



TESIS - RM 185401

**ANALISIS KLASIFIKASI CITRA SATELIT WORLDVIEW-  
2 MENGGUNAKAN MODEL *DEEP LEARNING***

(Studi Kasus : Peta Tutupan Lahan Perkotaan Terangun,  
Kabupaten Gayo Lues, Provinsi Aceh)

Ilyas  
03311750012004

Dosen Pembimbing  
Lalu Muhamad Jaelani, S.T., M.Sc., Ph.D.

Departemen Teknik Geomatika  
Fakultas Teknik Sipil, Perencanaan, dan Kebumihan  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2020



TESIS - RM 185401

**ANALISIS KLASIFIKASI CITRA SATELIT  
WORLDVIEW-2 MENGGUNAKAN MODEL *DEEP  
LEARNING***

**(Studi Kasus : Peta Tutupan Lahan Perkotaan Terangun,  
Kabupaten Gayo Lues, Provinsi Aceh)**

**Ilyas**

**03311750012004**

**Dosen Pembimbing**

**Lalu Muhamad Jaelani, S.T., M.Sc., Ph.D.**

**Departemen Teknik Geomatika  
Fakultas Teknik Sipil, Perencanaan, dan Kebumihan  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2020**

*Halaman ini sengaja dikosongkan*

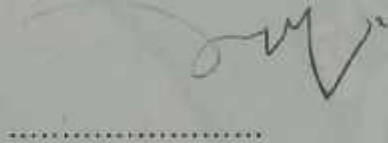
**LEMBAR PENGESAHAN TESIS**

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
**Magister Teknik (MT)**  
di  
**Institut Teknologi Sepuluh Nopember**  
oleh:  
**ILYAS**  
**NRP: 63311750012004**

Tanggal Ujian : 16 Januari 2020  
Periode Wisuda : Maret 2020

Disetujui oleh:  
Pembimbing:

1. Lalu Muhamad Jaelani, S.T., M.Sc.,  
Ph.D  
NIP: 198012212003121001



.....

Penguji:

1. Prof. Dr. Ir. Bangun Muljo Sukojo,  
DEA., DESS.  
NIP: 195305271983031001




.....

2. Dr. Ir. Muhammad Taufik  
NIP: 195509191986031001



.....

3. Hesti Hapsari Handayani, ST., M.Sc.,  
Ph.D.  
NIP:197812122005012001



.....

**Kepala Departemen Teknik Geomatika  
Fakultas Teknik Sipil, Perencanaan, dan Kebumihan**



**Danar Garuh Pratomo, S.T., M.T., Ph.D.**  
NIP: 198005072003121001



*Halaman ini sengaja dikosongkan*

**ANALISIS KLASIFIKASI CITRA SATELIT  
WORLDVIEW-2  
MENGUNAKAN MODEL *DEEP LEARNING***

Nama : Ilyas  
NRP : 03311750012004  
Pembimbing : Lalu Muhamad Jaelani, S.T., M.Sc., Ph.D.

**ABSTRAK**

Dalam menyusun sebuah dokumen Rencana Detil Tata Ruang (RDTR) maka diperlukan berbagai informasi peta dasar dari suatu wilayah yang akan direncanakan. Salah satu informasi tersebut adalah tutupan lahan. Informasi tutupan lahan merupakan informasi tahap awal yang harus tersedia agar para penyusun dokumen tata ruang memahami kondisi keruangan pada suatu area perencanaan.

Informasi tutupan lahan diperoleh dari hasil klasifikasi citra satelit WorldView-2 dengan resolusi spasial 0,5 meter. Penerapan dari klasifikasi model *deep learning* yaitu peta tutupan lahan perkotaan Terangun, Kabupaten Gayo Lues, Provinsi Aceh dengan skala 1:5.000 yang terdiri dari bangunan, jalan, lahan terbuka, sungai, sawah, ladang, kolam, dan kebun.

Uji akurasi klasifikasi model *deep learning* ini menunjukkan akurasi 65% *overall accuracy* dan 48,09% *kappa accuracy*, sedangkan metode *maximum likelihood classification* menunjukkan 49% *overall accuracy* dan *kappa accuracy* sebesar 24,56%. Sehingga dapat disimpulkan klasifikasi model *deep learning* mampu melakukan klasifikasi lebih baik dibandingkan dengan metode *maximum likelihood classification*.

**Kata Kunci** : Kecerdasan Buatan, Pembelajaran Mendalam, RDTR, Tutupan Lahan, WorldView-2.

*Halaman ini sengaja dikosongkan*

# **SATELLITE IMAGE CLASSIFICATION ANALYSIS OF WORLDVIEW-2 USING DEEP LEARNING MODEL**

Name : Ilyas  
ID : 03311750012004  
Supervisor : Lalu Muhamad Jaelani, S.T., M.Sc., Ph.D.

## **ABSTRACT**

In compiling a detailed Spatial Planning (RDTR) document, various basic map information from an area is needed. One such information is land cover. Land cover information is the initial information that must be available so that the drafters of spatial documents understand the spatial conditions in a planning area.

Land cover information was obtained from the classification of WorldView-2 satellite imagery with a spatial resolution of 0.5 meters. The application of the deep learning model classification is a map of urban land cover Terangun, Gayo Lues Regency, Aceh Province with a scale of 1: 5,000 consisting of buildings, roads, open land, rivers, rice fields, fields, ponds and gardens.

The classification accuracy test of this deep learning model shows the accuracy of 65% overall accuracy and 48.09% kappa accuracy, while the maximum likelihood classification method shows 49% overall accuracy and kappa accuracy of 24.56%. So it can be concluded that the classification of deep learning model is able to do a better classification than the maximum likelihood classification method.

**Keyword** : Artificial Intelligence, Deep Learning, Detail Spatial Planning, Landcover, WorldView-2.

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan atas qudrah dan iradah dari Allah SWT, sehingga penulis dapat menyelesaikan penyusunan laporan tesis ini yang berjudul “**Analisis Klasifikasi Citra Satelit WorldView-2 Menggunakan Model *Deep Learning* (Studi Kasus : Peta Tutupan Lahan Perkotaan Terangun, Kabupaten Gayo Lues, Provinsi Aceh)**”.

Penulis mengucapkan terimakasih yang tak terhingga kepada kedua orang tua dan adik penulis yang selalu memberikan dukungan moril, materil, dan doa kepada penulis. Terimakasih juga kepada untuk istri tercinta, yang sudah sangat sabar menemani penulis, selalu mendoakan, dan memberikan dukungan selama menempuh pendidikan, mohon maaf karena sudah meleset dari jadwal yang sudah kita sepakati bersama dalam menyelesaikan kuliah.

Tak lupa pula ucapan terimakasih penulis kepada Bapak Lalu Muhamad Jaelani, S.T., M.Sc., Ph.D., selaku pembimbing yang sudah bersedia membimbing penulis hingga selesai penyusunan tesis ini. Terimakasih juga kepada Prof. Dr. Ir. Bangun Muljo Sukojo, DEA, DESS, Dr.Ir. Muhammad Taufik, dan Ibu Hepi Hapsari Handayani, ST., M.Sc., Ph.D., yang sudah sangat banyak memberi masukan hingga selesai penyusunan tesis ini.

Tanpa bantuan dari berbagai pihak, penelitian ini tidak dapat terselesaikan, oleh karena itu pada kesempatan ini penulis ingin menyampaikan terimakasih sebesar-besarnya kepada:

1. Bapak Danar Guruh Pratomo, S.T., M.T., Ph.D., selaku Kepala Departemen Teknik Geomatika ITS Surabaya.
2. Ibu Ira Mutiara Anjasmara, S.T., M.Phil., Ph.D selaku koordinator tesis dan Kepala Prodi S2 Teknik Geomatika ITS Surabaya.
3. Seluruh Dosen Pengajar, Staf dan Karyawan Jurusan Teknik Geomatika ITS Surabaya.

4. Daniel Wilson, selaku *Senior Data Scientist at ESRI* yang sudah bersedia berdiskusi dengan penulis tentang penelitian ini.
5. Alim Misbullah yang telah membantu memberikan arahan disaat penulis mendapatkan kendala dalam membuat program.
6. Kepada Mas Fajar, Mbak Rere dan kawan-kawan S2 semuanya yang tidak dapat penulis sebutkan namanya satu persatu disini, terimakasih atas kebaikan kalian selama penulis menempuh pendidikan disini, semoga Allah membalas kebaikan kalian semua.

Penulis sadar bahwasanya laporan tesis ini belum sempurna dan masih banyak kekurangan, maka jika ditemukan kesalahan dalam penulisan, maka penulis siap menerima segala kritikan dan saran yang bersifat membangun demi perbaikan penulisan kedepannya. Harapan penulis semoga laporan tesis ini dapat memberikan manfaat kepada para pembaca.

Surabaya, 30 Januari 2020

Penulis

## DAFTAR ISI

<b>JUDUL PROPOSAL</b> .....	<b>i</b>
<b>LEMBARAN PENGESAHAN TESIS</b> .....	<b>iii</b>
<b>ABSTRAK</b> .....	<b>v</b>
<b>ABSTRACT</b> .....	<b>vii</b>
<b>KATA PENGANTAR</b> .....	<b>ix</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR GAMBAR</b> .....	<b>xv</b>
<b>DAFTAR TABEL</b> .....	<b>xviii</b>
<b>BAB 1 PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	5
<b>BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI</b> .....	<b>7</b>
2.1. Penginderaan Jauh .....	7
2.2. Klasifikasi .....	8
2.3. Klasifikasi Berbasis Pikel.....	9
2.3.1. Klasifikasi Terbimbing.....	10
2.3.1.1. <i>Maximum Likelihood Classification</i> .....	10
2.3.2. Klasifikasi Tidak Terbimbing .....	12
2.4. Klasifikasi Berbasis Objek.....	12
2.5. WorldView-2 Level 2A .....	13
2.5.1. Citra Pankromatik .....	14
2.5.2. Citra Multispektral .....	15
2.6. Orthorektifikasi .....	16
2.6.1. <i>Root Mean Square Error (RMSE)</i> .....	17
2.7. Interpretasi Citra .....	18
2.8. Ketelitian Hasil Klasifikasi .....	19



2.9. <i>Confusion Matix</i> .....	20
2.10. Metode Uji Ketelitian <i>Overall Accuracy</i> .....	22
2.11. Metode Uji Ketelitian <i>Kappa Accuracy</i> .....	23
2.12. Skala Peta.....	26
2.13. <i>Application Programming Interface (API)</i> .....	26
2.13.1. Python .....	26
2.13.2. Keras .....	27
2.13.3. TensorFlow .....	28
2.14. <i>Artificial Intelligency (AI)</i> .....	29
2.15. <i>Machine Learning</i> .....	29
2.16. <i>Deep Learning</i> .....	30
2.16.1. <i>Artificial Neural Network (ANN)</i> .....	31
2.16.2. <i>Deep Neural Network</i> .....	32
2.16.3. <i>Arsitektur Deep Neural Network</i> .....	32
2.17. <i>Training Neural Network</i> .....	33
2.17.1. <i>Feedforward</i> .....	35
2.17.2. <i>Activation Function</i> .....	35
2.17.3. <i>Loss Function</i> .....	39
2.17.4. <i>Backpropagation</i> .....	39
2.17.5. <i>Gradient Descent</i> .....	41
2.17.6. <i>Stochastic Gradiet Descent (SGD)</i> .....	44
2.18. <i>Convolutional Neural Network (CNN)</i> .....	45
2.18.1. <i>Feature Extraction (Learning) Layers</i> .....	47
2.18.2. <i>Convolutional Layer</i> .....	47
2.18.3. <i>Pooling Layer</i> .....	50
2.18.4. <i>Classification Layer</i> .....	51
2.18.5. <i>Flattening</i> .....	52
2.18.6. <i>Fully Connected Layer</i> .....	52
2.18.7. <i>Softmax</i> .....	53
2.19. <i>Augmentasi</i> .....	54
2.20. <i>Segmentasi</i> .....	54
2.20.1. <i>U-Net</i> .....	55

2.20.2. <i>Intersection over Union (IoU)</i> .....	57
2.21. Klasifikasi Tutupan Lahan Skala 1:5.000 .....	57
2.22. Penelitian Terdahulu .....	58
<b>BAB 3 METODOLOGI PENELITIAN</b> .....	<b>61</b>
3.1. Gambaran Umum Daerah Kajian.....	61
3.2.Peralatan dan Data.....	63
3.2.1. Peralatan.....	63
3.2.2. Data .....	63
3.3. Variabel Penelitian .....	63
3.4. Diagram Alir dan Alur Penelitan .....	65
<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	<b>69</b>
4.1. Citra Satelit WorldView-2 Level 2A Terorthorektifikasi .....	69
4.2. Hasil Orthorektifikasi.....	69
4.3. <i>Subset</i> Citra Satelit WorldView-2 Level 2A.....	71
4.4. <i>Training Sample</i> untuk Klasifikasi .....	71
4.5. Konversi Data Citra dan <i>Training Sample</i> Menjadi Data <i>Deep Learning</i> .....	73
4.6. Konversi Kelas Tutupan Lahan Kedalam Format HDF5 .....	76
4.7. Konversi Kelas Tutupan Lahan dan Citra Numpy Kedalam Format HDF5.....	84
4.8. Melatih Model U-Net.....	86
4.9. Augmentasi Kelas Tutupan Lahan.....	87
4.10. Menggabungkan File <i>Numpy</i> .....	89
4.11. Memisahkan File Citra dan Kelas Tutupan Lahan dari <i>Numpy Array</i> .....	92
4.12. Konversi File Numerik Citra dan Tutupan Lahan menjadi bilangan intiger .....	93
4.13. Mencetak File <i>Numpy</i> Citra dan Kelas Tutupan Lahan Kedalam Format HDF5 .....	94
4.14. Konversi Kelas Tutupan Lahan menjadi Vektor .....	95
4.15. Mendefinisikan Parameter <i>Training</i> dan Mendefinisikan Jumlah Kelas Tutupan Lahan .....	95

4.16. Membuat Arsitektur Model U-Net .....	96
4.17. Visualisasi Model <i>Deep Learning</i> .....	99
4.18. Hasil Klasifikasi Model <i>Deep Learning</i> .....	104
4.19. Hasil Klasifikasi Metode <i>Maximum Likelihood Classification</i> .....	105
4.19.1. <i>Training Sample</i> .....	105
4.20. Batas Wilayah Perencanaan (BWP) Perkotaan Terangun.....	108
4.21. Luas Tutupan Lahan.....	109
4.22. Selisih Luasan Tutupan Lahan .....	116
4.22.1. Selisih Luasan Klasifikasi Model <i>Deep Learning</i> dengan Tutupan Lahan RDTR Kota Terangun.....	116
4.22.2. Selisih Luasan Klasifikasi Metode <i>Maximum</i> <i>Likelihood Classification</i> dengan Tutupan Lahan RDTR Terangun .....	118
4.22.3. Selisih Luasan Klasifikasi Model <i>Deep Learning</i> dengan Metode <i>Maximum Likelihood Classification</i> .....	119
4.23. Uji Akurasi .....	120
4.23.1. Uji Akurasi Model <i>Deep Learning</i> dengan Tutupan Lahan RDTR Perkotaan Terangun .....	120
4.23.2. Uji Akurasi Metode <i>Maximum Likelihood Classification</i> dengan Tutupan Lahan RDTR Perkotaan Terangun .....	122
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>125</b>
5.1. Kesimpulan.....	125
5.2. Saran.....	126
<b>DAFTAR PUSTAKA .....</b>	<b>127</b>
<b>LAMPIRAN.....</b>	<b>131</b>
<b>BIODATA PENULIS.....</b>	<b>139</b>

## DAFTAR GAMBAR

Gambar 2.1. Spektrum Gelombang Elektromagnetik .....	8
Gambar 2.2. Satelit WorldView-2 .....	14
Gambar 2.3. Citra Pankromatik Perkotaan Terangun .....	15
Gambar 2.4. Citra Multispektral Perkotaan Terangun .....	15
Gambar 2.5. Orthorektifikasi .....	16
Gambar 2.6. Ketelitian Geometri Perka BIG No 15 Tahun 2014.....	16
Gambar 2.7. Ketelitian Geometri Perka BIG No 18 Tahun 2018.....	17
Gambar 2.8. Matrik Kesalahan .....	23
Gambar 2.9. Matrik <i>Confusion</i> .....	24
Gambar 2.10. Cabang ilmu dari AI.....	29
Gambar 2.11. Model <i>Artificial Neural Network</i> (ANN).....	31
Gambar 2.12. Arsitektur <i>Neural Network</i> dengan Satu <i>Hidden Layer</i> .....	32
Gambar 2.13. Proses Training pada <i>Neural Network</i> Sederhana.....	33
Gambar 2.14. Fungsi Aktivasi <i>Sigmoid</i> .....	36
Gambar 2.15. Fungsi Aktivasi <i>Hyperbolic Tangen (tanh)</i> .....	36
Gambar 2.16. Fungsi Aktivasi ReLU.....	37
Gambar 2.17. <i>Sigmoid Function</i> .....	38
Gambar 2.18. <i>Gradient</i> .....	42
Gambar 2.19. Arsitektur <i>Convolutional Neural Network</i> .....	45
Gambar 2.20. Operasi dari <i>Convolution</i> .....	47
Gambar 2.21. Ukuran <i>Filter</i> .....	47
Gambar 2.22. Ketebalan <i>Output</i> .....	48
Gambar 2.23. <i>Stride</i> dengan nilai 1 piksel .....	48
Gambar 2.24. <i>Padding</i> berukuran 1x1 piksel.....	49
Gambar 2.25. <i>Max Pooling</i> 2x2 dengan <i>stride</i> 2 piksel.....	50
Gambar 2.26. <i>Fully Connected Layer</i> .....	52
Gambar 2.27. <i>Intersection over Union (IoU)</i> .....	53
Gambar 2.28. Segmentasi Gambar.....	54
Gambar 2.29. Arsitektur U-Net.....	56

Gambar 2.30. <i>Intersection over Union (IoU)</i> .....	56
Gambar 2.31. Kelas Akurasi IoU .....	57
Gambar 3.1. Peta Lokasi Perkotaan Terangun.....	60
Gambar 3.2. Diagram Alir Penelitian .....	64
Gambar 4.1. Citra Satelit WorldView-2 .....	67
Gambar 4.2. Bagian Citra Satelit yang di <i>Subset</i> .....	69
Gambar 4.3. Sebaran <i>Training Sample</i> .....	71
Gambar 4.4. Proses Split Citra Satelit Worldview-2 .....	72
Gambar 4.5. Data Citra dan <i>Training Sample</i> Berukuran 128x128 Piksel .....	73
Gambar 4.6. Data Citra dan Kelas Tutupan Lahan Representasi Kebun.....	80
Gambar 4.7. File Kelas Tutupan Lahan dalam Format HDF5 .....	80
Gambar 4.8. Tampilan Citra Satelit dan Kelas Tutupan Lahan Representasi dari Kebun.....	82
Gambar 4.9. Proses Augmentasi Citra .....	85
Gambar 4.10. Tampilan Citra dan Kelas Tutupan Lahan Gabungan File <i>Numpy</i> Representasi Sawah .....	87
Gambar 4.11. Arsitektur model U-Net.....	91
Gambar 4.12. Jumlah Parameter Tiap Proses <i>Convolution</i> .....	93
Gambar 4.13. Grafik Model <i>Loss Validation</i> .....	94
Gambar 4.14. Grafik Model <i>Mean IoU Validation</i> .....	94
Gambar 4.15. Grafik Model <i>Loss Training</i> .....	95
Gambar 4.16. Grafik Model <i>Mean IoU Training</i> .....	95
Gambar 4.17. Peta Tutupan Lahan Hasil Klasifikasi model <i>Deep Learning</i> ....	98
Gambar 4.18. Sebaran <i>Training Sample</i> .....	100
Gambar4.19. Peta Tutupan Lahan Hasil Klasifikasi Metode <i>Maximum Likelihood Classification</i> .....	101
Gambar 4.20. Peta BWP Perkotaan Terangun.....	102
Gambar 4.21. Peta Tutupan Lahan Model <i>Deep Learning</i> Perkotaan Terangun .....	103
Gambar 4.22. Peta Tutupan Lahan Metode <i>Maximum Likelihood Classification</i> .....	105

Gambar 4.23. Peta Tutupan Lahan Hasil Interpretasi Berbasis Objek RDTR

Terangun ..... 102

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1. Band dan Lebar Spektrum Citra Multispektral .....	13
Tabel 2.2. Matrik <i>Confusion</i> .....	20
Tabel 2.3. Contoh Matrik <i>Confusion</i> .....	21
Tabel 2.4. Sumber Nilai Akurasi dari Matrik <i>Confusion</i> .....	21
Tabel 4.1. Jumlah Sampel Tutupan Lahan untuk Tiap Kelas Tutupan Lahan..	70
Tabel 4.2. Kategori Data <i>Deep Learning</i> .....	74
Tabel 4.3. Kelas Tutupan Lahan yang akan Dittraining .....	75
Tabel 4.4. Jumlah Sampel Kelas Tutupan Lahan.....	75
Tabel 4.5. Susunan Format Data <i>Numpy</i> .....	76
Tabel 4.6. Susunan Format Data <i>Numpy</i> Setelah Proses <i>Roolaxis</i> .....	76
Tabel 4.7. Pemisahan Data Citra dan Kelas Tutupan Lahan.....	77
Tabel 4.8. Normalisasi Data.....	78
Tabel 4.9. Normalisasi Data <i>Training Sample</i> .....	78
Tabel 4.10. Gabungan Data Citra dan Kelas Tutupan Lahan .....	79
Tabel 4.11. File Kelas Tutupan Lahan dalam Format HDF5.....	81
Tabel 4.12. Data Kelas Tutupan Lahan yang Akan Dilakukan Proses Augmentasi .....	82
Tabel 4.13. Jumlah Data Setelah Proses Augmentasi .....	84
Tabel 4.14. Gabungan File <i>Numpy</i> Kelas Tutupan Lahan yang Kurang .....	86
Tabel 4.15. Gabungan File <i>Numpy</i> Kelas Tutupan Lahan yang Banyak .....	86
Tabel 4.16. Gabungan File <i>Numpy</i> Kelas Tutupan Lahan yang Kurang dan Banyak.....	87
Tabel 4.17. File Citra yang Sudah Dipisahkan dari <i>Numpy</i> .....	88
Tabel 4.18. File Kelas Tutupan Lahan yang Sudah Dipisahkan dari <i>Numpy</i> ...	88
Tabel 4.19. Konversi Bilangan Data Citra dan Kelas Tutupan Lahan dari Bilangan <i>Floating</i> menjadi Bilangan <i>Intiger</i> .....	88
Tabel 4.20. File HDF5 Citra dan Kelas Tutupan Lahan .....	89
Tabel 4.21. Konversi Kelas Tutupan lahan dalam Format Vektor .....	89
Tabel 4.22. Jumlah Kelas Tutupan Lahan.....	90



Tabel 4.23. Perkembangan Nilai Akurasi Tiap <i>Epoch</i> .....	96
Tabel 4.24. Jumlah Sampel Tutupan Lahan untuk Tiap Kelas Tutupan Lahan.....	99
Tabel 4.25. Luas Tutupan Lahan Model <i>Deep Learning</i> .....	102
Tabel 4.26. Luas Tutupan Lahan Metode <i>Maximum Likelihood Classification</i> .....	104
Tabel 4.27. Selisih Luasan Kelas Tutupan Lahan.....	106
Tabel 4.28. Luas Tutupan Lahan Interpretasi Berbasis Objek.....	107
Tabel 4.29. Matrik <i>Confusion</i> Tutupan Lahan Model <i>Deep Learning</i> .....	109
Tabel 4.30. Matrik <i>Confusion</i> Tutupan Lahan Metode <i>Maximum Likelihood Classification</i> .....	111

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Pesatnya perkembangan teknologi dan informasi telah mendorong terjadinya revolusi industri 4.0 dan telah berdampak besar bagi kehidupan masyarakat khususnya dalam hal proses produksi menjadi lebih otomatis dan efisien. Revolusi industri 4.0 terjadi karena salah satu penyebabnya yaitu berkembangnya teknologi *Artificial Intelligence* (AI).

Menurut Minsky (1989) dalam kursini (2008) *Artificial Intelligence* adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Salah satu dari kemampuan yang dimiliki AI adalah kemampuan mengenali citra.

Menurut Tayyebi (2019) dalam penelitiannya menggunakan enam kelas tutupan lahan yang terdiri dari bangunan, jaringan jalan, tubuh air, ladang, kebun, dan hutan. Citra satelit yang digunakan mempunyai resolusi 1 meter, sedangkan *epoch* yang digunakan sebanyak 30 *epoch* . Akurasi yang dihasilkan dari klasifikasi tutupan lahan tersebut cukup bagus, akurasi tersebut meliputi *precision* yaitu bangunan (82,50%), jaringan jalan (84,78%), tubuh air (86,14%), ladang (90,38%), kebun (89,05%), dan hutan (91,46%). Sedangkan akurasi *recall* diperoleh bangunan (81,28%), jaringan jalan (85,13%), tubuh air (85,55%), ladang (91,88%), kebun (88,19%), dan hutan (92,65%).

Menurut Sutanto (1986) citra satelit merupakan gambaran rekaman objek yang dihasilkan dengan cara optik, elektro-optik, optik-mekanik, dan elektronik. Gambar yang dihasilkan mirip dengan objek sesungguhnya di alam. Citra satelit merupakan salah satu data dasar yang digunakan dalam membuat peta tutupan lahan. Banyak sekali sumber data citra satelit yang digunakan untuk membuat peta tutupan lahan selain citra satelit Worldview-2 level 2A.

Citra satelit Worldview-2 level 2A merupakan salah satu citra satelit resolusi tinggi yang mempunyai resolusi spasialnya pankromatik yaitu 0,5 m dan resolusi spasial multispektral yaitu 1,85 m. Adapun resolusi spasial yang dimiliki

oleh citra satelit worldview-2 level 2A sudah mendukung dalam membuat peta tutupan lahan dengan skala 1:5.000. Peta tutupan lahan merupakan salah satu peta yang dibutuhkan dalam dokumen Rencana Tata Ruang Wilayah (RTRW).

Undang-undang Nomor 26 tahun 2007 mengamanatkan bahwa tiap-tiap daerah perlu menyusun Peraturan Daerah (PERDA) tentang Rencana Tata Ruang Wilayah (RTRW). Secara hirarki, RTRW terdiri dari RTRW Nasional dengan skala 1:1.000.000, RTRW Pulau dengan skala 1:500.000, RTRW Provinsi dengan skala 1:250.000, RTRW Kabupaten dengan skala 1:50.000, RTRW Kota dengan skala 1:25.000, dan Rencana Detail Tata Ruang (RDTR) dengan skala 1:5.000. Skala peta tersebut tertuang dalam Peraturan Pemerintah Nomor 8 Tahun 2013 tentang ketelitian peta rencana tata ruang.

Qanun (Perda) Kabupaten Gayo Lues, No.15, Tahun 2013, tentang Rencana Tata Ruang Wilayah (RTRW) Kabupaten Gayo Lues 2012-2032, Perkotaan Terangun ditetapkan sebagai salah satu kawasan strategis kabupaten berdasarkan sudut pandang ekonomi. Perkotaan Terangun termasuk perkotaan yang cepat berkembang dan diproyeksikan akan menjadi sebuah kota transit.

Adapun studi kasus dalam penelitian ini yaitu peta tutupan lahan perkotaan Terangun, Kabupaten Gayo Lues, Provinsi Aceh dengan skala 1:5.000. Berdasarkan hasil observasi citra satelit WorldView-2 level 2A maka dapat disimpulkan bahwa kelas tutupan lahan di Perkotaan Terangun terdiri dari bangunan, jaringan jalan, kebun campuran, kolam, pekarangan, sawah, sungai, dan ladang.

Berdasarkan permasalahan yang telah dijabarkan diatas maka dalam penelitian ini akan dilakukan analisa klasifikasi citra satelit worldview-2 level 2A menggunakan salah satu cabang ilmu dari *artificial intelligence* yaitu *deep learning*. *Deep learning* merupakan model yang secara dramatis telah meningkatkan seni dalam mengenali suara, objek, dan banyak tugas lainnya (LeCun, Bengio, dan Hinton, 2015).

