .....	67
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>	<b>69</b>
5.1. KESIMPULAN.....	69
5.2. SARAN.....	69
DAFTAR PUSTAKA .....	71
LAMPIRAN.....	73

## DAFTAR GAMBAR

Gambar 2.1.Korosi Pada Pelat kapal.....	9
Gambar 2.2. <i>Hue Saturation Value</i> .....	13
Gambar 2.3. <i>Machine Learning</i> (mathworks.com, 2016).....	14
Gambar 2.4.Neuron pada otak (medium.com, 2019).....	17
Gambar 2.5.Karakteristik <i>Neural Network</i> .....	18
Gambar 2.6. <i>Single Layer Neural Net</i> .....	21
Gambar 2.7. <i>Multilayer Neural Net</i> .....	21
Gambar 2.8. <i>Backpropagation</i> .....	23
Gambar 2.9.ArsitekturCNN .....	27
Gambar 2.10. <i>Stride</i> .....	29
Gambar 2.11. <i>Padding</i> .....	30
Gambar 4.1. <i>Training Progress</i> .....	59
Gambar 4.2.Log <i>Training Data</i> .....	60
Gambar 4.3. Hasil <i>Training Data</i> .....	60
Gambar 4.4.Tampilan awal Aplikasi .....	63
Gambar 4.5.Tampilan Awal Aplikasi .....	64
Gambar 4.6.Hasil Prediksi Aplikaszi.....	64
Gambar 4.7.Aplikasi Mendeteksi luasan Area Korosi.....	65

**DAFTAR TABEL**

Tabel 2. 1 IMO clarification of coating conditions.....	10
Tabel 4. 1. Hasil Uji Validasi Toolbox Color Threshold.....	52
Tabel 4.2. Percobaan Terhadap Jumlah Epoch .....	66
Tabel 4.3. Percobaan Terhadap Nilai <i>Learning Rate</i> .....	67
Tabel 4.4. Percobaan Terhadap Ukuran Citra.....	68

## **BAB 1. PENDAHULUAN**

### **1.1. Latar Belakang**

Pada industri perkapalan, baja sering digunakan dalam pembuatan suatu material, salah satunya adalah pelat kapal dengan kandungan baja karbon yang rendah. Pelat baja merupakan material yang bisa dikatakan paling efisien, karena proses penyambungan dengan menggunakan teknologi las dan proses reparasi memakan waktu yang relatif efisien [1]. Logam ini terdiri dari unsur besi dan karbon dimana di dalam proses pencetakan pelat baja lembaran tentunya untuk pelat kapal digunakan perbandingan jumlah karbon yang cukup banyak sehingga sering juga disebut dengan *marine plate*, tentunya sudah sesuai dengan standar yang dikeluarkan oleh Biro Klasifikasi Indonesia.

Pelat kapal merupakan material yang digunakan sebagai kulit kapal yang berhubungan langsung dengan faktor lingkungan sekitar seperti suhu, salinitas air, udara dan lain-lain. Faktor inilah yang mempengaruhi adanya korosi pada pelat kapal.

Setiap logam yang teraliri oleh arus listrik maka secara terus menerus akan mengalami pelepasan struktur elektron yang terkandung, inilah yang disebut dengan korosi. Korosi yaitu proses kembalinya ke alam senyawa elektron yang terkandung dalam sebuah logam. Korosi terjadi melalui reaksi redoks, dimana oksigen mengalami reduksi, dan logam mengalami oksidasi yang menghasilkan karat. Karat adalah lapisan merah (kekuning-kuningan) yang melekat pada besi dan sebagainya, dari akibat proses korosi. Besi yang berkarat mudah dikenali melalui warna, serta sifatnya yang rapuh dan mudah larut.

Korosi merupakan salah satu musuh terbesar dalam dunia industri dikarenakan kerugian yang ditimbulkan yaitu penurunan kekuatan material dan biaya perbaikan yang mahal. Korosi yang terjadi pada permukaan logam akibat reaksi kimia karena pH air yang rendah dan udara yang lembab, sehingga semakin lama logam akan menipis [2]. Hal ini juga terjadi pada lambung kapal yang bergesekan langsung dengan air laut saat berlayar, sehingga pada daerah tersebut sering terjadi korosi. Kerugian yang cukup besar akibat korosi

mengharuskan adanya upaya-upaya pencegahan terjadinya karat seperti melakukan perlindungan secara elektrokimia menggunakan anoda korban atau melakukan pelapisan anti karat (*anti rust coatings*).

Pada penelitian sebelumnya [1],[2],[3], untuk mendeteksi korosi pada pelat baja di kapal sering dilakukan dengan kasat mata, kimiawi, dan juga menggunakan alat uji *non-destructive*. Hal ini menyebabkan ketidakefisienan waktu dalam proses pengerjaan reparasi kapal. Mengingat Pentingnya hal-hal tersebut, maka dibuatlah suatu usulan judul penelitian “Aplikasi Pendeteksi Korosi Pada Pelat Kapal Dengan Metode *Convolutional Neural Network*” yang diharapkan pada penelitian ini, dengan aplikasi dari salah satu algoritma cerdas *Convolutional Neural Network* kedepannya, proses reparasi kapal dengan pelat korosi dapat lebih cepat, efisien, dan aman.

### **1.2. Rumusan Masalah**

1. Bagaimana merancang suatu Aplikasi Pendeteksi korosi pada pelat lambung kapal?
2. Bagaimana parameter dan arsitektur Neural Network terbaik pada proses training CNN?

### **1.3. Batasan Masalah**

Mengingat ruang lingkup permasalahan yang begitu luas, maka pada penelitian ini terdapat beberapa batasan masalah agar pembahasan yang akan dilakukan bisa terfokus. Diantaranya adalah:

1. Data yang digunakan adalah data gambar yang diambil dari galangan kapal.
2. Terdapat proses *cropping* menggunakan *software* Picasa 3.0 dikarenakan ukuran data yang terlalu besar.
3. Deteksi kondisi pelat dibagi menjadi 3 kelas yaitu, *Good*, *Fair*, dan *Poor*.
4. Jenis korosi yang dideteksi adalah korosi seragam (*Uniform Attack*).
5. Program mendeteksi luas korosi berdasarkan warna korosi.
6. Warna cat pada kapal harus selain dari warna yang menyerupai karat.
7. Jumlah data per kelas adalah minimal 100 gambar.
8. Data yang diambil harus berada dalam kondisi spektrum cahaya sempurna.

9. Penelitian ini menggunakan laptop Zyrex sky 232 mini, dengan spesifikasi prosessor *intel Celeron N4020*, RAM 4 GB, VGA intel HD 2GB.
10. Penelitian ini dibuat menggunakan *software* Matlab versi R2020a.

#### **1.4. Tujuan penelitian**

Berdasarkan rumusan masalah yang ada, tujuan dalam penelitian ini adalah:

1. Mampu merancang suatu aplikasi pendeteksi korosi pada pelat lambung kapal.
2. Mendapatkan arsitektur terbaik dan hasil dari klasifikasi korosi pelat pada gambar menggunakan *Convolutional Neural Network (CNN)*.

#### **1.5. Manfaat Penelitian**

Bagi akademik:

1. Dapat digunakan sebagai referensi untuk pengembangan ilmu pengetahuan dan teknologi kedepannya.
2. Sebagai bahan kajian dalam pembelajaran *Artificial Intelligence*.

Bagi Industri perkapalan:

1. Dapat digunakan sebagai alat bantu inspeksi kapal.
2. Meningkatkan kualitas reparasi kapal dan mengurangi tingkat resiko kecelakaan inspektor.

#### **1.6. Sistematika Penulisan**

Secara garis besar penyusunan proposal skripsi dan pembaca memahami uraian dan makna secara sistematis, maka skripsi disusun pada pola berikut:

##### **BAB I: PENDAHULUAN**

Pada bab ini konsep dasar penyusunan skripsi yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

##### **BAB II: TINJAUAN PUSTAKA**

Pada bab ini diberikan penjelasan mengenai teori dasar yang digunakan dalam menyelesaikan skripsi ini yaitu teori dasar tentang karakteristik pelat

korosi, *Artificial Neural Network*, karakteristik *Neural Network*, Arsitektur *Neural Network*, dan *Convolutional Neural Network*.

