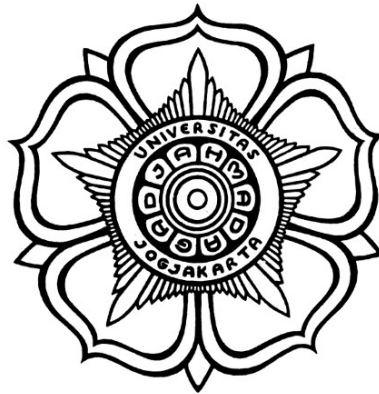


**USULAN PENELITIAN S2**

**PERBAIKAN PENGENALAN GESTUR TANGAN TERHADAP KONDISI  
LINGKUNGAN BERINTENSITAS CAHAYA RENDAH MENGGUNAKAN  
RETINEX**



**ANTHONIUS ADI NUGROHO**  
19/448690/PPA/05773

**PROGRAM MAGISTER ILMU KOMPUTER  
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2020**

## HALAMAN PENGESAHAN USULAN PENELITIAN S2

Judul Tesis : Perbaikan Pengenalan Gestur Tangan Terhadap Kondisi  
Lingkungan Berintensitas Cahaya Rendah Menggunakan  
Retinex  
Nama Mahasiswa : Anthonius Adi N  
NIM : 19/448690/PPA/05773

Proposal telah diuji pada tanggal \_\_\_\_\_ dan sudah diperbaiki sesuai saran  
penguji dan sudah disetujui para penguji.

### Nama Penguji

### Tanda Tangan

1. .

1.

2. .

2.

Yogyakarta, .....2020

Mengetahui,  
Pembimbing

Pengusul



Dr. Raden Sumiharto, S.Si., M.Kom.  
NIP. 197706252005011001

Anthonius Adi N.  
NIM. 19/448690/PPA/05773

## DAFTAR ISI

<b>HALAMAN PERNYATAAN</b>	<b>ii</b>
<b>DAFTAR ISI</b>	<b>iii</b>
<b>DAFTAR TABEL</b>	<b>v</b>
<b>DAFTAR GAMBAR</b>	<b>vi</b>
<b>INTISARI</b>	<b>vii</b>
<b>I PENDAHULUAN</b>	<b>1</b>
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
<b>II TINJAUAN PUSTAKA</b>	<b>5</b>
<b>III LANDASAN TEORI</b>	<b>9</b>
3.1. Computer Vision	9
3.2. Citra Digital	9
3.2.1. Citra RGB	10
3.2.2. Citra HSV	10
3.3. Operasi Morfologi	11
3.3.1. Erosi	11
3.3.2. Dilasi	12
3.3.3. <i>Opening</i>	12
3.3.4. <i>Closing</i>	12
3.4. <i>Hand Gesture Recognition</i>	12
3.4.1. <i>American Sign Language(ASL)</i>	12
3.5. Retinex(Single Scale Retinex)	13
3.5.1. Multiscale Retinex(MSR)	14
3.5.2. Multiscale Retinex Color Restoration(MSRCR)	15
3.6. MobileNet	15
3.6.1. Depthwise Separable Convolution	16
3.7. MobileNetV2	17
3.8. Convolutional Neural Network(CNN)	19
3.8.1. Local Receptive Fields	19
3.8.2. Shared Weight	21

3.8.3.	Konvolusi . . . . .	21
3.8.4.	Fungsi Aktivasi . . . . .	22
3.9.	Evaluasi . . . . .	23
3.9.1.	Confussion Matrix . . . . .	23
3.9.2.	Average Precicion(AP) . . . . .	24
<b>IV</b>	<b>METODOLOGI PENELITIAN . . . . .</b>	<b>26</b>
4.1.	Studi Literatur . . . . .	26
4.2.	Alat dan Bahan . . . . .	26
4.2.1.	Alat . . . . .	26
4.2.2.	Bahan . . . . .	26
4.3.	Prosedur Kerja . . . . .	26
4.3.1.	Analisis dan Perancangan Sistem . . . . .	26
4.4.	Pengumpulan Data . . . . .	28
4.5.	Proses Pelatihan . . . . .	29
4.5.1.	Pelatihan Gesture Recognition . . . . .	30
4.5.2.	Pelatihan Object Detection . . . . .	30
4.6.	Pengujian dan Evaluasi . . . . .	31
4.6.1.	Evaluasi Deteksi Tangan . . . . .	31
4.6.2.	Evaluasi Pengenalan Gestur Tangan . . . . .	32
4.6.3.	Pengujian SNR . . . . .	32
4.6.4.	Pengujian Deteksi Tangan Menggunakan Retinex . . . . .	32
4.6.5.	Pengujian Pengenalan Gestur Tangan Menggunakan Retinex . . . . .	34
4.6.6.	Pengujian Sistem Keseluruhan . . . . .	36
<b>V</b>	<b>JADWAL PENELITIAN . . . . .</b>	<b>37</b>
	<b>DAFTAR PUSTAKA . . . . .</b>	<b>38</b>

## DAFTAR TABEL

2.1	Tinjauan Pustaka . . . . .	7
2.2	Lanjutan Tabel . . . . .	8
3.1	Blok Arsitektur <i>Bottleneck MobileNetV2</i> (Sandler et al., 2018) . . .	18
4.1	Pengujian Deteksi Tangan . . . . .	33
4.2	Pengujian Pengenalan Gestur Tangan . . . . .	35
5.1	Jadwal Penelitian . . . . .	37

## DAFTAR GAMBAR

3.1	Koordinat Citra Digital (Sarifudin.,2015) . . . . .	9
3.2	Citra RGB . . . . .	10
3.3	Ruang warna HSV (Kolkur et al., 2017) . . . . .	11
3.4	American Sign Language (Barczak et al., 2011) . . . . .	13
3.5	Convolutional standard dan depthwise separable (Howard et al., 2017)	17
3.6	Struktur Dasar Pada <i>MobileNetV2</i> (Google AI Blog., 2018) . . . . .	18
3.7	Ilustrasi Citra 28x28 Piksel (Nielsen., 2015) . . . . .	19
3.8	Ilustrasi <i>local receptive fields</i> (Nielsen., 2015) . . . . .	20
3.9	Ilustrasi Pergesaran <i>local receptive fields</i> (Nielsen., 2015) . . . . .	20
3.10	Ilustrasi Pergeseran dengan Stride = 1 (Nielsen., 2015) . . . . .	21
3.11	Confusion Matrix (Leonard., 2017) . . . . .	23
3.12	Ilustrasi IoU(Hui., 2018) . . . . .	24
3.13	Metrik Evaluasi COCO (Chen et al., 2015) . . . . .	25
4.1	Alur Kegiatan Penelitian . . . . .	27
4.2	Rancangan Pengujian Sistem . . . . .	27
4.3	Sample Dataset ASL Massey University (Barczak et al., 2011) . . . .	28
4.4	Setup Pengambilan Dataset (Barczak et al., 2011) . . . . .	29
4.5	Transfer Learning Pelatihan Gestur . . . . .	30
4.6	Arsitektur Mobilenet V2 (Sandler et al., 2018) . . . . .	31
4.7	Skema Pengujian Deteksi Tangan . . . . .	33
4.8	Skema Pengujian Pengenalan Gestur Tangan . . . . .	34

## INTISARI

### **PERBAIKAN PENGENALAN GESTUR TANGAN TERHADAP KONDISI LINGKUNGAN BERINTENSITAS CAHAYA RENDAH MENGGUNAKAN RETINEX**

Oleh

ANTHONIUS ADI NUGROHO

19/448690/PPA/05773

Bahasa isyarat adalah salah satu bentuk komunikasi non-verbal antar manusia yang memiliki makna tersendiri. *American Sign Language*(ASL) merupakan bentuk dari bahasa isyarat tangan yang digunakan oleh penyandang disabilitas untuk berkomunikasi satu sama lain. Penggunaan bahasa isyarat membantu penyediaan hak atas informasi yang diberikan. Perkembangan ilmu yang sangat maju membantu seseorang dapat memahami suatu bahasa isyarat tanpa harus mempelajari hal tersebut. Salah satu bagian disiplin ilmu tersebut adalah pengenalan gestur, dimana di dalamnya terdapat bahasa isyarat yang dapat diterjemahkan dengan bantuan kamera.

Kamera digunakan sebagai alat yang mengimitasi mata manusia dalam mengenali sebuah gestur. Proses pengenalan bahasa isyarat memiliki beberapa hal penting di dalamnya. Intensitas cahaya merupakan salah satu faktor penting dalam pengambilan sebuah citra yang ditangkap oleh kamera, dimana merupakan suatu tantangan tersendiri ketika informasi citra tidak dapat ditangkap dengan jelas yang menyebabkan penurunan performa. Permasalahan ini dapat diatasi dengan menggunakan teori *Retinex* yang diambil dari kata retina dan cortex. *Retinex* mampu meningkatkan tingkat kecerahan citra dengan cara layaknya mata manusia yang dapat melihat walaupun dalam ruangan minim cahaya. Citra yang telah dilakukan perbaikan kontras akan dilakukan deteksi tangan kemudian akan dikenali bahasa isyarat tersebut menggunakan teknik *Convolution Neural Network*.

***Kata kunci - Retinex, Hand Gesture Recognition, Object Detection***

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Disabilitas adalah kelompok masyarakat yang memiliki keterbatasan yang dapat menghambat partisipasi dan peran serta mereka dalam kehidupan bermasyarakat. Penyandang disabilitas memiliki berbagai kategori, yaitu disabilitas fisik, intelektual, mental dan sensorik. menurut data dari kemenkes tahun 2015, persentase 3 teratas penyandang disabilitas di provinsi indonesia adalah 6.36% kesulitan melihat, 3.76% kesulitan berjalan dan 3.35% kesulitan mendengar (Kemenkes., 2018). Pemerintah Indonesia telah menandatangani konvensi tentang Hak-Hak Penyandang Disabilitas pada tanggal 30 Maret 2007 di New York. Adanya penandatanganan tersebut menunjukkan bangsa indonesia menghormati, melindungi, memenuhi dan memajukan hak-hak penyandang disabilitas. Untuk itu perlu adanya dukungan dari masyarakat dalam mewujudkannya.

Bagi penyandang disabilitas, mereka memiliki hambatan akses dalam melakukan aktivitas sehari-hari. Dengan adanya perkembangan pengetahuan dan teknologi, mereka mulai terbantu dan dapat melakukan aktivitas layaknya masyarakat pada umumnya. Mulai banyak penyandang disabilitas yang melakukan mobilitas tinggi dengan kursi roda, mengakses informasi dengan adanya penerjemah bahasa. Dengan adanya hal tersebut, penyandang disabilitas mendapat tempat dan peranan yang sama dengan masyarakat lainnya.