Klasifikasi citra satelit WorldView-2 level 2A dengan model *deep learning* dan *maximum likelihood classification* untuk mendapatkan informasi luas tutupan lahan di Perkotaan Terangun. Hasil klasifikasi kedua metode tersebut kemudian dibandingkan dengan klasifikasi tutupan lahan berbasis objek yaitu peta

tutupan lahan Perkotaan Terangun yang sudah mendapatkan rekomendasi dari Badan Informasi Geospasial (BIG).

## **1.2. Perumusan Masalah**

Mengacu pada latar belakang penelitian, permasalahan yang muncul penelitian ini adalah:

- a. Bagaimana melakukan klasifikasi citra satelit WorldView-2 level 2A dengan model *deep learning*.
- b. Bagaimana melakukan klasifikasi citra satelit WorldView-2 level 2A dengan metode *maximum likelihood classification*.
- c. Bagaimana akurasi informasi tutupan lahan hasil klasifikasi model *deep learning* dengan klasifikasi berbasis objek yang sudah mendapat rekomendasi dari BIG.
- d. Bagaimana akurasi informasi tutupan lahan hasil klasifikasi metode *maximum likelihood classification* dengan klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG
- e. Bagaimana akurasi informasi tutupan lahan hasil klasifikasi *deep learning* dan metode *maximum likelihood classification*.

## **1.3. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah :

- a. Studi kasus dibatasi pada tutupan lahan yang terdiri dari delapan kelas tutupan lahan dari bangunan, jalan, lahan terbuka, sungai, sawah, ladang, kolam, dan kebun.
- b. Data raster yang digunakan adalah citra satelit WorldView-2 level 2A yang telah terortorektifikasi dan telah mendapatkan rekomendasi dari Badan Informasi Geospasial (BIG).
- c. Data vektor yang digunakan berupa data batas administrasi (kabupaten dan kecamatan) yang bersumber dari Dinas PUPR Kabupaten Gayo Lues dengan skala 1:50.000, dan data vektor BWP Terangun dan peta tutupan lahan lahan Perkotaan Terangun dengan skala 1:5.000.

d. Dalam membuat model *deep learning* ukuran data yang digunakan yaitu 128x128 piksel, dan *stride* berukuran 64 piksel.

e. Dalam membuat model *deep learning* jumlah *epoch* yang digunakan adalah 50 *epoch*.

f. Mengetahui kesesuaian klasifikasi tutupan lahan model *deep learning* dan metode *maximum likelihood classification* dengan tutupan lahan berbasis objek yang sudah mendapatkan rekomendasi dari BIG, Tahun 2017.

g. Mengetahui perbedaan luasan klasifikasi tutupan lahan model *deep learning* dan metode *maximum likelihood classification* dengan peta tutupan lahan hasil klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG, Tahun 2017.

#### **1.4. Tujuan Penelitian**

a. Untuk merancang program klasifikasi citra Worldview-2 level 2A berbasis Python, Keras dan TensorFlow.

b. Untuk mengetahui perbedaan hasil klasifikasi tutupan lahan model *deep learning* dan metode *maximum likelihood classification* dengan klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG.

c. Untuk menganalisa hasil klasifikasi citra WorldView-2 level 2A dengan model *deep learning*.

d. Untuk menganalisa hasil klasifikasi citra WorldView-2 level 2A dengan metode *maximum likelihood classification*.

e. Untuk mengetahui perbedaan hasil klasifikasi tutupan lahan model *deep learning* dengan klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG.

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

Adapun manfaat yang diharapkan dalam penelitian ini adalah :

a. Tersedianya peta tutupan lahan hasil klasifikasi model *deep learning* dengan skala 1:5.000.

b. Tersedianya peta tutupan lahan hasil klasifikasi metode *maximum likelihood classification* dengan skala 1:5.000.

c. Tersedianya informasi perbedaan luasan tutupan lahan yang dihasilkan dari klasifikasi model *deep learning* dengan peta tutupan lahan hasil klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG.

d. Tersedianya informasi perbedaan luasan tutupan lahan yang dihasilkan dari klasifikasi metode *maximum likelihood classification* dengan peta tutupan lahan hasil klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG dengan skala 1:5.000.

*Halaman ini sengaja dikosongkan*

## **BAB 2**

### **KAJIAN PUSTAKA DAN DASAR TEORI**

#### **2.1. Penginderaan Jauh**

Penginderaan jauh (*remote sensing*) adalah ilmu dan seni untuk memperoleh informasi tentang suatu objek, daerah atau fenomena melalui analisis data yang diperoleh dengan suatu alat tanpa kontak langsung dengan objek daerah atau fenomena yang dikaji (Lillesand dan Kiefer, 1999).

Menurut S. Martin (2004) penginderaan jauh (*remote sensing*) adalah penggunaan gelombang radiasi elektromagnetik untuk memperoleh informasi tentang lautan, daratan dan atmosfer tanpa kontak langsung dengan objek, permukaan atau fenomena yang dikaji. Data penginderaan jauh dapat berupa citra, grafik, dan data numerik. Data tersebut dapat dianalisis untuk mendapatkan informasi tentang objek daerah atau fenomena yang diteliti. Interpretasi citra atau penafsiran citra penginderaan jauh dilakukan untuk mengidentifikasi objek yang tergambar dalam citra dan menilai arti pentingnya objek tersebut.

Empat komponen dasar dari system penginderaan jauh adalah target, sumber energy, alur transmisi, dan sensor. Komponen dalam system ini bekerja bersama untuk mengukur dan mencatat informasi target tanpa menyentuh objek tersebut. Sumber energi yang menyinari atau memancarkan energi elektromagnetik pada target mutlak diperlukan.

Energi berinteraksi dengan target dan sekaligus berfungsi sebagai media untuk meneruskan informasi dari target kepada sensor. Sensor adalah sebuah alat yang mengumpulkan dan mencatat radiasi elektromagnetik. Setelah dicatat, data akan dikirimkan ke stasiun penerima dan diproses menjadi format yang siap dipakai, diantaranya berupa citra. Citra ini kemudian diinterpretasikan untuk memperoleh informasi mengenai target, objek atau fenomena yang dikaji.

Prinsip utama dari system peninderaan jauh adalah interaksi gelombang elektromagnetik dengan suatu objek. Energi elektromagnetik yang mengenai suatu objek akan mengalami tiga bentuk interaksi. Ketiga Interaksi ini dapat ditulis dengan :

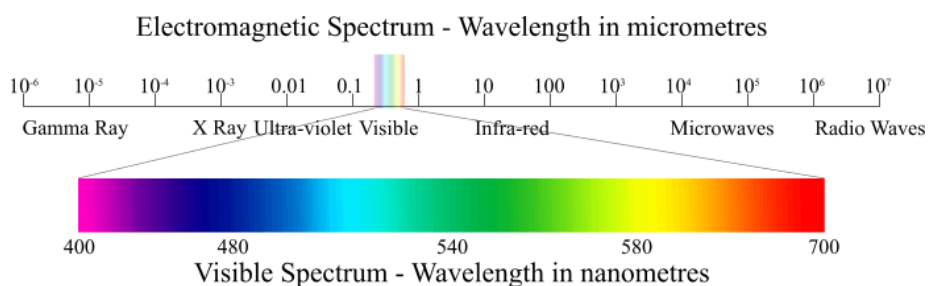


$$EI(\lambda) = ER(\lambda) + EA(\lambda) + ET(\lambda) \quad (2.1)$$

Keterangan :

- EI = Energi EM yang mengenai objek (nm)
- ER = Energi EM yang dipantulkan objek (nm)
- EA = Energi EM yang diserap objek (nm)
- ET = Energi EM yang diteruskan objek (nm)
- $\lambda$  = Panjang gelombang (nm)

Energi elektromagnetik merupakan sebuah medium untuk pengiriman informasi dari target kepada sensor. Energi elektromagnetik merambat dalam gelombang dengan beberapa karakter yang bias diukur, yaitu: panjang gelombang (*wavelength*), frekuensi, amplitudo, kecepatan. Amplitudo adalah tinggi gelombang, sedangkan panjang gelombang adalah jarak antara dua puncak. Frekuensi adalah jumlah gelombang yang melalui suatu titik dalam satu satuan waktu. Frekuensi tergantung dari kecepatan merambatnya gelombang. Karena kecepatan energi elektromagnetik adalah konstan (kecepatan cahaya), panjang gelombang dan frekuensi berbanding terbalik. Semakin panjang suatu gelombang, semakin rendah frekuensinya, dan semakin pendek suatu gelombang semakin tinggi frekuensinya.



Gambar 2.1. Spektrum Gelombang Elektromagnetik

(Sumber : <https://www.sat.dundee.ac.uk/gif/spectrum.png>)

## 2.2. Klasifikasi

Menurut Prahasta (2008) klasifikasi citra digital merupakan suatu proses penyusunan, pengurutan, atau pengelompokan semua piksel yang terdapat dalam

*bands* citra yang bersangkutan kedalam beberapa kelas atau kelompok berdasarkan suatu kriteria atau kategori objek hingga menghasilkan peta tematik dalam bentuk raster.

Setiap piksel yang terdapat dalam setiap kelas (hasil klasifikasi) diasumsikan memiliki karakteristik yang homogen. Tujuan dari klasifikasi yaitu untuk mengekstrak pola-pola respon spektral (terutama yang dominan) yang terdapat di dalam citra itu sendiri pada umumnya merupakan kelas-kelas penutupan lahan (*landcover*).

Klasifikasi citra satelit pada umumnya terdiri dari dua bentuk klasifikasi, yaitu klasifikasi berdasarkan piksel dan klasifikasi berdasarkan objek.

### 2.3. Klasifikasi Berbasis Piksel

Menurut Amalisana dkk (2017) klasifikasi berbasis piksel telah banyak digunakan dalam banyak kajian untuk mendapatkan informasi tentang tutupan lahan. Metode berbasis piksel termasuk sebuah algoritma yang dirancang untuk mengumpulkan informasi tentang objek menurut nilai piksel dari gambar (Kete dkk, 2019). Metode klasifikasi berbasis piksel digunakan untuk membimbing klasifikasi dengan algoritma *Support Vector Machine* (SVM).

SVM adalah sebuah klasifikasi non parametrik yang mampu menyeleksi vektor dari garis sebagai sebagai dasar untuk memisahkan dua kelas atau lebih dengan mengoptimalkan batas kelas antara tutupan lahan menurut prinsip klasifikasi linier. Algoritma SVM digambarkan sebagai berikut.

$$D(X^T) = \sum_i^l y_i a_i X_i X^T \tag{2.2}$$

Dimana :

- $y_i$  = Label kelas dari SVM
- $X_i X^T$  = Data testing
- $A_i$  dan  $b_0$  = Parameter Numerik yang ditentukan secara otomatis melalui algoritma SVM dengan mengoptimalkan
- 1 = Jumlah dari *support vector*

Klasifikasi berbasis piksel terdiri dari klasifikasi terbimbing dan klasifikasi tidak terbimbing.

### 2.3.1 Klasifikasi Terbimbing

Menurut Danoedoro (1996) klasifikasi terbimbing meliputi sekumpulan algoritma yang didasari pemasukan contoh objek (berupa nilai spektral) oleh operator. Contoh ini disebut sampel, dan lokasi geografis kelompok piksel sampel ini disebut sebagai daerah contoh (*training area*).

Sebelum sampel diambil, operator (atau pengguna) harus mempersiapkan sistem klasifikasi yang akan diterapkan. Seperti halnya klasifikasi manual. Dua hal penting yang harus dipertimbangkan dalam klasifikasi ialah sistem klasifikasi dan kriteria sampel. Pengambilan sampel secara digital oleh operator pada dasarnya merupakan cara ‘melatih’ komputer untuk mengenali objek berdasarkan kecenderungan spektralnya.

Banyak sekali algoritma yang digunakan untuk melakukan klasifikasi terbimbing diantaranya ialah :

- a. Jarak Minimum Terhadap Rata-rata (*Minimum Distance to Mean Algorithm*).
- b. Algoritma paralelepiped (*Box Classification Algorithm*).
- c. Algoritma Kemiripan Maksimum (*Maximum Likelihood Algorithm*).
- d. Algoritma Tetangga Terdekat (*K-Nearest Neighbour Algorithm*).

Dalam penelitian ini akan digunakan salah satu algoritma sebagaimana yang telah disebutkan diatas. Adapun algoritma tersebut yaitu algoritma kemiripan maksimum (*Maximum Likelihood Algorithm*).

Dalam penelitian ini akan digunakan salah satu dari algoritma tersebut. Adapun algoritma tersebut yaitu *maximum likelihood classification*.

#### 2.3.1.1 *Maximum Likelihood Classification*

Algoritma *maximum likelihood classification* merupakan algoritma yang secara statistik paling mapan. Asumsi dari algoritma ini ialah, objek homogen selalu menampilkan histogram yang terdistribusi normal (*Bayesian*). Pada

algoritma ini piksel diklaskan sebagai objek tertentu tidak karena jarak *Euclidean* nya, melainkan oleh bentuk, ukuran dan orientasi sampel pada *feature space* yang berupa elipsoida (Danoedoro, 1996).

Pada algoritma klasifikasi kemiripan maksimum ini, diasumsikan bahwa probabilitas untuk semua kelas dipandang sama. Pada kenyataannya, tidak semua kelas dapat diperlakukan dengan probabilitas sama untuk dipresentasikan pada citra. Suatu gugus sampel yang jauh lebih kecil dari gugus-gugus sampel yang lain, tentunya mempunyai probabilitas yang lebih kecil untuk muncul, sehingga perlu adanya faktor pembobot untuk masing-masing kelas yang ada.

Gugus sampel yang kecil ini secara logis dapat diberi bobot yang lebih rendah dibandingkan gugus-gugus yang lain. Pengambilan keputusan yang memperhatikan kemiripan maksimum dilakukan berdasarkan persamaan berikut:

$$D = \ln(a_c) - [0,5 \ln(|\text{Cov}_c|)] - [0,5(X - M_c)^T (\text{Cov}_c^{-1})(X - M_c)] \quad (2.3)$$

Dimana :

- $D$  = jarak yang diberi bobot
- $c$  = suatu kelas tertentu
- $X$  = vektor piksel yang diklasifikasi
- $M_c$  = vektor rata-rata sampel kelas  $c$
- $a_c$  = persentase probabilitas sembarang piksel untuk dapat menjadi anggota kelas  $c$ , dimana nilai *default* (yang ditawarkan komputer) adalah 1,0, berasal dari pengetahuan/informasi apriori.
- $\text{Cov}_c$  = matriks kovariansi piksel-piksel pada sampel kelas  $c$
- $|\text{Cov}_c|$  = determinan  $\text{Cov}_c$  (aljabar matriks)
- $\text{Cov}_c^{-1}$  = inversi  $\text{Cov}_c$  (aljabar matriks)
- $\text{Ln}$  = fungsi logaritma natural
- $T$  = fungsi transposisi (aljabar matriks)

Melalui persamaan ini, suatu piksel akan dimasukkan sebagai kelas  $c$ , apabila nilai  $D$  untuk kelas  $c$  adalah yang terendah. Secara teoritis, algoritma kemiripan maksimum ini akan berfungsi dengan baik, apabila bentuk histogram

saluran-saluran yang dilibatkan dalam proses klasifikasi menyajikan bentuk distribusi normal, karena algoritma ini mempertimbangkan paling banyak variabel statistik dibanding yang lain. Dalam praktek, selain faktor bentuk histogramnya, bentuk gugus-gugus sampel yang cenderung elipsoid juga akan lebih akurat apabila diklasifikasi dengan algoritma ini.

Perlu diperhatikan bahwa algoritma ini masih mengandalkan diri pada asumsi bahwa berbagai objek dapat dibedakan semata-mata berdasarkan kecenderungan spektralnya. Kenyataan menunjukkan bahwa objek terlalu kompleks untuk dapat diklasifikasikan dengan asumsi ini.

### **2.3.2 Klasifikasi Tidak Terbimbing**

Menurut Danoedoro (1996) klasifikasi tak terbimbing secara otomatis diputuskan oleh komputer, tanpa campur tangan operator (kalaupun ada, proses interaksi ini sangat terbatas). Proses ini merupakan suatu proses iterasi sampai menghasilkan pengelompokan akhir gugus-gugus spektral. Campur tangan operator terutama setelah gugus-gugus spektral terbentuk, yaitu dengan menandai setiap gugus dengan objek tertentu.

Terdapat tidak algoritma klasifikasi tak terbimbing yaitu :

- a. Jarak minimum ke pusat gugus (*minimum distance to cluster center*).
- b. Penggugusan statistik (*Statistical clustering*).
- c. Algoritma campuran (*Hybrid algorithm*)

### **2.4. Klasifikasi Berbasis Objek**

Menurut Kete dkk (2019) klasifikasi berbasis objek mampu mengidentifikasi tiap tampilan objek dalam gambar menurut kemiripan spektral, yang kemudian menyatukan menjadi beberapa kelas dari tutupan lahan. Teknik klasifikasi berbasis objek terdiri dari segmentasi dan klasifikasi. kedua proses tersebut identifikasi lahan diperoleh dari ekstraksi bentuk, tekstur dan spektral.

Algoritma dalam proses segmentasi adalah Multi Resolution Segmentation (MRS), dengan skala 0,5 bentuk 0,01 dan kepadatan 0,03. Nilai ini ditentukan dengan menggunakan percobaan *try* dan *error* dalam lokasi studi. Oleh karena itu kita secara langsung dapat fokus pada percobaan tersebut. Selain itu konsep

algoritma dalam klasifikasi adalah *Support Vector Machine* (SVM), yang dulu membimbing teknik klasifikasi berdasarkan klasifikasi linier memungkinkan mencari *hyperplane* yang tepat dari keduanya.

## 2.5. WorldView-2 level 2A

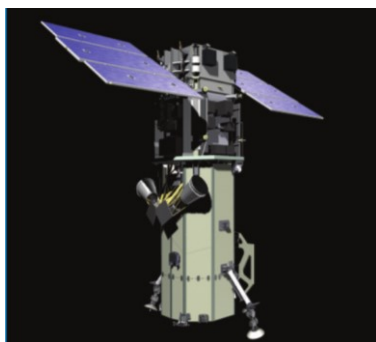
Satelit WorldView-2 level 2A diluncurkan pada tanggal 8 Oktober tahun 2009, yang merupakan satelit resolusi tinggi pertama yang mempunyai delapan band citra multispektral. Satelit WorldView-2 level 2A mengorbit bumi pada ketinggian 770 kilometer. Selain menyediakan citra multispektral dengan resolusi spasial 1,85 meter, satelit WorldView-2 level 2A juga menyediakan citra pankromatik dengan resolusi spasial 0,5 m. Satelit WorldView-2 level 2A sanggup merekam citra permukaan bumi hingga 975.000 kilometer persegi per harinya (DigitalGlobe, 2009).

Satelit WorldView-2 level 2A dapat menghasilkan citra pankromatik dengan lebar spektrumnya yaitu 4500-800 nm, dan resolusi spasial hingga 0,46 cm pada kondisi tegak, dan 0,52 cm pada kondisi miring. Selain citra pankromatik, WorldView-2 level 2A juga menghasilkan citra multispektral dengan resolusi spasial 1,85 m pada kondisi tegak dan 2,07 m pada kondisi miring. Citra multispektral ini terdiri dari 8 band, yaitu:

Tabel 2.1. Band dan Lebar Spektrum Citra Multispektral

No	Band	Lebar Spektrum
1	<i>Coastal</i>	400 - 450 nm
2	<i>Blue</i>	450 - 510 nm
3	<i>Green</i>	510 - 580 nm
4	<i>Yellow</i>	585 - 625 nm
5	<i>Red</i>	630 -690 nm
6	<i>Red Edge</i>	705 - 745 nm
7	<i>Near-IR1</i>	770 - 895 nm
8	<i>Near-IR2</i>	860 - 1040 nm

Adapun lebar sapuan tiap *scene* dari citra WorldView-2 level 2A adalah 16,4 km persegi.



Gambar 2.2. Satelit WorldView-2

(Sumber : [https://space.skyrocket.de/doc\\_sdat/worldview-2.htm](https://space.skyrocket.de/doc_sdat/worldview-2.htm))

Kelebihan dari satelit WorldView-2 yaitu menyediakan citra dengan resolusi tinggi untuk membuat peta detil, dapat mendeteksi perubahan lahan dan dapat menganalisis citra lebih mendalam. Khusus untuk penggunaan diluar pemerintahan Amerika Serikat, maka citra yang dihasilkan oleh satelit WorldView-2 yang seharusnya 46 cm akan diproses kembali menjadi 50 cm.

### 2.5.1. Citra Pankromatik

Menurut Utoyo (2007) citra pankromatik peka terhadap panjang gelombang 0,36 mikrometer hingga 0,72 mikrometer. Kepekaannya hampir sama dengan kepekaan mata manusia. Menurut Collwell dan Lo dalam Utoyo (2007) terdapat empat keunggulan foto pankromatik hitam putih, yaitu sebagai berikut:

- a. Kesan rona objek serupa dengan kesan mata yang memandang objek aslinya, karena kepekaan film sama dengan kepekaan mata manusia.
- b. Resolusi spasialnya halus, resolusi yang halus memungkinkan pengenalan objek yang berukuran kecil.
- c. Stabilitas dimensional tinggi, sehingga banyak digunakan dalam bidang fotogrametri.

Menurut Lillesand dan Kiefer (1999) penggunaan foto pankromatik sangat luas antara lain, pemetaan geologi, pemetaan tanah, pemetaan penutup dan

penggunaan lahan, bidang pertanian, kehutanan, sumber daya air, perencanaan kota dan wilayah, ekologi hewan liar, evaluasi dampak lingkungan, dan sistem informasi lahan.



Gambar 2.3. Citra Pankromatik Perkotaan Terangun  
(Sumber : Dinas PUPR Kabupaten Gayo Lues)

#### 2.5.2. Citra Multispektral

Menurut Utoyo (2007) citra multispektral adalah citra yang dibuat dengan sensor jamak, tetapi salurannya sempit. Citra multispektral masih dibedakan ke dalam dua jenis, yaitu sebagai berikut:

a. Citra RBV (*Return Beam Vidicon*), yaitu citra yang menggunakan sensor kamera dan hasilnya tidak dalam bentuk citra karena detektornya bukan film dan prosesnya non citra grafik.

b. Citra MSS (*Multi Spektral Scanner*), yaitu citra yang menggunakan sensornya dapat berupa spektrum tampak, maupun spektrum inframerah termal. Citra ini dapat dibuat dari pesawat udara.

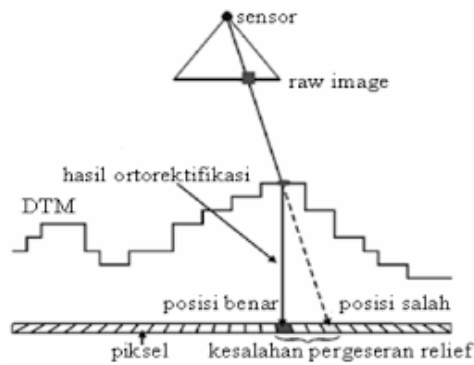


Gambar 2.4. Citra Multispektral Perkotaan Terangun  
(Sumber : Dinas PUPR Kabupaten Gayo Lues)



## 2.6. Orthorektifikasi

Berdasarkan *Standar Operating Procedures* (SOP) Pengolahan Citra Inderaja Digital, orthorektifikasi merupakan kegiatan mengoreksi geometri citra pada arah X, Y, dan Z.



Gambar 2.5. Orthorektifikasi

(Sumber:

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQIvQkaMp7IjU1ftI-ya1y5lMr3DxsPa5nHFUVoW07Fewce2rNh&s>)

Berdasarkan Peraturan Kepala Badan Informasi Geospasial No 15 Tahun 2014 Tentang Pedoman Teknis Ketelitian Peta Dasar maka kelas ketelitian peta RBI terbagi kepada tiga kelas. Untuk lebih jelasnya dapat dilihat pada Gambar 2.6. Ketelitian Geometri Perka BIG No 15 Tahun 2014.

No.	Skala	Interval kontur (m)	Ketelitian Peta RBI					
			Kelas 1		Kelas 2		Kelas 3	
			Horizontal (CE90 dalam m)	Vertikal (LE90 dalam m)	Horizontal (CE90 dalam m)	Vertikal (LE90 dalam m)	Horizontal (CE90 dalam m)	Vertikal (LE90 dalam m)
1.	1:1.000.000	400	200	200	300	300,00	500	500,00
2.	1:500.000	200	100	100	150	150,00	250	250,00
3.	1:250.000	100	50	50	75	75,00	125	125,00
4.	1:100.000	40	20	20	30	30,00	50	50,00
5.	1:50.000	20	10	10	15	15,00	25	25,00
6.	1:25.000	10	5	5	7,5	7,50	12,5	12,50
7.	1:10.000	4	2	2	3	3,00	5	5,00
8.	1:5.000	2	1	1	1,5	1,50	2,5	2,50
9.	1:2.500	1	0,5	0,5	0,75	0,75	1,25	1,25
10.	1:1.000	0,4	0,2	0,2	0,3	0,30	0,5	0,50

Gambar 2.6. Ketelitian Geometri Perka BIG No 15 Tahun 2014

Pada tahun 2018 kembali dikeluarkan perka BIG yang terbaru dengan maksud untuk merubah kebijakan yang ada pada perka BIG No 15 Tahun 2014. Terjadi perubahan perka BIG No 15 Tahun 2014 dalam peraturan ini lebih melonggarkan ketelitian terhadap peta RBI. Adapun ketelitian yang dimaksud dapat dilihat pada Gambar 2.7. Ketelitian Geometri Perka BIG No 18 Tahun 2018

No	Skala	Interval Kontur (m)	Ketelitian Peta RBI					
			Kelas 1		Kelas 2		Kelas 3	
			Horisontal (CE90 dalam m)	Vertikal (LE90 dalam m)	Horisontal (CE90 dalam m)	Vertikal (LE90 dalam m)	Horisontal (CE90 dalam m)	Vertikal (LE90 dalam m)
1	1:1.000.000	400	300	200	600	300	900,0	400
2	1:500.000	200	150	100	300	150	450,0	200
3	1:250.000	100	75	50	150	75	225,0	100
4	1:100.000	40	30	20	60	30	90,0	40
5	1:50.000	20	15	10	30	15	45,0	20
6	1:25.000	10	7,5	5	15	7,5	22,5	10
7	1:10.000	4	3	2	6	3	9,0	4
8	1:5.000	2	1,5	1	3	1,5	4,5	2
9	1:2.500	1	0,75	0,5	1,5	0,75	2,3	1
10	1:1.000	0,4	0,3	0,2	0,6	0,3	0,9	0,4

Gambar 2.7. Ketelitian Geometri Perka BIG No 18 Tahun 2018

### 2.6.1. *Root Mean Square Error (RMSE)*

RMSE adalah jarak yang terbentuk dari antara koordinat yang dimasukkan dalam proses rektifikasi dengan koordinat setelah proses rektifikasi pada lokasi yang sama (ERDAS, 1999). RMSE dihitung pada saat transformasi koordinat selesai dilakukan. Secara umum persamaan untuk menghitung besarnya RMSE yang terjadi dalam bidang dua dimensi sebagai berikut:

$$RMSE = \sqrt{\frac{(x' - x_{orig})^2 + (y' - y_{orig})^2}{n}} \quad (2.4)$$

Dimana:

$x', y'$  = Koordinat citra hasil koreksi geometrik

$(x, y)_{orig}$  = Koordinat titik kontrol tanah

n = Jumlah GCP

## 2.7. Interpretasi Citra

Interpretasi citra merupakan perbuatan mengkaji foto udara dan atau citra dengan maksud mengidentifikasi obyek dan menilai arti pentingnya obyek tersebut (Estes dan Simonet (1975) dalam Sutanto (1986)). Adapun unsur-unsur interpretasi pada citra atau foto udara terdiri atas sembilan kunci interpretasi, yaitu sebagai berikut:

### a. Rona dan Warna Rona (*Tone*)

Yaitu tingkat kegelapan atau kecerahan suatu objek pada citra. Adapun Warna (*Colour*), yaitu wujud yang tampak pada mata dengan menggunakan spektrum tampak yang lebih sempit. Misalnya, warna biru, hijau, merah, dan warna yang lainnya.