### BAB III: METODE PENELITIAN

Pada bab ini akan dijelaskan tahapan-tahapan berupa proses yang dimulai dari mengidentifikasi masalah yang ada hingga hasil akhir yang diharapkan.

### BAB IV: HASIL PENELITIAN

Pada bab ini akan dijelaskan hasil dari percobaan penelitian berupa *training* data dan *test* data, juga seberapa besar tingkat akurasi dan *lost* dalam mengklasifikasikan gambar.

### BAB V: PENUTUP

Pada bab ini akan disajikan secara singkat kesimpulan yang diperoleh dari pembahasan dan juga memuat saran-saran bagi pihak yang berkepentingan untuk pengembangan penelitian lebih lanjut.

## **BAB 2. TINJAUAN PUSTAKA**

### **2.1. Pelat Kapal**

Baja telah lama digunakan dalam industri kapal sebagai komponen pelat utama didalam badan maupun lambung kapal. Penggunaan material baja di dalam lingkungan industri perkapalan, menempati urutan pertama sebagai komponen bangunan kapal, mesin utama maupun mesin bantu kapal [4].

Pelat kapal dibuat dari peleburan biji besi dalam tungku sembur yang mempunyai struktur kerucut dan tungku tersebut tentunya terbuat dari bahan tahan api. Panas peleburan menggunakan kokas dan batu kapur agar kerak pada bijih besi dapat terangkat dan tidak tercampur. Kandungan dalam tiap lembar pelat adalah 92-97 % merupakan besi. Sisanya terdapat kandungan karbon, silikon, mangan, belerang, dan fosfor. Tentunya dalam cetakan pelat kotoran yang terbawa harus diminimalisir untuk menjaga kualitas dari pelat tersebut [4].

Baja secara luar dapat diartikan sebagai paduan antara besi dan karbon. Kandungan karbon bervariasi berkisar antara 0,1 % dan ketika baja telah mengeras menjadi 1,8 % dari kandungan seluruh pelat. Proses pengasaman digunakan untuk memperbaiki pelat besi yang rendah dengan memasukkan unsur fosfor dan unsur sulfur. Kedua unsur tersebut kaya akan silikon dan menghasilkan kerak asam yang dibutuhkan pelat. Unsur fosfor merupakan kapur yang menghasilkan kerak dasar dari 85 % unsur baja [4].

### **2.2. Korosi**

Korosi adalah kerusakan atau degradasi logam atau besi akibat reaksi redoks antara suatu logam dengan berbagai zat di lingkungannya yang menghasilkan senyawa yang tidak dikehendaki seperti karat. Besi yang mengalami korosi membentuk karat dengan rumus  $Fe_2O_3 \cdot xH_2O$ . Pada proses terjadinya korosi, besi (Fe) bertindak sebagai pengoksidasi dan oksigen ( $O_2$ ) yang terlarut dalam air bertindak sebagai pereduksi. Besi yang mengalami reaksi oksidasi akibat adanya beda potensial negatif (anoda) pada besi, menghasilkan ion besi dan elektron. Ion besi kemudian terlarut dalam air sehingga menyebabkan besi terkikis dan ion elektron bermigrasi ke katoda. Reaksi reduksi

oksigen yang terjadi karena adanya oksigen ( $O_2$ ) dan ion Hidronium ( $4H^+$ ) pada katoda menghasilkan molekul-molekul air ( $2H_2O$ ).

Sementara itu, ion besi ( $Fe^{2+}$ ) yang terlarut akan di oksidasi lebih lanjut oleh oksigen ( $O_2$ ) sehingga membentuk senyawa ( $Fe_2O_3$ ). Senyawa tersebut kemudian terhidrasi oleh molekul air sehingga menghasilkan karat yang berwarna coklat ke merah atau merah ke kuning ( $Fe_2O_3 \cdot xH_2O$ ) dan juga ion hidronium ( $4H^+$ ) kembali. Karena ion Hidronium dihasilkan, hal ini menyebabkan proses korosi akan terus berlangsung karena ion Hidronium merupakan reaktan pada reaksi reduksi oksigen di katoda. Oleh karena itu, proses oksidasi pun terus berjalan dan menyebabkan besi terus menerus terkikis sehingga kualitas ketahanan besi terus menurun. Hal ini juga berlaku pada pelat besi kapal sebagai akibat interaksi dengan lingkungan sekitarnya [4].

Beberapa prinsip pencegahan korosi yang telah berkembang umumnya disesuaikan dengan jenis peralatan, tempat, maupun jenis lingkungan yang korosif. Pencegahan korosi pada baja yang cukup luas dikenal adalah dengan cara melapisi baja dengan lapisan penghalang (*coatings*). Lapisan penghalang dimaksudkan terutamas untuk memisahkan permukaan baja dari lingkungan mikro pada permukaan baja dari lingkungan, mengendalikan lingkungan mikro pada permukaan baja, maupun untuk tujuan keindahan atau penampilan (dekoratif). Banyak cara pelapisan yang digunakan untuk maksud tersebut antara lain cat, lak (*laquers*), vernis, dan lapisan baja. Sejauh ini penggunaan cat sebagai pelapis dan pelindung korosi [4].

### **2.2.1. Korosi Seragam (*Uniform Attack*)**

Korosi seragam adalah korosi yang terjadi pada permukaan logam akibat reaksi kimia karena pH air rendah dan udara yang lembab, sehingga semakin lama logam akan semakin menipis. Umumnya terjadi pada pelat baja atau profil, logam homogen. Korosi jenis ini bisa dicegah dengan cara diberi lapis lindung yang mengandung inhibitor. Berikut contoh gambar korosi pada pelat kapal dilihat pada gambar 2.1 dibawah.



Gambar 2. 1.Korosi Pada Pelat kapal

### 2.3. Regulasi IMO

International Maritime Organization (IMO) merupakan badan khusus PBB yang bertanggung jawab untuk untuk keselamatan dan keamanan aktivitas pelayaran dan pencegahan polusi di laut oleh kapal. Secara teknis, IMO memiliki tugas dalam pemutakhiran legislasi yang ada atau untuk mengembangkan dan mengadopsi peraturan baru, melalui pertemuan yang dihadiri oleh ahli maritim dari negara anggota, serta organisasi antar pemerintah dan non-pemerintah lain seperti BIMCO, CMI, Greenpeace, IALA.

#### 2.3.1. *Assessment of coating condition (GOOD, FAIR, POOR)*

Lapisan pada pelat kapal akan dinilai oleh surveyor menjadi tiga kelas yaitu kondisi GOOD, FAIR, atau POOR. Kondisi tersebut dibagi berdasarkan visual inspection dan persentase dari luas area dengan kondisi lapisan yang rusak dan daerah yang penuh dengan karat. Definisi dari kondisi karat GOOD, FAIR, dan POOR berdasarkan aturan IMO adalah sebagai berikut:

- GOOD: kondisi dengan karat hanya berkisar dibawah 3%
- FAIR: Kondisi dengan lapisan pelindung yang rusak atau penetrasi karat kurang 20% dari luas area.
- POOR: kondisi kerusakan pada lapisan atau penyebaran karat lebih dari 20%

Berikut adalah tabel 2.1. IMO yang mendefinisikan kondisi permukaan pelat kapal:

Tabel 2. 1 IMO clarification of coating conditions

	GOOD	FAIR	POOR
Breakdown of coating od area rusted	<3%	3-20%	>20%
Area of hard rust scale	-	<10%	>10%
Local breakdown of coating or rust on edges or weld lines	<20%	20-50%	>50%
Notes:			
1) % is the percentage calculated on basis of the area under consideration or of the critical structural area			
2) % is the percentage calculatedon basis of edges or weld lines in the area under considerationor the critical structural area spot rusting, i.e. rustin in spot without visible failure of coating			

#### 2.4. Spektrum Cahaya

Pada tahun 1873, J.C. Maxwell secara teori menjabarkan kemungkinan adanya gelombang elektromagnetik di alam yang menjalar dengan kecepatan sebesar kecepatan cahaya. Kemudian secara eksperimen Heinrich Hertz pada tahun 1888, dengan memakai osilasi dipol listrik, berhasil memperoleh gelombang elektromagnetik, yakni gelombang mikro, yang ternyata dapat dipantulkan, dibiaskan, difokuskan dengan lensa dan seterusnya sebagaimana lazimnya cahaya. Sejak itu, cahaya diyakini sebagai gelombang elektromagnetik transversal. Yang dimaksud dengan gelombang elektromagnetik ialah gelombang medan listrik dan kuat medan magnet di setiap titik yang dilalui gelombang elektromagnetik itu berubah-ubah terhadap waktu secara periodis dan perubahan itu di jalankan sepanjang arah menjalarnya gelombang [5].