Teknologi yang semakin dewasa membuat mobilitas penyandang disabilitas menjadi lebih tinggi. Kursi roda mungkin dapat digunakan untuk membantu penyandang bergerak dari suatu tempat ke tempat lain, namun bagi beberapa penyandang disabilitas tertentu yang tidak memiliki kemampuan normal pada kondisi tangan atau lumpuh sebagian tidak dapat menggunakan kursi roda tersebut. Akibatnya perlu adanya orang lain untuk membantu mennggerakan kursi roda tersebut.



Teknologi komputer dan robotika saat ini memiliki peranan penting dalam membantu sebuah permasalahan dari mulai kegiatan industri hingga kegiatan masyarakat. (Posada-Gómez et al., 2007) Membuat kursi roda pintar dengan kontrol gestur tangan, namun pada penelitian tersebut memiliki kelemahan terhadap cahaya. Intensitas cahaya yang rendah membuat sistem kurang mampu mendeteksi kontrol dari gestur tangan, sehingga hanya dapat digunakan dalam keadaan cahaya terang. Penelitian tersebut menggunakan teknologi pemrosesan citra yang dikombinasikan dengan elektronika dan mekanika untuk pergerakan kursi roda.

Pemrosesan citra memiliki beberapa hal fundamental permasalahan diantaranya adalah proses perbaikan citra. Perbaikan citra digunakan untuk memperbaiki sebuah citra yang bermasalah agar informasi citra terlihat lebih jelas secara visual maupun perhitungan. Penggunaan *image processing* sangat dibutuhkan untuk membantu menyelesaikan permasalahan dalam kehidupan sehari-hari. Aplikasi dari implementasi *image processing* beberapa diantaranya adalah pengenalan dan deteksi pada sebuah objek. Untuk menyelesaikan permasalahan tersebut perlu membuat sistem yang tahan terhadap kondisi cahaya berintensitas rendah.

Pengenalan dan deteksi sebuah objek memiliki ruang lingkup yang sangat luas untuk dikembangkan, pada penelitian ini berfokus pada *gesture recognition*. Dalam pemrosesan citra algoritme pada *gesture recognition* dapat diimplementasikan pada komunikasi non verbal ataupun suatu gerakan yang dapat membantu seseorang menyelesaikan permasalahan. Pola pada *gesture* dapat dikenali oleh seseorang dengan cara melihat *gesture* tersebut, hal yang sama terjadi pada kamera yang mengadopsi apa yang dilakukan mata manusia untuk mengenali sebuah objek. *Hand gesture* salah satu contoh implementasi *image processing* yang dapat dikembangkan untuk membantu seseorang menyelesaikan permasalahan dengan menggunakan gestur tangan.

Salah satu faktor yang dapat menurunkan kualitas suatu citra yaitu pencahayaan dari sebuah citra, proses pengambilan citra dalam intensitas rendah akan menghasilkan citra yang buruk (Saputra., 2016). Implementasi algoritme untuk meningkatkan kualitas citra terhadap kondisi cahaya merupakan permasalahan yang

menarik untuk diteliti. Kondisi cahaya pada image processing adalah sesuatu topik yang menantang dalam permasalahan *image processing* dan *computer vision* untuk meningkatkan visibilitas maupun kualitas yang lebih baik dari suatu citra. Beberapa penelitian terdahulu telah melakukan peningkatan algoritme pada kondisi cahaya yang minim untuk suatu citra (Loh et al., 2019). Berkembangnya algoritme dalam ruang lingkup kondisi cahaya yang minim bertujuan mendapatkan kualitas kontras yang lebih jelas untuk dapat dilakukan komputasi lebih lanjut.

*Retinex* merupakan metode yang diusulkan oleh Land dengan memodelkan pencahayaan dan persepsi warna berdasarkan penglihatan mata manusia. Mata manusia dapat membedakan sebuah objek sekalipun dalam kondisi intensitas cahaya yang rendah. Metode *Retinex* terus mengalami berbagai pengembangan dari *Single Scale Retinex* hingga *Multiscale Retinex* berupaya untuk memperoleh keseimbangan kontras dalam pencahayaan berintensitas rendah (Saputra., 2016).

Peningkatan kontras menjadi salah satu solusi untuk meningkatkan visibilitas dari sebuah citra. Penelitian ini menggunakan algoritme *Multiscale Retinex Color Restoration* untuk meningkatkan kontras serta *object detection* untuk melakukan segmentasi pada suatu citra, kemudian untuk melakukan pengenalan sebuah gesture yang telah dilakukan perbaikan kontras, akan dilanjutkan dengan metode *Convolutional Neural Network*.

Implementasi dari peningkatan kontras untuk pengenalan gestur tangan ini diharapkan dapat dibawa untuk menyelesaikan permasalahan pada sistem yang bergantung pada cahaya. Beberapa sistem yang dimaksud seperti kursi roda pintar, ataupun sistem lainnya seperti pengenalan wajah dan pengenalan gestur.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang, deteksi dan pengenalan sebuah objek sangat dipengaruhi oleh cahaya. Intensitas cahaya yang rendah dapat menyebabkan rendahnya akurasi pengenalan gestur tangan.

### 1.3. Batasan Masalah

Penelitian ini memiliki batasan masalah yang bertujuan untuk tidak memperluas pokok bahasan. Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Acuan dataset menggunakan *American Sign Language*, dengan sepuluh klasifikasi yaitu angka 0 hingga angka 9.
2. Data yang digunakan menggunakan dataset private dan dataset public.
3. Proses pengujian akan dilakukan dengan 3 subjek dengan warna kulit yang berbeda.
4. Penurunan intensitas cahaya dikurangi sebesar 50% lux sebelumnya hingga nilai lux kurang dari 50 lux.

### 1.4. Tujuan Penelitian

Penelitian yang dilakukan bertujuan untuk meningkatkan performa pengenalan gestur tangan dalam tingkat intensitas cahaya yang bervariasi menggunakan metode *Multiscale Retinex Color Restoration* dengan menggabungkan *Convolutional Neural Network* sebagai algoritme pengenalan gestur tangan.

### 1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberi manfaat berupa:

1. Memberikan fitur tambahan untuk sistem yang memiliki permasalahan dengan pencahayaan dalam intensitas cahaya rendah.
2. Meningkatkan akurasi pengenalan gestur tangan pada kondisi cahaya berintensitas rendah.

## BAB II

### TINJAUAN PUSTAKA

Penelitian untuk menghasilkan algoritme deteksi dan pengenalan gestur sudah banyak dilakukan, sehingga banyak metode yang memiliki tingkat keberhasilan tinggi. Namun pada pemrosesan citra tingkat keberhasilan tidak hanya didukung pada algoritme deteksi dan pengenalan saja, namun kualitas pada citra tersebut harus memiliki kualitas yang bagus pula sehingga dapat menghasilkan akurasi yang tinggi (Saputra., 2016).

Salah satu hal yang mempengaruhi kualitas tersebut adalah kondisi cahaya yang mengarah pada objek, sehingga mempengaruhi hasil yang ditangkap oleh kamera. Peningkatan kontras sangat dibutuhkan untuk meningkatkan kualitas image dalam kondisi cahaya yang minim. Salah satu pendekatan yang populer dipakai adalah Retinex. (Tanaka et al., 2019) mengimplementasikan Retinex pada preprocessing untuk meningkatkan kontras citra. Percobaan yang dilakukan dengan membandingkan hasil segmentasi citra asli dan citra dengan proses preprocessing. Hasil yang didapat secara kualitatif terlihat setelah dilakukan segmentasi Gaussian Mixture Model. Hasil segmentasi objek dengan citra preprocessing mendapatkan foreground yang jelas daripada citra asli. Namun pada citra dengan Retinex peningkatan kontras berubah menghasilkan warna yang tidak natural.

Peningkatan pencahayaan pada citra dapat dilakukan dengan banyak metode. Selain Retinex, metode yang paling populer adalah *Histogram Equalization*. (Srinivasan, 2016) Melakukan perbandingan antara algoritme *Retinex* dan *Histogram Equalization*. Pada penelitian tersebut *Retinex* yang di implementasikan adalah *Single Scale Retinex* (SSR) dan *Multiscale Retinex* (MSR). Kemudian *Histogram Equalization* yang diimplementasikan adalah BBHE, DSIHE dan RLBHE. Kedua algoritme tersebut diuji menggunakan citra yang sama kemudian dilakukan perbandingan.

Pada citra keluaran *Histogram Equalization* (BBHE dan DSIE), citra mengalami meningkatkan kontras dengan keterbatasan beberapa fitur yang tidak terlihat, Keluaran RLBHE mereduksi kualitas dari piksel. Kemudian dengan citra keluaran *Retinex*, citra mengalami peningkatan kontras dengan fitur yang terlihat lebih jelas daripada citra *Histogram Equalization*.

Algoritme *Retinex* telah mengalami pengembangan untuk meningkatkan kualitas citra pada kondisi lingkungan tertentu. Saputra ditahun 2016 melakukan perbandingan variasi *Retinex* untuk peningkatan deteksi wajah yang dilakukan pada kondisi ruangan berintensitas rendah. Algoritme *Adaptive Multiscale Retinex* (AMSR), *Multiscale Retinex Color Restoration*(MSRCR) di implementasikan pada 4 kondisi cahaya. Hasil peningkatan maksimal yang didapatkan pada MSRCR dapat meningkatkan 1.46 kali dan ASMR mampu meningkatkan 1.11 kali. Namun peningkatan tersebut terjadi pada parameter intensitas 273.25 lux(Saputra., 2016).

Pengembangan MSR juga dilakukan oleh (Shen et al., 2017) dengan menambah layer preprocessing/post processing pada proses konvolusi citra pada CNN menjadi MSR-Net. Kemudian dibandingkan dengan MSRCR dan beberapa metode lainnya. Hasil MSR-Net mengalami peningkatan kontras citra dengan warna natural dibandingkan MSRCR dan beberapa metode lainnya.

Pengenalan sebuah gestur adalah yang menentukan hasil akhir dari sistem. CNN merupakan salah satu algoritme yang sering dijadikan solusi untuk proses klasifikasi dalam *machine learning*. Beberapa penelitian menggunakan CNN dengan beberapa variasi parameter. Penelitian (Yingxin et al., 2017) menggunakan CNN untuk mengenali sebuah gestur tangan dengan dataset *Cambridge hand gesture datasets* (CHGD). Pada penelitian ini dilakukan parameter iluminasi cahaya pada beberapa kondisi. Pengenalan gestur menghasilkan angka presentase yang tinggi sebesar 94.1%. Proses dalam preprocessing menggunakan *canny edge* untuk menghilangkan efek iluminasi. *Canny edge* sangat membantu untuk kondisi iluminasi cahaya dibandingkan dengan algoritme CNN saja yang menggunakan citra asli di dapat hasil 70.0%.