### b. Tekstur (*Texture*)

Frekuensi perubahan rona pada citra yang dinyatakan dengan kasar, sedang, dan halus. Misalnya, hutan bertekstur kasar, semak belukar bertekstur sedang, sedangkan sawah bertekstur halus

### c. Bentuk (*Shape*)

Konfigurasi atau kerangka gambar dari suatu objek yang mudah dikenali. Misalnya, persegi empat teratur dapat diidentifikasi sebagai kompleks perkantoran, sedangkan bentuk persegi tidak teratur dapat diidentifikasi sebagai kompleks permukiman penduduk. Bentuk lainnya antara lain gedung sekolah pada umumnya berbentuk huruf I, L, dan U atau persegi panjang.

### d. Ukuran (*Size*)

Ciri objek berupa jarak, luas, lereng, dan volume. Ukuran objek pada citra dikalikan dengan skala menghasilkan jarak yang sebenarnya.

### e. Pola (*Pattern*)

Susunan keruangan yang dapat menandai bahwa suatu objek merupakan bentukan oleh manusia atau bentukan alamiah. Misalnya, pola garis teratur merupakan pola jalan, sedangkan pola garis yang berkelok-kelok merupakan sungai. Permukiman transmigrasi dikenali dengan pola yang teratur, yaitu ukuran rumah dan jaraknya seragam, serta selalu menghadap ke jalan. Kebun karet, kebun kelapa, dan kebun kopi mudah dibedakan dengan hutan atau vegetasi lainnya dengan polanya yang teratur, yaitu dari pola serta jarak tanamnya.

f. Situs (*Size*)

Letak suatu objek terhadap objek lain di sekitarnya. Misalnya, permukiman pada umumnya memanjang pada pinggir pantai, tanggul alam, atau sepanjang tepi jalan. Adapun persawahan banyak terdapat di daerah dataran rendah dan berdekatan dengan aliran sungai. Jadi, situs sawah berdekatan dengan situs sungai.

g. Bayangan (*Shadow*)

Sifat yang menyembunyikan detail atau objek yang berada di daerah gelap. Bayangan juga dapat merupakan kunci pengenalan yang penting dari beberapa objek yang justru dengan adanya bayangan menjadi lebih jelas. Misalnya, lereng terjal tampak lebih jelas dengan adanya bayangan, begitu juga cerobong asap dan menara, tampak lebih jelas dengan adanya bayangan. Foto-foto yang sangat condong biasanya memperlihatkan bayangan objek yang tergambar dengan jelas.

h. Asosiasi (*Association*)

Keterkaitan antara objek yang satu dengan objek yang lainnya. Misalnya, stasiun kereta api berasosiasi dengan jalan kereta api. Adapun permukiman penduduk berasosiasi dengan jalan.

i. Konvergensi Bukti

Bukti-bukti yang mengarah kepada kebenaran, artinya semakin banyak unsur interpretasi yang digunakan dalam menginterpretasi suatu citra maka semakin besar kemungkinan kebenaran interpretasi yang dilakukan.

## **2.8. Ketelitian Hasil Klasifikasi**

Menurut Danoedoro (1996) hasil klasifikasi multispektral yang berupa peta tutupan lahan mempunyai tingkat ketelitian (akurasi) tertentu yang dapat diukur secara kuantitatif. Sebenarnya evaluasi data ketelitian hasil klasifikasi ini dapat diterapkan pada dua aspek, yaitu aspek kedalaman isi dan aspek kebenaran dilapangan.

Aspek kedalaman isi berkaitan dengan tingkat kerincian informasi secara tematik, misalnya peta penggunaan lahanyang menyajikan informasi berupa kelas-kelas sawah, tegalan, permukiman akan mempunyai tingkat kerincian yang

lebih rendah dari pada informasi berupa kelas-kelas sawah dengan padi dua kali, sawah dengan padi diselingi palawija, tegalan, permukiman desa dan permukiman perkotaan.

Sedangkan aspek kebenaran dilapangan biasanya lebih ditekankan pada aspek tepat tidaknya penutup yang diberi label atau kelas pada suatu posisi di koordinat tertentu. Misalnya, piksel  $A(x,y)$  menurut hasil klasifikasi multispektral adalah hutan, dan hasil pengujian di lapangan pada posisi  $(x,y)$  merupakan kebun campuran kerapatan tinggi. Hal ini dapat diartikan bahwa hasil klasifikasi piksel  $A(x,y)$  tersebut tidaklah akurat.

## 2.9. *Confusion Matrix*

Menurut Faisal dan Nugrahadi (2019) *confusion matrix* merupakan tabel untuk mengukur kinerja algoritma klasifikasi, model klasifikasi atau *classifier*. Jika digunakan data hasil prediksi dan data test/aktual data maka *confusion matrix* dapat digambarkan sebagai berikut.

Tabel 2.2. Matrik *Confusion*

N = 4	Prediksi kelas 1	Prediksi kelas 2
Aktual kelas 1	2	1
Aktual kelas 2	0	1

Matrik diatas memberikan informasi sebagai berikut :

- a. Nilai baris 1 dan kolom 1 : jumlah kelas 1 yang diprediksi sebagai kelas 1 adalah 2
- b. Nilai baris 1 dan kolom 2 : jumlah kelas 1 yang diprediksi sebagai kelas 2 adalah 1, artinya terdapat satu kesalahan prediksi
- c. Nilai baris 2 kolom 1 : jumlah kelas 2 yang diprediksi sebagai kelas 1 adalah 0.
- d. Nilai baris 2 dan kolom 2 : jumlah kelas 2 yang diprediksi sebagai kelas 2 adalah 1.

Pada contoh diatas adalah contoh klasifikasi 2 kelas atau binari kelas. Artinya data hanya memiliki 2 kelas/kategori. Pada *confusion matrix* terdapat beberapa istilah yang umum digunakan pada kasus klasifikasi binari kelas, yaitu :

- a. *True positives* (TP), kasus dimana seseorang diprediksi sebagai penderita kanker dan kenyataannya orang itu adalah penderita kanker.
- b. *True negative* (TN), kasus dimana seseorang diprediksi tidak menderita kanker dan kenyataannya orang tersebut tidak menderita kanker.
- c. *False positive* (FP), kasus dimana seseorang diprediksi sebagai penderita kanker dan ternyata orang tersebut tidak menderita kanker.
- d. *False negative* (FN), kasus dimana seseorang diprediksi tidak menderita kanker, tetapi ternyata orang tersebut menderita kanker.

Dan berikut ini adalah contoh *confusion matrix* untuk kasus diatas.

Tabel 2.3. Contoh Matrik *Confusion*

N = 165	Prediksi TIDAK KANKER	Prediksi KANKER
Aktual TIDAK KANKER	TN=50	FP=10
Aktual KANKER	FN=5	TP=100

Dari nilai TP, TN, FP, FN maka dapat dihitung beberapa nilai lain yang dapat dijadikan nilai kinerja *classifier*. Nilai-nilai tersebut adalah :

Tabel 2.4. Sumber Nilai Akurasi Dari Matrik *Confusion*

No	Nama	Rumus	Perhitungan
1	<i>Accuracy</i> Prosentase <i>classifier</i> benar melakukan prediksi	$\frac{(TP + TN)}{n}$	$\frac{(100 + 50)}{165} = 0.91$
2	<i>Misclassification Rate/Error Rate</i> Prosentasi <i>classifier</i> melakukan kesalahan prediksi	$\frac{(FP + FN)}{n}$	$\frac{(10 + 5)}{165} = 0.09$
3	True Positive Rate/Sensitivity/Recall Persentase data positif yang diprediksi sebagai positif	$\frac{TP}{(TP + FN)}$	$\frac{100}{(100 + 5)} = 0.95$ Prosentase data aktual KANKER yang diprediksi sebagai KANKER.

4	<i>False Positive Rate</i> Prosentase data negatif diprediksi sebagai positif	$\frac{FP}{(TN + FP)}$	$\frac{10}{(10 + 50)} = 0.17$ Prosentase data aktual TIDAK KANKER yang diprediksi sebagai KANKER.
5	<i>Specificity</i> Prosentase data negatif diprediksi sebagai negatif	$\frac{TN}{(TN + FP)}$	$\frac{50}{(50 + 10)} = 0.83$ Prosentase data aktual TIDAK KANKER yang diprediksi sebagai TIDAK KANKER.
6	<i>Precision</i> Prosentase prediksi data sebagai positif yang benar	$\frac{TP}{(FP + TP)}$	$\frac{100}{(10 + 100)} = 0.91$
7	<i>Prevalence</i> Prosentase jumlah instance positif pada data	$\frac{Actual\_positive}{n}$	$\frac{105}{165} = 0.64$

## 2.10. Metode Uji Ketelitian *Overall Accuracy*

Menurut Danoedoro (1996) sebenarnya terdapat dua varian metode uji ketelitian secara statistik, merujuk pada (Mather, 1987) varian yang pertama adalah sebagai berikut.

a. Operator mengambil kembali piksel-piksel yang telah dipilih sebagai sampel dengan label 1,2,3,...,n, dimana n merupakan jumlah label atau kelas yang telah ditentukan. Masing-masing label ini berisi sejumlah piksel dengan variasi nilai tertentu, yang memberikan satu himpunan parameter statistik (rata-rata, simpangan baku, varian, kovarian, dan sebagainya).

b. Gugus-gugus piksel yang sama, tetapi sebagai hasil proses klasifikasi juga diambil. Piksel-piksel ini, sekalipun pada posisi koordinat yang sama persis dengan langkah (a), biasanya mengandung komposisi label yang berbeda, sebagai hasil penurunan informasi secara statistik

Kedua himpunan label ini kemudian disusun dalam bentuk matriks yang disebut sebagai matriks kesalahan (*error matrix*, atau *confusion matrix*). Himpunan matriks tersebut ada yang berfungsi sebagai label data acuan atau label

data lapangan dan ada juga sebagai label hasil klasifikasi. Ketelitian dari seluruh hasil klasifikasi dapat dinyatakan dalam persen.

Menurut Lunetta dan Lyon (2005) sebuah matrik kesalahan adalah sebuah himpunan persegi dari nilai-nilai yang disusun dalam baris dan kolom yang menyatakan jumlah dari sampel misalnya piksel, kelompok piksel atau poligon yang ditugaskan untuk kategori tertentu relatif terhadap kategori sebenarnya. Secara khusus kolom merepresentasikan data sedangkan baris mengindikasikan peta yang dihasilkan dari data penginderaan jauh. Data referensi dianggap benar yang dapat dikumpulkan dari berbagai sumber, termasuk interpretasi foto udara, hasil pengamatan dan pengukuran dilapangan.

Matrik kesalahan setelah dibuat dengan benar, maka dapat digunakan sebagai tahap awal untuk sebuah rangkaian deskriptif dan teknik analisis statistik. Statistik deskriptif paling umum dan paling sederhana adalah *overall accuracy*, yang menjumlahkan hasil dengan membagi total benar dari sampel, yaitu jumlah dari diagonal utama dengan jumlah unit sampel dalam matrik kesalahan. Selain itu, kategori akurasi secara individual dapat dihitung dalam sebuah masalah yang serupa. Matrik kesalahan dapat dilihat pada Tabel 2.5. Matrik kesalahan.

Tabel 2.5. Matrik Kesalahan

		Reference Data					
		D	C	AG	SB	Row Total	
Classified Data	D	63	4	22	24	113	D=Desiduous
	C	6	79	8	8	101	C=Conifer
	AG	0	11	85	11	107	AG=Agriculture
	SB	4	7	3	89	103	SB=Shrub
Column Total		73	101	118	132	424	

<u>PRODUCERS ACCURACY</u>	<u>USERS ACCURACY</u>	<u>OVERALL ACCURACY</u>
D= 63/73=86%	D= 63/113=86%	(63+79+85+89)/424
C= 79/101=78%	C= 79/101=78%	316/424=75%
AG= 85/118=72%	AG= 85/107=72%	
SB= 89/132=67%	SB= 89/103=67%	

(Sumber : Lunetta dan Lyon, 2005.)



### 2.11. Metode Uji Ketelitian *Kappa Accuracy*

Menurut Lowell dan Jatou (1999) perbandingan dari akurasi peta digunakan untuk menentukan yang mana dari beberapa klasifikasi cocok untuk proyek pemetaan, untuk menentukan dua hasil skema klasifikasi tutupan lahan dengan akurasi tinggi, atau untuk menilai jika akurasi berbeda diantara perbedaan batas spasial. Peta boleh jadi dibandingkan sebagai dasar dari beberapa akurasi yang diukur. Karena pengukuran ini mencerminkan perbedaan komponen dari akurasi.

Untuk menentukan dua peta terbaik harus mempertimbangkan informasi yang disediakan oleh berbagai pengukuran. Sebagai contoh perbandingan peta boleh jadi berdasarkan pada *user's accuracy* dan *producer's accuracy* untuk kelas tutupan lahan yang berbeda. Namun, kadang-kadang perbandingan peta menggunakan sekali pengukuran akurasi, salah satunya untuk kenyamanan, atau karena sangat banyak jumlah dari perbandingan yang dibutuhkan untuk dilakukan dalam mode semi otomatis.

Untuk kasus tersebut yang mana perbandingan adalah berdasarkan sekali pengukuran akurasi, bahwa mengukur harus dipilih untuk mencerminkan tujuan untuk membangun motivasi dan penggunaan peta. Dengan kata lain ukuran digunakan untuk membandingkan dua peta yang bagaimanapun harus ditampilkan nilai dari peta untuk penggunaan khusus.

Hasil dari *accuracy assesment* sering diringkas melalui sebuah matrik kesalahan. Dianggap bahwa tiap piksel atau poligon yang ditampilkan di atas peta adalah nama sebagai kelas tutupan lahan, dan sebuah referensi ('benar') tutupan lahan adalah juga diperoleh piksel atau poligon tersebut. Ringkasan matrik kesalahan kemudian dibandingkan antara peta dan referensi klasifikasi untuk mengumpulkan piksel atau poligon.

Jika representasi matrik kesalahan adalah  $q$  kelas tutupan lahan, maka pengukuran akurasi yang umum digunakan adalah *Kappa coefficient of agreement*.

$$K = \frac{P_c - \sum_{i=1}^q P_{i+} P_{+i}}{1 - \sum_{i=1}^q P_{i+} P_{+i}} \quad (2.5)$$

dimana

$$P_{i+} = \sum_{k=1}^q P_{ik} \quad (2.6)$$

Dan

$$P_{+i} = \sum_{k=1}^q P_{ki} \quad (2.7)$$

Tabel 2.6. Matrik *Confusion Kappa Accuracy*

	<u>Refecence</u>				
	1	2	...	q	
1	P <sub>11</sub>	P <sub>12</sub>	...	P <sub>1q</sub>	P <sub>1+</sub>
2	P <sub>21</sub>	P <sub>22</sub>	...	P <sub>2q</sub>	P <sub>2+</sub>
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
q	P <sub>q1</sub>				P <sub>q+</sub>
	P <sub>+1</sub>	P <sub>+2</sub>	...	P <sub>+q</sub>	

(Sumber : Lowell dan Jatou, 1999)

$P_{ij}$  adalah adalah bagian dari area dalam tutupan lahan kelas  $i$  yang dipetakan dan tutupan lahan kelas  $j$  sebagai referensinya.

$$p_{i+} = \sum_{j=1}^q p_{ij}, p_{+j} = \sum_{i=1}^q p_{ij} \quad (2.8)$$

Kappa dengan peluang acak, sebagaimana disarankan oleh (Foody, 1992).

$$K^* = \frac{P_c - 1/q}{1 - 1/q} \quad (2.9)$$

## 2.12. Skala Peta

Skala peta adalah perbandingan jarak antara dua tempat yang digambarkan pada peta dengan jarak sebenarnya di permukaan bumi. Skala peta dapat dinyatakan sebagai berikut :

$$\text{Skala Peta} = \frac{\text{Jarak di Peta}}{\text{Jarak Sebenarnya}} \quad (2.10)$$

Skala angka dan skala garis saling melengkapi . skala angka memudahkan untuk mengetahui skala peta yang akan digunakan sedangkan skala grafis membantu ketelitian peta karena ukuran “batang/bar” dapat dijadikan dalam penentuan jarak (Widyatmanti dan Natalia, 2006).

Skala peta 1:5.000 memiliki arti satu satuan dipeta bernilai 5.000 satuan di lapangan yang berarti satu *centimeter* dipeta sama dengan 5.000 *centimeter* dilapangan. Peta skala 1:5.000 dapat digunakan oleh instansi terkait seperti PUPR dan BAPPEDA untuk penyusunan dokumen Rencana Detail Tata Ruang (RDTR). Pada peta skala 1:5.000 maka semua objek yang tampak dari citra satelit dan ukurannya 6,25 m<sup>2</sup> harus digambarkan dalam bentuk area (*polygon*).

## 2.13. *Application Programming Interface* (API)

Menurut Ramadhani (2019) API merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan programmer untuk membongkar suatu perangkat lunak, kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain.

API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem fungsi. Proses ini dikelola melalui sistem operasi. Keunggulan dari API adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi.

### 2.13.1. Python

Menurut Supardi (2017) Python adalah salah satu bahasa pemrograman yang populer di dunia kerja Indonesia. Selain itu, di ranah akademik juga banyak

akademisi yang menggunakan python untuk menyelesaikan penelitiannya di bidang komputasi sains, robotika, data science, ekonomi, antariksa, dan berbagai macam bidang lainnya. Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC.

Python secara *default* telah terpasang di beberapa sistem operasi berbasis Linux, seperti Ubuntu, Linux Mint, dan Fedora. Untuk sistem operasi lain, sudah tersedia *installer* yang disediakan untuk sistem operasi tersebut. Banyak hal yang dapat dipelajari ketika menggunakan bahasa pemrograman python. Beberapa *package* yang populer di python antara lain:

- a. Django, sebagai web framework
- b. Scipy dan Scikit, pustaka untuk membuat aplikasi *machine learning* dan kecerdasan buatan (*Artificial Intelligence*).
- c. Tornado pustaka untuk membuat aplikasi web, websocket, dan asynchronous programming.
- d. Celery, pustaka untuk membuat asynchronous task
- e. OpenCV, pustaka untuk membuat aplikasi computer vision
- f. Matplotlib, pustaka untuk membuat grafik untuk keperluan saintifik.
- g. BioPython, pustaka untuk menganalisa DNA dan Genome makhluk hidup.
- h. TensorFlow, pustaka untuk membuat aplikasi yang ditenagai oleh *deep learning*.

Selain itu, python pun memiliki sebuah *package manager* yang populer dan unggul yang dinamakan dengan PIP. Dengan menggunakan PIP, dapat memasang dan menghapus pustaka python. Di Indonesia sendiri banyak sekali website yang sudah berbasis python. Beberapa diantaranya yaitu KelasKita, CodeSaya, Kargo.co.id dan banyak lainnya.

### 2.13.2. *Keras*

*Keras* ditulis oleh Francois Chollet yang memiliki sebuah modular, minimalis, dan arsitekturnya mudah dikembangkan. *Keras* merupakan salah satu *Application Programming Interface* (API) jaringan saraf tiruan tingkat tinggi. *Keras* ditulis dalam bahasa pemrograman Python dan mampu berjalan bersama

API yang lain seperti TensorFlow, CNTK, atau Theano. Keras mudah digunakan dan sering digunakan bersama TensorFlow dibelakang layar (Gulli dan Pal, 2017).

Dengan menggunakan Keras akan mempermudah dan mempercepat pengguna dalam membuat *prototype*. Keras mendukung jaringan konvolusional dan jaringan berulang dan juga mendukung kombinasi dari kedua jaringan tersebut. Keras berasal dari bahasa Yunani yang berarti tanduk. Keras pada awalnya dikembangkan sebagai bagian dari upaya penelitian proyek ONEIROS (*Open-ended Neuro-Electronic Intelligent Robot Operating System*).

### 2.13.3. TensorFlow

Menurut Rungta (2019), akhir-akhir ini pustaka *deep learning* paling terkenal di dunia adalah Google's TensorFlow. Produk Google menggunakan *machine learning* dalam semua produknya untuk meningkatkan kemampuan mesin pencari, terjemahan, keterangan gambar atau rekomendasi.

Untuk memberikan contoh yang pasti, pengguna google dapat merasakan proses pencarian dengan AI lebih cepat dan lebih halus. Jika pengguna mengetik sebuah kata kunci pada lokasi pencarian, google menyediakan sebuah rekomendasi tentang apa yang mungkin menjadi kata selanjutnya.

Google ingin menggunakan *machine learning* untuk mengambil keuntungan dari kumpulan data secara besar-besaran untuk memberikan pengguna pengalaman terbaik. Terdapat tiga kelompok dalam penggunaan *machine learning* yaitu peneliti, ilmuwan data, dan para pembuat program. Mereka dapat menggunakan peralatan yang sama untuk saling bekerja sama satu sama lain dan meningkatkan efisiensi mereka dalam bekerja.

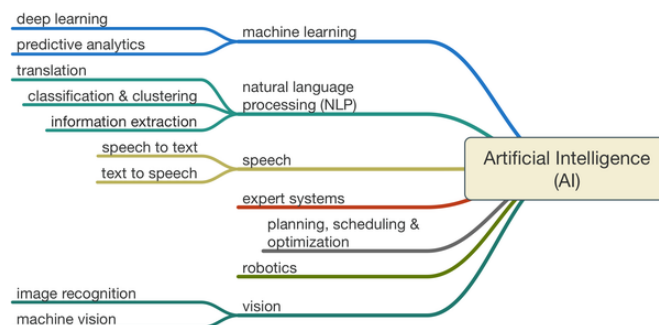
Google bukan hanya punya data, tetapi mereka juga komputer yang paling besar di dunia, jadi TensorFlow diciptakan untuk skala. TensorFlow adalah pustaka yang dibangun oleh Google Brain Team untuk mempercepat *machine learning* dan riset jaringan *deep learning*. Tensorflow dibangun untuk menjalankan berbagai macam *Central Processing Unit* (CPU) dan *Graphics Processing Unit* (GPU).

## 2.14. *Artificial Intelligency (AI)*

Kecerdasan buatan adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia (Minsky, 1989 dalam Kursini (2006)). Sedangkan menurut Simon (1987) kecerdasan buatan merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemograman komputer untuk melakukan suatu hal yang dalam pandangan manusia adalah cerdas.

Menurut Winston dan Prendergast (1994) ada tiga tujuan kecerdasan buatan, yaitu: membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat lebih berguna. Adapun yang dimaksud dengan kecerdasan adalah kemampuan untuk belajar atau mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru dengan menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif.

AI terdiri dari banyak konsep, yaitu : *Machine Learning, Natural Language Processing, Speech, Expert Systems, Planning, Scheduling & Optimization, Robotic, dan Vision.*



Gambar 2.10. Cabang ilmu dari AI

(Sumber:

<https://qph.fs.quoracdn.net/main-qimg-1d8e331be2cf10df66d4bfaaa6cfc025>)

## 2.15. *Machine Learning*

Menurut Withanawasam (2015) *machine learning* adalah cabang ilmu dari *artificial intelligence* yang fokus pada proses pembelajaran (*learning*) dari

sejumlah data yang ada untuk membuat prediksi pada data yang belum diketahui tanpa memerlukan programming yang eksplisit. *Machine learning* secara khusus dibagi menjadi dua kategori yaitu *supervised learning* dan *unsupervised learning*.

Menurut Gulli (2015) *supervised learning* pasangan dari contoh yang dibuat oleh (data *input*, dan output yang diharapkan) tersedia dan komputer mempelajari sebuah model sesuai *input* yang diberikan. sebuah *ouput* yang diharapkan mempunyai kesalahan yang minimal merupakan hasil prediksi. *Classification*, *neural network*, dan *regression* merupakan semua contoh dari *supervised learning*. Untuk semua teknik dari *supervised learning*, diasumsikan bahwa ada peramal atau guru yang dapat mengajarkan kepada komputer apa yang harus dilakukan agar komputer dapat menerapkan pelajaran yang dipelajari tersebut pada data baru yang belum diketahui.

Sedangkan dalam *unsupervised learning* komputer tidak ada guru yang mengajarkan dan komputer dibiarkan sendiri dalam mencari struktur, pola dan kelainan dalam data. *Clustering* dan *Density Estimations* merupakan contoh khusus dari *unsupervised learning*.

## **2.16. Deep Learning**

Menurut Chollet (2017) *Deep learning* adalah sub-bagian dari *Machine Learning*. Pandangan baru tentang representasi *learning* dari data yang menekankan pada *learning layer* yang semakin meningkat. Banyaknya *layer* yang berkontribusi dalam model disebut dengan *deep model*. *Deep learning* moderen sering melibatkan puluhan hingga ratusan *layer* dan semua *layer* belajar secara otomatis dari paparan data pelatihan.

Menurut LeCun, Bengio, dan Hinton (2015) *deep learning* memungkinkan model komputasi yang terdiri dari berbagai *layer* pemrosesan untuk belajar representasi data dengan berbagai tingkatan abstraksi. Model ini secara dramatis telah meningkatkan seni dalam mengenali suara, mengenali objek, dan banyak tugas lainnya.

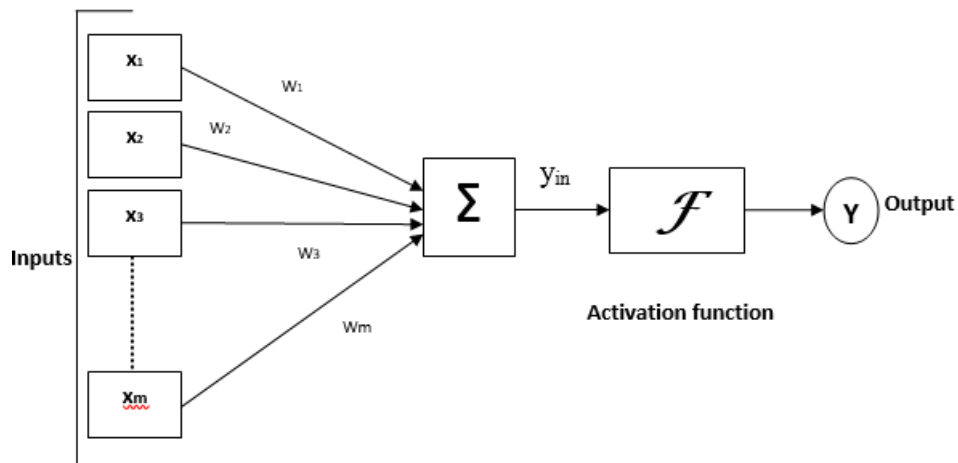
*Deep learning* menemukan struktur yang rumit dalam kumpulan data yang besar dengan menggunakan *backpropagation* untuk menunjukkan

bagaimana mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi di setiap lapisan dari representasi lapisan sebelumnya.

Akhir-akhir ini deep learning menjadi sorotan dalam pengembangan machine learning, alasannya karena deep learning telah mencapai hasil yang luar biasa dalam computer vision. Deep learning merupakan cabang dari machine learning yang terinspirasi dari korteks manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak hidden layer (Santoso dan Ariyanto, 2018).

### 2.16.1. *Artificial Neural Network (ANN)*

Menurut Kukreja dkk (2015) dalam istilah yang paling sederhana, ANN merupakan sebuah jaringan tiruan dari otak makhluk hidup. ANN terdiri unit proses yang disebut dengan neuron. ANN mencoba meniru struktur dan tingkah laku jaringan otak makhluk hidup. Sebuah neuron terdiri dari input (*dendrite*) dan satu output (*synapse via axon*). ANN mempunyai sebuah fungsi untuk menentukan aktivasi dari sebuah neuron.