Gelombang elektromagnetik meliputi cahaya, gelombang radio, gelombang mikro, inframerah, cahaya tampak, ultraviolet, sinar X dan sinar gamma [6]. Berbagai jenis gelombang elektromagnetik tersebut hanya berbeda dalam panjang gelombang dan frekuensinya, yang dihubungkan dengan persamaan:

$$f = c/\lambda \dots\dots\dots (2.1)$$

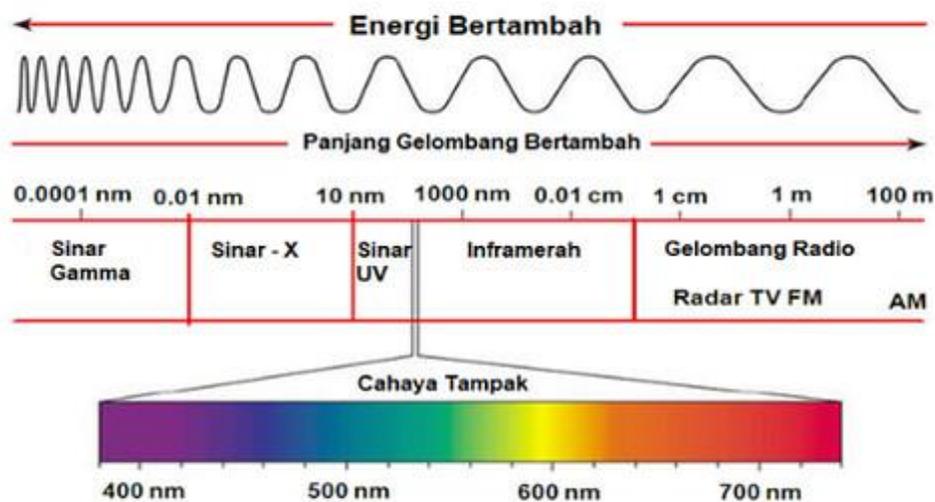
dimana:  $f$  = frekuensi (Hz)

$c$  = kecepatan cahaya (m/s)

$\lambda$  = panjang gelombang (m)

Cahaya merupakan sebagian dari gelombang elektromagnetik yang dapat dilihat mata dengan komponennya yaitu cahaya merah, jingga, kuning, hijau, biru, nila dan ungu. Panjang gelombang cahaya berada pada kisaran antara 0,2  $\mu\text{m}$  sampai dengan 0,5  $\mu\text{m}$ , yang bersesuaian dengan frekuensi antara  $6 \times 10^{15}$  Hz hingga  $20 \times 10^{15}$  Hz [7]

Dua properti cahaya yang paling jelas dapat langsung dideskripsikan dengan teori gelombang untuk cahaya adalah intensitas (atau kecerahan) dan warna. Intensitas cahaya merupakan energi yang dibawanya persatuan waktu dan sebanding dengan kuadrat amplitudo gelombang. Warna cahaya berhubungan dengan panjang gelombang atau frekuensi cahaya tersebut. Cahaya tampak yaitu cahaya yang sensitif pada mata kita jatuh pada kisaran 400 nm sampai 750 nm. Kisaran ini dikenal sebagai spektrum tampak, dan di dalamnya terdapat warna ungu sampai merah [7].



Gambar 2. 2.Spektrum Cahaya

Cahaya matahari merupakan gabungan cahaya dengan panjang gelombang dan spektrum warna yang berbeda-beda. Spektrum gelombang elektromagnetik meliputi gelombang radio dan televisi, gelombang mikro, gelombang inframerah, gelombang tampak (*visible light*), gelombang ultraviolet, sinar X dan sinar gamma. Dari spektrum gelombang elektromagnetik tersebut hanya bagian yang sangat kecil yang dapat ditangkap oleh indera penglihatan yaitu cahaya tampak (*visible light*) (gambar 2.2). Pada gambar 2.2 dapat dilihat perbedaan panjang gelombang dan frekuensi dari cahaya tampak menimbulkan warna yang berbeda yaitu merah, jingga, kuning, hijau, biru, nila dan ungu yang disebut juga spektrum tampak.

## **2.5. *Color Threshold***

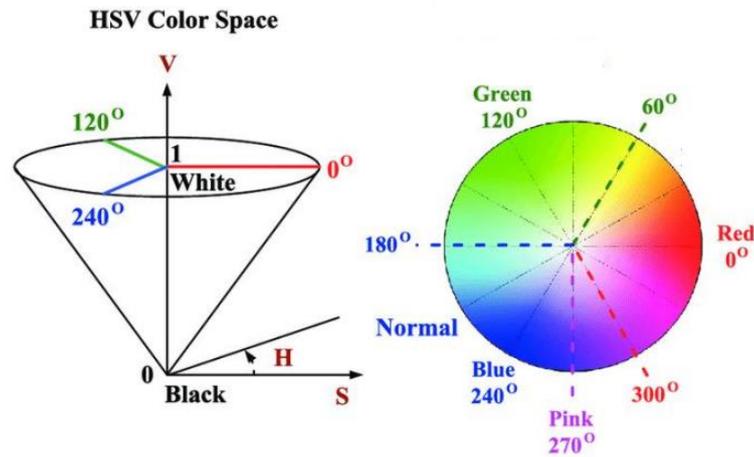
*Color Threshold* atau *Color Model* merupakan cara standar untuk menspesifikasikan suatu warna tertentu, dengan mendefinisikan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk ke dalam suatu model tertentu. Pada manusia, warna ditentukan dengan atribut kecerahan, corak dan keragamannya. Sebuah komputer mendeskripsikan warna berdasarkan atribut warna pokok seperti merah (R), hijau (G), biru (B) untuk mendapatkan warna yang sesuai dengan warna di dunia nyata.

*Color threshold* mempunyai banyak variasi, hal ini bergantung pada mesin yang dipakai untuk menampilkan atau memproduksi warna tersebut. Contohnya ada beberapa printer yang menggunakan ruang warna CMYK dan LCD yang memakai ruang warna RGB. Beberapa contoh *color threshold* adalah RGB (*Red Green Blue*), CMYK (*Cyan Magenta Yellow Black*), HSL (*Hue Saturation Lightness*), *Hue Saturation Value*, *Luminance-Chrominance* dan sebagainya. Beberapa mesin terikat pada satu model warna tertentu.

### **2.5.1. HSV (Hue Saturation Value)**

Model warna HSV pertama kali dideskripsikan oleh Alvy Ray Smith pada tahun 1978. Model warna HSV mempunyai 3 variabel berhubungan antara *Hue*, *Saturation*, dan *Value*. *Color Threshold* atau ruang warna HSV berdasarkan pada koordinat polar. HSV adalah versi kubus RGB dengan

transformasi linear

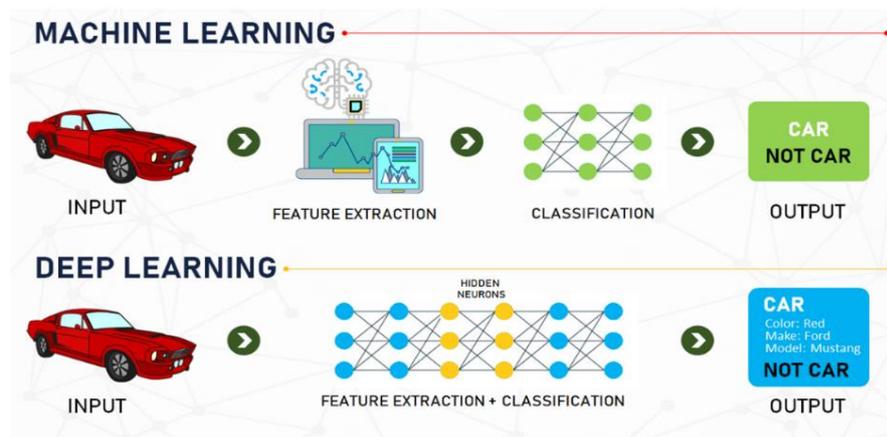


Gambar 2.3. *Hue Saturation Value*

Pada Gambar 2.3 merupakan penggambaran komponen warna HSV yang mempunyai spesifikasi deskripsi pada masing-masing komponennya. *Hue* adalah deskripsi untuk derajat warna pada lingkaran yang menunjukkan perbedaan warna, walaupun putaran derajat 360, namun nilai *hue* dinormalisasi dalam jarak 0–25 dengan awal titik 0 ialah warna merah. *Saturation* adalah representasi kroma/saturasi pada warna, semakin rendah saturasi maka warna akan memudar dan mendekati putih sedangkan jika semakin tinggi maka warna akan semakin dalam atau jelas (perbedaan akan semakin mencolok). *Value* adalah representasi dari kecerahan warna, dengan jarak 0 – 255 dimana 0 adalah warna hitam sedangkan 255 adalah warna putih.