Pada citra gelap informasi atau fitur penting dari sebuah citra akan ter-

sembunyi. Informasi dalam sebuah citra penting untuk merepresentasikan sebuah citra itu sendiri. Untuk mendapatkan informasi tersebut pada penelitian (Loh et al., 2019) berfokus pada perbaikan citra dengan tujuan memperoleh fitur untuk mendukung sistem visi otomatis dimana sebuah citra memiliki kontras dan pencahayaan yang rendah. Dalam penelitian ini memodelkan sebuah citra dengan cahaya rendah sebagai distribusi peningkatan fungsi lokal menggunakan proses gaussian yang dilatih pada saat runtime menggunakan data referensi yang dihasilkan dari sebuah CNN. CNN sendiri dilatih menggunakan dengan data yang sangat besar berdasarkan statistik pencahayaan. Sehingga proses pembelajaran dapat mempelajari hubungan antara fitur dengan piksel. Dengan demikian refrensi yang dihasilkan melatih gaussian proses untuk melakukan representasi fitur dengan benar.

Dasar-dasar penelitian sebelumnya yang menjadi tinjauan pustaka pada penelitian ini dirangkum dalam Tabel 2.1.

**Tabel 2.1 Tinjauan Pustaka**

No	Nama	Penelitian	Metode	Hasil
1	(Loh et al., 2019)	<i>Low-light image enhancement using Gaussian Process for features retrieval</i>	<i>Gaussian process and Convolutional Neural Network</i>	Citra dengan kontras yang sangat rendah dapat di perbaiki menggunakan gaussian proses dan CNN untuk memperoleh detail informasi dari sebuah objek
2	(Tanaka et al., 2019)	<i>Retinex Foreground Segmentation for Low Light Environments</i>	<i>Retinex dan Gaussian Mixture Model</i>	Secara kualitatif citra yang dihasilkan setelah melalui <i>preprocessing</i> algoritme Retinex dapat meningkatkan pencahayaan dari sebuah citra
3	(Yingxin et al., 2017)	<i>A Robust Hand Gesture Recognition Method via Convolutional Neural Network</i>	<i>Edge detection dan Convolutional Neural Network</i>	Pengenalan gestur tangan mendapatkan hasil 94.1% digabungkan dengan proses deteksi tepi.

Tabel 2.2 Lanjutan Tabel

4	(Shen et al., 2017)	<i>MSR-Net:Low-light Image Enhancement using Deep Convolutional Network</i>	MSR-net	Hasil dari implementasi algoritme MSR-Net di bandingkan dengan MSRCR dan beberapa algoritme lain mendapatkan kontras yang lebih tinggi dengan warna yang natural dibandingkan algoritme lain.
5	(Saputra., 2016)	Perbandingan Varian <i>Metode Multiscale Retinex</i> Untuk Peningkatan Akurasi Deteksi Wajah <i>Adaboost HAAR-like</i>	Variasi metode <i>Multiscale Retinex</i>	Kondisi 439.75 lux MSRCR meningkatkan akurasi 1.31 kali dan AMSR hanya 1,11 kali. Kondisi 273.25 lux MSRCR 1.46 kali dan AMSR 1.31 kali. Kondisi 150 lux MSRCR 1.38 kali dan AMSR 0.97 kali. Kondisi 9 lux kedua algoritme tidak dapat mendeteksi wajah sebuah citra.
6	(Srinivasan, 2014)	Perbandingan antara <i>Retinex</i> dan <i>Histogram Equalization</i>	SSR, MSR, BBHE, DSIHE, RLBHE	Citra keluaran <i>Retinex</i> memiliki <i>output</i> kontras yang baik tanpa menghilangkan fitur pada citra.
7	(Arabi, 2019)	Mendeteksi peralatan konstruksi yang diaplikasikan pada embedded system dan PC	<i>SSD MobileNet</i>	Untuk semua device mAP > 90%
8	(Huang et al., 2019)	<i>Hand Gesture Recognition with Skin Detection and Deep Learning Method</i> <i>Hand Gesture Recognition with Skin Detection and Deep Learning Method</i>	<i>Skin detection</i> dan <i>Convolutional Neural Network</i>	Menghasilkan akurasi 98.41%

## BAB III

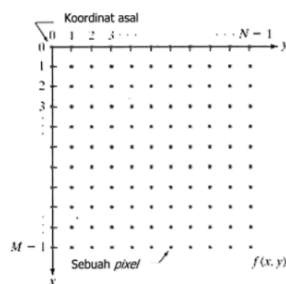
### LANDASAN TEORI

#### 3.1. Computer Vision

*Computer vision* adalah teknologi yang membuat computer dapat melihat layaknya mata manusia. Penggunaan computer vision tanpa kita sadari telah di implementasikan dalam membantu menyelesaikan persoalan sehari hari. Contoh implementasi *computer vision* antara lain *face recognition*, *object classification*, *medical imaging*, *gesture recognition*, *video surveillance*, *3D reconstruction*. Pada pengolahan citra, sebuah citra memiliki fitur-fitur penting yang digunakan sebagai informasi saat pengolahan. Namun untuk mendapatkan suatu hasil, beberapa tahap proses harus dilakukan seperti *preprocessing*, ekstraksi ciri, *post-processing* dan sebagainya.

#### 3.2. Citra Digital

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 tingkat keabuan dari suatu citra. Apabila nilai  $(x, y)$  dan  $f$  secara keseluruhan berhingga dan bernilai diskrit maka citra tersebut adalah citra digital. Citra digital dalam bentuk matrik dapat dilihat pada Persamaan 3.1 dan posisi koordinat citra digital dapat dilihat pada gambar 3.1 (Sarifudin.,2015).



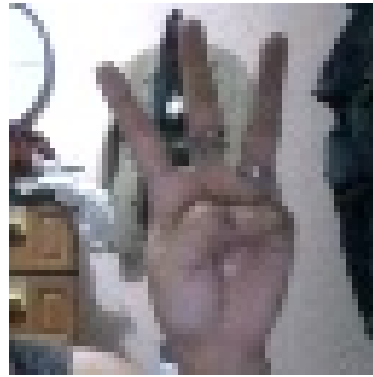
Gambar 3.1 Koordinat Citra Digital (Sarifudin.,2015)



$$f(x, y) \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, n-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1, 0) & f(m-1, 1) & \dots & f(m-1, n-1) \end{bmatrix} \quad (3.1)$$

### 3.2.1. Citra RGB

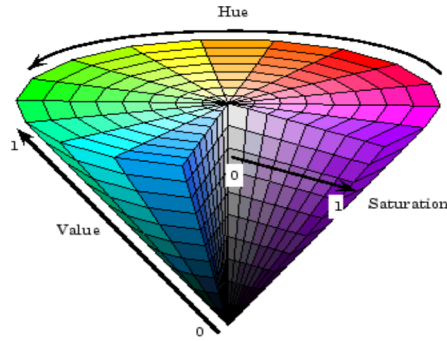
Citra RGB disebut juga sebagai citra berwarna, citra ini menyajikan tiga layer warna yaitu Red, Green dan Blue. Setiap piksel dari citra RGB merupakan gabungan dari variasi nilai intensitas tiga warna dasar yaitu merah(R), hijau(G), biru(B). Tiga warna tersebut dikodekan dengan 8 bit, dengan total ketiganya  $3 \times 8 = 24$  bit. Sehingga variasi warna sebanyak  $2^{24} = 16.777.216$  variasi warna (Sarifudin., 2015). Contoh citra RGB dapat dilihat pada Gambar 3.2.



Gambar 3.2 Citra RGB

### 3.2.2. Citra HSV

Ruang warna HSV memiliki tiga komponen warna, yaitu Hue(H), Saturation(S) dan Value(V). Hue memiliki variasi warna yang digambarkan secara melingkar, dimana merepresentasikan warna dari merah, kuning, hijau, cyan, biru, magenta, dan kembali lagi ke merah. Saturation memiliki variasi nilai 0 hingga 1, dimana merepresentasikan saturasi warna dari merah ke merah muda dan Value memiliki nilai 0 hingga 1 yang merepresentasikan intensitas warna atau tingkat kecerahan dari hitam ke putih, dimana nilai semakin tinggi semakin cerah (Kolkur et al., 2017). Ruang warna HSV dapat di gambarkan seperti Gambar 3.3.



**Gambar 3.3 Ruang warna HSV (Kolkur et al., 2017)**

Pada RGB suatu warna di representasikan dengan nilai tiap komponen, namun untuk ruang warna HSV memiliki nilai yang berbeda dari RGB. Hue memiliki range dari  $0^\circ$  sampai  $360^\circ$ . Transformasi warna dari RGB ke HSV dapat dilihat dalam Persamaan 3.2, 3.3, 3.4, 3.5.

$$V = \max(R, G, B) \quad (3.2)$$

$$S = \begin{cases} \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} & V \neq 0 \\ 0 & \text{lainnya} \end{cases} \quad (3.3)$$

$$H = \begin{cases} \frac{60(G-B)}{V - \min(R, G, B)} & V = R \\ 120 + \frac{60(B-R)}{V - \min(R, G, B)} & V = G \\ 240 + \frac{60(R-G)}{V - \min(R, G, B)} & V = B \end{cases} \quad (3.4)$$

$$H = H + 360 \quad (3.5)$$

Penggunaan citra dengan ruang HSV mampu memisahkan informasi warna sesuai dengan kemampuan mata manusia (Afrianto & Amalia., 2016).