Gambar 2.11. Model *Artificial Neural Network (ANN)*

(Sumber:

[https://www.tutorialspoint.com/artificial\\_neural\\_network/images/model.jpg](https://www.tutorialspoint.com/artificial_neural_network/images/model.jpg))

Nilai input disimbolkan dengan  $X_1..X_m$  yang akan diinputkan kedalam neuron. Sebuah bias juga akan ditambahkan kedalam neuron pada saat proses input. Biasanya nilai dari bias adalah 1. Bobot disimbolkan dengan  $W_{k1}...W_{kp}$ .



sebuah bobot akan terkoneksi dengan sinyal. Hasil nilai input dan bobot yang diberikan akan menghasilkan kekuatan sinyal tersebut. Sebuah neuron akan menerima nilai input dari berbagai sumber yang berbeda, dan mempunyai satu output.

### **2.16.2. *Deep Neural Network (DNN)***

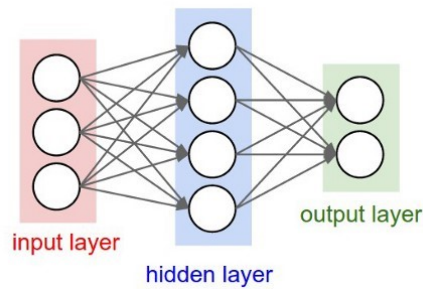
Menurut Wiley (2016) DNN adalah *neural network* dengan berbagai *hidden layer*. Meskipun ekstensi konseptual yang relatif sederhana dari *neural network* seperti *deep architecture* memberikan kemajuan yang berharga dalam hal kemampuan model dan tantangan baru dalam melatihnya.

Menggunakan berbagai *hidden layer* memungkinkan membangun lebih canggih dari elemen sederhana menjadi elemen yang lebih kompleks. Untuk data yang memiliki *deep architectures*, sebuah *deep neural network* dapat menyesuaikan dengan lebih akurat dengan sedikit parameter daripada *neural network*, karena lebih banyak *layer* dapat merepresentasikan lebih efisien dan lebih akurat.

Sebuah *neural network* dengan berbagai *hidden layer* dinamakan dengan *deep neural network* (DNN) dan praktek dari *training network* tersebut kadang-kadang disebut sebagai *deep learning* (Lantz, 2015).

### **2.16.3. *Arsitektur Deep Neural Network***

Sebuah arsitektur *neural network* terdiri dari *input layer*, *hidden layer*, dan *output layer*. *Input layer* terdiri dari neuron yang mempunyai nilai sesuai dengan data yang diinput. *Hidden layer* terdiri dari sejumlah *hidden layer* dan neuron dari tiap-tiap layer tergantung dari kompleksitasnya. Sedangkan *Output layer* berisi berbagai macam output biasanya memiliki satu *neuron* yang mempunyai kisaran nilai dari 0 hingga 1 (Kuldeep dan Anitha, 2015)

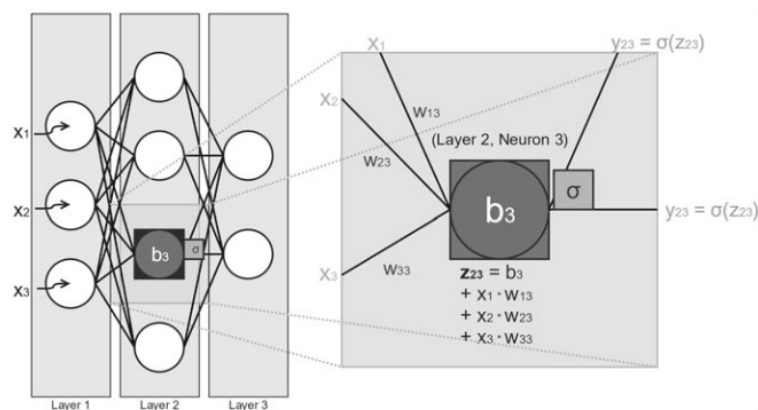


Gambar 2.12. Arsitektur *neural network* dengan satu *hidden layer*

(Sumber: [https://miro.medium.com/max/1000/1\\*HKfrSssprukPyb1SFtISA.jpeg](https://miro.medium.com/max/1000/1*HKfrSssprukPyb1SFtISA.jpeg))

### 2.17. *Training Neural Network*

Menurut Skansi (2018) dalam melakukan sebuah *training neural network* maka aliran informasi yang mengalir dalam sebuah jaringan *neural network* berasal dari neuron pada layer pertama (*input layer*) melalui *neuron* pada layer kedua (*hidden layer*) dan menuju kepada layer ketiga (*output layer*).



Gambar 2.13. Proses training pada *neural network* sederhana

(Sumber : Skansi, 2018)

Seperti pada gambar 2.13. *input layer* terdiri dari tiga neuron dan tiap-tiap neuron dapat menerima satu nilai input. Nilai tersebut direpresentasikan dalam bentuk variabel  $X_1$ ,  $X_2$ , dan  $X_3$ . Penerimaan input data hanya dilakukan oleh layer pertama. Setiap *neuron* dalam input layer dapat membawa satu *output layer*. Data yang diinputkan boleh lebih kecil dari jumlah *neuron*, dengan syarat

*neuron* yang tidak ada nilainya harus diberikan nilai nol. Nilai masukan tidak boleh lebih banyak dari jumlah *neuron*.

Input dapat direpresentasikan sebagai sebuah bilangan  $X_1, X_2, \dots, X_n$ . Setiap *neuron* pada *input layer* terhubung dengan *neuron* pada *output layer*, akan tetapi *neuron* pada layer yang sama tidak saling terhubung. Setiap hubungan antara *neuron*  $J$  dalam layer  $K$  dan *neuron*  $M$  dalam layer  $N$  mempunyai sebuah bobot ( $W$ ) yang dinyatakan dengan  $W_{JM}^{KM}$ , bobot mengatur berapa banyak nilai awal akan diteruskan kepada *neuron* pada layer selanjutnya. Jika *input* adalah 12 dan bobot pada *neuron* tujuan adalah 0.25, maka *neuron* yang dituju akan menerima nilai 3. Bobot dapat ditingkatkan nilainya dan juga dapat diturunkan karena bobot tidak dibatasi antara 0 dan 1.

Pada gambar 2.13. bagian kanan menjelaskan bahwa (*neuron* 3 pada layer 2), menerima nilai input dari hasil penjumlahan perkalian antara input data dan bobot dari layer sebelumnya. Pada kasus ini input datanya adalah  $X_1, X_2, X_3$ , dan bobotnya adalah  $W_{13}, W_{23}, W_{33}$ . Tiap *neuron* memiliki nilai modifikasi di dalamnya yang disebut dengan bias yang direpresentasikan dalam kasus ini adalah  $b_3$ , bias ini telah ditambahkan pada penjumlahan sebelumnya. Hasil dari perkalian ini disebut dengan *logit*. Secara biasanya dinyatakan dengan  $z$  dalam kasus ini ( $z_{23}$ ).

Pada model yang sederhana cukup diberikan *logit* sebagai outputnya. Tetapi kebanyakan model menerapkan sebuah fungsi nonlinear yang sering disebut dengan *nonlinearity* atau *activation function* terhadap *logit* untuk menghasilkan output. Output biasanya dinyatakan dengan  $y$  seperti pada gambar 2.13, yaitu  $y_{23}$ . *Nonlinearity* secara umum berkenaan dengan  $S(x)$  atau nama dari sebuah fungsi. Fungsi yang paling umum digunakan adalah *sigmoid function* dan *logistic function*.

Fungsi logistik mengambil *logit*  $z$  dan mengembalikannya sebagai output  $\sigma(z) = \frac{1}{1 + e^{-z}}$  (Skansi, 2018). Fungsi logistik menerima nilai antara 0 dan 1, dan menghitung peluang *output* yang diberikan oleh *input*. Pada gambar 2.13. untuk mendapatkan nilai dari  $z_{31}$  maka,  $z_{31}$  dihitung sebagai  $z_{31} = b_{31} + w_{11}^{23} y_{21} + w_{21}^{23} y_{22} + w_{31}^{23} y_{23} + w_{41}^{23} y_{24}$  langkah yang sama juga berlaku

untuk mendapatkan nilai  $z_{32}$  dan kemudian menerapkan fungsi *nonlinearity* yang dipilih sehingga diperoleh hasil akhir dari suatu proses training.

### 2.17.1. *Feedforward*

Fungsi prediksi untuk *neural network* juga dikatakan dengan *feedforward*, dan ini hanyalah perkalian *dot product* sederhana dan fungsi *nonlinear* pada tiap *layer* dari *neural network*. Fungsi *nonlinear* biasanya digunakan dalam *hidden layer*, fungsi yang biasa digunakan adalah sebuah ReLU, Sigmoid, atau Tanh. Menurut Gulli dan Kapoor (2017) aliran informasi dalam *multi layer perceptrons* selalu bersumber dari *input layer* ke *output layer*. tidak ada arah mundur, maka jaringan ini dikenal sebagai *feedforward*.

Sedangkan menurut Heaton (2015) istilah *feedforward* menggambarkan bagaimana *neural network* memproses dan mengingat kembali pola. Dalam sebuah *neural network*, tiap *layer* dari *neural network* terdiri dari koneksi terhadap *layer selanjutnya*. Sebagai contoh, koneksi ini memanjang dari *input layer* ke *hidden layer*, tapi tidak ada koneksi yang bergerak mundur.

Sebuah *feedforward neural network* adalah sebuah *artificial neural network* yang terdiri dari beberapa perceptron, yang terdiri dari *layer*, seperti *input layer*, *output layer*, dan satu atau lebih *hidden layer*. Tiap *layer perceptron* juga dikenal sebagai neuron, mempunyai koneksi langsung terhadap *perceptron* dalam layer berikutnya. Dimana, koneksi antara dua neuron memiliki sebuah bobot yang mirip dengan bobot *perceptron*.

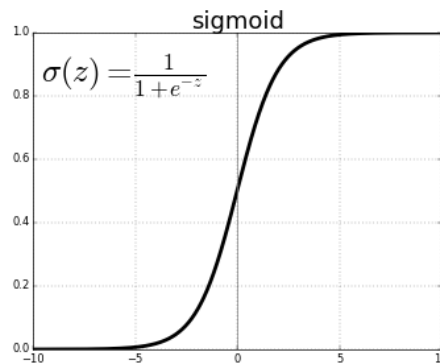
### 2.17.2. *Activation Function*

*Activation function* membuat *neural network* tidak linier. Sebuah *activation function* memutuskan apakah sebuah *perceptron* harus aktif atau tidak. Selama *training activation*, fungsi berperan penting dalam menyesuaikan *gradien*. Prilaku *nonlinear* dari *activation function* memberikan *neural network* untuk mempelajari fungsi yang kompleks. Sebagian besar dari *activation function* adalah kontinyu dan diferensial, kecuali *retified unit* pada 0. Fungsi kontinyu memiliki perubahan kecil dalam *output* untuk setiap perubahan kecil dalam *input*. Sebuah fungsi diferensial memiliki sebuah turunan di setiap titik pada domain.

Untuk melatih sebuah *neural network*, fungsinya harus dapat dibedakan. Berikut ini adalah beberapa fungsi aktivasi.

a. *Sigmoid*

Fungsi sigmoid dapat dianggap sebagai sebuah fungsi yang diperhalus dan karenanya dapat dibedakan. Sigmoid berguna untuk mengkonversi nilai apapun menjadi probabilitas dan dapat digunakan untuk klasifikasi bilangan biner. Sigmoid memetakan nilai input menjadi nilai yang berkisar antara 0 hingga 1. Sebagaimana yang ditunjukkan pada Gambar 2.14 berikut ini.



Gambar 2.14. Fungsi aktivasi *sigmoid*

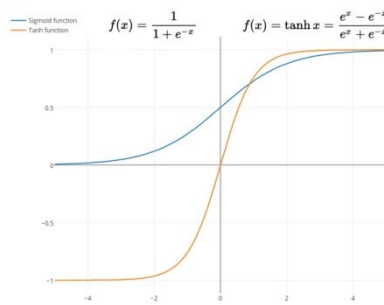
(Sumber :

<https://i0.wp.com/www.stokastik.in/wp-content/uploads/2017/07/sigmoid.png?resize=400%2C300>)

Perubahan nilai Y terhadap X akan menjadi kecil, dan karena itu akan terjadi *vanishing gradient*. Setelah proses *learning*, perubahannya mungkin mengecil.

b. *Hyperbolic Tangen (tanh)*

Fungsi *hyperbolic tangen (tanh)* merupakan fungsi *sigmoid* yang diskalakan. Seperti *sigmoid* fungsi ini halus dan dapat dibedakan. Tanh menginput nilai antara -1 hingga 1, seperti yang ditunjukkan pada Gambar 2.15. berikut ini.



Gambar 2.15. Fungsi aktivasi *hyperbolic tangen* (tanh)

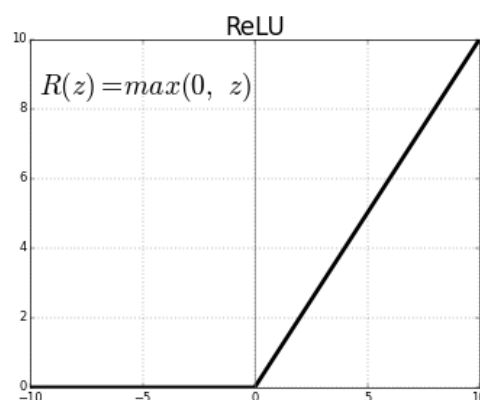
(sumber :

[https://miro.medium.com/max/840/1\\*RasZ\\_z7pBtOT-djlyYmS8Q.jpeg](https://miro.medium.com/max/840/1*RasZ_z7pBtOT-djlyYmS8Q.jpeg))

*Gradient* lebih stabil dibandingkan *sigmoid*, oleh karena itu memiliki sedikit masalah terjadi seperti *vanishing gradient*. *Sigmoid* dan *tanh* aktif sepanjang waktu membuat *artificial neural network* benar-benar berat.

c. *Rectified Linear Unit* (ReLU)

ReLU membiarkan hanya nilai yang tinggi yang melewatinya. Hal ini membuat beberapa *neuron* tidak berfungsi. ReLU memetakan *input* X ke  $\max(0, X)$ , dan memetakan input negatif ke 0. Nilai *input* yang bernilai positif akan menjadi nilai *output* tanpa melakukan perubahan seperti yang ditunjukkan pada gambar 2.16. berikut ini.



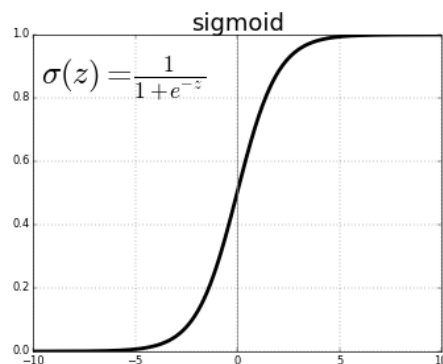
Gambar 2.16. Fungsi aktivasi ReLU

(Sumber:

[https://miro.medium.com/max/347/1\\*QindKA4Dt7O13CbICMSxWw.png](https://miro.medium.com/max/347/1*QindKA4Dt7O13CbICMSxWw.png))

Karena ReLU tidak aktif sepanjang waktu, maka dapat dilatih lebih cepat. Karena fungsinya sederhana, secara komputasi ini merupakan fungsi yang sangat mudah. Memilih fungsi aktivasi sangat tergantung pada aplikasi. Namun, ReLU bekerja dengan baik untuk sejumlah besar masalah.

Menurut Kukreja dkk (2015) ada banyak fungsi yang digunakan dalam proses aktivasi, salah satu fungsi yang paling umum digunakan adalah fungsi aktivasi sigmoid.



Gambar 2.17. Sigmoid function

(Sumber:

<https://i0.wp.com/www.stokastik.in/wp-content/uploads/2017/07/sigmoid.png?resize=400%2C300>)

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.11}$$

(Sumber : Kukreja dkk., 2015)

Dimana :

$$z = \sum_{i=1}^n x_i W_i \tag{2.12}$$

(Sumber : Kukreja dkk., 2015)

Keterangan:

$z$  = Jumlah nilai dari input dikalikan dengan bobot antara satu layer dengan layer selanjutnya.

$x_i$  = Nilai input

$W_i$  = Bobot

Adapun fungsi lainnya yang digunakan adalah *Step function*, *Linear function*, *Rump function*, dan *Hyperbolic tangen function*. *Hyperbolic tangen function* mirip dengan *Sigmoid function* tapi batasnya dari -1 ke 1, sedangkan *Sigmoid function* dari 0 ke 1.

### 2.17.3. Loss Function

Hasil dari proses propagasi biasanya tidak sesuai dengan target yang diinginkan. Oleh sebab itu perlu dilakukan perhitungan perbedaan nilai target dan hasil propagasi apakah sudah sesuai, jika hasilnya belum sesuai maka perlu diketahui berapa selisih nilai yang belum sesuai tersebut. Proses ini biasanya dinamakan dengan *loss function* atau *cost function* dan dinyatakan dengan  $E(x)$  atau  $J(x)$ .

Dengan menggunakan rumus *mean squared error* berikut ini, maka akan diketahui berapa selisih perbedaan tersebut.

$$Loss = \frac{1}{2} \sum_{n \in \text{train}} (t^{(n)} - y^{(n)})^2 \quad (2.13)$$

(Sumber : Skansi, 2018).

Dimana :

$$\begin{aligned} t^{(n)} &= \text{Target yang dicapai} \\ y^{(n)} &= \text{Hasil yang diharapkan} \end{aligned}$$

*Mean square error* menjumlahkan seluruh *error* yang terjadi dari hasil propagasi dan setelah didapatkan nilai *loss* maka dilakukan proses *update* bobot (Skansi, 2018).

### 2.17.4. Backpropagation

Menurut Skansi (2018) *backpropagation* dari *error* pada dasarnya hanya *gradien descent*. Secara matematika, *backpropagation* adalah sebagai berikut:

$$W_{updated} = W_{old} - \eta \nabla E \quad (2.14)$$

(Sumber : Skansi, 2018)



Dimana :

$w$  = *Weigh*

$\eta$  = *Learning rate*

$E$  = *Cost function*

Penulisan terhadap  $w_{updated}$  sebagai sebuah aturan yang menugaskan nilai baru terhadap  $w$  dapat juga ditulis dengan notasi berikut ini (Skansi, S., 2018).

$$w \leftarrow w - \eta \nabla E \quad (2.15)$$

(Sumber : Skansi, 2018)

Notasi diatas berarti bahwa pertama kali yang harus dihitung adalah yang paling kanan dan kemudian hasil perhitungan ditugaskan ke  $w$  sebagai nilai baru. Dalam proses *backpropagation* sering muncul tiga permasalahan yaitu yang pertama proses *backpropagation* memakan waktu yang lama. Setelah bobot diperbaharui, maka diperlukan proses beberapa *training sample* untuk melihat tiap-tiap bobot apakah nilainya lebih baik atau lebih buruk dari sebelumnya.

Permasalahan yang kedua yaitu dengan mengubah bobot secara individual maka tidak akan pernah menemukan apakah kombinasi dari bobot dapat berkerja dengan baik atau tidak, bisa jadi akan membuat hasil klasifikasi menjadi lebih parah dari sebelumnya.

Permasalah ketiga yaitu menjelang akhir proses pembelajaran, perubahan nilai harus kecil. Permasalahan yang terjadi dalam proses *backpropagation* ini biasanya diselesaikan dengan sebuah *learning rate* yang dinamis, dimana akan mendapatkan nilai yang semakin kecil seiring dengan perkembangan dari proses *backpropagation*.

Sedangkan menurut Setti dan Wanto (2018) *backpropagation* salah satu metode dalam *artificial neural network* dan mempunyai kelebihan dan kemampuan dalam proses *training*. *Backpropagation* digunakan dalam proses pelatihan. Algoritma *backpropagation* merupakan sebuah algoritma yang mudah dan sederhana yang berkinerja baik bahkan dengan data yang kompleks. Tidak seperti algoritma lainnya seperti algoritma (*Bayesian Learning*).

*Backpropagation* memiliki sifat komputasi yang bagus, terutama pada saat data skala besar disajikan. *Training* dengan algoritma *backpropagation*

seperti *training* dengan *neural network* lainnya. Selama proses *training* dengan *backpropagation* bobot diatur secara iteratif untuk meminimalisir *error* yang terjadi.

#### 2.17.5. *Gradient Descent*

Menurut Ruder (2017) *gradient descent* adalah salah satu algoritma yang paling terkenal untuk melakukan optimasi dan sejauh ini merupakan cara yang paling umum untuk mengoptimasi *neural network*. Pada saat yang bersamaan, setiap *library deep learning* berisi implementasi berbagai macam algoritma untuk mengoptimalkan *gradient descent*. *Gradient descent* terdiri dari berbagai macam yaitu: *batch gradient descent*, *stochastic gradient descent*, dan *minibatch gradient descent*.

Sedangkan menurut Heaton (2015) *backpropagation* adalah tipe dari *gradient descent*. *Gradient descent* berkenaan dengan perhitungan sebuah *gradient* pada tiap bobot elemen dalam *neural network*. Karena tidak akan menghasilkan nilai yang diharapkan untuk tiap-tiap *training* elemen, *gradient* dari tiap bobot akan memberikan sebuah petunjuk tentang perubahan tiap bobot untuk mencapai nilai yang diharapkan. Jika *neural network* memberikan nilai yang pasti sebagaimana diharapkan, maka *gradient* untuk tiap bobot akan menjadi nol, menunjukkan bahwa tidak perlu perubahan bobot.

*Gradient* merupakan turunan dari fungsi *error* pada nilai bobot. Fungsi *error* mengukur jarak *output* dari *neural network* dengan *output* yang diharapkan. Sebenarnya, kita dapat menggunakan *gradient descent*, dimana sebuah proses yang tiap *gradient* nilai dari bobot dapat mencapai nilai lebih rendah dari fungsi *error*.

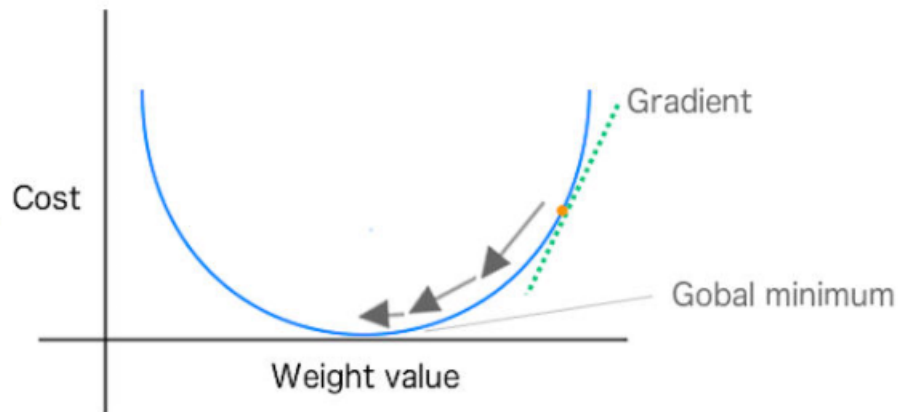
Berkenaan dengan fungsi *error*, *gradient* pada dasarnya adalah turunan parsial dari tiap bobot dalam *neural network*. Tiap bobot memiliki sebuah *gradient* yang merupakan kemiringan dari sebuah fungsi *error*. Sebuah bobot adalah koneksi antara dua buah neuron. Menghitung *gradient* dari fungsi *error* memungkinkan metode *training* untuk menentukan apakah bobot harus ditingkatkan atau diturunkan. Pada gilirannya, penentuan ini akan menurunkan *error* dari *neural network*.

*Error* merupakan perbedaan antara hasil yang diharapkan dengan nilai pasti dari *neural network*. Banyak metode training yang berbeda yang disebut dengan algoritma *training propagation* yang memanfaatkan *gradient*. Dari semua algoritma tersebut memberikan informasi yaitu *zero gradient* berarti bobot tidak memberikan kontribusi terhadap *error* dalam *neural network*. *Negative gradient* berarti bobot harus ditingkatkan untuk mencapai *error* terendah. Sedangkan *positive gradient* berarti bobot harus diturunkan untuk mencapai *error* terendah.

Sedangkan menurut Paterson dan Gibson (2017) dalam *gradient descent*, kita dapat membayangkan kualitas dari hasil prediksi jaringan (sebagai sebuah fungsi dari bobot atau nilai parameter) sebagai sebuah lanskap. Bukit merepresentasikan lokasi (nilai parameter atau bobot) yang memberikan banyak kesalahan prediksi. Sedangkan lembah merepresentasikan lokasi dengan sedikit kesalahan prediksi.

Kita dapat memilih salah satu titik pada lanskap tersebut untuk menempatkan satu bobot awal. Tujuannya adalah untuk memindahkan bobot agar turun ke lembah, ke tempat yang area paling rendah errornya secepat mungkin. Sebuah algoritma seperti *gradient descent* dapat merasakan kemiringan yang sebenarnya dari setiap bobot dan tahu ke arah mana akan turun.

*Gradient descent* mengukur kemiringan (perubahan *error* yang disebabkan oleh perubahan bobot) dan membawa bobot satu langkah ke bagian bawah lembah. Hal itu dilakukan dengan mengambil sebuah turunan dari *loss function* untuk menghasilkan *gradient*. *Gradient* memberikan arah kepada algoritma untuk langkah selanjutnya dalam melakukan optimasi algoritma, sebagaimana yang digambarkan dalam gambar berikut ini :



Gambar 2.18. *Gradient*

(Sumber : Abrahams dkk, 2016)

*Gradient descent* adalah algoritma untuk menemukan titik dimana sebuah fungsi mencapai nilai minimum. Perlu diingat bahwa kita mendefinisikan *learning* sebagai peningkatan parameter model untuk meminimalkan *loss* melalui sebuah langkah *training*. Dengan konsep tersebut, menerapkan *gradient descent* untuk menemukan fungsi *loss* minimum akan menghasilkan model *learning* dari input data.

*Gradient* merupakan operasi matematika secara umum direpresentasikan dengan simbol  $\nabla$  (nabla). Ini dianalogikan sebuah turunan, tapi diterapkan untuk fungsi yang diinput sebuah vektor dan *output* sebuah nilai tunggal, seperti fungsi *loss*. *Output* dari *gradient* adalah sebuah vektor dari turunan parsial.

$$\nabla \equiv \left( \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_n} \right) \quad (2.16)$$

Dimana :

$\nabla$  = *Gradient*

$\partial$  = Turunan Parsial

$w_1 \dots w_n$  = bobot

### 2.17.6. Stochastic Gradient Descent (SGD)

SGD adalah algoritma optimasi *machine learning*. Dalam gradient descent kita menghitung seluruh *loss* terhadap semua *training sample* sebelum menghitung gradient dan memperbaharui parameter vektor. Dalam SGD, kita menghitung gradient dan parameter vektor yang diupdate setelah *training sampel* (Patterson dan Gibson, 2017).

Menurut Bhanu dan Kumar (2017) dengan *dataset* yang besar, menghitung *cost* dan *gradient* untuk seluruh data *training* biasanya sangat lambat. Permasalahan ini diselesaikan dengan SGD dengan mengikuti *gradient* yang *negatif* dari fungsi yang objektif setelah melihat beberapa contoh *training*. SGD digunakan dalam *neural network* untuk mengurangi *cost* yang tinggi dari proses *backpropagation* terhadap seluruh *training dataset*. Diberikan sebuah fungsi *objective*  $J(\phi)$ , algoritma standar *gradient descent* memperbaharui parameter  $\phi$  sebagai berikut (Bhanu dan Kumar, 2017).

$$\phi = \phi - \alpha \nabla_{\phi} E[J(\phi)] \quad (2.17)$$

(Sumber : Bhanu dan Kumar, 2017).