## 2.6. Metode Classification

Ada beberapa metode yang digunakan dalam *Image Classification*. Beberapa metode yang termasuk Teknik pada *Machine Learning* dan *Deep Learning* telah menjadi pendekatan yang paling populer digunakan untuk masalah pengenalan objek ini. Metode pada kedua Teknik tersebut sama-sama teknik untuk mempelajari dan mengidentifikasi objek pada gambar, namun mereka berbeda dalam pelaksanaannya.



Gambar 2.4. *Machine Learning* (mathworks.com, 2016)

Gambar 2.4 merupakan gambaran perbedaan dari Teknik *Machine Learning* dan *Deep Learning*. Pada Teknik *Machine Learning*, membutuhkan pendefinisian gambar terlebih dahulu agar dapat dibaca oleh komputer. Sedangkan pada Teknik *Deep Learning*, metode yang dijalankan lebih sederhana tapi proses pembelajaran lebih panjang dan mampu menghasilkan akurasi yang tinggi.

## 2.7. *Machine Learning*

*Machine Learning* adalah Teknik untuk menganalisis data yang mengajarkan komputer untuk melakukan kemampuan alami manusia dan hewan (pembelajaran dari pengalaman). Algoritma *Machine Learning* menggunakan metode komputasi untuk “mempelajari” informasi dari data tanpa bergantung pada persamaan yang telah ditentukan sebagai model. Algoritma akan beradaptasi untuk meningkatkan kinerja mereka ketika jumlah sampel atau data *training* yang tersedia untuk pembelajaran meningkat.

Dengan populernya istilah *Big Data*, teknik *Machine Learning* merupakan teknik kunci untuk mengatasi berbagai permasalahan tersebut, seperti:

- Sistem keuangan, untuk *Credit Scoring*, *Trading*, dan berbagai masalah akuntansi.
- *Image Processing* dan *Computer Vision*, untuk pengenalan wajah, *Motion Gesture* dan *Object Recognizing*.
- Biomedika, untuk mendeteksi tumor, penyesuaian DNA, Pengenalan sidik jari

dan untuk berbagai bidang lainnya.

Dalam proses pembelajaran, *Machine Learning* ini umumnya menggunakan dua skenario pembelajaran yaitu *Supervised Learning* dan *Unsupervised Learning*. Perbedaan keduanya ada pada bagaimana cara mereka belajar [8].

- *Supervised Learning*

Pada skenario *Supervised Learning*, algoritma yang digunakan seolah-olah akan dilatih terlebih dahulu untuk melakukan prediksi maupun klasifikasi menggunakan data latih hingga mencapai target yang ditetapkan oleh user. Biasanya algoritma akan dilatih dengan beberapa macam data latih berbeda untuk mencapai beberapa target tertentu sesuai dengan kualifikasi yang diberikan user. Ketika algoritma sudah mencapai atau sangat mendekati target, maka algoritma sudah siap untuk diuji.

Sebagai contoh untuk klasifikasi buah mangga matang dan mangga busuk berbasis gambar. Sebelum melakukan klasifikasi, perlu disiapkan data latih berupa beberapa foto mangga matang dan mangga busuk. Kedua data latih tersebut kemudian dilatih menggunakan algoritma *Neural Network* dengan dua target yang berbeda. Setelah data latih yang ditentukan mencapai target, maka algoritma tersebut siap digunakan untuk menguji gambar. *Supervised Learning* menggunakan dua teknik untuk membangun sebuah model yaitu, model klasifikasi dan model regresi. Ada beberapa algoritma yang termasuk dalam *supervised Learning* ini seperti: *Neural Network*, *Naïve Bayes Classifier*, *Decision Tree*, Regresi linear berganda, Analisis waktu dan beberapa algoritma lainnya.

- *Unsupervised Learning*

Berbeda dengan Teknik *supervised*, Teknik *unsupervised* tidak membutuhkan data latih untuk melatih algoritma. Pendekatan yang digunakan oleh Teknik *Unsupervised Learning* ini adalah dengan melakukan pembelajaran pada pola data tersebut terbentuk, algoritma akan mengelompokkan data sesuai dengan pola data lain yang sejenis. Berdasarkan model matematikanya, Teknik *unsupervised* ini tidak memiliki

target variabel.

Pemodelan yang paling sering digunakan oleh Teknik *unsupervised* ini adalah *Clustering Model*. Model ini digunakan untuk menganalisis suatu data untuk menemukan pola tersembunyi atau mengelompokkan suatu data. *Model Clustering* ini biasanya diaplikasikan untuk *market research*, tes psikologi dan lainnya. Algoritma yang biasa digunakan untuk *unsupervised Learning* ini seperti: *Fuzzy C-means*, *K-means*, *Hierarchical Clustering* dan sebagainya.

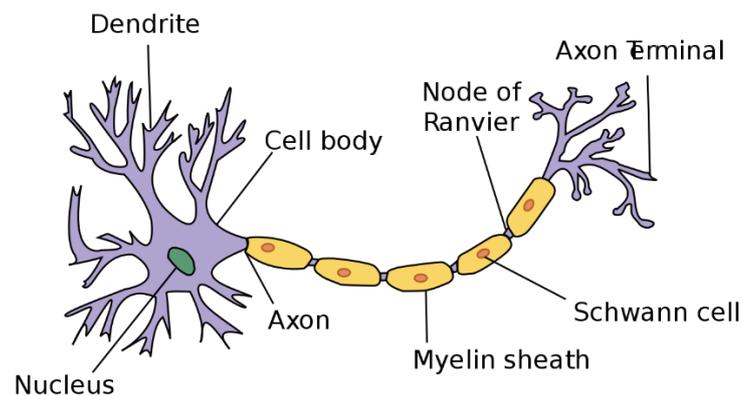
Dalam kasus *object recognition*, saat ini proses *Learning* sebenarnya bisa melakukan keduanya, tergantung ekstraksi seperti apa yang dijalankan. Tapi dalam beberapa penelitian sebelumnya, Teknik *unsupervised* dengan metode klasifikasi memberikan hasil dengan akurasi tinggi. Salah satu metode klasifikasi *unsupervised Learning* yang cukup sering digunakan untuk *object recognition* adalah model klasifikasi menggunakan algoritma *Neural Network* (Jaringan Syaraf Tiruan). Algoritma *Neural Network* ini sering digunakan untuk pengenalan pola seperti kasus *object recognition* ini karena kemampuannya yang unik, yaitu kemampuan untuk meniru proses pembelajaran seperti otak manusia [9].

### **2.7.1. Artificial Neural Network**

Algoritma *Artificial Neural Network* (ANN) muncul pertama kali ketika manusia ingin menghubungkan antara kemampuan otak manusia yang mampu untuk belajar dan komputer yang memiliki kemampuan untuk memproses banyak data dan menyimpannya. Karena banyak sekali masalah yang tidak mampu dirumuskan oleh algoritma biasa. Contohnya untuk menentukan harga sebuah perumahan atau *real estate*. Otak manusia mampu untuk memperkirakan kalkulasi harga yang sesuai, tapi algoritma komputer tidak mampu melakukan hal yang sama seperti itu. Sementara, saat ini segala hal yang sudah dikomputerisasi, untuk meringankan beban manusia. Untuk itulah, algoritma *Artificial Neural Network* (ANN) kini banyak digunakan diberbagai bidang,

baik itu untuk klasifikasi, pengenalan pola ataupun untuk prediksi [10].