### 3.3. Operasi Morfologi

#### 3.3.1. Erosi

Operasi erosi adalah operasi penipisan objek yang terdapat pada citra biner. Operasi erosi dilakukan dengan cara mengurangi piksel pada kontur dari objek citra sesuai dengan kernel. Operasi erosi dinotasikan pada Persamaan 3.6 (Hidayatullah., 2017).

$$A \ominus B = A^c \oplus B^c \quad (3.6)$$

### 3.3.2. Dilasi

Operasi dilasi adalah operasi penebalan objek yang terdapat pada citra biner. Operasi ini dilakukan dengan menambah piksel pada kontur dari objek sesuai dengan kernel. Operasi ini berguna untuk menghaluskan citra dan menutupi lubang-lubang yang kosong. Operasi dilasi dinotasikan pada Persamaan 3.7 (Hidayatullah., 2017).

$$A \oplus B = \{t \in Z^2 : t = a + b, a \in A, b \in B\} \quad (3.7)$$

### 3.3.3. Opening

Operasi *opening* merupakan operasi yang biasa digunakan untuk memperhalus kontur citra serta menghilangkan lubang-lubang kecil pada citra. Operasi ini terdiri dari 2 dua tahap yaitu erosi kemudian dilanjutkan dilasi. Operasi erosi berguna untuk menghilangkan noise pada citra karena struktur latar depan yang berukuran kecil tereliminasi, sedangkan dilasi digunakan untuk menebalkan citra. Operasi opening dinotasikan dalam Persamaan 3.8(Hidayatullah., 2017).

$$A \bullet B = (A \ominus B) \oplus B \quad (3.8)$$

### 3.3.4. Closing

Operasi *closing* merupakan kebalikan dari operasi *opening*. Terdiri dari dua tahap yaitu operasi dilasi kemudian dilanjutkan dengan operasi erosi. Kegunaan operasi ini adalah untuk menutupi lubang yang kosong pada citra dengan menggunakan dilasi kemudian dilakukan erosi citra untuk menipiskan suatu citra. Operasi closing dinotasikan pada Persamaan 3.9 (Hidayatullah., 2017).

$$A \bullet B = (A \oplus B) \ominus B \quad (3.9)$$

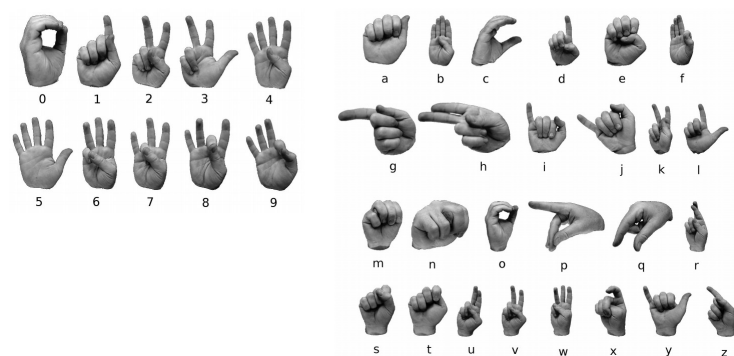
## 3.4. Hand Gesture Recognition

### 3.4.1. American Sign Language(ASL)

Bahasa isyarat merupakan media komunikasi utama bagi kaum difabel khususnya penyandang tunarungu di dunia. Setiap negara mempunyai bahasa isyarat

masing masing. Dengan adanya bahasa isyarat, penyandang tunarungu dapat melakukan komunikasi antar sesama penderita maupun berkomunikasi dengan setiap orang. Bahasa isyarat mulai banyak digunakan pada siaran televisi, seperti acara debat ataupun channel berita dengan dibantu penerjemah bahasa isyarat. Dengan begitu memberi mereka hak yang sama untuk mendapatkan informasi.

Salah satu bahasa isyarat yang digunakan adalah *American Sign Language*(ASL). *American Sign Language* menjadi salah satu alat bantu pembelajaran komunikasi untuk penggunanya. penggunaan bahasa isyarat dilakukan dengan gerakan gestur tangan yang memiliki makna untuk setiap pose. Pada penelitian ini, bahasa isyarat yang digunakan untuk menguji sistem adalah *American Sign Language*. Pada gestur ASL, mempunyai beberapa bentuk gestur tangan huruf dan angka seperti pada Gambar 3.4.



**Gambar 3.4 American Sign Language (Barczak et al., 2011)**

### 3.5. Retinex(Single Scale Retinex)

Algoritme *Retinex* merupakan algoritme yang berusaha untuk mempertahankan ketetapan warna dimana warna suatu objek yang dilihat dalam keadaan pencahayaan yang berbeda(Aribowo et al., 2009). Algoritme *Retinex* ini sering juga disebut dengan *Single Scale Retinex* (SSR) karena hanya memiliki satu kanal. Sebuah citra terbentuk sebagai hasil jumlahan produk radiasi dan reflektansi. Berdasarkan teori *Retinex* dapat dituliskan secara matematis seperti Persamaan 3.10 berikut(Saputra., 2016):

$$I(x, y) = L(x, y)xR(x, y) \quad (3.10)$$

dimana

$$L(x, y) = I(x, y)xF(x, y) \quad (3.11)$$

$$F(x, y) = K.e^{(-\frac{x^2+y^2}{\sigma})} \quad (3.12)$$

Keterangan :

$I(x, y)$  : representasi dari kumpulansinyal dari citra asli.

$R(x, y)$  : representasi dari komponen refleksi dari objek target.

$L(x, y)$  : representasi komponen pencahayaan yang memenuhi Persamaan 3.10.

$F(x, y)$  : *gaussian surround function* yang memenuhi Persamaan 3.11.

Bentuk logaritmik pada Persamaan 3.13 digunakan untuk memisahkan antara komponen pencahayaan dan komponen refleksinya. Pemisahan ini dilakukan untuk mendapatkan informasi dari citra tersebut sehingga hasil *Single Scale Retinex* dapat dilihat pada Persamaan 3.14, dimana  $R_{SSR}$  adalah hasil *Single Scale Retinex* pada kanal  $i$  (Saputra., 2016).

$$\log I(x, y) = \log L(x, y) + \log(x, y) \quad (3.13)$$

$$R_{SSR} = \log I(x, y) - \log[f(x, y) * I_i(x, y)] \quad (3.14)$$

### 3.5.1. Multiscale Retinex(MSR)

*Multiscale Retinex* merupakan pengembangan dari *Single Scale Retinex* dengan menggabungkan beberapa scale yang berbeda dengan pembobotan tertentu. Sehingga nilai sangat mempengaruhi detail warna dari sebuah citra. Bentuk matematis dari *multiscale Retinex* dapat dilihat pada Persamaa 3.15 dan 3.16 berikut(Saputra., 2016):

$$R_{MSRi} = \sum_{n=1}^N \omega_n R_{ni} \quad (3.15)$$

$$R_{ni}(x, y) = \log I(x, y) - \log[f_n(x, y) * I_i(x, y)] \quad (3.16)$$

Keterangan :

$R_{MSRi}$  : hasil *Multiscale Retinex* pada kanal  $i$

$\omega_n$  : bobot pada skala  $n$

Pada SSR dapat menyediakan kompresi rentang dinamis dan penampakan warna, namun keduanya tidak secara bersamaan. Oleh karena itu *Multiscale Retinex* menggabungkan kompresi rentang dinamis dari *Single Scale Retinex* dengan penampakan warna (Aribowo et al., 2009).

### 3.5.2. Multiscale Retinex Color Restoration (MSRCR)

MSRCR merupakan pengembangan dari metode MSR yang mampu memperbaiki kualitas citra yang berhubungan dengan pencahayaan yaitu dengan mempertahankan *color constancy*. *Color constancy* berfungsi untuk mempertahankan komposisi warna suatu citra tetap terlihat sama walaupun kondisi pencahayaan yang berbeda-beda (Saputra., 2016). Perhitungan MSRCR dapat dilihat pada Persamaan 3.17, 3.18, 3.19, 3.20, 3.21 (Hanumantharaju et al., 2014).

$$R'_{MSRCR}(x, y) = C_i(x, y)R_{MSR_i}(x, y) \quad (3.17)$$

$$C_i(x, y) = f[I'_i(x, y)] \quad (3.18)$$

$$I'(x, y) = I_i(x, y) / \sum_{i=0}^S I_i(x, y) \quad (3.19)$$

$$C_i(x, y) = \beta \log[\alpha I_i] - \log\left[\sum_{i=0}^S I_i(x, y)\right] \quad (3.20)$$

$$R_{MSRCR}(x, y) = G \times [R'_{MSRCR}(x, y) + b] \quad (3.21)$$

Keterangan :

$S = 3$  kanal warna RGB

$\alpha$  = konstanta kontrol

$\beta$  = konstanta gain

$G, b$  = gain dan offset

### 3.6. MobileNet

*MobileNet* merupakan salah satu arsitektur CNN yang dibangun oleh Google dan dikhususkan untuk *mobile device* karena proses komputasi yang ringan.

Perbedaan arsitektur *MobileNet* dan CNN pada umumnya terdapat pada penggunaan layer konvolusi dengan ketebalan sesuai input dari citra. Arsitektur SSD *MobileNet* menggunakan *Depthwise layer* dan *Pointwise layer*.

### 3.6.1. Depthwise Separable Convolution

*Depthwise separable convolution* merupakan kunci utama untuk membangun arsitektur *neural network*. Ide utamanya adalah mengganti operator konvolusi secara menyeluruh dengan memisahkan konvolusi menjadi dua lapisan terpisah. Layer pertama adalah *depthwise convolution*, layer ini memiliki *light-weight filter* dengan menggunakan *single convolutional filter* untuk setiap input channel. Layer kedua menggunakan  $1 \times 1$  convolution yang disebut dengan *pointwise convolution* dimana layer ini membentuk fitur-fitur baru melalui perhitungan kombinasi linear dari input channel.

Pada konvolusi biasa memiliki input citra  $D_F \times D_F \times M$  fitur map pada  $F$  dan menghasilkan  $D_G \times D_G \times N$  fitur map pada  $G$ , dimana  $D_F$  adalah tinggi dan lebar dari fitur map.  $M$  merupakan jumlah input channel (input depth).  $D_G$  merupakan tinggi dan lebar keluaran fitur map dan  $N$  adalah keluaran channel dari fitur map (output depth). Kemudian kernel pada konvolusi biasa  $D_K \times D_K \times M \times N$ , dimana  $D_K$  adalah dimensi kernel,  $M$  jumlah input channel dan  $N$  jumlah keluaran channel. Total cost dari konvolusi biasa memiliki ketergantungan pada  $M$  pada Persamaan 3.22.

$$D_K \bullet D_K \bullet M \bullet N \bullet D_F \bullet D_F \quad (3.22)$$

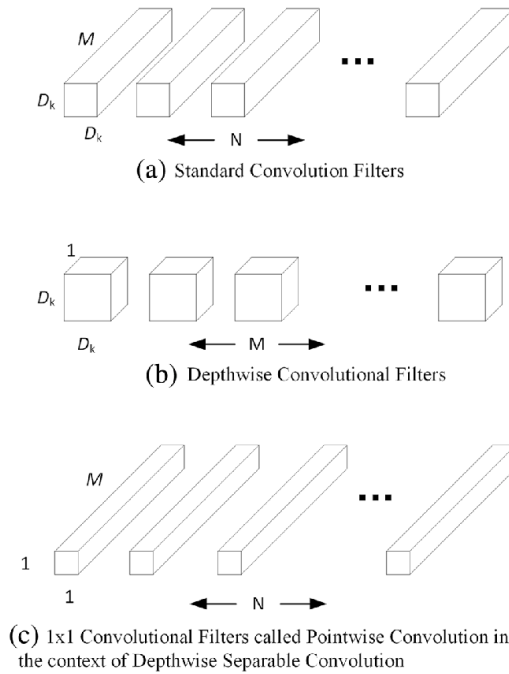
*Depthwise separable convolution* terbentuk dari layer *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* sangat efisien daripada konvolusi biasa, namun pada layer *depthwise* tersebut tidak menghasilkan fitur baru, oleh karena itu digunakan tambahan layer *pointwise convolution* untuk menghitung linier kombinasi dari keluaran *depthwise convolution* menggunakan  $1 \times 1$  convolution, sehingga menghasilkan fitur baru. Total cost dari *depthwise convolution* pada Persamaan 3.23.

$$D_K \bullet D_K \bullet M \bullet D_F \bullet D_F \quad (3.23)$$

Kombinasi kedua layer tersebut dinamakan *depthwise separable convolution* yang memiliki total cost Persamaan 3.24 :

$$D_K \bullet D_K \bullet M \bullet N \bullet D_F \bullet D_F + M \bullet N \bullet D_F \bullet D_F \quad (3.24)$$

*MobileNet* menggunakan  $3 \times 3$  *depthwise separable convolution* yang mana mereduksi komputasi 8 hingga 9 kali lebih cepat dari konvolusi biasa. Perbedaan konvolusi biasa dan *depthwise separable convolution* digambarkan pada Gambar 3.5 (Howard et al., 2017).