Dimana nilai yang diharapkan  $E[J(\phi)]$  diperoleh melalui sebuah proses evaluasi *cost* dan *gradient* terhadap seluruh data *training*. Dengan SGD, *gradient* dari parameter dihitung menggunakan beberapa *training sample* tanpa nilai yang diharapkan. Parameter yang diperbaharui sebagai berikut (Bhanu, B., Kumar, A., 2017).

$$\phi = \phi - \alpha \nabla_{\phi} J(\phi; x^{(i)}, y^{(i)}) \quad (2.18)$$

(Sumber : Bhanu dan Kumar, 2017)

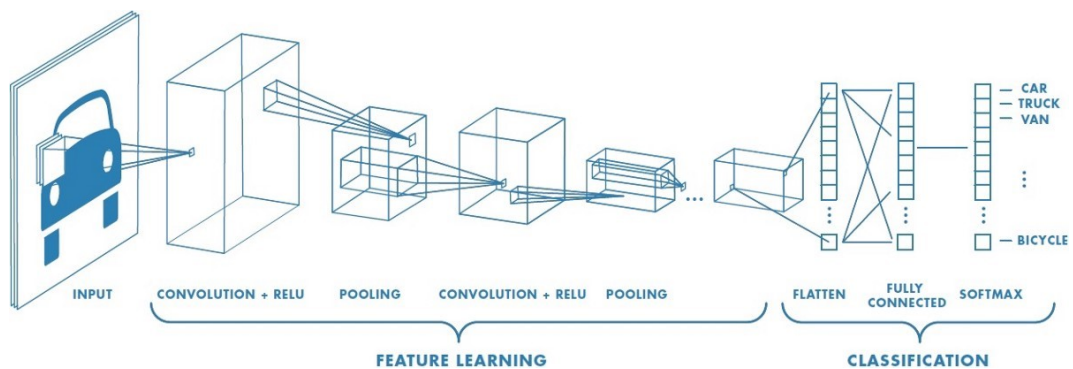
Pasangan  $(x^{(i)}, y^{(i)})$  adalah dari *training set*. Tiap parameter yang diperbaharui dihitung menggunakan beberapa *training sample*. Ini mengurangi *variance* dalam parameter yang diperbaharui dengan potensi mengarah ke konvergensi yang lebih stabil. Sebelumnya tiap *training epoch*, kita acak data *training* untuk menghindari bias dan *gradient*. Menyajikan data *training* ke dalam jaringan tanpa diacak dapat menyebabkan bias dan *gradient* dan mengarah ke konvergensi yang buruk.

Salah satu masalah *learning* dengan SGD adalah kecenderungan *gradient* menurun karena dilakukan *backpropagation* melalui berbagai lapisan tidak linier. Kita mengatasi masalah ini dengan menggunakan *learning rate* yang berbeda untuk tiap layer dalam jaringan yang diusulkan.

### 2.18. *Convolutional Neural Network (CNN)*

Menurut Patterson dan Gibson (2019) tujuan dari CNN adalah untuk mempelajari fitur tingkat tinggi dalam data melalui proses konvolusi. CNN sangat cocok untuk mengenali objek pada gambar secara konsisten. CNN dapat mengenali wajah, orang, rambu jalan, dan banyak aspek lainnya dari data visual termasuk teks dan suara. Kemampuan CNN dalam mengenali gambar menjadi salah satu alasan utama mengapa dunia mengakui keunggulan dari deep learning.

CNN mendorong kemajuan besar dalam bidang *computer vision* yang memiliki aplikasi yang nyata untuk mobil yang dapat mengemudi sendiri, robotika, drone, dan pengobatan gangguan penglihatan. CNN juga banyak digunakan dalam tugas-tugas lain seperti terjemahan bahasa alami, dan analisis sentimen. Konvolusi adalah sebuah konsep yang manjur yang membantu dalam membangun fitur ruang yang lebih kuat berdasarkan sinyal. CNN memiliki tiga layer yaitu *input layer*, *feature extraction (learning layer)*, dan *classification layer*.



Gambar 2.19. Arsitektur *Convolutional Neural Network*

(Sumber :

[https://miro.medium.com/max/1000/1\\*XbuW8WuRrAY5pC4t-9DZAO.jpeg](https://miro.medium.com/max/1000/1*XbuW8WuRrAY5pC4t-9DZAO.jpeg))

Menurut Osinga (2018) penelitian awalnya digunakan *fully connected network* untuk menyelesaikan berbagai macam masalah. Tetapi ketika data yang diinput sebuah gambar, maka *fully connected network* menjadi pilihan yang buruk. CNN menyediakan cara untuk kita untuk melatih klasifikasi gambar menggunakan parameter yang jauh lebih sedikit. CNN melakukan ini dengan meniru bagaimana manusia dan hewan melihat.

Pada dasarnya operasi dalam CNN adalah konvolusi. Malahan menerapkan sebuah fungsi terhadap seluruh gambar yang diinput, konvolusi memindai dengan menjalankan jendela kecil terhadap gambar sekaligus. Pada tiap lokasi diterapkan sebuah *kernel* (biasanya sebuah matrik perkalian yang diikuti oleh sebuah fungsi aktivasi, seperti pada *fully connected network*). *Kernel* tersebut sering disebut sebagai *filter*.

Hasil dari penerapan *kernel* terhadap seluruh gambar merupakan gambar baru yang lebih kecil. Sebagai contoh, sebuah bentuk filternya adalah (3,3). Jika kita terapkan 32 filter terhadap gambar yang di *input*, maka kita akan membutuhkan  $3*3*3(\text{input image/RGB})*32 = 864 \text{ parameter}$  sebuah penghematan yang besar melalui sebuah *fully connected network*.

Sebuah CNN adalah susunan dari urutan *layer*, dimana setiap *layer* dari jaringan berjalan melalui sebuah fungsi yang dapat dibedakan untuk mentransformasikan dirinya dari salah satu gumpalan dari aktivasi menjadi yang lainnya. Empat tipe utama dari *layer* yang digunakan untuk membangun sebuah CNN, yaitu *Convolutional layer*, *Rectified Linear Unit layer*, *Pooling layer*, dan *Fully Connected layer*. Semua *layer* ini disusun secara bersamaan untuk membentuk sebuah CNN yang utuh (Dev, 2017).

Arsitektur CNN sederhana terdiri dari *Input-Conv-ReLU-Pool-Fully Connected*. Namun dalam *deep CNN* umumnya lebih banyak *layer* yang diselipkan diantara lima *layer* dasar ini. Sebuah arsitektur *deep neural network* klasik terdiri dari *Input-Conv-ReLU-Conv-ReLU-Pooling-ReLU-Conv-ReLU-Pooling-Fully Connected*.

CNN melakukan transformasi data dari *input layer* melalui semua koneksi *layer* kedalam sekumpulan nilai kelas yang diberikan oleh *output layer*. Terdapat banyak variasi dari arsitektur CNN, tapi variasi tersebut berdasarkan pola dari

layer. sebuah CNN terdiri dari *input layer*, *feature extraction (learning) layers*, dan *classification layers*.

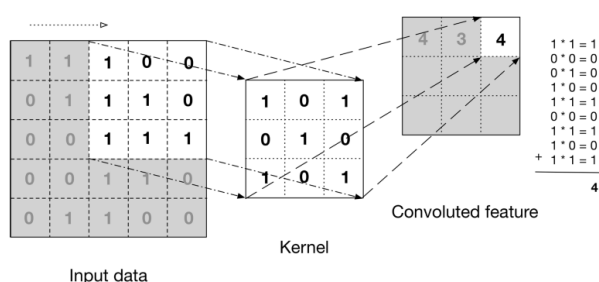
*Input layer* secara umum menerima input tiga dimensi dalam bentuk spasial dengan ukuran (panjang x lebar) dari gambar dan memiliki kedalaman yang mewakili dari saluran warna (umumnya terdiri dari tiga saluran yaitu , merah, hijau, dan biru atau sering disebut RGB)

### 2.18.1. Feature Extraction (Learning) Layer

*Feature extraction layers* secara umum memiliki pola pengulangan secara berurutan yang terdiri dari *convolutional layer* dan *pooling layer*.

### 2.18.2. Convolutional Layer

Menurut Patterson dan Gibson (2017) sebuah *convolutional layer* mendefinisikan sebuah operasi matematika yang menggambarkan sebuah aturan bagaimana menyatukan dua kumpulan informasi. Operasi *convolution* dikenal sebagai pendeteksi fitur dari sebuah CNN. *Input* dari sebuah *convolution* bisa jadi data mentah atau sebuah *output feature map* dari *convolution* lainnya. *Convolution* sering diartikan sebagai sebuah *filter* dimana kernel melakukan *filter input* data terhadap informasi jenis tertentu. Misalnya sisi kernel, maka memungkinkan data hanya lewat melalui tepi gambar.



Gambar 2.20. Operasi dari *convolution*

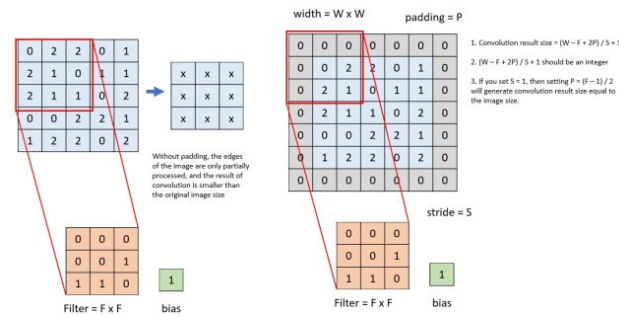
(Sumber : Patterson dan Gibson, 2017)

*Hyperparameter* menentukan pengaturan spasial dan ukuran *output* dari *convolutional layer* yang terdiri dari *filter size*, *output*, *stride*, dan *zero padding*.



a. *Filter Size*

Setiap *filter* kecil secara spasial berkenaan dengan lebar dan tinggi ukuran *filter*. Misalnya bagaimana *convolutional layer* dapat memiliki *filter* yang berukuran 5x5x3. Ini berarti ukuran dari *filter* tersebut lebarnya 5, tingginya 5, dan memiliki 3 saluran warna yang terdiri dari RGB (Patterson dan Gibson, 2017).



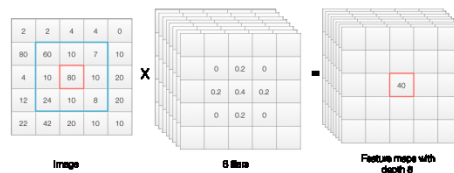
Gambar 2.21. Ukuran *Filter*

(Sumber

[https://jamesmccaffrey.files.wordpress.com/2018/05/convolution\\_math.jpg?w=640&h=358](https://jamesmccaffrey.files.wordpress.com/2018/05/convolution_math.jpg?w=640&h=358))

b. *Output Depth*

Kita secara manual dapat memilih kedalaman dari *output volume*. Kedalaman *hiperparameter* mengontrol jumlah *neuron* dalam *convolutional layer* yang terhubung ke wilayah yang sama dari *input volume* (Patterson dan Gibson, 2017).



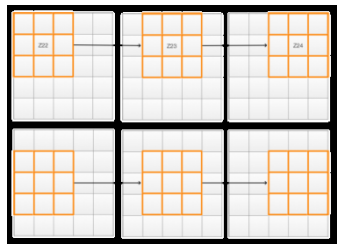
Gambar 2.22. Ketebalan *Output*

(Sumber : [https://jhui.github.io/assets/cnn/filter\\_m.png](https://jhui.github.io/assets/cnn/filter_m.png))

c. *Stride*

*Stride* melakukan konfigurasi seberapa jauh jendela *filter* bergeser akan berpindah sesuai aplikasi dari fungsi *filter*. Tiap kali kita menerapkan fungsi *filter* ke kolom input, kita membuat kedalaman kolom baru dalam *volume output*. Pengaturan yang paling rendah untuk *stride* misalnya 1, akan mengalokasikan lebih banyak *depth colom* dalam *output volume*.

Ini juga akan menghasilkan lebih berat bidang reseptif yang tumpang tindih diantara kolom, mengarah ke *output volume* yang lebih besar. Sebaliknya adalah benar jika kita menentukan nilai *stride* yang tinggi. Nilai *stride* yang tinggi akan memberikan sedikit tumpang tindih dan *volume output* yang kecil secara spasial (Patterson dan Gibson, 2017).



Gambar 2.23. *Stride* dengan nilai satu piksel

(Sumber : <https://jhui.github.io/assets/cnn/stride.png>)

d. *Zerro Padding*

Dengan menggunakan *zerro padding* kita dapat mengontrol ukuran spasial dari *output volume*. Kita menggunakan ini untuk menangani kasus yang mana kita mau mempertahankan ukuran spasial *volume input* dalam *volume output* (Patterson dan Gibson, 2017).

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

Gambar 2.24. *Padding* berukuran 1x1 piksel

(Sumber :

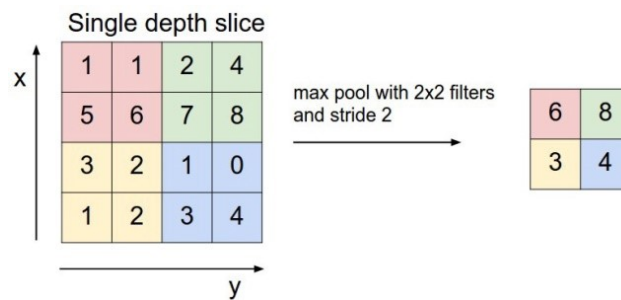
<https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2017/06/28093553/zero-padding.png>)

### 2.18.3. *Pooling Layer*

Menurut Hope, Resheff dan Lieder (2017) secara teknis *pooling layer* berarti mengurangi ukuran dari data dengan beberapa fungsi agregasi lokal, biasanya dalam tiap *feature map*. Alasan dibalik ini bersifat teknis dan lebih teoritis. Aspek teknisnya adalah bahwa *pooling* mengurangi ukuran dari data yang akan diproses di akhir. Secara drastis dapat mengurangi jumlah dari seluruh parameter dalam model. Khususnya jika kita menggunakan *fully connected layer* setelah melakukan *convolutional*.

*Pooling layer* umumnya disisipkan di antara *convolutional layer* secara berturut-turut. *Pooling layer* mengurangi representasi data secara progresif melalui jaringan dan membantu mengontrol *overfitting*. Lapisan *pooling layer* beroperasi secara independen. Operasi yang umum digunakan adalah *max pooling* dan *average pooling*.

*Pooling layer* menggunakan operasi *max()* untuk mengurangi *input data* secara spasial (lebar dan tinggi). Operasi ini merujuk sebagai *max pooling*. Dengan *filter* yang berukuran 2x2, operasi *max()* mengambil nilai terbesar dari empat nilai dalam area *filter*. Operasi ini tidak berpengaruh terhadap kedalaman dimensi.



Gambar 2.25. *Max Pooling* 2x2 dengan *stride* 2 piksel

(sumber :

<https://qph.fs.quoracdn.net/main-qimg-8afedfb2f82f279781bfefa269bc6a90.webp>)

*Pooling layer* menggunakan *filter* untuk melakukan proses *downsampling* pada *input volume*. *Layer* ini melakukan *downsampling* sepanjang dimensi spasial dari data *input*. Ini berarti jika gambar yang di *input* memiliki lebar 32 piksel dan tinggi 32 piksel, lebar dan tinggi yang dihasilkan akan lebih kecil (misalnya lebar menjadi 16 piksel dan tinggi 16 piksel). Pengaturan yang paling umum digunakan untuk *pooling layer* adalah menerapkan *filter* berukuran 2x2 dengan *stride* 2. Ini akan menurunkan tiap kedalaman dari sampel yang beririsan dalam *volume input* oleh sebuah faktor dari dua dimensi spasial (lebar dan tinggi). *Downsampling* menghasilkan 75 persen data dari aktivasi yang dibuang.

#### 2.18.4. *Classification Layer*

Dalam proses *classification* kita memiliki satu atau lebih *fully connected layer* (FCL) untuk mengambil fitur tingkat tinggi dan menghasilkan kelas probabilitas atau peluang dan score. *Layer* ini terhubung ke semua *neuron* dalam *layer* sebelumnya. Hasil dari *layer* ini biasanya menghasilkan *output* dua dimensi dari dimensi (bxN), dimana b adalah jumlah contoh dalam *minibatch* dan N adalah jumlah kelas yang penting dalam *scoring*.

### 2.18.5. *Flattening*

Sebelum menerapkan *fully connected layer*, maka perlu dilakukan proses *flatten* gambar kembali menjadi bentuk vektor satu dimensi, karena *fully connected layer* tidak lagi membutuhkan aspek spasial. Perlu diperhatikan bahwa ukuran gambar yang diikuti oleh dua *convolution* dan *pooling layer* adalah  $7 \times 7 \times 64$ .

Ukuran gambar aslinya adalah  $28 \times 28$  piksel, pengurangan pertama menjadi  $14 \times 14$ , dan kemudian menjadi  $7 \times 7$  dalam dua kali operasi *pooling layer*. nilai 64 merupakan jumlah dari *feature map* yang dibuat dalam *convolutional layer* yang kedua. Ketika mempertimbangkan jumlah total dari parameter yang dipelajari dalam model, sebagian besar akan berada dalam *fully connected layer* (mulai dari  $7 \times 7 \times 64$  hingga 1.024 memberi kita 3,2 juta parameter). Jumlah ini akan menjadi 16 kali lebih besar ( $28 \times 28 \times 64 \times 1.024$ , kira-kira berjumlah 51 juta) jika kita tidak menggunakan *max pooling* (Hope, Resheff dan Lieder, 2017 ).

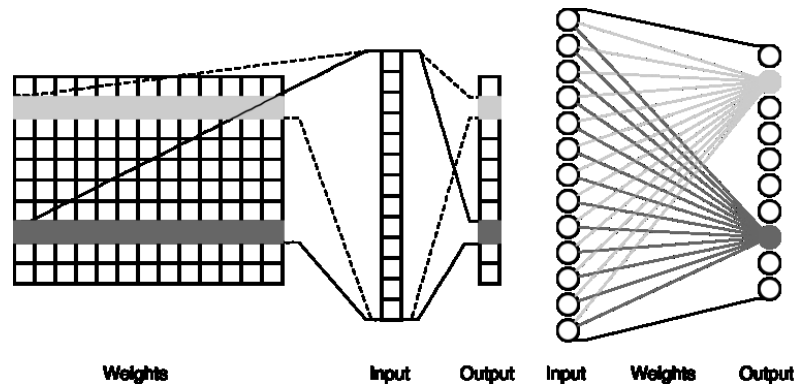
Menurut Sugomori dkk (2017 ) semua algoritma *machine learning* dan *deep learning* secara tidak langsung menyatakan bahwa tipe dari data *input* adalah berdimensi satu. Ketika kita lihat aplikasi dunia nyata, data tidak mesti satu dimensi. Contohnya adalah gambar. Meskipun kita masih dapat mengkonversi menjadi dua dimensi atau lebih tinggi kedalam array satu dimensi. Dilihat dari sudut pandang implementasi, akan lebih baik untuk membangun sebuah model yang dapat menangani data dua dimensi seperti apa adanya. Selain itu, beberapa informasi yang tertanam dalam data, seperti hubungan posisi, mungkin hilang ketika dilakukan *flatten* menjadi satu dimensi.

### 2.18.6. *Fully Connected Layer*

Kita menggunakan *layer* ini untuk menghitung *score* dari kelas yang akan digunakan sebagai *output* dari sebuah *network*. *Output layer* merupakan akhir dari sebuah *network*. Ukuran dari *output volume* adalah  $1 \times 1 \times N$ , dimana N adalah jumlah dari kelas *output* yang telah kita evaluasi. *Layer* ini mempunyai koneksi antar semua *neuron* pada *layer* sebelumnya.

*Fully connected layer* mempunyai parameter normal untuk layer dan *hyperparameter*. *Fully connected layer* melakukan transformasi pada data *input*

*volume* yang merupakan sebuah fungsi dari aktivasi data *input volume* dan parameter (bobot dan bias dari *neuron*).



Gambar 2.26. *Fully Connected Layer*

(Sumber :

<https://www.oreilly.com/library/view/machine-learning-with/9781787121515/assets/88732ed1-f431-4e77-a2cc-3e8e20b414a0.png>)

### 2.18.7. *Softmax*

Menurut Osinga (2018) *softmax* merupakan fungsi aktivasi yang memastikan bahwa jumlah dari *output* vektor benar-benar 1. Ini adalah fungsi aktivasi yang sesuai untuk jaringan yang menghasilkan *output* tepat satu label untuk tiap *input* yang diberikan, sebagai contoh yaitu klasifikasi gambar. *Output vector* kemudian akan direpresentasikan sebagai distribusi peluang.

Fungsi *softmax* digunakan pada klasifikasi *neural network*. *Softmax* biasanya ditemukan pada *output neural network*. *Neuron* yang memiliki nilai tertinggi menyatakan *input* sebagai anggota dari kelas tersebut. Oleh karena itu, metode ini lebih disukai, fungsi aktivasi *softmax* memaksa *output* dari *neural network* untuk merepresentasikan peluang bahwa *input* termasuk ke dalam tiap-tiap kelas. Tanpa *softmax*, *neuron output* hanyalah nilai numerik, dengan mengindikasikan yang tertinggi sebagai kelas pemenang (Heaton, 2015). Fungsi dari *softmax* dapat dilihat sebagai berikut:

$$\phi_i = \frac{e^{z_i}}{\sum_{j \in \text{group}} e^{z_j}} \quad (2.19)$$

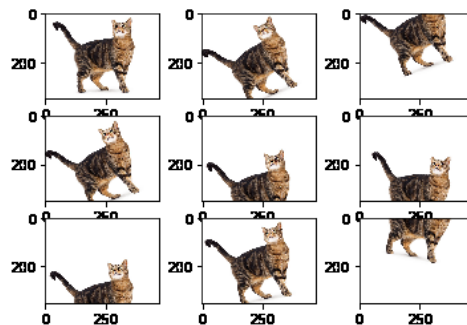
(Sumber : Heaton, 2015)

Dimana :

- $\phi$  = Fungsi *softmax*
- $i$  = Indeks *neuron output* yang dihitung
- $e$  = Fungsi eksponensial
- $j$  = Indeks *neuron output* dalam grup
- $z$  = Array dari *neuron output*

## 2.19. Augmentasi

Menurut Asperti dan Mastronardo (2017) augmentasi adalah sebuah teknik kunci dari *machine learning*. Augmentasi terdapat dalam peningkatan jumlah data, dengan menyatukan data baru secara buatan dari data yang sudah ada melalui perubahan kecil pada gambar tersebut. Contohnya dalam augmentasi gambar maka operasi yang dilakukan adalah rotasi, modifikasi pencahayaan, penyekalaan ulang, pemangkasan, dan sebagainya. Bahkan pegacakan suara dapat dikatakan sebagai sebuah bentuk dari data augmentasi.



Gambar 2.27. Augmentasi gambar

( Sumber :

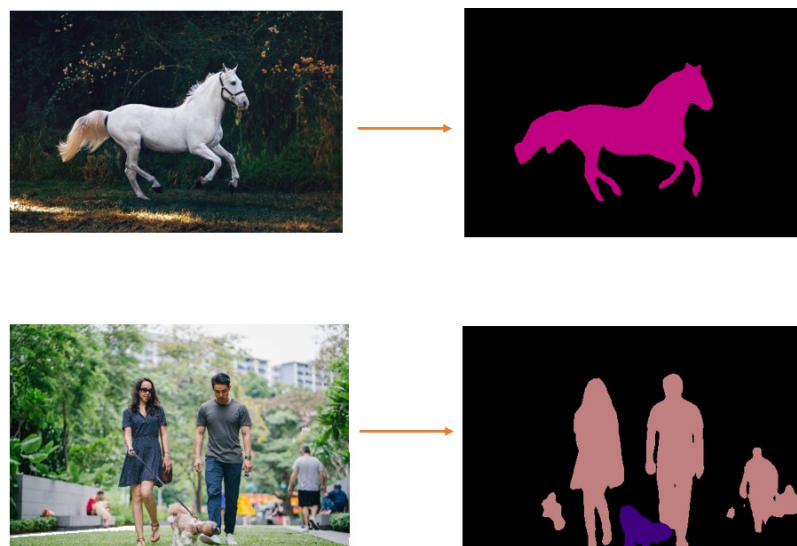
[https://miro.medium.com/max/366/1\\*5RXJ2OIVJp4\\_sd1XzhK6pQ.png](https://miro.medium.com/max/366/1*5RXJ2OIVJp4_sd1XzhK6pQ.png) )

## 2.20. Segmentasi

Menurut He dan Huang (2017) segmentasi gambar adalah sebuah langkah proses data yang sangat penting dalam bidang *computer vision* dan pengenalan pola. Pada dasarnya segmentasi hanya sebuah proses membagi sebuah

gambar menjadi area atau objek yang berbeda dan mengekstraksi objek yang penting dan bermakna. Lebih tepatnya, segmentasi gambar adalah proses membagi sebuah gambar kedalam beberapa wilayah yang tidak tumpang tindih (*overlapping*) berdasarkan skala keabuan, warna, tekstur, bentuk, ukuran, dan posisi dari gambar.

Tiap piksel dalam wilayah yang sama mempunyai beberapa karakteristik yang cocok atau ciri yang diperhitungkan, tetapi piksel pada area yang berdekatan memiliki karakteristik atau ciri yang berbeda. Saat ini segmentasi gambar telah banyak digunakan dalam berbagai bidang aplikasi.



Gambar 2.28. Segmentasi Gambar

( Sumber :

<https://www.learnopencv.com/wp-content/uploads/2019/06/semantic-segmentation-examples.png> )

### 2.20.1. U-Net

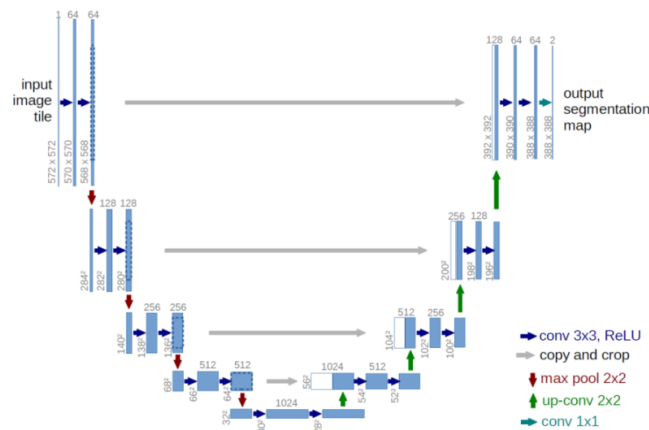
U-Net adalah *Convolutional Neural Network* untuk segmentasi gambar pada bidang biomedis yang diciptakan oleh departemen ilmu komputer Universitas Freiburg Jerman. U-Net juga disebut *Fully Connected Network* (FCN) untuk melakukan segmentasi gambar.



Menurut Ronneberger, Fischer dan Brox (2015) U-Net merupakan sebuah jaringan yang terdiri dari jalur kontrak yang terdapat sebelah kiri dan jalur ekspansif yang terdapat sebelah kanan. Jalur kontrak mengikuti arsitektur khusus dari jaringan konvolusi. Jalur tersebut terdiri dari aplikasi berulang dari dua konvolusi yang terdiri dari 3x3 (baris x kolom). Tiap konvolusi diikuti oleh fungsi aktivasi Rectified Linear Unit (ReLU) dan 2x2 max pooling dengan stride 2 untuk downsampling. Tiap langkah downsampling digandakan jumlah dari fitur channel.

Tiap langkah pada jalur ekspansif, maka terdiri dari sebuah upsampling dari feature map yang diikuti oleh 2x2 konvolusi (up-convolution), yang membagi dua jumlah dari fitur channel. Gabungan dengan yang sejalan dengan feature map yang dipotong dari jalur kontrak, dan dua konvolusi yang berukuran 3x3, masing-masing diikuti oleh sebuah ReLU.

Pemotongan diperlukan karena hilangnya piksel pada batas gambar dalam setiap konvolusi. Pada lapisan akhir, sebuah konvolusi yang berukuran 1x1 digunakan untuk memetakan masing-masing 64 fitur komponen vektor ke jumlah kelas yang diinginkan. Total jaringan memiliki 23 lapisan konvolusi.



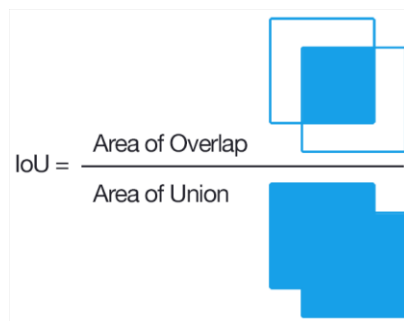
Gambar 2.29. Arsitektur U-Net

(Sumber :

<https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/u-net-architecture.png>)

### 2.20.2. *Intersection over Union (IoU)*

Menurut Rahman dan Wang (2016) ukuran standar yang paling umum digunakan dalam permasalahan segmentasi objek dinamakan dengan *Intersection over Union (IoU)*. Apabila diberikan sebuah gambar, IoU mengukur kesesuaian dari batas gambar, yaitu antara batas yang diprediksi dengan batas yang sebenarnya di lapangan untuk tiap-tiap objek yang ada dalam gambar tersebut. IoU mendefinisikan ukuran dari *intersection* dibagi dengan *union* dari dua batas tersebut.