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



Gambar 2. 5. Neuron pada otak (medium.com, 2019)

Struktur pada gambar 2.5 adalah bentuk standar satuan unit jaringan otak manusia yang telah disederhanakan (satu neuron). Bentuk standar ini mungkin akan berubah dikemudian hari jika ada ilmuwan lain yang menemukan bentuk standar yang lebih baik ataupun memperbaiki bentuk standar ini. Cara kerja dari satuan sistem syaraf diatas adalah bermula pada sinyal masuk melalui dendrit yang kemudian diproses pada *cell body* dengan fungsi tertentu. Informasi tersebut kemudian akan dikirim ke neuron lain melewati sinapsis (pertemuan antar dua sel). Jika sinyal hasil proses memenuhi nilai ambang batas (*threshold*) maka informasi tersebut akan diterima. Pada kasus tersebut neuron bisa dikatakan sudah diaktifasi. ANN merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan suatu masalah berdasarkan informasi internal maupun eksternal. Seperti halnya otak manusia, selalu memiliki kemampuan untuk belajar dengan melakukan adaptasi [9].

### 2.7.2. Karakteristik *Neural Network*

*Neural Network* memiliki beberapa tipe yang berbeda, akan tetapi hampir semua komponen dan karakteristik yang dimiliki sama. Sama halnya dengan jaringan otak manusia, neural network juga terdiri dari beberapa neuron dan antar neuron juga berhubungan antar satu sama lain. Neuron-neuron tersebut akan melakukan transformasi informasi yang diterima melalui sambungan keduanya menuju neuron yang lain berdasarkan bobot yang dimiliki antar neuron. Berikut pada gambar 2.6 merupakan komponen umum yang ada pada neural network.



Gambar 2.6. Karakteristik *Neural Network*

Penyelesaian masalah dengan jaringan syaraf tiruan tidak memerlukan pemrograman. Jaringan syaraf tiruan menyelesaikan masalah melalui proses belajar dari contoh-contoh pelatihan yang diberikan. Biasanya pada jaringan syaraf tiruan diberikan sebuah himpunan pola pelatihan yang terdiri dari sekumpulan contoh pola. Proses belajar jaringan syaraf tiruan berasal dari serangkaian contoh-contoh pola yang diberikan. Metode pelatihan yang sering dipakai adalah metode belajar terbimbing. Selama proses belajar itu pola masukan disajikan bersama-sama dengan pola keluaran yang diinginkan. Jaringan akan menyesuaikan nilai bobotnya sebagai tanggapan atas pola masukan dan sasaran yang disajikan tersebut [11].

- Faktor bobot

Bobot merupakan suatu nilai yang mendefinisikan tingkat atau kepentingan hubungan antara suatu node dengan node yang lain. Semakin besar bobot suatu hubungan menandakan semakin pentingnya hubungan kedua node

tersebut. Bobot merupakan suatu hubungan berupa bilangan real maupun integer, tergantung dari jenis permasalahan dan model yang digunakan. Bobot-bobot tersebut bisa ditentukan untuk berada didalam interval tertentu. Selama proses pelatihan, bobot tersebut dapat menyesuaikan dengan pola-pola input. Jaringan dengan sendirinya akan memperbaiki diri terus menerus karena adanya kemampuan untuk belajar. Setiap ada suatu masalah baru, jaringan dapat belajar dari masalah baru tersebut, yaitu dengan mengatur Kembali nilai bobot untuk menyesuaikan karakter nilai [8].

- Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas internal (*summation function*) yang mungkin berbentuk linear ataupun non-linear, berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* (bobot dari hasil penjumlahan) dari input. Secara umum, ada 2 *activation function* yang sering digunakan dalam *neural network*, diantaranya [12]:

- 1) Fungsi Sigmoid and Tanh (Non-Linear)

Fungsi sigmoid dan fungsi tanh merupakan salah satu fungsi yang sering digunakan untuk fungsi aktivasi pada *neural network multi layer*. Kedua fungsi aktivasi ini, sama-sama fungsi aktivasi non-linear yang fungsi ini biasanya digunakan untuk klasifikasi 2 *class* atau kelompok data.

Fungsi sigmoid mempunyai rentang antara 0 hingga 1 sedangkan rentang dari fungsi tanh adalah -1 hingga 1. Kedua fungsi ini hampir sama, hanya saja fungsi tanh merupakan pengembangan dari fungsi sigmoid. Tapi kedua fungsi ini memiliki kelemahan yaitu, dapat mematikan gradient, Ketika aktivasi dari neuron mengeluarkan nilai yang berada pada range 0 atau 1, dimana *gradient* di wilayah ini hampir bernilai 0. Kemudian *output* dari sigmoid tidak *zero-centered* [12].

- 2) Fungsi ReLu (non-linear)

ReLu atau *Rectified Linear Unit* menjadi salah satu *activation function*

yang populer dibelakang ini, Vincent Vanhoucke dalam *course Deep Learning*-nya di udacity mengatakan bahwa ReLU merupakan *activation function* favorit para *engineer* yang malas. Karena ReLU pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila  $Z \leq 0$  maka  $Z = 0$  dan apabila  $Z > 0$  maka  $Z = z$ .

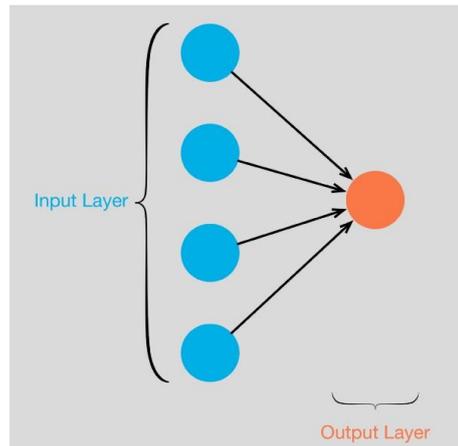
ReLU bisa diimplementasikan hanya dengan membuat pembatas (*threshold*) pada bilangan nol. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan dengan fungsi Sigmoid dan Tanh. Namun aktivasi ini juga memiliki kelemahan yaitu aktivasi ini bisa menjadi rapuh pada proses *training* dan bisa membuat unit tersebut mati [12].

### 2.7.3. Arsitektur Neural Network

Pada *neural network*, neuron-neuron yang ada pada lapisan yang sama memiliki keadaan yang sama. Terdapat faktor penting dalam menentukan sifat suara neuron yaitu bobot (*weight*) dan penggunaan fungsi aktivasi dari neuron tersebut. Arsitektur yang dapat dibentuk ANN bermacam-macam. Dari yang paling sederhana terdiri satu neuron (*single neuron*) sampai yang paling rumit menjadi multi neuron (*multiple neuron*) dalam satu lapis (*single layer*). Ada juga arsitektur dengan jaringan *multiple* neuron dalam *multiple layers*. Beberapa jaringan tersebut memiliki kemampuan yang berbeda-beda. Semakin rumit suatu jaringan, maka persoalan yang dapat diselesaikan menjadi lebih luas. Namun terdapat kelemahan yaitu kerumitan tersebut dapat menimbulkan persoalan tersendiri pada kebutuhan proses *training* dan simulasi (*testing*) yang akan memerlukan waktu lebih lama. Secara umum, algoritma *neural network* dibagi menjadi 3 arsitektur berdasarkan jumlah lapisannya, yaitu [13]:

#### 1. *Single Layer Neural Network*

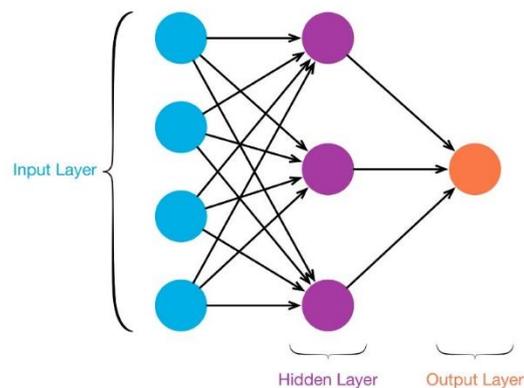
Dari namanya bisa diketahui bahwa arsitektur pada gambar 2.6 ini hanya terdiri dari satu lapisan input dan satu lapisan *output*. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnnya menjadi *output* tanpa harus melalui lapisan tersembunyi.