**Gambar 3.5 Convolutional standard dan depthwise separable (Howard et al., 2017)**

### 3.7. MobileNetV2

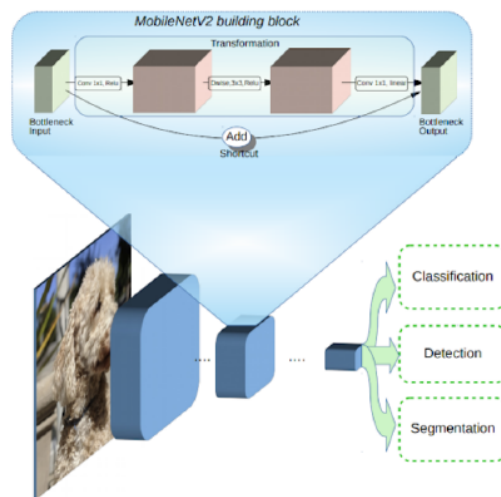
Perbedaan *MobileNet* versi kedua merupakan pengembangan dari *MobileNet* versi pertama. Pada *MobileNet* memiliki modifikasi arsitektur linier *bottleneck* dan *shortcut* koneksi antar *bottleneck* seperti Gambar 3.6. Sehingga arsitektur dalam *MobileNetV2* menjadi *bottleneck depth-separable convolution residuals*. Arsitektur *bottleneck* dituliskan pada Tabel 3.1.



**Tabel 3.1 Blok Arsitektur *Bottleneck MobileNetV2* (Sandler et al., 2018)**

Input	Operator	Output
$h \times w \times k$	$1 \times 1$ Conv2D, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	$3 \times 3$ dwse $s=s$ , ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear $1 \times 1$ Conv2D	$\frac{h}{s} \times \frac{w}{s} \times k'$

Tabel 3.1 menunjukan blok arsitektur *bottleneck MobileNetV2* terdiri dari *fully convolution* dengan 32 filter diikuti 19 layer *residual bottleneck*. Kernel yang digunakan menggunakan ukuran  $3 \times 3$  (standard untuk modern network) dan menggunakan dropout dan normalisasi selama pelatihan(Sandler et al., 2018).



**Gambar 3.6 Struktur Dasar Pada *MobileNetV2* (Google AI Blog., 2018)**

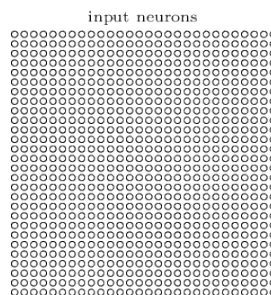
Pada bagian *bottleneck* terdapat *input* dan *output*, sedangkan layer di bagian dalam mengenkapsulasi kemampuan model untuk merubah input dari piksel ke gambar. Tambahan *shortcut* antar *bottleneck* memungkinkan saat melakukan *training* menjadi lebih cepat dengan peningkatan akurasi yang lebih baik(Google AI Blog., 2018).

### 3.8. Convolutional Neural Network(CNN)

*Convolutional Neural Network* (CNN) merupakan jaringan syaraf tiruan yang memiliki arsitektur khusus yang dapat bekerja dengan baik dalam penggunaannya pada pemrosesan data berbentuk larik, seperti data runtun waktu dan citra digital. Dalam arsitekturnya CNN menggunakan operasi matematika konvolusi pada layer jaringan. Operasi konvolusi antara  $a$  dan  $b$  disimbolkan dengan  $(a \times b)$ . *Convolutional neural network* pada penerapannya menggunakan tiga ide dasar yaitu local receptive fields, shared weights, dan pooling(Nielsen., 2015).

#### 3.8.1. Local Receptive Fields

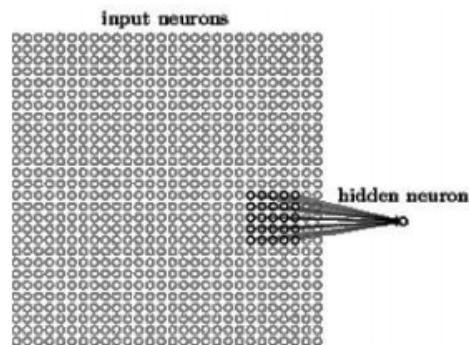
Pada jaringan syaraf tiruan dengan *fully-connected*, input dari jaringan digambarkan dengan garis vertikal dari kumpulan neuron. Pada CNN, input digambarkan sebagai persegi dengan ukuran  $o \times o$  sesuai dengan ukuran input yang diberikan. Misalkan terdapat input dengan ukuran  $28 \times 28$  neuron, yang berkesesuaian dengan  $28 \times 28$  intensitas piksel pada citra seperti terlihat pada Gambar 3.7.



**Gambar 3.7 Ilustrasi Citra 28x28 Piksel (Nielsen., 2015)**

Seperti jaringan syaraf tiruan pada umumnya, setiap neuron dari input terhubung dengan layer dari hidden neuron. Sedikit berbeda, pada CNN neuron input tidak terhubung secara *fully-connected* dengan setiap hidden neuron, tetapi hanya region lokal kecil dari input terhubung dengan sebuah hidden neuron (Nielsen., 2015). Pada Gambar 3.8 terlihat bahwa hanya sebagian kecil region yang *localized* dari input neuron yang terhubung dengan hidden neuron.

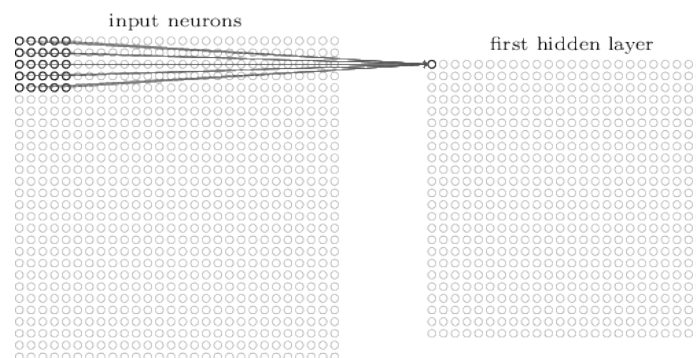
Pada Gambar 3.9 setiap neuron pada hidden layer terhubung dengan region berukuran  $5 \times 5$  piksel pada neuron input. Region pada input layer inilah yang dina-



**Gambar 3.8 Ilustrasi *local receptive fields* (Nielsen., 2015)**

makan dengan *local receptive fields* dan setiap koneksi memiliki bobot yang akan disesuaikan seiring dengan proses pelatihan. Selain itu hidden neuron juga memiliki dan mempelajari bias secara keseluruhan. Oleh karena itu, setiap hidden neuron dilatih untuk menganalisa masing-masing *local receptive fields* yang bersesuaian.

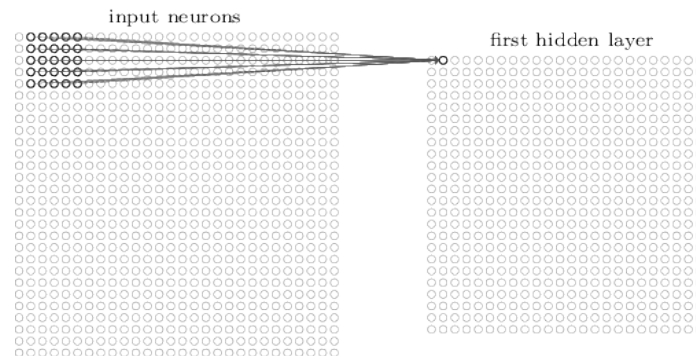
Tahap selanjutnya, *local receptive fields* akan digeser (slide) sepanjang ukuran citra dari posisi paling kiri atas hingga kanan bawah. Setiap *local receptive fields* akan memiliki pasangan hidden neuron yang berbeda pada hidden layer. Ilustrasi proses pergeseran *local receptive fields* terdapat pada Gambar 3.9 dan Gambar 3.10 berikut.



**Gambar 3.9 Ilustrasi Pergeseran *local receptive fields* (Nielsen., 2015)**

Selanjutnya *local receptive fields* akan digeser sebanyak satu piksel ke kanan seperti pada Gambar 3.8 dan terhubung dengan hidden neuron yang berbeda dengan *local receptive fields* sebelumnya.

Pergeseran piksel dengan  $\text{stride} = 1$  di ilustrasikan pada Gambar 3.9. Proses ini berlangsung hingga *local receptive fields* berada pada posisi piksel paling kanan



**Gambar 3.10 Ilustrasi Pergeseran dengan Stride = 1 (Nielsen., 2015)**

bawah. Sehingga setelah proses slide dilakukan akan terbentuk hidden layer dengan ukuran sesuai dengan ukuran local receptive fields dan panjang pergeseran (stride) yang digunakan (Nielsen, 2015). Maka jika citra dengan ukuran  $28 \times 28$  piksel dan digunakan *local receptive fields* dengan ukuran  $5 \times 5$ , dan digeser dengan stride = 1, maka akan terbentuk hidden layer dengan ukuran  $24 \times 24$  neuron.

### 3.8.2. Shared Weight

Pada CNN, sebuah neuron pada hidden layer yang terhubung dengan  $5 \times 5$  neuron pada input layer (sesuai dengan ukuran *local receptive fields* yang digunakan). Hal ini menunjukkan bahwa neuron tersebut memiliki sebuah bias dan matriks bobot dengan ukuran  $5 \times 5$  yang menghubungkan  $5 \times 5$  neuron inputnya. Matriks bobot ini, dalam CNN, disebut sebagai kernel. Matriks bobot pada jaringan syaraf tiruan biasa sedikit berbeda dengan matriks bobot pada CNN, yaitu nilai matriks bobot pada CNN bernilai sama untuk setiap  $24 \times 24$  neuron pada hidden layer. Hal tersebut menunjukkan bahwa semua neuron pada hidden layer akan mendeteksi fitur yang sama dengan lokasi pada input citra yang berbeda (Nielsen., 2015).