Gambar 2.30. *Intersection over Union (IoU)*

(Sumber :

[https://www.pyimagesearch.com/wp-content/uploads/2016/09/iou\\_equation.png](https://www.pyimagesearch.com/wp-content/uploads/2016/09/iou_equation.png))



Gambar 2.31. Kelas akurasi IoU

(Sumber :

[https://www.pyimagesearch.com/wp-content/uploads/2016/09/iou\\_examples.png](https://www.pyimagesearch.com/wp-content/uploads/2016/09/iou_examples.png) )

### 2.21. **Klasifikasi Tutupan Lahan Skala 1:5.000**

Klasifikasi tutupan lahan untuk keperluan Rencana Detail Tata Ruang (RDTR) adalah klasifikasi detail skala besar yang disesuaikan guna memenuhi

kebutuhan klasifikasi pola ruang (zonasi) RDTR dan kebutuhan analisis untuk RDTR. Klasifikasi harus menggunakan citra satelit resolusi tinggi yang sudah terkoreksi.

Dalam melakukan klasifikasi tutupan lahan untuk skala 1:5.000, maka perlu diperhatikan tema yang harus dipetakan yaitu, bangunan, permukiman, komersial, perkantoran, industri, pelayanan umum, pariwisata, hankam, transportasi, perairan, area terbuka, hutan, persampahan, perkebunan, pertanian dan peternakan, dan pertambangan (Modul Validasi Peta Rencana Tata Ruang, 2017).

## **2.22. Penelitian Terdahulu**

Penelitian yang dilakukan oleh Tayyebi (2019) dalam penelitiannya yang berjudul *High Resolution Land Cover Mapping using Deep Learning* menggunakan citra satelit *National Agriculture Imagery Program* (NAIP). Dalam penelitian tersebut menerapkan model *deep learning* untuk membuat peta tutupan lahan di negara bagian Alabama menggunakan *Keras* dan *ArcgisPro*.

Resolusi spasial yang digunakan dalam penelitian ini yaitu 1m x 1m. Terdapat 12 citra satelit yang digunakan untuk dilakukan klasifikasi dengan model *deep learning*. Dalam penelitian tersebut informasi tutupan lahan yang dihasilkan terdiri dari enam kelas, meliputi bangunan, jaringan jalan, tubuh air, lahan terbuka, kebun, dan hutan.

Adapun jumlah *training sample* yang digunakan dalam penelitian ini terdiri dari bangunan 420 sampel, jaringan jalan 438 sampel, tubuh air 702 sampel, kebun 1008 sampel, hutan 837 sampel, dan sawah 891 sampel. Ukuran citra yang digunakan dalam penelitian ini yaitu 256 piksel x 256 piksel dengan jarak tiap gambar yang di *capture* sebesar 64 piksel (*stride*).

Jumlah model yang dilatih sebanyak 30 *epoch* dan tingkat kesabaran (*patient*) yang digunakan 20, artinya jika dalam 20 kali *training* model tidak ada peningkatan akurasi, maka *epoch* akan berhenti. Dari hasil penelitian tersebut maka diperoleh informasi akurasi *precision* dan *recall*. Untuk akurasi *precision* maka persentasenya diperoleh yaitu bangunan (82,50%), jaringan jalan (84,78%), tubuh air (86,14%), lahan terbuka (90,38%), kebun (89,05%), dan hutan

(91,46%). Sedangkan akurasi *recall* diperoleh bangunan (81,28%), jaringan jalan (85,13%), tubuh air (85,55%), lahan terbuka (91,88%), kebun (88,19%), dan hutan (92,65%).

Dalam penelitian “Analisis Klasifikasi Citra Satelit WorldView-2 level 2A menggunakan Model *Deep Learning*” ini model yang digunakan sebanyak 50 *epoch*, tingkat kesabaran yang digunakan sama dengan jumlah *epoch*. Citra satelit yang digunakan resolusinya 0,5 meter. Sedangkan ukuran citra yang digunakan dalam proses *training* berukuran 128 piksel x 128 piksel dan *stride* sebesar 64 piksel.

*Halaman ini sengaja dikosongkan*

## **BAB 3**

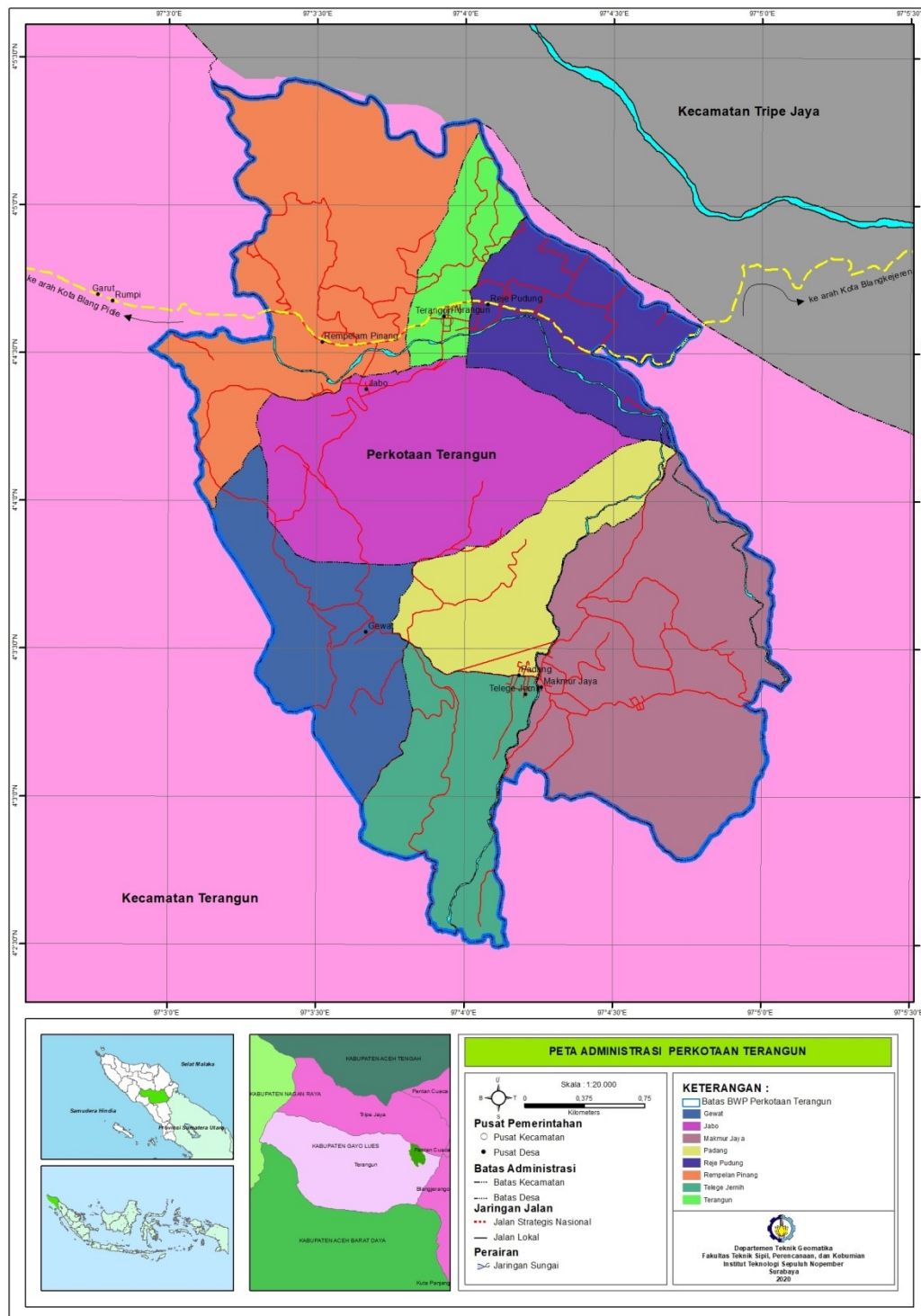
### **METODA PENELITIAN**

#### **3.1. Gambaran Umum Daerah Kajian**

Berdasarkan Qanun Kabupaten Gayo Lues No.15 Tahun 2013 tentang Rencana Tata Ruang Wilayah (RTRW) Tahun 2012-2032, Pemerintah Kabupaten Gayo Lues menetapkan Bagian Wilayah Perencanaan (BWP) Terangun sebagai salah satu wilayah perkotaan yang menjadi prioritas untuk dikembangkan. BWP Terangun terletak di Kecamatan Terangun, dimana Kecamatan Terangun terletak pada lokasi yang sangat strategis, karena berbatasan langsung dengan Kabupaten Aceh Barat Daya dan Taman Nasional Gunung Leuser (TNGL). Kawasan perkotaan Terangun terdiri dari 8 gampong/desa yaitu:

- a. Gampong Gewat
- b. Gampong Jabo
- c. Gampong Makmur Jaya
- d. Gampong Padang
- e. Gampong Reje Pudung
- f. Gampong Rempelan Pinang
- g. Gampong Telege Jernih
- h. Gampong Terangun

Secara astronomis BWP Terangun terletak pada koordinat  $04^{\circ} 02' 29''$  LU sampai  $04^{\circ} 05' 25''$  LS dan  $97^{\circ} 02' 56''$  BT sampai  $97^{\circ} 05' 06''$  BT dengan luas kawasannya seluas adalah 1.211 Ha.



Gambar 3.1. Peta Lokasi Perkotaan Terangun  
(Sumber : Dinas PUPR Kabupaten Gayo Lues)

### **3.2. Peralatan dan Data**

Untuk menunjang penelitian “Analisis Klasifikasi Citra Satelit WorldView-2 Menggunakan Model *Deep Learning*” maka diperlukan peralatan dan data yang memadai.

#### **3.2.1. Peralatan**

Adapun peralatan yang digunakan dalam penelitian ini meliputi :

- a. Seperangkat komputer dengan spesifikasi processor core I-7, *Read Access Memory* (RAM) berkapasitas 16 GB, Harddisk berkapasitas 2 TB, *Video Graphics Array* (VGA) NVIDIA berkapasitas 2 GB.
- b. Aplikasi Pengolahan Data Citra Satelit dan Pemetaan.
- c. Jupyter Notebook.

#### **3.2.2. Data**

Data yang digunakan dalam penelitian ini adalah data sekunder yang terdiri dari data raster dan data vektor. Data raster berupa citra satelit Worldview-2 level 2A yang telah terorthorektifikasi dan telah mendapatkan rekomendasi dari Badan Informasi Geospasial (BIG). Sedangkan data vektor berupa data batas administrasi (batas kabupaten, kecamatan, desa), batas wilayah perencanaan (BPW) Perkotaan Terangun, dan peta tutupan lahan hasil klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG skala 1:5.000. Semua data tersebut bersumber dari dinas PUPR Kabupaten Gayo Lues.

### **3.3. Variabel Penelitian**

Adapun variabel penelitian yang merupakan kelas klasifikasi yang digunakan terbagi menjadi delapan, yaitu bangunan, jaringan jalan, perkebunan campuran, kolam, pekarangan, sawah, sungai, dan tegalan. Pembagian kelas ini berdasarkan hasil observasi citra satelit WorldView-2 level 2A Perkotaan Terangun. Semua variabel akan digambarkan dalam bentuk polygon dengan ukuran terkecil adalah 6,25 meter. Berikut ini merupakan penjelasan masing masing variabel.



a. Bangunan

Bangunan merupakan wujud fisik hasil pekerjaan konstruksi yang menyatu dengan tempat kedudukannya, sebagian atau seluruhnya berada diatas dan/atau di dalam tanah dan/atau air, yang berfungsi sebagai tempat tinggal, kegiatan keagamaan, kegiatan usaha, kegiatan sosial, budaya, maupun kegiatan khusus (UU No.28 Tentang Bangunan Gedung, 2002).

b. Jalan

Jaringan prasarana transportasi yang diperuntukkan lalu lintas kendaraan (SNI 7645, 2010). Jaringan jalan yang akan digambarkan adalah yang mempunyai lebar  $\geq 2,5$  meter.

c. Lahan Terbuka

Lahan terbuka merupakan lahan tanpa tutupan, baik bersifat alami, semi alami, maupun artifisial menurut karaktersistik permukaannya (SNI 7645, 2010).

d. Sungai

Tempat mengalirnya air yang bersifat alamiah (SNI 7645, 2010). Jaringan sungai yang akan digambarkan adalah yang mempunyai lebar  $\geq 2,5$  meter.

e. Sawah

Areal pertanian yang digenangi air atau diberi air baik dengan teknologi pengairan, tadah hujan, lebak, atau pasang surut yang dicirikan oleh pola pematang, dengan ditanami jenis tanaman pangan berumur pendek atau padi (SNI 7645, 2010).

f. Ladang

Area yang digunakan untuk kegiatan pertanian dengan jenis tanaman semusim di lahan kering (SNI 7645, 2010).

g. Kolam

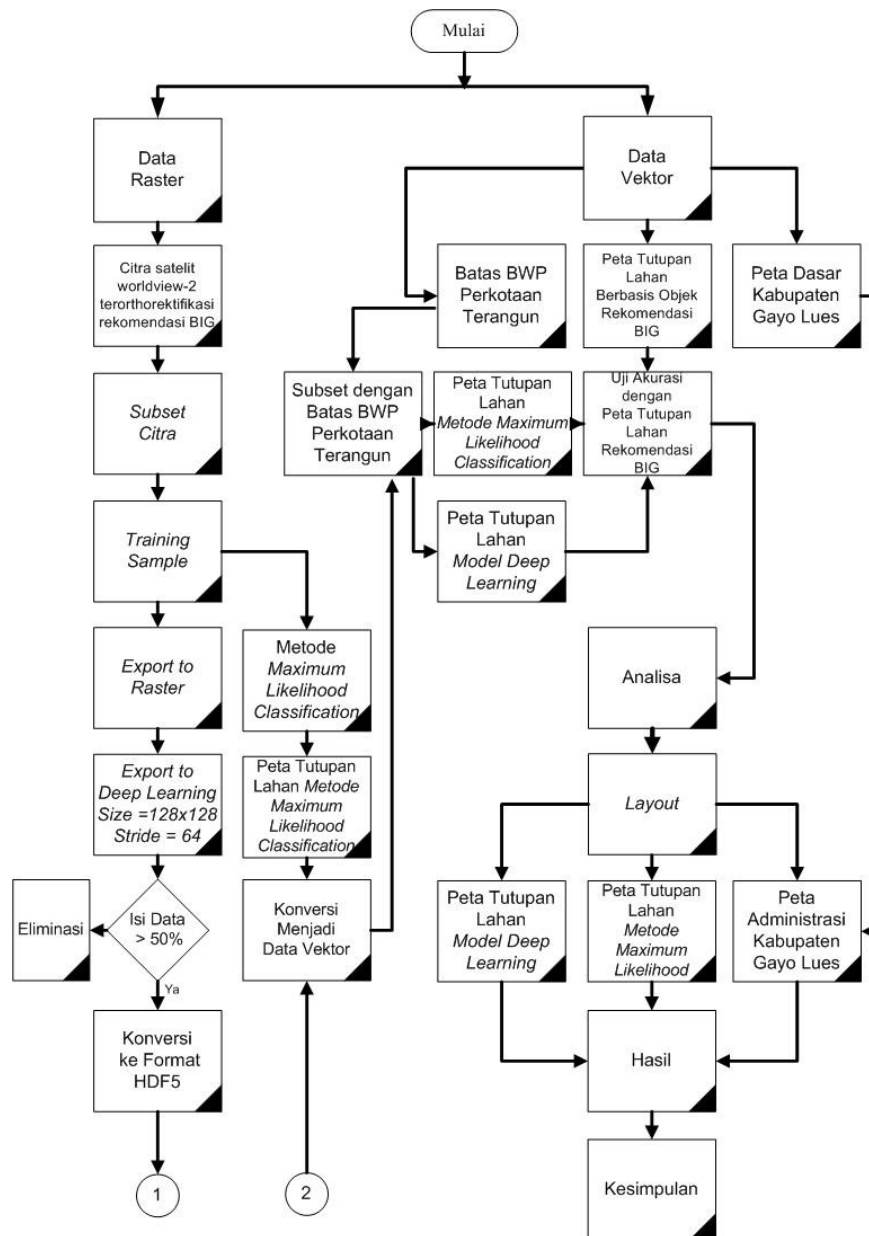
Kolam merupakan tempat atau wadah budidaya yang paling tua digunakan oleh manusia untuk wadah pembudidayaan ikan hias atau ikan konsumsi hampir diseluruh penjuru dunia. Kolam umumnya digunakan untuk ikan yang berukuran yang lebih besar, seperti ikan koi, ikan belida, dan ikan konsumsi (Satyani dan Priono, 2012).

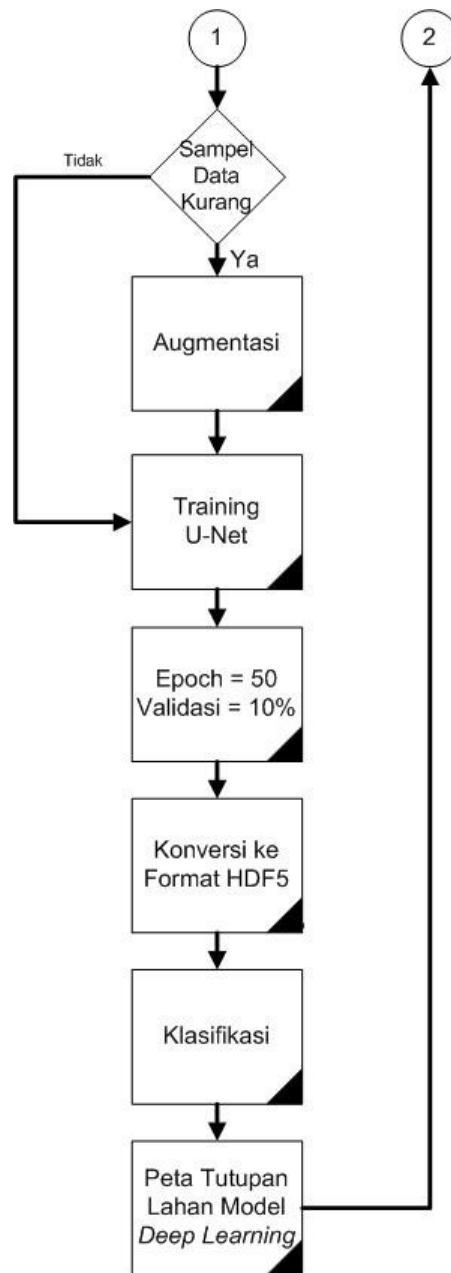
h. Perkebunan Campuran (SNI 7645, 2010).

Lahan yang ditanami tanaman keras lebih dari satu jenis atau tidak seragam yang menghasilkan bunga, buah, dan getah dan cara pengambilan hasilnya bukan dengan cara menebang pohon (SNI 7645, 2010).

### 3.4. Diagram Alir dan Alur Penelitian

Adapun diagram alir dari penelitian ini dapat dilihat pada Gambar 3.2. Diagram alir penelitian.





Gambar 3.2. Diagram alir penelitian

Penelitian diawali dengan studi literatur dan studi literatur dari penelitian terdahulu agar diketahui parameter apasaja yang digunakan untuk klasifikasi tutupan lahan menggunakan model *deep learning*. Tahap pengolahan data dan analisa citra dan tutupan lahan dijabarkan sebagai berikut ini:

- a. Melakukan analisis deskriptif untuk data raster dan data vektor.
- b. Menyiapkan data raster berupa data citra satelit Worldview-2 level 2A.

- c. Menyiapkan data vektor berupa batas administrasi kabupaten, kecamatan, kawasan Perkotaan Terangun, dan peta tutupan lahan berbasis objek yang telah mendapatkan rekomendasi dari BIG.
- d. *Subset* citra.
- e. Membuat *training sample* dan memberikan nama untuk delapan kelas tutupan lahan citra satelit WorldView-2 level 2A.
- f. Mengkonversi *training sample* menjadi data raster agar dapat dilakukan segmentasi gambar.
- g. Mengkonversi *training sample* menjadi data *deep learning* dengan ukuran cita 128x128 piksel dan ukuran *stride* 64 piksel.
- h. Menentukan *pooling* dengan menggunakan *max pooling*.
- i. Chip yang bernilai tidak nol < 50%, dikeluarkan dari proses training.
- j. Konversi chip kedalam format *Hierarchical Data Format 5* (HDF5).
- k. Melakukan proses *augmentasi* untuk meningkatkan jumlah *training sample* yang masih sedikit dengan melakukan metode yang umum dipakai yaitu *shifting*, *rotation*, dan *zoom*.
- l. Melakukan proses *augmentasi* dengan metode *shifting* yaitu dengan menambahkan data *training* yang baru dengan ukuran data 128x128 piksel, dan *stride* dengan ukuran 64.
- m. Melakukan proses *augmentasi* dengan metode *rotation* yaitu tiap-tiap data training dilakukan proses rotasi yaitu -180°, -90°, 90°, 180°.
- n. Melakukan proses *augmentasi* dengan metode *zoom* yaitu tiap-tiap data *training* dilakukan proses *zoom* yang berkisar antara 0.005 hingga 0.45
- o. Melatih model U-Net, dalam proses training 90% data digunakan untuk kalibrasi model, sedangkan 10% digunakan untuk validasi.
- p. Jalankan model dengan 50 *epochs*, hentikan model tiap 20 *epochs* jika akurasi dari validasi tidak meningkat.
- q. Konversi model U-Net ke dalam format *Hierarchical Data Format 5* (HDF 5).
- r. Menggunakan model yang sudah dilatih untuk mengklasifikasi citra WorldView-2 level 2A.

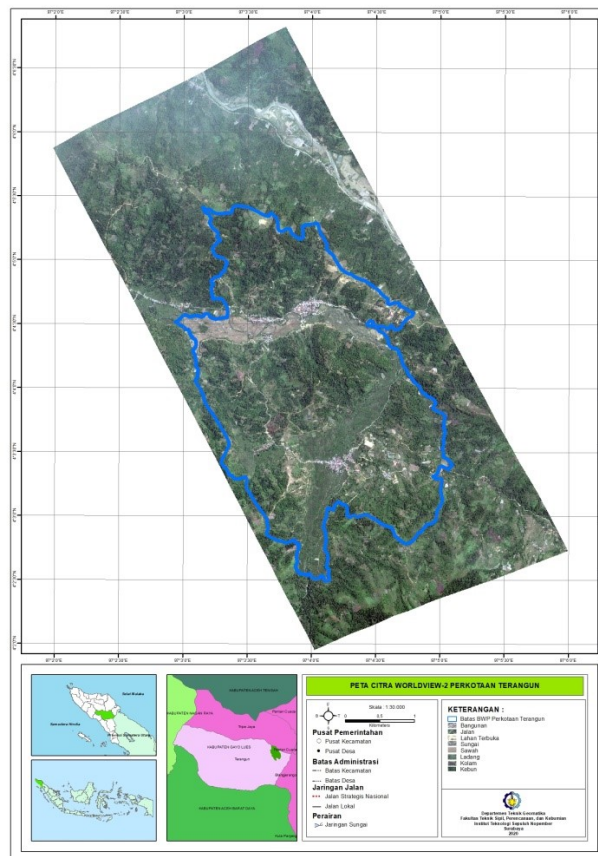
- s. Menggunakan *training sample* pada langkah (e) untuk melakukan klasifikasi metode *maximum likelihood classification*.
- t. Konversi hasil klasifikasi model *deep learning* dan hasil klasifikasi metode *maximum likelihood classification* ke dalam format data vektor.
- u. *Subset* hasil klasifikasi dengan batas Perkotaan Terangun.
- v. Menguji akurasi data vektor hasil klasifikasi model *deep learning* dan hasil klasifikasi metode *maximum likelihood classification* dengan peta tutupan lahan perkotaan terangun hasil interpretasi berbasis objek yang sudah mendapat rekomendasi dari Badan Informasi Geospasial (BIG).
- w. Analisa hasil dari uji akurasi terkait kesesuaian tutupan lahan hasil klasifikasi model *deep learning* dan metode *maximum likelihood classification* dengan peta tutupan lahan hasil klasifikasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG.
- x. *Layout* peta.
- y. Hasil.
- z. Kesimpulan.

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1. Citra Satelit WorldView-2 level 2A Terorthorektifikasi

Agar data dan informasi tutupan lahan tidak berbeda jauh dengan kondisi eksisting pada saat penyusunan dokumen perencanaan, maka diperlukan informasi yang berkaitan dengan citra satelit WorldView-2 level 2A yang digunakan. Adapun informasi tersebut yaitu jenis citra satelit, sudut kemiringan pada saat perekaman, resolusi spasial, tanggal akuisisi data, dan tutupan awan. Citra satelit WorldView-2 level 2A yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4.1. Citra Satelit WorldView-2 level 2A.



Gambar 4.1. Citra Satelit WorldView-2 level 2A

Berdasarkan berita acara yang dikeluarkan oleh Badan Informasi Geospasial (BIG) yang bertanggal 1 Agustus 2017, maka terkait informasi citra satelit WorldView-2 level 2A dapat dilihat pada **Lampiran 1. Berita Acara Spesifikasi Citra Satelit WorldView-2.**

Dari berita acara yang dikeluarkan oleh BIG, maka citra satelit WorldView-2 level 2A tersebut dapat digunakan untuk membuat peta tutupan lahan dengan skala besar, yaitu 1:5.000. Dalam pembuatan peta tutupan lahan skala 1:5.000, maka citra satelit WorldView-2 level 2A tidak dapat langsung digunakan untuk proses interpretasi tutupan lahan, tetapi harus dilakukan terlebih dahulu proses orthorektifikasi.

#### **4.2. Hasil Orthorektifikasi**

Dari hasil proses orthorektifikasi citra satelit WorldView-2 level 2A yang digunakan dalam penelitian ini mempunyai tingkat akurasi 2 meter. Adapun informasi hasil orthorektifikasi dapat dilihat pada **Lampiran 2 Berita Acara Hasil orthorektifikasi.**

Dari berita acara hasil orthorektifikasi yang dikeluarkan oleh BIG yang bertanggal 6 September 2017 menyatakan bahwa dari hasil uji akurasi dengan menggunakan 13 titik *Independent Check Point* (ICP) maka akurasi yang dihasilkan dari proses orthorektifikasi yaitu sebesar 2 meter. Dari hasil uji akurasi tersebut maka dinyatakan citra satelit WorldView-2 level 2A yang telah dilakukan proses orthorektifikasi tersebut sudah dapat digunakan dalam penyusunan peta skala 1:5.000.

Jika berpedoman pada Perka BIG No 15 Tahun 2014 Tentang Pedoman Teknis Ketelitian Peta Dasar, maka citra satelit WorldView-2 level 2A yang telah dilakukan proses orthorektifikasi tersebut termasuk dalam kategori ketelitian kelas 3, karena mempunyai akurasinya sebesar 2 meter. Ketelitian ini terletak pada rentang antara 1,5 meter (kelas 2) hingga 2,5 meter (kelas 3).