Gambar 2.7. *Single Layer Neural Net*

## 2. *Multi layer Neural Network*

Arsitektur model ini sering disebut dengan *Multi Layer Perceptron (MLP)* yang terdiri dari banyak lapisan, umumnya memiliki salah satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan *output*, yang dikenal dengan lapisan tersembunyi (*hidden layer*). Umumnya, ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Arsitektur *neural network* dengan banyak lapisan ini umumnya digunakan untuk menyelesaikan permasalahan yang lebih sulit dengan pembelajaran yang lebih rumit. Dengan arsitektur seperti ini terbukti mampu untuk menyelesaikan banyak kasus dengan tingkat akurasi yang lebih tinggi.



Gambar 2.8. *Multilayer Neural Net*

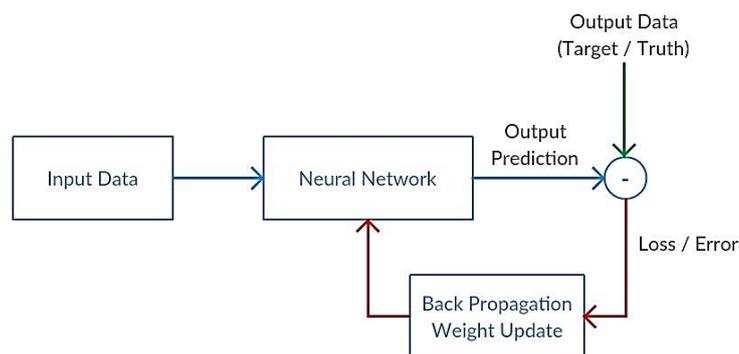
Pada gambar 2.8 tiap koneksi memiliki nilai bobot yang nilai dari setiap bobot itu berbeda-beda. Kemudian pada *hidden layer* dan *output layer* juga memiliki tambahan nilai “input” yang biasa disebut dengan bias. Proses untuk mendapatkan nilai *output* pada *neural network* bisa dihitung menggunakan rumus

$$\text{dot}_j = w_{ji}x_i + b_j$$

Persamaan diatas merupakan persamaan dari proses perkalian pada suatu node *output* pada arsitektur *neural network* yang menggunakan fungsi aktivasi ReLu. Dimana I adalah node pada *input layer*, kemudian j adalah node pada *hidden layer* atau *output layer* dan b adalah bias pada *hidden layer* atau *output layer*.

#### **2.7.4. Backpropagation**

*Neural network* merupakan suatu model komputasi yang sistemnya mengikuti syaraf otak manusia. Dan salah satu ciri khas dari otak manusia adalah melakukan pembelajaran berdasarkan pengalaman. *Neural network* juga begitu, model ini membutuhkan proses pembelajaran untuk mengenali pola dari data yang dipelajarinya. Pembelajaran ini bertujuan untuk melakukan suatu proses dalam menentukan nilai bobot (*weight*) yang tepat untuk masing-masing input. Proses untuk melakukan pembelajaran ini biasanya terjadi pada saat proses pelatihan data (*training data*) biasanya disebut dengan *backpropagation*. *Backpropagation* merupakan salah satu dari metode pelatihan pada jaringan syaraf tiruan, dimana ciri dari metode ini adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan [8]. Jadi *backpropagation* ini bekerja dengan cara melakukan update nilai bobot pada neuron di *layer* sebelumnya.



Gambar 2.9. *Backpropagation*

Seperti pada gambar 2.9, tugas dari metode *backpropagation* adalah dengan melakukan *update* bobot berdasarkan nilai *error* yang didapatkan dari perbandingan antara nilai *output* dan target *output*. Proses *training* menggunakan *backpropagation* secara garis besar dari dua tahap; tahap maju (*forward pass*) dan tahap mundur (*backward pass*) [14].

Pada tahap maju, akan dilakukan operasi dot antar nilai bobot pada node input dengan node pada *layer* didepannya. Operasi ini dilakukan untuk mendapatkan nilai *output*, proses ini disebut dengan *forward pass*. Kemudian hasil prediksi *output* akan dibandingkan dengan target dengan menggunakan sebuah fungsi yang biasa disebut dengan *loss function*. Secara sederhana *loss function* adalah fungsi yang digunakan untuk mengukur seberapa bagus performa dari *neural network* kita dalam melakukan prediksi terhadap target. Ada berbagai macam *loss function* yang bisa digunakan, namun yang paling sering digunakan adalah *squared error*, *loss function* yang sering digunakan untuk linear regresi. Sedangkan untuk klasifikasi dengan beberapa kelas yang biasa digunakan untuk mencari *loss function* menggunakan *squared error* adalah seperti berikut;

$$Loss = (Prediction - target)^2$$

Kemudian pada tahap kedua akan dilakukan propagasi mundur (*backward pass*). Tahapannya hampir sama dengan *forward pass*, pada

*backward pass* nilai *loss* yang didapatkan akan mengalir dan dijadikan sebagai bobot pengali menuju semua node pada *layer* sebelumnya untuk dicari *gradient* nya. Semisal ingin melakukan *update* pada parameter  $W_{k10}$ , pertama akan dicari seberapa besar perubahan *loss* berdasarkan *output* dengan cara mencari turunan parsial dari *loss function* terhadap *output*.

Setelah mendapatkan nilai *gradient loss* dari parameter yang dituju, maka akan diberlakukan *update* nilai dari parameter tersebut menggunakan algoritma *stochastic gradient descent* (SGD). Secara sederhana, algoritma SGD ini bertujuan untuk menurunkan nilai *loss* pada parameter dengan cara mengurangi bobot asli dari parameter tersebut dengan “Sebagian” dari nilai *gradient loss* yang sudah didapatkan sebelumnya. Sebagian disini diwakiri oleh nilai dari *Learning rate* [12]. Jadi SGD ini akan menurunkan nilai *loss* pada parameter secara terus menerus hingga mencapai titik minimum secara optimal sebanyak iterasi yang terjadi. Rumus untuk melakukan *update* parameter menggunakan SGD sebagai berikut.

$$W'_{k10} = W_{k10} - \alpha \left( \frac{\partial Loss}{\partial w_{k10}} \right)$$

Dari persamaan diatas, untuk melakukan *update* pada parameter  $W_{k10}$  maka nilai bobot asli dari parameter  $W_{k10}$  akan dikurangi dengan Sebagian dari nilai *gradient* yang didapat. Sebagian disini adalah nilai dari *Learning rate* ( $\alpha$ ) yang dikalikan dengan nilai *gradient loss* yang sudah dicari sebelumnya. Proses ini akan dilakukan terus menerus pada semua parameter yang terhubung hingga sampai pada node input. Setelah itu akan dilakukan proses *forward pass* dan *backward pass* lagi. Proses ini akan dilakukan terus menerus hingga proses iterasi selesai.

## 2.8. Deep Learning

Teknik *Deep Learning* adalah Teknik *Machine Learning* yang mengajarkan komputer agar memiliki kemampuan menyerupai kemampuan alami manusia, yaitu belajar dari pengalaman. Dengan adanya teknologi *Deep Learning*, produk-produk seperti *Self-Driving Car*, *Face Recognition* atau

*Voice Recognition* dapat dihasilkan. Ini adalah kunci untuk mengontrol suara diperangkat konsumen seperti ponsel, tablet, TV, dan lainnya. Teknik *Deep Learning* akhir-akhir ini mendapatkan banyak perhatian dari para peneliti karena pencapaiannya untuk melakukan klasifikasi langsung dari gambar, teks, atau suara. Ketika menggunakan *Deep Learning*. Penggunaan *Deep Learning* ini dapat mencapai akurasi ketepatan yang tinggi, terkadang melebihi kinerja tingkat manusia pada beberapa kasus. Model dilatih dengan menggunakan sejumlah data yang besar dan dipadukan dengan arsitektur *neural network* yang mengandung banyak lapisan [15].