### 3.8.3. Konvolusi

Lapisan yang pertama kali akan dilewati oleh data masukan adalah lapisan konvolusi, bertujuan untuk memperoleh feature map yang merepresentasikan masukan. Parameter yang digunakan untuk menentukan ukuran feature map keluaran berupa filter, ukuran dari filter, besarnya langkah pergeseran filter pada operasi

konvolusi atau biasa disebut stride, dan ukuran padding. Cara menghitung ukuran dari keluaran operasi konvolusi ditunjukkan pada Persamaan 3.25 dan 3.26 berikut(Nielsen., 2015):

$$H_0 = \frac{H - F + 2P}{S} + 1 \quad (3.25)$$

$$W_0 = \frac{W_i - F + 2P}{S} + 1 \quad (3.26)$$

Keterangan :

$H_0$  = tinggi fitur map keluaran

$W_0$  = lebar fitur map keluaran

$H_i$  = tinggi fitur map masukan

$W_i$  = lebar fitur map masukan

$F$  = ukuran filter

$P$  = ukuran padding

$S$  = ukuran stride

Nilai semua elemen pada persamaan tersebut harus merupakan bilangan bulat karena akan merepresentasikan suatu ukuran feature map. Apabila terdapat salah satu nilai yang bukan merupakan bilangan bulat, nilai tersebut harus dibulatkan dengan melakukan pembulatan kebawah. Adapun cara menghitung nilai keluaran dari proses konvolusi dapat dilihat pada Persamaan 3.27 berikut(Nielsen., 2015):

$$O_{mn} = \sum I_{i,j}^k \cdot F_{i,j} \quad (3.27)$$

Keterangan :

$O_{mn}$  = elemen matriks keluaran pada baris ke-m kolom ke -n.

$I_{i,j}^k$  = elemen matriks masukan bagian ke-k pada baris ke-i kolom ke-j.

$F_{i,j}$  = elemen matriks filter pada baris ke-i kolom ke -j.

### 3.8.4. Fungsi Aktivasi

Salah satu faktor signifikan mempengaruhi kinerja algoritme Convolutional Neural Network adalah penerapan fungsi aktivasi dalam jaringan. Fungsi ini membantu menyelesaikan permasalahan-permasalahan yang bersifat non-trivial dalam suatu jaringan dengan cara mengambil sebuah nilai dan melakukan operasi

matematika. Fungsi aktivasi ini diletakkan di perhitungan akhir dari keluaran feature map atau setelah layer konvolusi dan subsampling layer. Fungsi aktivasi yang sering digunakan adalah Rectified Linear Unit (ReLU) karena fungsi ini lebih cepat daripada fungsi aktivasi non-linear lainnya seperti sigmoid atau tanh. Fungsi ReLU dapat dilihat pada Persamaan 3.28 dan 3.29(Nielsen., 2015):

$$f(x) = ReLU(x) = \max(0, x) \quad (3.28)$$

$$f(x) = 0 \text{ jika } x \leq 0 \text{ atau } x \text{ jika } x > 0 \quad (3.29)$$

### 3.9. Evaluasi

#### 3.9.1. Confussion Matrix

Pengukuran evaluasi dari pengujian dapat dilakukan menggunakan *confusion matrix*. tabel *confusion matrix* merupakan tabel klasifikasi yang bersifat prediktif seperti ditunjukkan pada Gambar 3.11. Evaluasi menggunakan confusion matrix dapat digunakan untuk mengukur nilai akurasi dengan Persaman 3.30.

		Predicted values	
		Negative	Positive
Actual values	Negative	TN	FP
	Positive	FN	TP

Gambar 3.11 Confusion Matrix (Leonard., 2017)

$$akurasi = \frac{TP}{TP + FP} \quad (3.30)$$

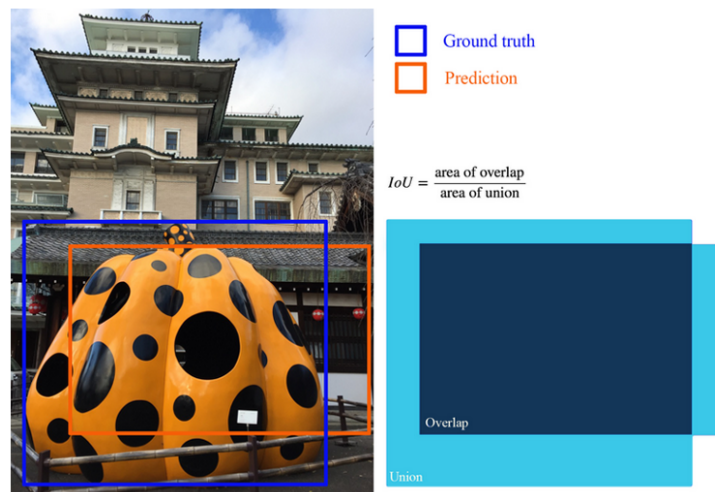
Dengan :

$TP$  = True Positif, objek berupa gestur tangan dan terkenali

$FP$  = False Positif, objek berupa gestur tangan tapi tidak terkenali

### 3.9.2. Average Precicion(AP)

*Average Precicion* adalah metrik populer untuk melakukan evaluasi pada objek detektor seperti *Faster-RCNN*, *SSD*, dan lainnya. AP menghitung nilai rata rata nilai presisi untuk nilai recall 0 hingga 1 (Hui., 2018). Evaluasi deteksi menggunakan konsep IoU(*intersection over union*) IoU menghitung interseksi kedua bounding boxes dimana terdiri dari ground truth dan prediksi dari bounding box digambarkan pada Gambar 3.12 dan perhitungan IoU pada Persamaan 3.31.



Gambar 3.12 Ilustrasi IoU(Hui., 2018)

$$IoU = \frac{area\_of\_overlap}{area\_of\_union} \quad (3.31)$$

IoU yang digunakan untuk memprediksi suatu bounding box, apabila nilai IoU melebihi *threshold* yang ditentukan maka akan terdeteksi sebuah objek. Sebaliknya jika IoU kurang dari batas yang ditentukan maka menandakan tidak terdeteksi apapun pada citra.

Terdapat 4 kategori dalam penentuan deteksi yaitu, FN(*False Negative*) terdapat objek namun tidak mendeteksi apapun. FP(*False Positive*) tidak terdapat objek namun hasilnya terdeteksi. TP(*True Positive*) terdapat objek dan terdeteksi. TN(*True Negative*) akan terjadi jika tidak ada objek dan tidak terdeteksi apapun. Berdasarkan nilai yang didapat maka diketahui presisi dan recall menggunakan Per-

samaan 3.32 dan Persamaan 3.33 (Hui., 2018).

$$presisi = \frac{TP}{TP + FP} \quad (3.32)$$

$$recall = \frac{TP}{TP + FN} \quad (3.33)$$

Metrik untuk mengukur detektor dari *object detection* menggunakan mAP (*mean average precision*). Nilai mAP dan AP dalam COCO dataset tidak memiliki perbedaan, dimana mAP adalah rata rata dari setiap AP yang memiliki nilai IoU sebagai *threshold*. Total metrik evaluasi pada COCO memiliki 12 metrik seperti pada Gambar 3.13 (Chen et al., 2015).

<b>Average Precision (AP):</b>	
AP	% AP at IoU=.50:.05:.95 ( <b>primary challenge metric</b> )
AP <sub>IoU=.50</sub>	% AP at IoU=.50 (PASCAL VOC metric)
AP <sub>IoU=.75</sub>	% AP at IoU=.75 (strict metric)
<b>AP Across Scales:</b>	
AP <sub>small</sub>	% AP for small objects: area < 32 <sup>2</sup>
AP <sub>medium</sub>	% AP for medium objects: 32 <sup>2</sup> < area < 96 <sup>2</sup>
AP <sub>large</sub>	% AP for large objects: area > 96 <sup>2</sup>
<b>Average Recall (AR):</b>	
AR <sub>max=1</sub>	% AR given 1 detection per image
AR <sub>max=10</sub>	% AR given 10 detections per image
AR <sub>max=100</sub>	% AR given 100 detections per image
<b>AR Across Scales:</b>	
AR <sub>small</sub>	% AR for small objects: area < 32 <sup>2</sup>
AR <sub>medium</sub>	% AR for medium objects: 32 <sup>2</sup> < area < 96 <sup>2</sup>
AR <sub>large</sub>	% AR for large objects: area > 96 <sup>2</sup>

**Gambar 3.13 Metrik Evaluasi COCO (Chen et al., 2015)**



## **BAB IV**

### **METODOLOGI PENELITIAN**

#### **4.1. Studi Literatur**

Studi literature dilakukan dengan mempelajari penelitian yang berkaitan dengan materi *hand gesture recognition*, *Retinex* dan *Convolutional Neural Network* yang diperoleh dari berbagai sumber seperti buku, artikel, jurnal dan sumber lain yang diperoleh dari internet.

#### **4.2. Alat dan Bahan**

##### **4.2.1. Alat**

1. PC/Laptop dengan spesifikasi processor Intel (R) Core i5-8300H CPU @2,4 GHz, GPU NVIDIA 1050, RAM 8 GB, sistem operasi Linux 64 bit.
2. Webcam Logitech C270
3. Dimmer
4. Kain
5. Lux Meter

##### **4.2.2. Bahan**

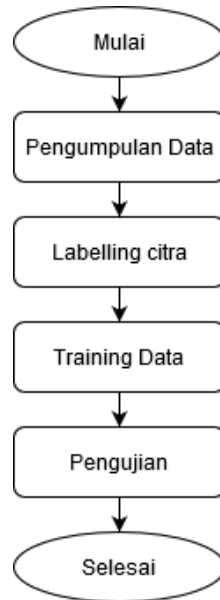
1. Dataset gestur tangan

#### **4.3. Prosedur Kerja**

##### **4.3.1. Analisis dan Perancangan Sistem**

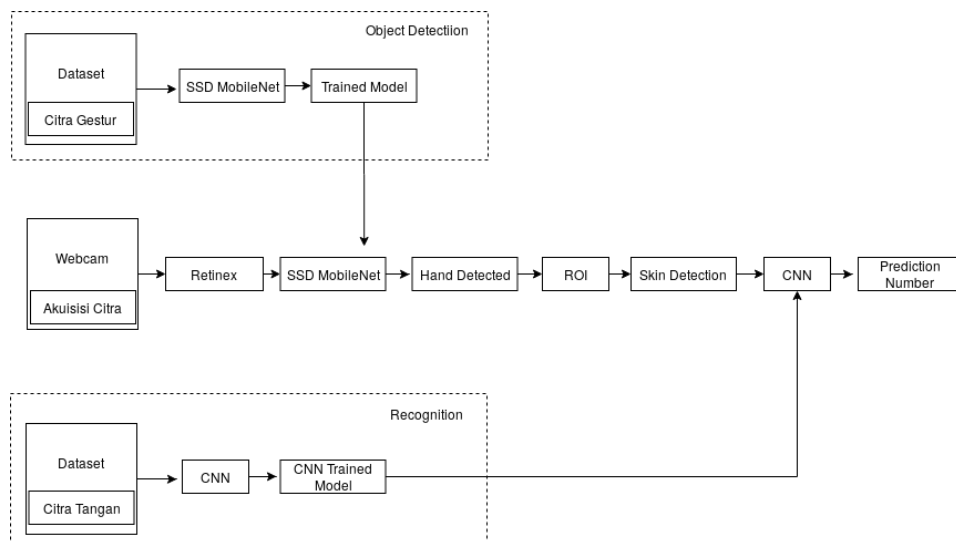
Alur penelitian yang akan dilakukan dalam penelitian ini memiliki beberapa tahapan diantaranya pengambilan dataset gestur, training data dan tahap paling

akhir adalah pengujian. Alur kegiatan penelitian yang akan dilakukan secara garis besar ditunjukkan pada Gambar 4.1.