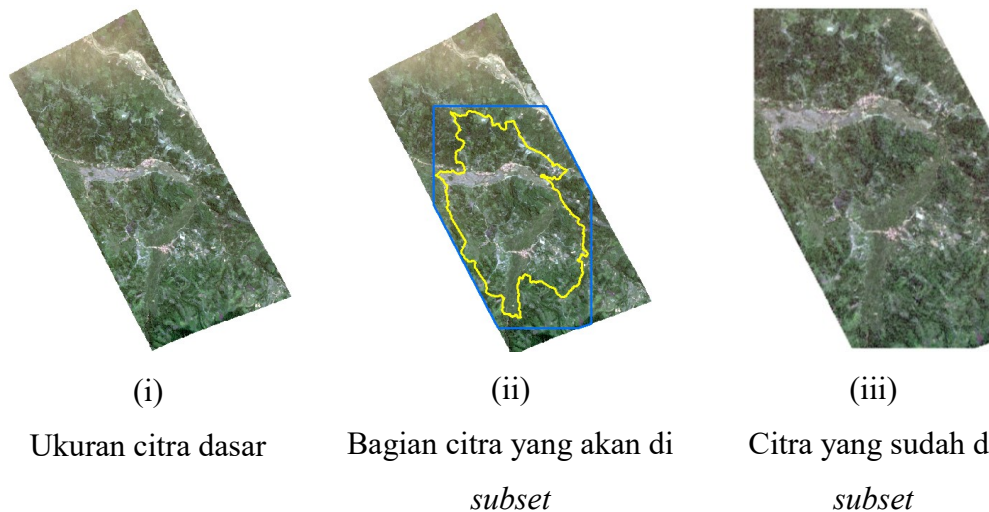
Sedangkan jika berpedoman pada Peraturan Kepala Badan Informasi Geospasial No 6 Tahun 2018 Tentang Perubahan atas Peraturan Kepala Badan Informasi Geospasial No 15 Tahun 2014 Tentang Pedoman Teknis Ketelitian Peta Dasar, ketelitian geometri citra satelit WorldView-2 level 2A yang digunakan

dalam penelitian ini mempunyai akurasi 2 meter, maka termasuk dalam ketelitian kelas 2.

Analisis klasifikasi citra satelit WorldView-2 level 2A dengan menggunakan model *deep learning* termasuk dalam metode klasifikasi terbimbing. Oleh karena itu diperlukan *training sampel* sebelum melakukan proses klasifikasi, agar komputer dapat mengikuti instruksi yang diberikan. Sehingga hasil klasifikasi dapat dilakukan dengan benar.

#### 4.3. **Subset Citra Satelit WorldView-2 Level 2A**

Sebelum membuat *training sample* maka citra satelit WorldView-2 level 2A perlu dilakukan *subset* terlebih dahulu agar *training sample* yang dibuat benar-benar mendekati ukuran lokasi penelitian, proses *subset* ini juga akan meningkatkan kinerja komputer karena citra satelit WorldView-2 level 2A ukurannya sudah lebih kecil. Citra satelit WorldView-2 level 2A yang sudah di *subset* dapat dilihat pada Gambar 4.2. Bagian citra satelit yang di *subset*.



Gambar 4.2. Bagian citra satelit yang di *subset*

#### 4.4. **Training Sample untuk klasifikasi**

*Training sample* diperlukan untuk memberikan informasi pada jaringan saraf agar mengenali objek-objek yang ada pada citra satelit. Adapun *training sample* yang digunakan dalam penelitian ini terdiri dari delapan kelas, yaitu



bangunan, jalan, lahan terbuka, sungai, sawah, ladang, kolam, dan kebun. *Training sample* digunakan untuk melatih jaringan saraf tiruan.

Dalam melakukan proses *training* jaringan saraf tiruan, maka diperlukan banyak data, masing-masing *training sample* untuk tiap kelas tutupan lahan dapat dilihat pada Tabel 4.1. Jumlah sampel tutupan lahan untuk tiap kelas.

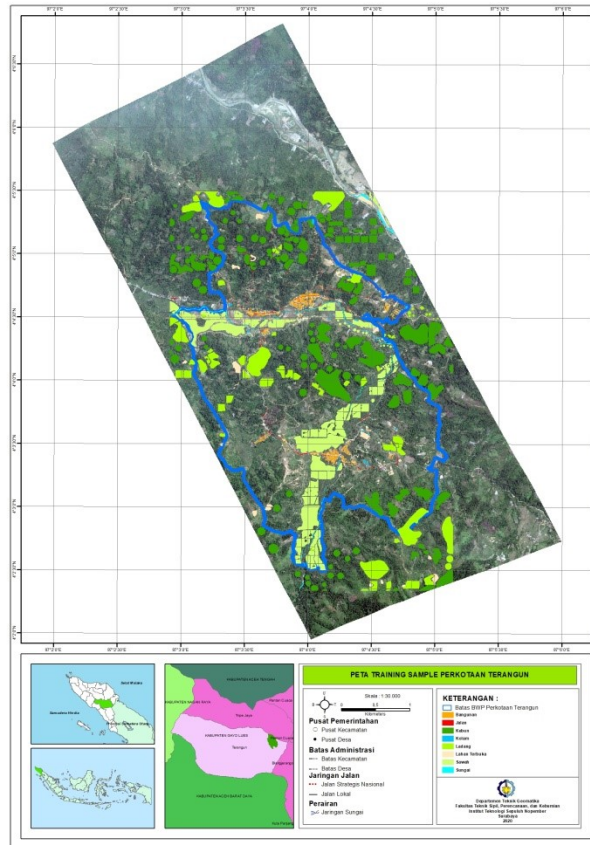
Tabel 4.1. Jumlah sampel Tutupan Lahan untuk Tiap kelas Tutupan Lahan

No	Kelas Tutupan Lahan	Jumlah Sampel	Jumlah Pikel
1	Bangunan	981	507.155
2	Jalan	93	204.488
3	Lahan Terbuka	40	376.617
4	Sungai	71	203.716
5	Sawah	130	4.259.372
6	Ladang	68	3.719.385
7	Kolam	41	74.964
8	Kebun	153	7.574.063
<b>Total</b>		<b>1577</b>	<b>16.919.760</b>

Jumlah total *training sample* yaitu sebanyak 1577 lokasi. Jumlah sampel terbanyak yaitu bangunan sebanyak 981 lokasi. Banyaknya sampel bangunan karena sampel dibuat tiap persil dari bangunan tersebut, disusul kebun sebanyak 153 lokasi dan sawah sebanyak 130 lokasi. Jumlah sampel kebun dan sawah juga relatif banyak karena pada lokasi penelitian banyak hamparan sawah dan kebun yang dijadikan masyarakat sebagai area bercocok tanam dan berkebun sereh wangi. Sedangkan sampel yang paling sedikit yaitu kolam, kolam diwilayah ini dijadikan sebagai tempat budidaya ikan air tawar dan jumlahnya yang digunakan sebagai sampel sebanyak 41 sampel.

Untuk melakukan klasifikasi tutupan lahan dengan model *deep learning* maka *training sample* dan citra satelit WorldView-2 level 2A harus dikonversi kedalam format data *deep learning*. Citra satelit WorldView-2 level 2A dan

sebaran dari *training sample* pada lokasi penelitian dapat dilihat pada Gambar 4.3. Sebaran *training sampel*.



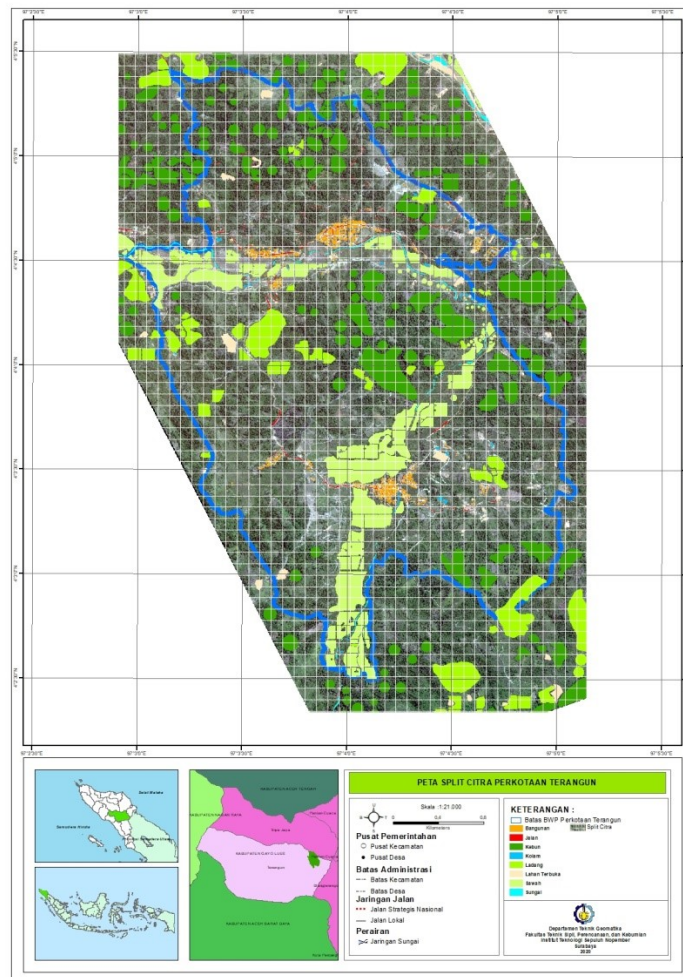
Gambar 4.3. Sebaran *training sample*

#### 4.5. Konversi Data Citra dan *Training Sample* Menjadi Data *Deep Learning*

Data citra satelit WorldView-2 level 2A yang digunakan dalam penelitian ini berukuran 4,1 km x 5,8 km atau 4100 m x 5800 m. Dari panjang dan lebar citra satelit tersebut maka memiliki luas 2.378 hektar. Dari luas tersebut mempunyai piksel 8.252 piksel x 11.639 piksel atau sebanyak 96.045.028 piksel. Untuk menjalankan proses *training* dalam satu kali proses tentu membutuhkan sumber daya komputer yang mempunyai spesifikasi sangat tinggi.

Untuk meringankan beban komputer dalam melakukan proses *training* dalam sebuah jaringan saraf tiruan, maka citra satelit WorldView-2 level 2A tersebut harus dipecahkan (*split*) menjadi bagian-bagian kecil (*chip*) sehingga

dapat dijalankan pada komputer yang mempunyai spesifikasi standar. Dalam melakukan proses *split* ukuran dari citra tersebut ditentukan ukurannya yaitu 128 piksel x 128 piksel.

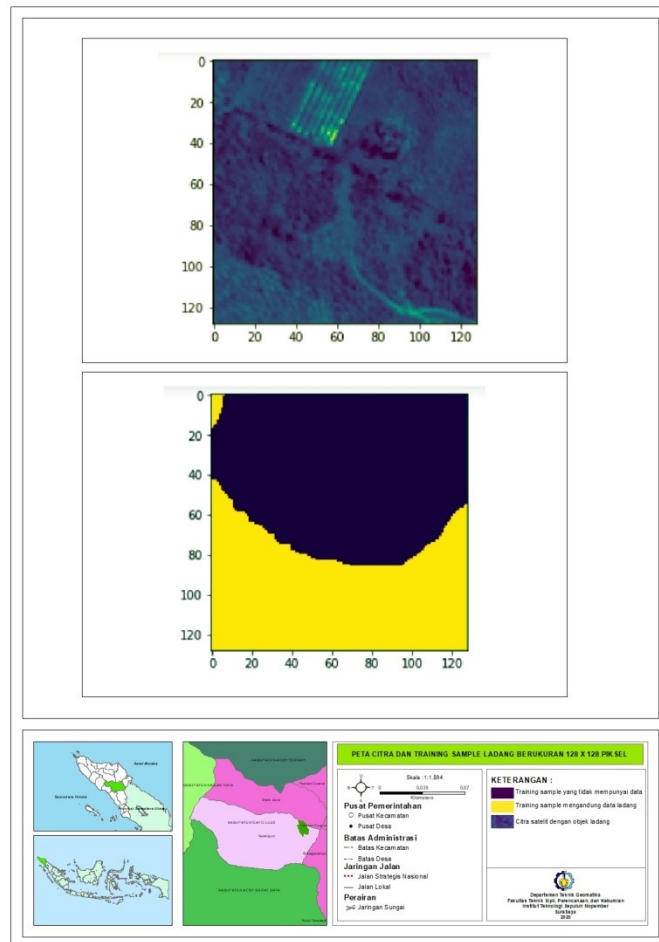


Gambar 4.4. Proses *Split* Citra Satelit WorldView-2

Selain ukuran, langkah (*stride*) juga harus ditentukan agar dalam proses *split* ini citra dapat terbagi dalam ukuran dan jarak yang sama. *Stride* berukuran 64 piksel x 64 piksel diterapkan dalam proses *split* ini, sehingga setiap jarak 64 piksel secara horizontal dan vertikal maka akan menghasilkan satu *chip* citra satelit worldview-2 level 2A berukuran 128x128 piksel.

Hal lain yang harus diperhatikan dalam melakukan konversi data citra menjadi format *deep learning* yaitu format output dari nama metadata, karena ada

lima pilihan formatnya yaitu *KITTI Rectangles*, *PASCAL VOC Rectangles*, *Classified Tiles* (kelas *training sample* dari peta), *RCNN Mask*, dan *Labeled Tiles*. Format *output metadata* yang digunakan dalam penelitian ini yaitu *Classified Tiles*. Format ini akan menghasilkan satu citra dan satu kelas *training sample* untuk tiap lokasi yang saling tumpang tindih.



Gambar 4.5. Peta citra dan *training sample* ladang berukuran 128x128 piksel

*Chip* yang dihasilkan merupakan data yang saling tumpang tindih (*overlay*) antara data citra satelit WorldView-2 level 2A dan data *training sampel*. Jumlah *chip* yang dihasilkan selain berukuran sama juga mempunyai jumlah yang sama antara *chip* citra satelit dan *chip training sample*. Jumlah *chip* berukuran 128x128 piksel berjumlah 18.781 data untuk data citra satelit dan 18.781 untuk *training sample*. Data yang sudah berukuran 128x128 piksel ini

akan digunakan untuk proses *training* dalam jaringan saraf tiruan untuk mendapatkan model yang dapat mengenali kelas tutupan lahan yang telah ditetapkan.

#### 4.6. Konversi Kelas Tutupan Lahan Kedalam Format HDF5

Untuk memudahkan dalam proses *training* maka data yang sudah dilakukan proses *split* dikonversikan menjadi format *Hierarchy Data Format 5* (HDF5). Format data HDF5 dapat menyimpan semua bentuk data dalam satu file, adapun data yang dapat disimpan dalam format ini berupa gambar, teks, angka, dan lainnya.

Data yang akan dikonversi kedalam format HDF5 harus diseleksi terlebih dahulu agar data yang akan dikonversi benar-benar data yang dapat memberikan informasi dalam proses *training*. Kategori data yang akan dikonversi kedalam format HDF5 meliputi :

- a. Data yang tidak ada informasi (*null*).
- b. Data yang mempunyai informasi.

Untuk data yang tidak ada informasi maka tidak dikonversikan kedalam format HDF5, hanya data yang memiliki informasi saja yang dikonversi. Untuk data yang memiliki informasi dikelompokkan menjadi dua kategori meliputi:

- a. Data berisi informasi tetapi mengandung *nodata (null)*.
- b. Data berisi informasi tetapi tidak mengandung *nodata (null)*.

Data yang memiliki informasi tetapi masih mengandung *nodata (null)* maka dikelompokkan lagi menjadi dua kategori, meliputi:

- a. Data berisi informasi lebih dari 50 persen
- b. Data berisi informasi kurang dari 50 persen

Data yang memiliki informasi, tetapi masih mengandung informasi kurang dari 50 persen akan dikeluarkan dari proses *training*. Terkait dengan jumlah tutupan lahan sesuai kategori sebagaimana tersebut diatas dapat dilihat pada Tabel 4.2. kategori data *deep learning*.

Tabel 4.2. Kategori Data *Deep Learning*

No	Kategori Data	Jumlah Data	Total
1	Data yang tidak berisi informasi	8754	18.781
2	Data yang berisi informasi	10027	
3	Chip yang berisi informasi, dan berisi <i>nodata</i>	9386	10.027
4	Chip yang berisi informasi, dan tidak berisi <i>nodata</i>	641	
5	Chip yang berisi informasi < 50%	6348	9.386
6	Chip yang berisi informasi > 50%	3038	
<b>Total</b>		<b>38194</b>	<b>38.194</b>

Tutupan lahan yang akan digunakan untuk melakukan training meliputi :

Tabel 4.3. Kelas Tutupan Lahan yang akan di *Training*

No	Tutupan lahan yang akan di <i>training</i>
1	<i>Background</i>
2	Bangunan
3	Jalan
4	Lahan Terbuka
5	Sungai
6	Sawah
7	Ladang
8	Kolam

9	Kebun
---	-------

Jumlah kelas yang digunakan untuk melakukan training dapat dilihat pada Tabel 4.4. Jumlah sampel kelas tutupan lahan.

Tabel 4.4. Jumlah Sampel Kelas Tutupan Lahan

No	Tutupan lahan yang akan di <i>training</i>	Jumlah Sampel	Jumlah <i>Channel</i>	Lebar (Piksel)	Tinggi (Piksel)
1	Bangunan	531	5	128	128
2	Jalan	92	5	128	128
3	Lahan Terbuka	99	5	128	128
4	Sungai	125	5	128	128
5	Sawah	1090	5	128	128
6	Ladang	862	5	128	128
7	Kolam	48	5	128	128
8	Kebun	1715	5	128	128

Untuk proses analisis maka format data dari HDF5 dikonversikan menjadi format data numerik yaitu format numerik python (*numpy*). Susunan data dalam format *numpy* dapat dilihat pada Tabel 4.5. Susunan format data *numpy*.

Tabel 4.5. Susunan Format Data *Numpy*

No	Tutupan lahan yang akan di <i>training</i>	Jumlah Sampel	Jumlah <i>Channel</i>	Lebar (Piksel)	Tinggi (Piksel)
1	Bangunan	531	5	128	128
2	Jalan	92	5	128	128
3	Lahan Terbuka	99	5	128	128
4	Sungai	125	5	128	128
5	Sawah	1090	5	128	128
6	Ladang	862	5	128	128
7	Kolam	48	5	128	128
8	Kebun	1715	5	128	128

Setelah susunan data *Numpy* tersusun maka harus dilakukan proses pergantian sumbu (*axis*) dari susunan format *Numpy*. Hal ini diperlukan agar susunan dari data berurutan yaitu (jumlah\_sampel, lebar\_citra, tinggi\_citra, jumlah\_band) Setelah dilakukan proses *roolaxis* maka susunannya berubah seperti pada Tabel 4.6. Susunan format *Numpy* setelah dilakukan proses *roolaxis*.

Tabel 4.6. Susunan Format Data *Numpy* Setelah Proses *Roolaxis*

No	Tutupan lahan yang akan di <i>training</i>	Sampel	Panjang (Piksel)	Lebar (Piksel)	<i>Channel</i>
1	Bangunan	531	128	128	5
2	Jalan	92	128	128	5
3	Lahan Terbuka	99	128	128	5
4	Sungai	125	128	128	5
5	Sawah	1090	128	128	5



6	Ladang	862	128	128	5
7	Kolam	48	128	128	5
8	Kebun	1715	128	128	5

Setelah proses *roolaxis* terhadap format data *numpy* selesai, maka perlu dilakukan proses pemisahan data. Data citra dan data kelas tutupan lahan dipisah sehingga berdiri sendiri masing-masing data tersebut. Data yang sudah dipisah tersebut dapat dilihat pada Tabel 4.7. Pemisahan data citra dan data kelas tutupan lahan.

Tabel 4.7. Pemisahan Data Citra dan Kelas Tutupan Lahan

No	Tutupan lahan yang akan di <i>training</i>	Sampel	Panjang (Piksel)	Lebar (Piksel)	<i>Channel</i>
1	Bangunan	531	128	128	5
	Citra	531	128	128	4
	<i>Training Sample</i>	531	128	128	1
2	Jalan	92	128	128	5
	Citra	92	128	128	4
	<i>Training Sample</i>	92	128	128	1
3	Lahan Terbuka	99	128	128	5
	Citra	99	128	128	4
	<i>Training Sample</i>	99	128	128	1
4	Sungai	125	128	128	5
	Citra	125	128	128	4
	<i>Training Sample</i>	125	128	128	1

5	Sawah	1090	128	128	5
	Citra	1090	128	128	4
	<i>Training Sample</i>	1090	128	128	1
6	Ladang	862	128	128	5
	Citra	862	128	128	4
	<i>Training Sample</i>	862	128	128	1
7	Kolam	48	128	128	5
	Citra	48	128	128	4
	<i>Training Sample</i>	48	128	128	1
8	Kebun	1715	128	128	5
	Citra	1715	128	128	4
	<i>Training Sample</i>	1715	128	128	1

Setelah memisahkan data citra dan data tutupan lahan maka tahapan selanjutnya dilakukan proses normalisasi data tiap tutupan lahan, sehingga data citra disusun sebagaimana tersusun pada tahap sebelumnya. Susunan data yang sudah dilakukan proses normalisasi dapat dilihat pada Tabel 4.8. Normalisasi data citra.

Tabel 4.8. Normalisasi Data Citra

<b>No</b>	<b>Tutupan lahan yang akan di <i>training</i></b>	<b>Jumlah Sampel</b>	<b>Panjang (Piksel)</b>	<b>Lebar (Piksel)</b>	<b>Jumlah <i>Channel</i></b>
1	Citra	531	128	128	4
2	Citra	92	128	128	4
3	Citra	99	128	128	4

4	Citra	125	128	128	4
5	Citra	1090	128	128	4
6	Citra	862	128	128	4
7	Citra	48	128	128	4
8	Citra	1715	128	128	4

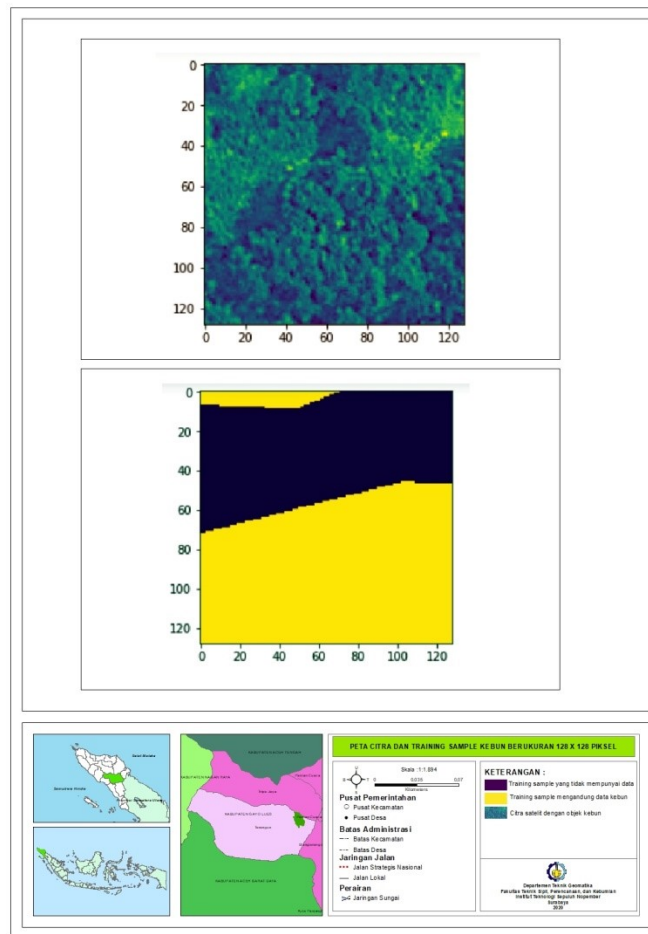
Tabel 4.9. Normalisasi Data *Training Sample*

<b>No</b>	<b>Data</b>	<b>Jumlah Sampel</b>	<b>Panjang (Piksel)</b>	<b>Lebar (Piksel)</b>	<b>Jumlah Channel</b>
1	<i>Training sample</i>	531	128	128	1
2	<i>Training sample</i>	92	128	128	1
3	<i>Training sample</i>	99	128	128	1
4	<i>Training sample</i>	125	128	128	1
5	<i>Training sample</i>	1090	128	128	1
6	<i>Training sample</i>	862	128	128	1
7	<i>Training sample</i>	48	128	128	1
8	<i>Training sample</i>	1715	128	128	1

Setelah proses normalisasi selesai dilakukan, maka kelas tutupan lahan digabung kembali kedalam data citra. Susunan data citra dan data kelas tutupan lahan dapat dilihat pada Tabel 4.10. Gabungan data citra dan kelas tutupan lahan.

Tabel 4.10. Gabungan Data Citra dan Kelas Tutupan Lahan

<b>No</b>	<b>Tutupan lahan yang akan di <i>training</i></b>	<b>Jumlah Sampel</b>	<b>Panjang (Piksel)</b>	<b>Lebar (Piksel)</b>	<b><i>Jumlah Channel</i></b>
1	Citra	531	128	128	4
	<i>Training Sample</i>	531	128	128	1
	Bangunan	531	128	128	5
2	Citra	92	128	128	4
	<i>Training Sample</i>	92	128	128	1
	Jalan	92	128	128	5
3	Citra	99	128	128	4
	<i>Training Sample</i>	99	128	128	1
	Lahan Terbuka	99	128	128	5
4	Citra	125	128	128	4
	<i>Training Sample</i>	125	128	128	1
	Sungai	125	128	128	5
5	Citra	1090	128	128	4
	<i>Training Sample</i>	1090	128	128	1
	Sawah	1090	128	128	5
6	Citra	862	128	128	4
	<i>Training Sample</i>	862	128	128	1
	Ladang	862	128	128	5
7	Citra	48	128	128	4
	<i>Training Sample</i>	48	128	128	1
	Kolam	48	128	128	5
8	Citra	1715	128	128	4
	<i>Training Sample</i>	1715	128	128	1
	Kebun	1715	128	128	5



Gambar 4.6. Data citra dan kelas tutupan lahan representasi kebun

#### 4.7. Konversi *Training Sample* dan Citra Numpy Menjadi HDF5

Data kelas tutupan lahan dan citra dalam format *Numpy* kemudian dilakukan proses konversi kedalam format HDF5, data dalam format ini akan digunakan untuk proses training.

Bangunan_HDF_.h5	06/01/2020 16:33	H5 File	339.842 KB
Jalan_HDF_.h5	06/01/2020 16:33	H5 File	58.882 KB
Kebun_HDF_.h5	06/01/2020 16:33	H5 File	1.097.602 KB
Kolam_HDF_.h5	06/01/2020 16:33	H5 File	30.722 KB
Ladang_HDF_.h5	06/01/2020 16:33	H5 File	551.682 KB
Lahan_Terbuka_HDF_.h5	06/01/2020 16:33	H5 File	63.362 KB
Sawah_HDF_.h5	06/01/2020 16:33	H5 File	697.602 KB
Sungai_HDF_.h5	06/01/2020 16:33	H5 File	80.002 KB

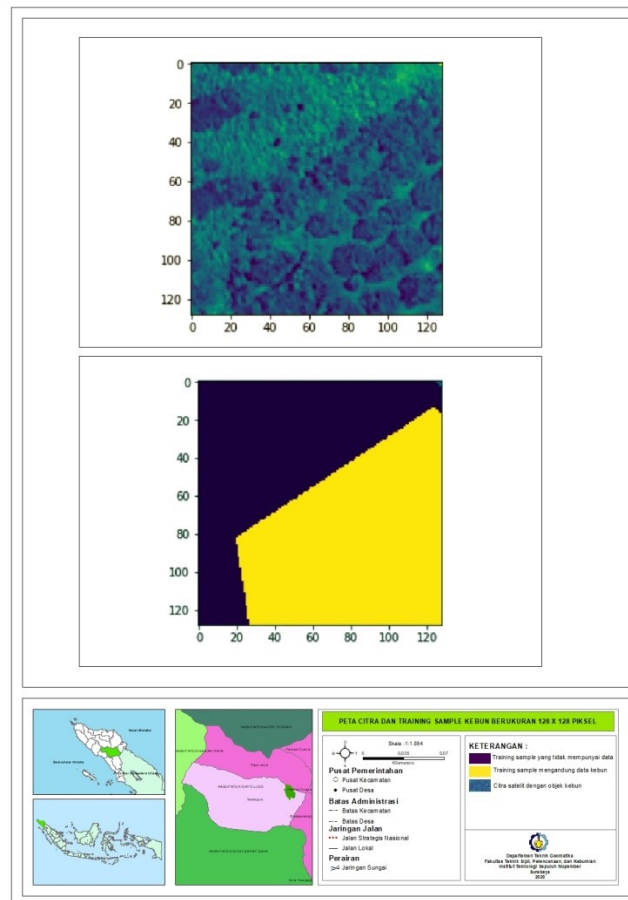
Gambar 4.7. File kelas tutupan lahan dalam format HDF5

Susunan format data HDF5 sama seperti susunan data format *Numpy* sebelum dilakukan proses *roolingaxis*.