Meskipun istilah *Deep Learning* pada *neural network* sudah ada sejak 2006, tapi baru beberapa tahun ke belakang istilah ini dimunculkan lagi. Hal itu dikarenakan dua hal, yang pertama metode *Deep Learning* membutuhkan dataset dalam jumlah besar, dengan begitu kemampuan akurasi yang tinggi dari *Deep Learning* ini baru terlihat benar-benar nyata. Alasan kedua, *Deep Learning* dengan dataset yang besar membutuhkan perangkat komputasi dengan performa yang tinggi untuk meningkatkan kecepatan pemrosesan. Alasan kedua inilah yang menjadi sebab metode *Deep Learning* kini menjadi sangat digemari oleh peneliti, karena saat ini perangkat komputasi seperti GPU dengan performa tinggi dan *cloud computing* yang semakin berkembang membuat metode ini cepat berkembang [15].

Salah satu metode *Deep Learning* yang memanfaatkan *Deep neural network* adalah *convolutional neural network* (CNN atau Convnet). Nama tersebut diambil dari operasi terpenting yang ada pada metode tersebut yaitu *Convolutional*, yang kemudian digabung dengan metode *neural network*. CNN ini merupakan model terbaru dari algoritma *Deep neural network* yang dikembangkan untuk memproses data dalam bentuk 2D data, seperti gambar. Berbeda dari algoritma pengolahan citra lainnya, algoritma CNN ini menghilangkan ekstraksi fitur secara manual. Jadi tidak perlu untuk mengekstraksi fitur untuk mengklasifikasikan gambar tersebut. Fitur-fitur yang relevan tidak perlu dilatih terlebih dahulu, mereka akan mempelajari fitur dari sebuah gambar. Ketika jaringan tersebut dijalankan. Ekstraksi fitur otomatis ini

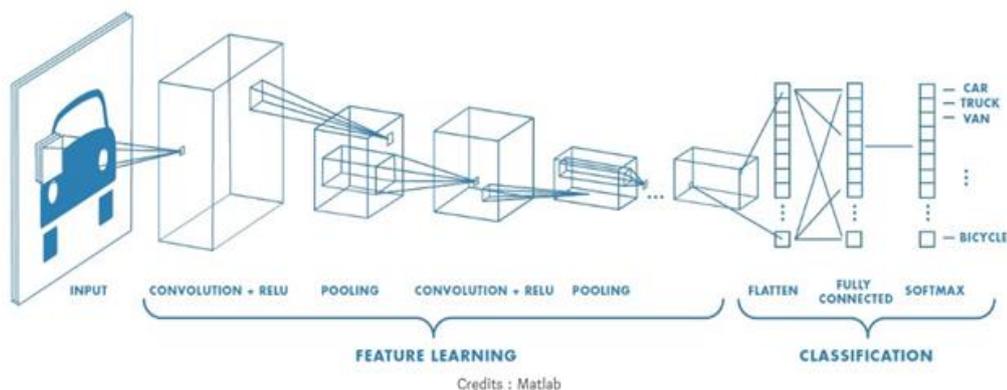
membuat model *Deep Learning* memiliki akurasi yang tinggi untuk melakukan tugas *computer vision* seperti klasifikasi objek [15].

### **2.9. Convolutional Neural Network**

*Convolutional Neural network* (CNN) adalah salah satu algoritma paling populer untuk *Deep Learning*, sebuah *Machine Learning* yang model pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada media dua dimensi seperti gambar, video, teks, atau suara. Algoritma CNN akan sangat berguna khususnya Ketika digunakan untuk mencari pola pada suatu gambar kemudian mengenali objek pada gambar tersebut. Bukan hanya pada objek atau benda saja, CNN ini sebenarnya juga bisa digunakan untuk mengenali wajah yang selama ini perlu segmentasi untuk meningkatkan akurasi. Penelitian awal mendasari penemuan CNN ini pertama kali dilakukan oleh Hubel dan Wiesel mengenai *visual cortex* pada indera penglihatan kucing. Pada dasarnya klasifikasi citra menggunakan MLP sudah bisa dilakukan, akan tetapi Ketika digunakan untuk melakukan klasifikasi data dalam jumlah banyak, akurasi yang didapatkan pun menurun. Oleh karena itu, algoritma CNN ini dikembangkan karena algoritma ini mampu untuk mempelajari langsung data yang ada pada gambar, kemudian menggunakan pola yang didapatkan untuk mengklasifikasi [16].

Berbeda dengan arsitektur MLP, arsitektur CNN ini sebenarnya lebih kompleks dan memiliki proses yang cukup panjang sebelum masuk tahap klasifikasi. Namun secara garis besar ada dua tahapan pemrosesan yang dilakukan oleh algoritma CNN, yaitu tahap *Feature Learning* dan tahap *classification*. Tahap *Feature Learning* merupakan tahap dimana gambar yang diinputkan akan diekstraksi untuk dipelajari *value* dari gambar tersebut. Proses inilah yang membedakan algoritma CNN dengan MLP. Jika MLP, hanya bisa menggunakan satu proses ekstraksi dalam sekali input untuk melakukan klasifikasi. Sedangkan CNN ini, untuk melakukan klasifikasi bisa menggunakan banyak sekali ekstraksi dalam sekali input. Banyak ekstraksi ini disimpan dalam bentuk kedalaman gambar (*depth*). Proses *Feature Learning* ini sangat

bergantung pada kedalaman suatu gambar. Semakin dalam dalam suatu gambar maka semakin banyak ekstraksi yang didapatkan sehingga pola yang didapat juga semakin jelas terbentuk [16]. Value inilah yang nantinya akan dikonversi menjadi vektor dan kemudian masuk pada tahap klasifikasi. Pada tahap klasifikasi ini, model *neural network* akan digunakan untuk melakukan klasifikasi objek berdasarkan kelasnya.



Gambar 2.10. Arsitektur CNN

Berdasarkan gambar 2.10, arsitektur dari algoritma dibagi menjadi dua tahapan pemrosesan. Pada tahap *Feature Learning*, secara umum ada tiga lapisan pemrosesan ekstraksi fitur. Lapisan-lapisan ini sering disebut dengan *convolutional layer*, *activation* dan *pooling layer*. Lapisan-lapisan ini akan melakukan operasi khusus untuk membentuk kedalaman data agar mendapatkan pola secara spesifik. Ketiga *layer* tersebut memiliki urutan proses yang tidak harus selalu sama, dalam artian prosesnya bisa dimodifikasi sesuai dengan kebutuhan. Konvolusi antara matriks input dengan kernel ukuran tertentu.

### 2.10. Convolutional Layer

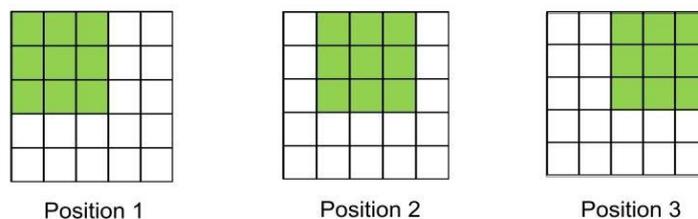
*Convolutional Layer* merupakan proses pertama yang harus dilakukan dalam tahapan *Feature Learning*. Pada *convolution layer*, akan dilakukan operasi konvolusi antara matriks input dengan kernel yang ada pada matriks *filter*. Konvolusi adalah operasi perkalian antara dua matriks yang kemudian hasilnya dijumlahkan [9]. Hasil dari proses konvolusi pada algoritma CNN ini disebut dengan *Feature map*.

*Convolution layer* ini adalah bagian terpenting dalam membentuk kedalaman (*depth*) data pada suatu *Feature*. Sebagai input-an, kedalaman suatu gambar didefinisikan dengan banyaknya chanel gambar tersebut. Sebagai contoh, jika gambar tersebut berukuran 32x32x3, angka 3 yang menotasikan jumlah *layer* warna pada gambar tersebut juga bisa dikatakan sebagai ukuran kedalaman dari gambar tersebut. Karena proses *Learning* pada algoritma CNN ini tergantung pada kedalaman data, untuk membuat data *inputan* lebih dalam lagi maka *inputan* akan dikonvolusi dengan sejumlah matriks  $K$  (kernel) yang disebut dengan *filter* pada tahapan *convolutional layer* ini. Dimana semua *filter* memiliki ukuran yang hampir selalu persegi dengan kedalaman sesuai dengan yang ditentukan.

Proses konvolusi antara gambar input dengan sejumlah *filter* akan menghasilkan satu set *Feature map* yang berisi berbagai variasi ekstraksi seperti *edge detection*, *smoothing*, *sharpening* dan sebagainya.