**Gambar 4.1 Alur Kegiatan Penelitian**

Rancangan pengujian sistem ditunjukkan pada Gambar 4.2 yang dimulai dengan *input* dari webcam secara *realtime* sehingga setiap frame akan di proses nantinya. Setiap frame dengan citra RGB akan dilewatkan pada algoritme *Retinex* untuk dilakukan perbaikan kualitas citra dengan harapan meningkatkan kontras pada sebuah citra.



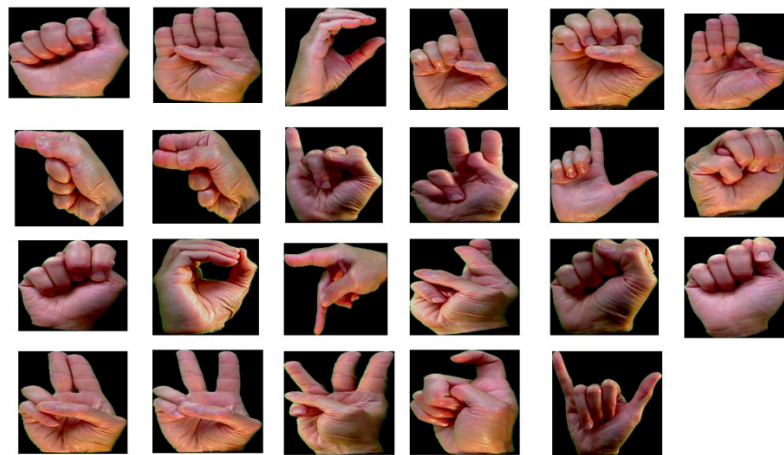
**Gambar 4.2 Rancangan Pengujian Sistem**

Setelah citra mengalami peningkatan kontras kemudian sistem akan melakukan deteksi tangan dengan *SSD MobileNet* yang kemudian citra tangan tersebut akan diambil sebagai ROI. Citra hasil ROI akan dilakukan segmentasi menggunakan *skin detection*. Hasil segmentasi akan diklasifikasikan dengan model CNN yang telah di training sebelumnya. Keluaran dari sistem ini akan menampilkan klasifikasi angka yang terdeteksi.

#### 4.4. Pengumpulan Data

Data pelatihan dibagi menjadi dua, yaitu dataset citra gestur tangan yang mengacu pada ASL dan dataset citra tangan biasa untuk *object detection*. Kedua dataset ini merupakan dua hal yang berbeda tujuan. Dataset ASL digunakan untuk proses pengenalan gestur dan dataset *object detection* digunakan untuk mendeteksi adanya tangan atau tidak.

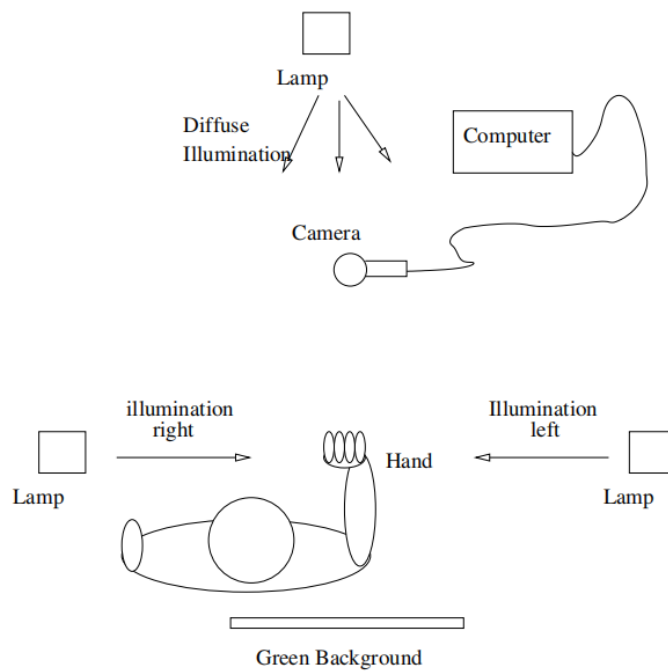
Dataset gestur tangan ASL menggunakan dataset public dari *Massey University*. Dataset tersebut berisi 2425 citra yang dibagi dalam 36 kelas, yaitu 26 untuk huruf A-Z dan 10 kelas untuk angka 0-9. Pengambilan dataset dilakukan oleh 5 individu dengan variasi cahaya yang berbeda. Contoh dataset *Massey University* dapat dilihat pada Gambar 4.3 (Barczak et al., 2011).



Gambar 4.3 Sample Dataset ASL Massey University (Barczak et al., 2011)

Acuan dalam penelitian ini menggunakan dataset angka 0 sampai 9, dengan citra yang sudah di segmentasi. Dataset ini akan dilatih dalam CNN untuk pengenalan-

an gestur tangan. Pengambilan dataset *Massey University* dilakukan menggunakan *green screen* sehingga mudah untuk disegmentasi, setup pengambilan dataset dilakukan dengan skema seperti Gambar 4.4 (Barczak et al., 2011).



**Gambar 4.4 Setup Pengambilan Dataset (Barczak et al., 2011)**

Pengambilan dataset berikutnya berbeda dengan dataset ASL, dataset berikutnya untuk deteksi tangan tidak memiliki klasifikasi, citra diambil dari 10 orang dengan random pose. dataset ini digunakan untuk pelatihan pada object detection. Pengambilan dataset dilakukan masing masing 100 capture untuk setiap orang, sehingga pada dataset ini diperoleh 1000 citra.

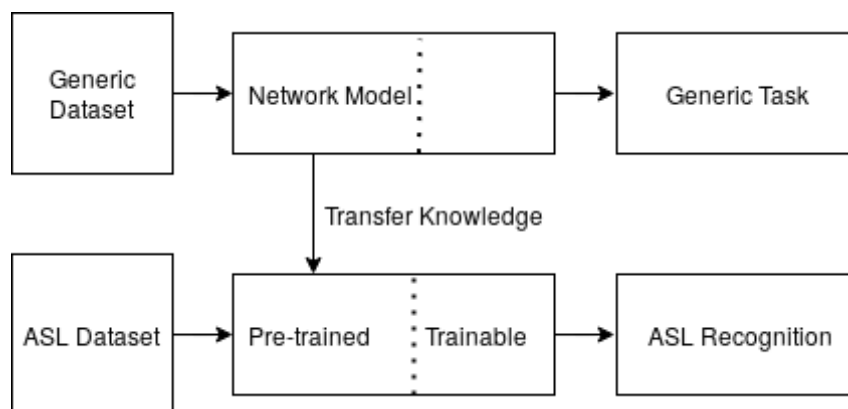
#### 4.5. Proses Pelatihan

Proses pelatihan dalam penelitian ini dibagi menjadi dua bagian, yaitu pelatihan object detection dan pelatihan gestur recognition.

#### 4.5.1. Pelatihan Gesture Recognition

Dataset pada pelatihan *gesture recognition* menggunakan dataset dari *Massey Universtisy* yang di labeli sesuai acuan ASL dari angka 0 hingga 9. Jumlah data-set adalah 700 citra kemudian dibagi yang akan dibagi dalam *train* dan *test* menggunakan *k-fold cross validation* dengan  $k = 5$ . Proses pelatihan untuk pengenalan gestur tangan menggunakan teknik *transfer learning* yang memiliki kelebihan pada waktu pelatihan dan penggunaan dataset yang cenderung sedikit. *Pre-trained* model yang digunakan adalah *MobileNetV2*. *Pre-trained* model memiliki pengetahuan dari dataset yang telah dilatih sebelumnya, pada *MobileNetV2* telah dilatih menggunakan dataset dari *ImageNet*, kemudian pengetahuan tersebut akan di pindahkan ke model baru dan dilatih sesuai dataset baru, sehingga memiliki keluaran dalam bentuk klasifikasi atau tugas sesuai yang diinginkan.

Masukan citra memiliki dimensi 224x224 piksel yang terdiri 3 channel RGB, sehingga node *input* berjumlah 50176 dengan range 0-255 untuk setiap piksel. Arsitektur jaringan pada pelatihan ini dilakukan dengan mengganti bagian blok klasifikasi dari *MobileNetV2* menjadi klasifikasi dataset ASL. Arsitektur transfer learning dapat dilihat pada Gambar 4.5.



Gambar 4.5 Transfer Learning Pelatihan Gestur

#### 4.5.2. Pelatihan Object Detection

Proses pelatihan pada *object detection* menggunakan *pre-trained model SSD MobilenetV2 COCO* yang sudah di latih sebelumnya dengan dataset COCO. Data-

set yang baru akan dilatih menggunakan *pre-trained* model, sehingga dengan pengetahuan yang sudah ada dilatih agar menghasilkan model baru sesuai dengan *output* yang diinginkan. Dataset yang akan dilatih sejumlah 1000 citra yang telah dilabeli secara manual dengan pembagian 80% *training* dan 20% *testing*. Arsitektur yang digunakan dalam pelatihan *object detection* dapat dilihat pada Gambar 4.6.

Input	Operator	$t$	$c$	$n$	$s$
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 4.6 Arsitektur Mobilenet V2 (Sandler et al., 2018)

#### 4.6. Pengujian dan Evaluasi

Pengujian dan evaluasi adalah tahap untuk mengukur performa dari sebuah sistem. Pengujian ini akan dibagi menjadi 6 tahap, yaitu evaluasi deteksi tangan, evaluasi pengenalan gestur, pengujian deteksi tangan menggunakan *Retinex*, pengujian pengenalan gestur tangan menggunakan *Retinex*, pengujian SNR dan pengujian keseluruhan sistem. Setiap pengujian yang menggunakan *Retinex* dilakukan hal yang sama, yaitu dengan menurunkan intensitas cahaya.

##### 4.6.1. Evaluasi Deteksi Tangan

Evaluasi model deteksi tangan dilakukan setelah *training* menggunakan metrik *mean average precicion* (mAP). *Mean average precicion* merupakan metrik yang populer untuk mengevaluasi model dari detektor. *Mean average precicion* merupakan rata rata dari *Average Precicion* (AP) pada setiap nilai IoU. Sesuai de-

ngan acuan COCO dataset untuk mengevaluasi model pada objek detection dengan menghitung mAP dari IoU=0.5 hingga 0.95 dengan penambahan 0.05.