- *Stride*

Berdasarkan gambar diatas, *Feature map* didapatkan dari proses konvolusi antara gambar dengan sejumlah *filter*  $K$  *filter*. Proses perhitungan untuk mendapatkan *Feature map* tersebut adalah dengan cara melakukan operasi perkalian antara matriks pada gambar dengan matriks yang kecil pada  $K$  *filter*. Proses perkalian ini dimulai dari bagian pojok kiri atas matriks pada gambar. Hasil perkalian dari setiap elemen pada matriks tersebut kemudian akan dijumlahkan untuk menghasilkan satu elemen baru dari hasil perhitungan tersebut. Setelah itu matriks  $K$  *filter* tersebut akan digeser ke kanan, sejumlah “*stride*” yang ditentukan. *Stride* ( $S$ ) adalah parameter yang menentukan berapa jumlah pergeseran *filter* yang akan digunakan. Jika nilai *stride* adalah 1, maka  $K$  *filter* akan bergeser sebanyak 1 piksel secara horisontal ke kanan hingga selesai, kemudian bergeser secara vertikal ke bawah.

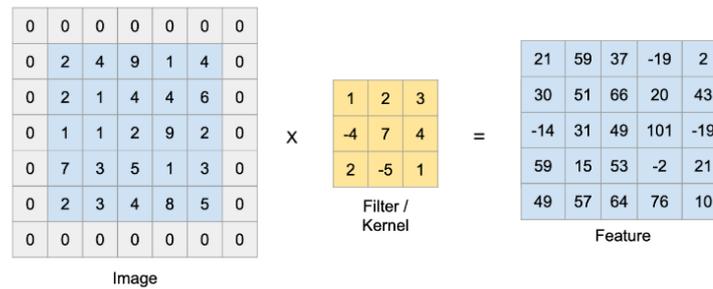


Gambar 2.11.Stride

Seperti pada gambar 2.11, bagian pada matriks gambar yang sedang terjadi proses konvolusi ditandai oleh blok yang berwarna hijau. Pada konvolusi dengan kernel seluas volume kernel tersebut dan konvolusi ini diawali dari bagian pojok kiri matriks gambar. Setelah hasil konvolusi selesai dijumlahkan, kernel akan digeser ke kanan sebesar jumlah *stride* yang ditentukan seperti pada step dua. Dalam menentukan jumlah *stride* ini juga tidak boleh sembarangan, harus memastikan jumlah *stride* yang ditentukan tersebut sesuai dengan luas matriks gambar.

- *Padding*

*Padding* adalah parameter yang menentukan jumlah piksel (berisi nilai 0) yang akan ditambahkan disetiap sisi dari matriks *input*. *Padding* ini digunakan untuk memanipulasi dimensi *output* dari *feature map*. *Padding* (P) ini bisa dibilang salah satu parameter yang cukup penting untuk mendapatkan hasil konvolusi yang cukup dalam. Karena dimensi *output* dari proses konvolusi ini selalu lebih kecil dari dimensi inputnya. Sementara *output* akan digunakan lagi pada *convolution layer* berikut, otomatis informasi pada gambar akan makin banyak terbuang, apalagi jika *filter size* yang digunakan cukup besar. Dengan menggunakan dimensi yang sama dengan input atau setidaknya tidak berkurang secara drastic. Sehingga bisa menggunakan *convolution layer* yang lebih *deep* dan mendapatkan lebih banyak ekstraksi dari input.

Gambar 2.12. *Padding*

Seperti gambar 2.12, matriks gambar yang awalnya berukuran 5x5 kemudian ditambah lagi dengan *padding* 1 ( $P=1$ ) dengan begitu pada setiap sisi-sisi matriks akan ditambahkan 1 piksel dengan nilai 0 pada setiap sisinya. Penggunaan *padding* ini selain untuk memanipulasi *feature map* nantinya, Teknik ini juga bermanfaat untuk meningkatkan performa *convolution layer*. Karena dengan menambahkan *padding*, maka *filter* akan focus pada informasi yang berada diantara *padding* tersebut.

Seperti yang sudah dijelaskan sebelumnya dalam menentukan *stride* haruslah pas dengan volume matriks gambar. Salah satu cara untuk melihat bahwa *stride* yang akan digunakan sesuai atau tidak, bisa menggunakan rumus untuk menghitung dimensi dari *feature map* ini. Jika hasil *output* berupa bilangan integer, maka *stride* yang digunakan sudah sesuai.

$$Output = \frac{W - N + 2P}{S} + 1$$

Sebagai contoh dari proses konvolusi pada *convolution layer* ini, misalnya memiliki input dengan ukuran 4x4 yang akan dikonvolusi dengan kernel berukuran 3x3 dengan *stride* 2 dan *padding* 1. Jika dihitung menggunakan rumus diatas maka *feature map* yang didapatkan adalah 2, berarti *feature map* yang didapatkan adalah matriks berukuran 2x2. Setelah semua matriks pada input selesai di konvolusi, proses selanjutnya adalah menentukan bobot dari *output* tersebut harus “aktif” atau tidak menggunakan fungsi tertentu. Pada

kasus CNN ini, fungsi aktivasi yang sering digunakan untuk mengaktifkan hasil dari proses konvolusi adalah ReLu [17].

### **2.11. Pooling Layer**

Ada dua metode untuk mengurangi ukuran dari volume input, yang pertama *convolutional layer* dengan *stride*  $> 1$ , seperti contoh diatas dan yang kedua adalah *pooling layer*. Merujuk pada arsitektur CNN diawal pembahasan tadi, tahap *pooling layer* ini terletak setelah *convolution layer*. Pada dasarnya *pooling layer* ini terdiri dari sebuah *filter* dengan ukuran tertentu yang akan dioperasikan dengan *stride* tertentu ke *feature map* hasil *convolution layer*.

Ada dua metode *pooling* yang biasa digunakan di *pooling layer* yaitu *average pooling* dan *max pooling*, nilai yang diambil adalah nilai rata-rata dari matriks yang mengalami operasi *pooling*. Sedangkan *max pooling* adalah dengan mengambil nilai tertinggi. Lapisan *pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume *output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*.

Lapisan *pooling* bekerja disetiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Sebagai contoh menggunakan *max pooling 2x2* dengan *stride 2*, maka pada setiap pergeseran *filter*, nilai maksimum pada area  $2 \times 2$  piksel tersebut yang akan dipilih. Pada *feature map* hasil konvolusi dilakukan operasi *max pooling*. Operasi untuk mengambil nilai tertinggi pada batasan ukuran *filter* tertentu. Dari proses *max pooling* dengan *filter 2x2* dan *stride 2* tersebut didapatkan *feature map* baru dengan ukuran yang lebih kecil yaitu  $2 \times 2$  [17].

### **2.5.3. Fully Connected Layer**

*Fully connected layer* merupakan lapisan dimana semua neuron dari lapisan-lapisan sebelumnya dijadikan satu untuk dilakukan proses klasifikasi menggunakan *neural network*. Pada dasarnya lapisan ini sama halnya dengan

lapisan *neural network* biasa, bisa dalam bentuk *single net* ataupun MLP. Tapi sebelum dilakukan proses klasifikasi, *feature map* yang dihasilkan dari *feature Learning* tersebut masih berbentuk multi dimensional *array*, sehingga perlu mengubah menjadi bentuk vektor, Teknik ini disebut *flatten*. *Flatten* merupakan Teknik untuk *reshape feature map* menjadi sebuah vektor agar bisa digunakan sebagai input dari *fully connected layer*. Jadi input dari *fully connected layer* terdiri dari satu neuron hasil *reshape feature map* tadi menjadi vektor. Setelah dilakukan *flatten* semua bobot tersebut akan diklasifikasi sesuai dengan banyaknya kelas [9].

Pada *fully connected layer* ini, tidak ada ketentuan pasti untuk menggunakan jaringan *single net* ataupun MLP. Karena proses ekstraksi dan *feature Learning* sudah dilakukan pada *layer* sebelumnya, beban jaringan *neural network* untuk melakukan proses klasifikasi relatif lebih ringan. Tapi dalam beberapa penelitian sebelumnya, menunjukkan dengan menggunakan jaringan MLP dapat meningkatkan akurasi klasifikasi meskipun perbedaannya tidak terlalu signifikan [9].