#### 4.6.2. Evaluasi Pengenalan Gestur Tangan

Evaluasi model CNN untuk pengenalan gestur tangan didapatkan saat *training* dan *testing* menggunakan *k-fold cross validation* dengan  $k = 5$ , dimana dilakukan validasi sebanyak 5 kali dengan skenario pengujian yang berbeda pada proses *training* dan *testing*. *Training accuracy* dan *testing accuracy* didapatkan pada setiap *fold*.

#### 4.6.3. Pengujian SNR

SNR(*Signal to Noise Ratio*) adalah ukuran untuk mengukur kualitas citra terhadap citra yang dilakukan perbaikan. Citra hasil perbaikan dibandingkan dengan citra asli untuk mendapatkan nilai SNR nya. Nilai SNR yang tinggi mengindikasikan kualitas citra yang semakin baik karena rasio sinyal terhadap metode juga tinggi, sebaliknya nilai SNR yang rendah berarti kualitas citra yang di hasilkan semakin buruk atau semakin kecil dalam peningkatan kualitas citra. Pengujian SNR ini dilakukan untuk menentukan nilai dari parameter *retinex* yang akan dipakai pada sistem.

#### 4.6.4. Pengujian Deteksi Tangan Menggunakan Retinex

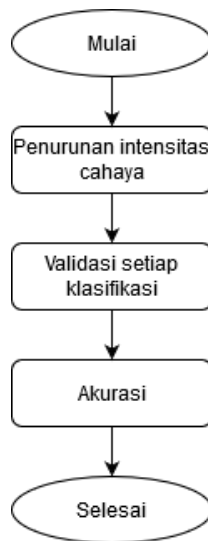
Tujuan pengujian deteksi tangan dilakukan untuk menguji sistem dalam melakukan deteksi tangan terkait dengan penurunan intensitas cahaya. Skema pengujian deteksi tangan dapat dilihat pada Gambar 4.7.





#### 4.6.5. Pengujian Pengenalan Gestur Tangan Menggunakan Retinex

Pengujian tahap kedua yaitu pengenalan gestur tangan dengan *retinex*. Tujuan dari pengujian ini untuk mengukur performa sistem dalam melakukan pengenalan gestur tangan dengan acuan ASL sesuai dengan dataset yang telah dilatih. Skema pengujian pengenalan gestur tangan dapat dilihat pada Gambar 4.8.



**Gambar 4.8 Skema Pengujian Pengenalan Gestur Tangan**

Pengujian pengenalan dilakukan secara manual dan terpisah dari deteksi objek, tahap ini memiliki perlakuan sama dengan tahap sebelumnya, yaitu menggunakan 3 subjek yang berbeda, kemudian dilakukan pengenalan gestur tangan sebanyak 10 kali untuk setiap klasifikasi. Total citra yang didapat dalam satu kondisi lux oleh satu subjek adalah 100 citra. Tahap berikutnya dilakukan penurunan intensitas cahaya, dengan mengatur cahaya lampu di ruangan uji sebesar 50% lux menggunakan dimmer. Pengujian dilakukan hingga batas lux kurang dari 50 lux.

Tabel 4.2 Pengujian Pengenalan Gestur Tangan

Nilai Lux(X)																																		
Klasifikasi	Pengenalan Subjek_1									Pengenalan Subjek_2									Pengenalan Subjek_3															
Angka 0																																		
Angka 1																																		
Angka 2																																		
...																																		
Angka 9																																		

Nilai Lux(...)																																		
Klasifikasi	Pengenalan Subjek_1									Pengenalan Subjek_2									Pengenalan Subjek_3															
Angka 0																																		
Angka 1																																		
Angka 2																																		
...																																		
Angka 9																																		

Nilai Lux( $X \leq 50$ )																																		
Klasifikasi	Pengenalan Subjek_1									Pengenalan Subjek_2									Pengenalan Subjek_3															
Angka 0																																		
Angka 1																																		
Angka 2																																		
...																																		
Angka 9																																		

Hasil dari pengujian akan di catat dalam bentuk tabel seperti pada Tabel 4.2. Evaluasi pengujian dilakukan dengan *confussion matrix*, sehingga memperoleh nilai akurasi dari hasil pengujian. Perhitungan akurasi dilakukan pada setiap nilai lux yang di uji.

#### 4.6.6. Pengujian Sistem Keseluruhan

Pengujian tahap terakhir adalah pengujian keseluruhan sistem yang dilakukan secara manual, dengan menggabungkan *retinex*, deteksi objek dan pengenalan gestur tangan dalam satu program utuh. Pengenalan akan terjadi apabila sebuah tangan dideteksi terlebih dahulu, jika tidak terdeteksi maka tidak akan dilakukan pengenalan. Penurunan intensitas dilakukan sama seperti pengujian sebelumnya. Tabel pengujian yang digunakan mengacu pada Tabel 4.2. Evaluasi dari akurasi sistem keseluruhan dapat diperoleh menggunakan *confusion matrix*.

## BAB V

### JADWAL PENELITIAN

**Tabel 5.1 Jadwal Penelitian**

No	Kegiatan	Target	Waktu Penelitian
1	Studi literatur	Menyelesaikan review literature terkait dengan penelitian yang relevan	Oktober - Januari
2	Pengumpulan Data	Mengumpulkan dataset gestur jari tangan	Maret – April 2020
3	Perancangan Model	Melakukan uji coba dan perancangan model Neural network dalam sistem untuk training data	Maret – Juni 2020
4	Implementasi	Melakukan training data untuk menghasilkan model	Juni – Agustus 2020
5	Pengujian	Menguji model secara realtime	Agustus – Oktober 2020
6	Penulisan Laporan	Menyusun laporan secara bertahap	Maret – Desember 2020

## DAFTAR PUSTAKA

- Aribowo, E., Yustina, E., Studi, P., Informatika, T., Teknologi, F., Universitas, I. & Dahlan, A., 2009, *Implementasi Metode Retinex Untuk Pencerahan Citra*, Jurnal Informatika, 3, 2, 323–330.
- Loh, Y.P., Liang, X. & Chan, C.S., 2019, *Low-light image enhancement using Gaussian Process for features retrieval*, Signal Processing: Image Communication, 74, 175–190. <https://doi.org/10.1016/j.image.2019.02.001>,.
- Saputra, L.K.P., 2016, *Perbandingan Varian Metode Multiscale Retinex Untuk Peningkatan Akurasi Deteksi Wajah Adaboost HAAR-like*, Jurnal Teknik Informatika dan Sistem Informasi, 2, 1, 89–98.
- Shen, L., Yue, Z., Feng, F., Chen, Q., Liu, S. & Ma, J., 2017, *MSR-net: Low-light Image Enhancement Using Deep Convolutional Network*, , , January. <http://arxiv.org/abs/1711.02488>,.
- Tanaka, Y., Yamashita, Y., Nishikawa, K., Yamaguchi, T. & Nishitani, T., 2019, *Retinex Foreground Segmentation for Low Light Environments*, 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2018 - Proceedings, , November, 285–290.
- Yingxin, X., Jinghua, L., Lichun, W. & Dehui, K., 2017, *A Robust Hand Gesture Recognition Method via Convolutional Neural Network*, Proceedings - 2016 International Conference on Digital Home, ICDH 2016, 64–67.
- Madenda, Sarifudin., 2015, *Pengolahan & Video Digital*, Erlangga, Jakarta [BUKU].
- Nielsen, M., 2015, *Neural Networks and Deep Learning*, Determination Press., [Daring]. tersedia di <http://neuralnetworksanddeeplearning.com>.
- Rinaldi, Munir., 2004, *Pengolahan Citra Digital*, Bandung, Informatika [BUKU].

- Hidayatullah, Priyanto,. 2017,*Pengolahan Citra Digital Teori dan Aplikasi Nyata*, Bandung, Informatika[BUKU].
- Barczak, A.L.C., Reyes, N.H., Abastillas, M., Piccio, A. & Susnjak, T., 2011, A *New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures*, Res. Lett. Inf. Math. Sci., 15, 12-20.<http://iims.massey.ac.nz/research/letters/>.
- Kolkur, S., Kalbande, D., Shimpi, P., Bapat, C. & Jatakia, J., 2017, *Human Skin Detection Using RGB , HSV and YCbCr Color Models*, , 137, 324-332.
- Arabi, S., Haghighat, A. & Sharma, A., 2019, *A deep learning based solution for construction equipment detection: from development to deployment* , , April.
- Srinivasan, R., 2016, *Implementing Histogram Equalization and Retinex*, , July.
- Huang, H., Chong, Y., Nie, C. & Pan, S., 2019, *Hand Gesture Recognition with Skin Detection and Deep Learning Method* *Hand Gesture Recognition with Skin Detection and Deep Learning Method*, J. Phys.: Conf. Ser. 1213 022001.
- Posada-Gómez, Rubén & Sanchez Medel, Luis & Alor-Hernández, Giner & Martinez Sibaja, Albino & Aguilar-Laserre, A. & Leija-Salas, L.. 2007. *A Hands Gesture System Of Control For An Intelligent Wheelchair*. 68 - 71. 10.1109/ICEE.2007.4344975.
- Kemenkes, 2018, *Indonesia Inklusi dan Ramah Disabilitas*[Daring], tersedia di <https://www.kemkes.go.id/resources/download/pusdatin/infodatin/infodatin-disabilitas.pdf>.
- Afrianto, T. & Amalia, F., 2016, *Pengaruh Komponen Krominan Pada Ruang Warna*, Prosiding Seminar Nasional Teknologi Terapan (SNTT), , August 2017, 282-285.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H., 2017, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, <http://arxiv.org/abs/1704.04861>.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.C., 2018, *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 45104520.
- Google AI Blog., 2018, *MobileNetV2: The Next Generation of On-Device Computer Vision Networks*[Daring], tersedia di [ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html?m=1](https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html?m=1).
- Leonard, L.C., 2017, *Web-Based Behavioral Modeling for Continuous User Authentication (CUA)*, 1 edisi, Elsevier Inc., [Daring]. tersedia di DOI:10.1016/bs.adcom.2016.12.001.
- Hui, Jonathan., 2018, *mAP (mean Average Precision) for Object Detection*[Daring], tersedia di [www.medium.com/@jonathan\\_hui/map-mean-average-precision-for-object-detection-45c121a31173](https://www.medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173).
- Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P & Zitnick, C.L., 2015, *Microsoft COCO Captions: Data Collection and Evaluation Server*, arXiv:1504.00325
- Hanumantharaju, M.C., Ravishankar, M., Rameshbabu, D.R., Manjunath Aradhya V.N., 2014, *A new framework for retinex-based colour image enhancement using particle swarm optimisation*, International Journal of Swarm Intelligence, 2014 Vol.1 No.2, pp.133-155, DOI:10.1504/IJSI.2014.060241.