Tabel 4.11. Susunan Data Citra dan Tutupan Lahan Dalam Format Data HDF5

<b>No</b>	<b>Tutupan lahan yang akan di <i>training</i></b>	<b>Jumlah Sampel</b>	<b>Panjang (Piksel)</b>	<b>Lebar (Piksel)</b>	<b>Jumlah <i>Channel</i></b>
1	Bangunan	531	128	128	5
2	Jalan	92	128	128	5
3	Lahan Terbuka	99	128	128	5
4	Sungai	125	128	128	5
5	Sawah	1090	128	128	5
6	Ladang	862	128	128	5
7	Kolam	48	128	128	5
8	Kebun	1715	128	128	5

Setelah data citra dan tutupan lahan dikonversi kedalam format HDF5 maka data tersebut diperiksa kembali untuk memastikan bahwa data tersebut valid dan dapat digunakan untuk proses *training*.



Gambar 4.8. Tampilan citra satelit dan kelas tutupan lahan representasi dari kebun

#### 4.8. Melatih Model U-Net

Model U-Net ini mampu mengenali objek-objek gambar dengan baik. Oleh sebab itu, model ini diterapkan dalam penelitian ini agar objek yang ada pada citra satelit WorldView-2 level 2A dapat dikenali dengan jelas. Sebelum proses training data dengan model U-Net dilakukan, maka perlu dilakukan proses augmentasi.

Tabel 4.12. Data Kelas Tutupan Lahan yang akan Dilakukan Augmentasi

No	Tutupan lahan yang akan di <i>training</i>	Jumlah Sampel	Panjang (Piksel)	Lebar (Piksel)	Jumlah <i>Channel</i>
1	Bangunan	531	128	128	5
2	Jalan	92	128	128	5
3	Lahan Terbuka	99	128	128	5
4	Sungai	125	128	128	5
5	Sawah	1090	128	128	5
6	Ladang	862	128	128	5
7	Kolam	48	128	128	5
8	Kebun	1715	128	128	5

#### 4.9. Augmentasi Kelas Tutupan Lahan

Proses augmentasi diperlukan dalam klasifikasi data citra digital agar sampel yang digunakan dalam pengolahan data tersebut dapat ditingkatkan berdasarkan data awal tanpa mengurangi karakteristik citra tersebut secara signifikan. Model augmentasi yang digunakan meliputi model putar (*rotation*), model geser (*shear*), *shift*, dan pembesaran (*zoom*).

Augmentasi metode rotasi yaitu melakukan proses pemutaran gambar sebesar  $-180^\circ$ ,  $-90^\circ$ ,  $90^\circ$ , dan  $180^\circ$ . sedangkan metode geser (*shear*) melakukan proses penggeseran gambar sebesar 0.1, 0.2, 0.3, 0.4, dan 0.5. Metode augmentasi *shift* melakukan proses penggeseran 5 kolom dan 5 baris. Sedangkan metode augmentasi pembesaran (*zoom*) melakukan pembesaran sebesar 0.05 dan 0.45.

Hasil dari proses augmentasi gambar sampel maka terjadi peningkatan jumlah sampel yang signifikan tergantung pada faktor nilai augmentasi yang diberikan. Adapun faktor nilai augmentasi yang diberikan yaitu bangunan = 0, jalan = 5, lahan terbuka = 5, sungai = 4, sawah = 0, ladang = 0, kolam = 10, dan



kebun = 0. Peningkatan jumlah sampel dapat dilihat pada Tabel 4.13. Sampel sebelum dan sesudah augmentasi.

Kelas tutupan lahan sungai sebelum dilakukan proses augmentasi berjumlah 125 sampel, setelah dilakukan proses augmentasi dengan memberikan faktor pengali bernilai 5 maka jumlahnya berubah menjadi 2500 sampel.

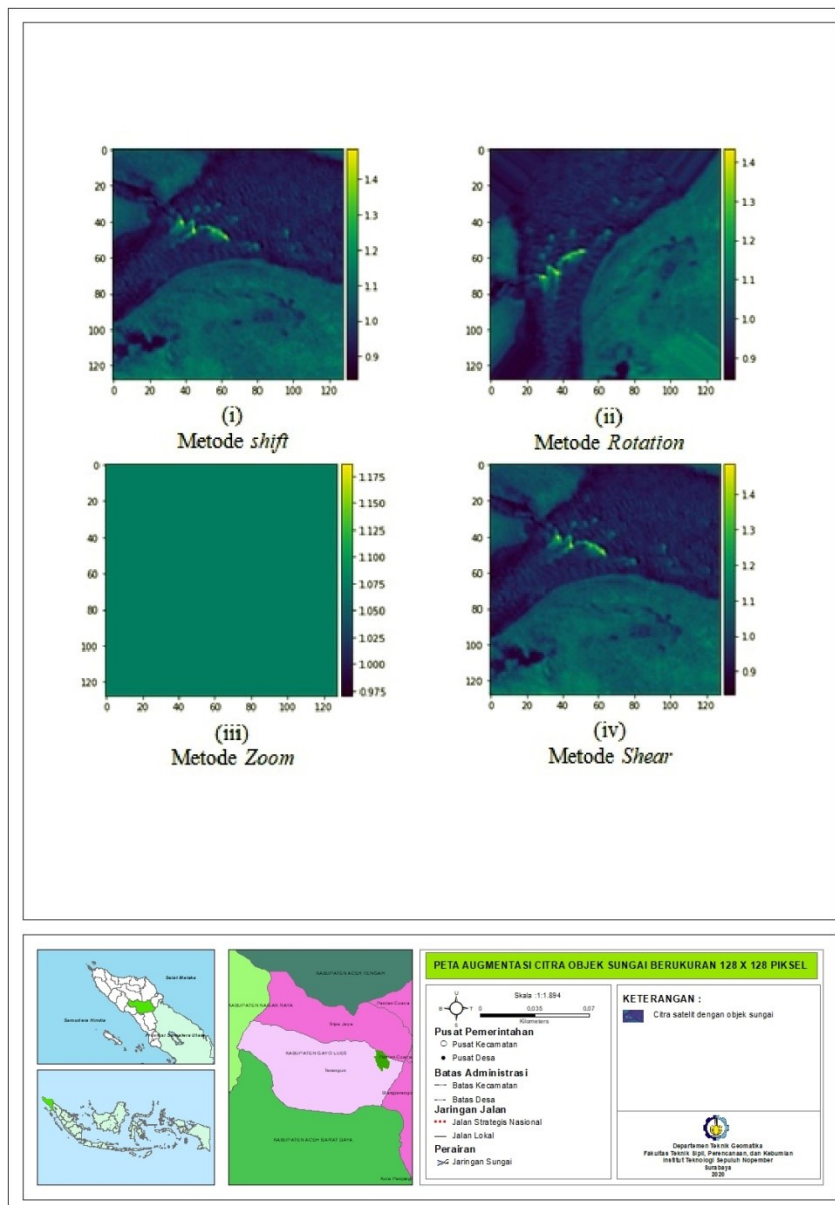
Hal yang sama juga dilakukan terhadap jalan dengan faktor pengalinya bernilai 4, maka jumlah sampel meningkat menjadi 2300 sampel dari sebelumnya hanya 92 sampel. Lahan terbuka terjadi peningkatan jumlah sampel dari 99 sampel menjadi 495 sampel dengan faktor pengalinya bernilai 5. Sampel terakhir yang dilakukan proses augmentasi yaitu kolam, yang pada dasarnya hanya berjumlah 48 sampel, dengan menggunakan faktor pengalinya 10 maka meningkat drastis menjadi 2400 sampel. Peningkatan empat kelas tutupan lahan yang dilakukan proses augmentasi dapat dilihat pada tabel 4.13. Jumlah data setelah proses augmentasi.

Tabel 4.13. Jumlah Data Sebelum dan Setelah Proses Augmentasi

No	Tutupan lahan yang akan di <i>training</i>	Jumlah Sampel Sebelum augmentasi	Jumlah Sampel Setelah augmentasi	Panjang (Piksel)	Lebar (Piksel)	Jumlah <i>Channel</i>
1	Bangunan	531	531	128	128	5
2	Jalan	92	2300	128	128	5
3	Lahan Terbuka	99	495	128	128	5
4	Sungai	125	2500	128	128	5
5	Sawah	1090	1090	128	128	5
6	Ladang	862	862	128	128	5
7	Kolam	48	2400	128	128	5

8	Kebun	1715	1715	128	128	5
Total		4562	11893			

Berikut ini merupakan contoh augmentasi dari objek sungai. Citra objek sungai yang telah dilakukan proses augmentasi dapat dilihat pada Tabel 4.13. Proses augmentasi citra.



Gambar 4.9. Proses augmentasi citra

#### 4.10. Menggabungkan File *Numpy*

Semua file *numpy* kelas tutupan lahan yang dianggap masih kurang dan telah dilakukan proses augmentasi harus disatukan kembali agar filenya tersusun dengan rapi dan dapat dibaca oleh program. Total semua sampel kelas tutupan lahan yaitu 7.695 sampel dapat dilihat pada Tabel 4.14. Gabungan file *numpy* kelas tutupan lahan yang masih kurang

Tabel 4.14. Gabungan File *Numpy* Kelas Tutupan Lahan yang Masih Kurang

No	Tutupan lahan yang akan di <i>training</i>	Jumlah Sampel Sebelum Augmentasi	Jumlah Sampel Setelah Augmentasi	Panjang (Piksel)	Lebar (Piksel)	Jumlah <i>Channel</i>
1	Jalan	92	2300	128	128	5
2	Lahan Terbuka	99	495	128	128	5
3	Sungai	125	2500	128	128	5
4	Kolam	48	2400	128	128	5
Total		364	7695			

Hal yang sama juga dilakukan terhadap file *numpy* yang kelas tutupan lahannya masuk dalam kategori banyak, walaupun tidak dilakukan proses augmentasi, tetapi tetap harus digabung ulang susunannya. Adapun jumlah total sampel kelas tutupan lahan yang tidak dilakukan proses augmentasi yaitu 4.198 sampel dapat dilihat pada Tabel 4.15. Gabungan file *numpy* kelas tutupan lahan yang banyak

Tabel 4.15. Gabungan File *Numpy* Kelas Tutupan Lahan yang Banyak

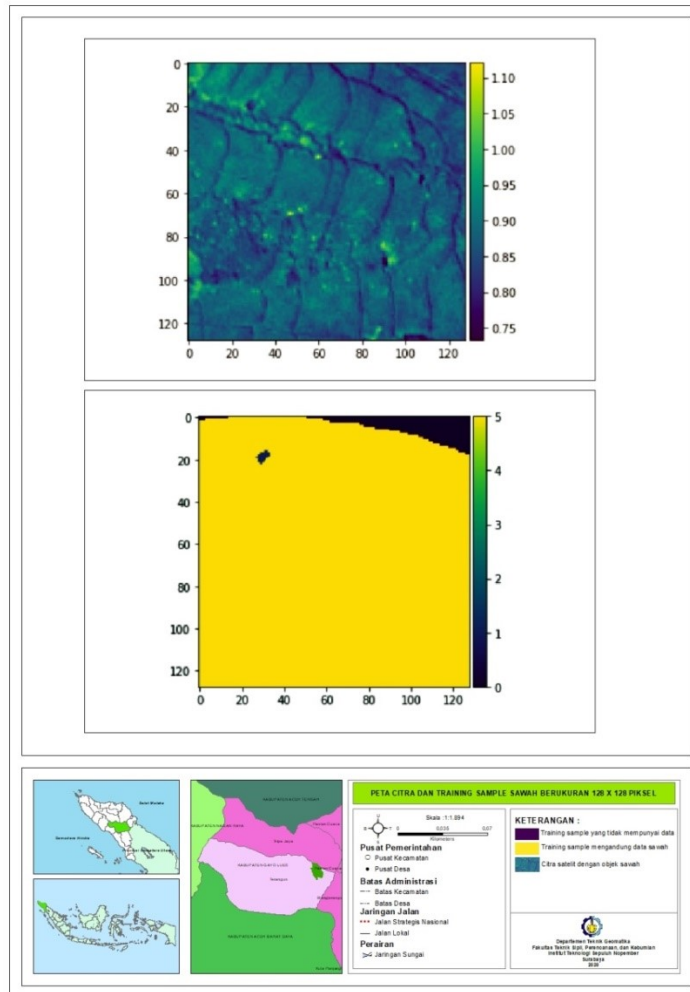
No	Tutupan lahan yang akan di <i>training</i>	Sebelum Augmentasi	Setelah Augmentasi	Panjang (Piksel)	Lebar (Piksel)	Jumlah <i>Channel</i>
1	Bangunan	531	531	128	128	5
2	Sawah	1090	1090	128	128	5
3	Ladang	862	862	128	128	5
4	Kebun	1715	1715	128	128	5
Total		4198	4198			

Setelah susunan file *numpy* yang kategori sedikit dan banyak telah digabung masing-masing sesuai kategori. Maka tahap berikutnya sampelnya harus digabung semuanya baik yang kategori sedikit maupun banyak, sehingga total sampel yang akan dilakukan training berubah dari 4.562 sampel menjadi 11.839 sampel. Jumlah sampelnya terjadi peningkatan hampir tiga kali lipat. Jumlah sampel yang sedikit maupun banyak dapat dilihat pada Tabel 4.16. Gabungan file *numpy* kelas tutupan lahan yang kurang dan banyak.

Tabel 4.16. Gabungan File *Numpy* Kelas Tutupan Lahan yang Kurang dan Banyak

No	Sampel	Jumlah Sampel
1	Tidak Augmentasi	4198
2	Augmentasi	7695
Total		11893

Berikut ini merupakan contoh data gabungan file *numpy* yang sudah digabungkan antara yang sudah dilakukan augmentasi maupun yang belum dilakukan augmentasi. Citra objek ditampilkan secara acak dapat dilihat pada Gambar 4.10. Citra dan kelas tutupan lahan gabungan file *numpy* representasi sawah.



Gambar 4.10. Citra dan kelas tutupan lahan gabungan file *numpy* representasi sawah

#### 4.11. Memisahkan File Citra dan Kelas Tutupan Lahan dari *Numpy Array*

Selanjutnya dilakukan pemisahan file *numpy array* dari data yang sudah tergabung antara citra dan kelas tutupan lahan. Hasil pemisahan file tersebut dapat dilihat pada Tabel 4.17. File citra yang sudah dipisahkan dari *numpy array*.

Tabel 4.17. File Citra yang Sudah Dipisahkan dari *Numpy*

No	Numpy Array	Jumlah Sampel	Panjang (Piksel)	Lebar (Piksel)	Jumlah Channel
1	Data	11893	128	128	5
2	Citra	11893	128	128	4

Hal yang sama juga dilakukan terhadap kelas tutupan lahan. Hasil pemisahan file tersebut dapat dilihat pada Tabel 4.18. File kelas tutupan lahan yang sudah dipisahkan dari *numpy array*.

Tabel 4.18. File Kelas Tutupan Lahan yang sudah Dipisahkan Dari *Numpy*

No	Numpy Array	Jumlah Sampel	Panjang (Piksel)	Lebar (Piksel)	Jumlah Channel
1	Data	11893	128	128	5
2	<i>Training sample</i>	11893	128	128	1

#### 4.12. Konversi File Numerik Citra dan Tutupan Lahan Menjadi bilangan Intiger

Dalam melakukan proses *training*, maka file *numpy* citra maupun kelas tutupan lahan harus dirubah formatnya dari bilangan *floating* menjadi bilangan bulat (*intiger*). Format bilangan tersebut dapat dilihat pada Tabel 4.19. Konversi bilangan data citra dan kelas tutupan lahan dari *floating* menjadi *intiger*.

Tabel 4.19. Konversi Bilangan Data Citra dan Kelas Tutupan Lahan dari Bilangan *Floating* Menjadi Bilangan *Intiger*

Bilangan Floating					Bilangan Intiger				
8.	8.	8.	8.	8.	8	8	8	8	8
8.	8.	8.	8.	8.	8	8	8	8	8
8.	8.	8.	8.	8.	8	8	8	8	8
8.	8.	8.	8.	8.	8	8	8	8	8
8.	8.	8.	8.	8.	8	8	8	8	8
8.	8.	8.	8.	8.	8	8	8	8	8

#### 4.13. Mencetak File *Numpy* Citra dan Kelas Tutupan Lahan kedalam Format HDF5

File *numpy* citra dan kelas tutupan lahan selanjutnya dikonversi menjadi HDF5. Setelah formatnya menjadi HDF5, maka dilakukan proses pembacaan data untuk melihat valid atau tidaknya data tersebut. Proses tersebut dilakukan untuk kedua file baik citra maupun kelas tutupan lahan dapat dilihat pada Tabel 4.20. File HDF5 citra dan kelas tutupan lahan

Tabel 4.20. File HDF5 Citra dan Kelas Tutupan Lahan

No	Numpy Array	Jumlah Sampel	Panjang (Piksel)	Lebar (Piksel)	Jumlah Channel
1	Citra	11893	128	128	4
2	<i>Training sample</i>	11893	128	128	1

#### 4.14. Konversi Kelas Tutupan Lahan Menjadi Vektor

File citra dan kelas tutupan lahan dalam format HDF5 selanjutnya dikonversi menjadi data vektor. Hal ini dilakukan untuk mengarahkan hasil *training* dan melihat kecenderungan dari hasil *training* tersebut mengarah ke 8 kelas tutupan lahan yang sudah ditentukan. Dalam hal ini dibuat 9 kelas untuk mewaspadaikan apabila ada kelas yang tidak dapat diklasifikasikan. Satu kelas tambahan ini nantinya akan berisi dengan kelas tutupan lahan yaitu *background*.

Tabel 4.21. Konversi Kelas Tutupan Lahan dalam Format Vektor

No	Numpy Array	Jumlah Sampel	Panjang (Piksel)	Lebar (Piksel)	Jumlah Channel
1	<i>Training sample</i>	11893	128	128	1
2	<i>Training sample Setelah dikonversi</i>	11893	128	128	9

#### 4.15. Mendefinisikan Parameter *Training* dan Mendefinisikan Jumlah Kelas Tutupan Lahan

Sebelum melakukan *training*, maka harus dipersiapkan parameter-parameter yang dibutuhkan dalam proses *training*. Adapun parameter tersebut yaitu ukuran citra yaitu lebarnya 128 piksel dan tingginya juga 128 piksel. Parameter selanjutnya yaitu jumlah *band* yang ingin diklasifikasi kelas tutupan lahan dan yang terakhir yaitu jumlah kelas tutupan lahan yang diinginkan.

Kelas tutupan lahan yang ditargetkan yaitu delapan kelas yang terdiri dari bangunan, jalan, lahan terbuka, sungai, sawah, ladang, kolam, dan kebun. Dikhawatirkan ada objek yang tidak dapat dilakukan klasifikasi maka jumlah kelas harus ditambah satu lagi yaitu *background*, sehingga total dari kelas tutupan lahan yaitu sembilan kelas.



Tabel 4.22. Jumlah Kelas Tutupan Lahan

No	Tutupan lahan yang akan di <i>training</i>
1	<i>Background</i>
2	Bangunan
3	Jalan
4	Lahan Terbuka
5	Sungai
6	Sawah
7	Ladang
8	Kolam
9	Kebun

#### 4.16. Membuat Arsitektur Model U-Net

Arsitektur model yang digunakan dalam model ini mengikuti arsitektur U-Net. Penggunaan arsitektur U-Net dalam proses *training* ini dengan harapan dapat melakukan segmentasi objek dengan baik. Dalam membuat arsitektur model U-Net maka harus ditentukan parameter seperti *convolution*, *activation*, *kernel*, *stride*, *padding*, *dropout*, dan *maxpooling*.

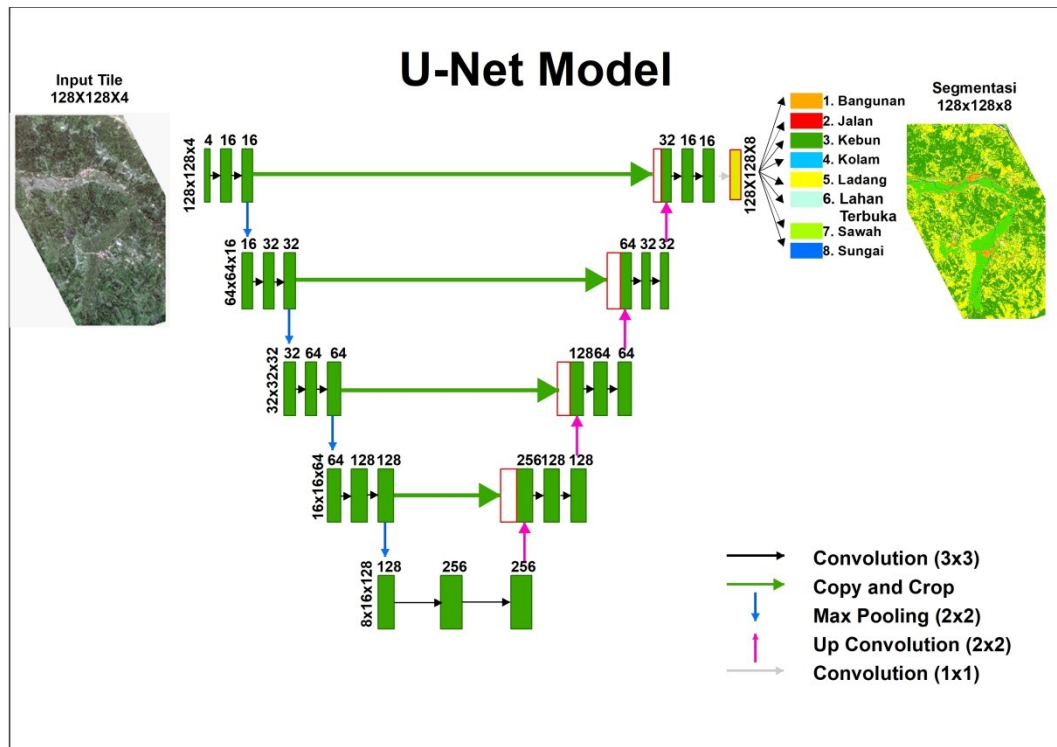
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 128, 128, 4)	0	
conv2d_1 (Conv2D)	(None, 128, 128, 16)	592	input_1[0][0]
dropout_1 (Dropout)	(None, 128, 128, 16)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 16)	2320	dropout_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 16)	0	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 64, 64, 32)	4640	max_pooling2d_1[0][0]
dropout_2 (Dropout)	(None, 64, 64, 32)	0	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 64, 64, 32)	9248	dropout_2[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 32)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 64)	18496	max_pooling2d_2[0][0]
dropout_3 (Dropout)	(None, 32, 32, 64)	0	conv2d_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 64)	36928	dropout_3[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 64)	0	conv2d_6[0][0]
conv2d_7 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_3[0][0]
dropout_4 (Dropout)	(None, 16, 16, 128)	0	conv2d_7[0][0]
conv2d_8 (Conv2D)	(None, 16, 16, 128)	147584	dropout_4[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 8, 8, 128)	0	conv2d_8[0][0]
conv2d_9 (Conv2D)	(None, 8, 8, 256)	295168	max_pooling2d_4[0][0]
dropout_5 (Dropout)	(None, 8, 8, 256)	0	conv2d_9[0][0]
conv2d_10 (Conv2D)	(None, 8, 8, 256)	590080	dropout_5[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 16, 16, 128)	131200	conv2d_10[0][0]
concatenate_1 (Concatenate)	(None, 16, 16, 256)	0	conv2d_transpose_1[0][0] conv2d_8[0][0]
conv2d_11 (Conv2D)	(None, 16, 16, 128)	295040	concatenate_1[0][0]
dropout_6 (Dropout)	(None, 16, 16, 128)	0	conv2d_11[0][0]
conv2d_12 (Conv2D)	(None, 16, 16, 128)	147584	dropout_6[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 32, 32, 64)	32832	conv2d_12[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 128)	0	conv2d_transpose_2[0][0] conv2d_6[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 64)	73792	concatenate_2[0][0]
dropout_7 (Dropout)	(None, 32, 32, 64)	0	conv2d_13[0][0]
conv2d_14 (Conv2D)	(None, 32, 32, 64)	36928	dropout_7[0][0]
conv2d_transpose_3 (Conv2DTrans	(None, 64, 64, 32)	8224	conv2d_14[0][0]
concatenate_3 (Concatenate)	(None, 64, 64, 64)	0	conv2d_transpose_3[0][0] conv2d_4[0][0]
conv2d_15 (Conv2D)	(None, 64, 64, 32)	18464	concatenate_3[0][0]
dropout_8 (Dropout)	(None, 64, 64, 32)	0	conv2d_15[0][0]

conv2d_16 (Conv2D)	(None, 64, 64, 32)	9248	dropout_8[0][0]
conv2d_transpose_4 (Conv2DTrans	(None, 128, 128, 16)	2064	conv2d_16[0][0]
concatenate_4 (Concatenate)	(None, 128, 128, 32)	0	conv2d_transpose_4[0][0] conv2d_2[0][0]
conv2d_17 (Conv2D)	(None, 128, 128, 16)	4624	concatenate_4[0][0]
dropout_9 (Dropout)	(None, 128, 128, 16)	0	conv2d_17[0][0]
conv2d_18 (Conv2D)	(None, 128, 128, 16)	2320	dropout_9[0][0]
conv2d_19 (Conv2D)	(None, 128, 128, 9)	153	conv2d_18[0][0]
=====			
Total params: 1,941,385			
Trainable params: 1,941,385			
Non-trainable params: 0			

Gambar 4.11. Jumlah parameter tiap proses *convolution*

Citra yang dimasukkan kedalam model U-Net untuk dilakukan *training* berukuran 128x128 piksel dengan jumlah *channel* 4. Proses *convolution* dilakukan sebanyak 19 kali. *Convolution* menggunakan *kernel* berukuran 3x3 piksel dengan ketebalan kernel ditentukan sebanyak 16, 32, 64, dan 128 lapisan. Fungsi aktivasi yang digunakan pada proses *training* ini yaitu ReLU. *Kernel\_initializer* yang digunakan adalah *he\_normal*.

Sedangkan *padding* menggunakan *same*, artinya tidak ada penambahan nilai nol pada sekeliling gambar, karena ukuran 128x128 piksel dapat terserap semua informasi dengan *stride* 2x2 piksel. Nilai *dropout* yang diberikan sebesar 0,1, dan *pooling layer* menggunakan *maxpooling* dengan ukuran 2x2 piksel. Total parameter yang dihasilkan dalam model U-Net ini yaitu 1.941.385 parameter. Arsitektur model U-Net dapat dilihat pada Gambar 4.12. Arsitektur model U-Net.



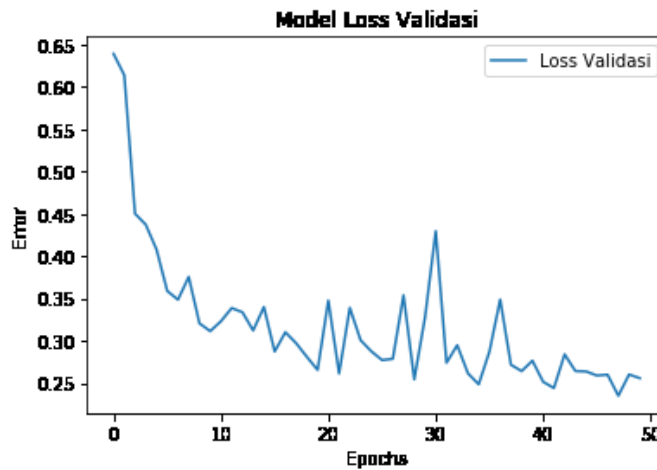
Gambar 4.12. Arsitektur model U-Net

#### 4.17. Visualisasi Model *Deep Learning*

Model *deep learning* yang terbentuk diperoleh dari hasil *training* yang dilakukan sebanyak 50 *epoch*. Akurasi yang diukur untuk melihat kemampuan sebuah model meliputi *loss validation*, *mean IoU validation*, *loss training*, dan *mean IoU training*. dengan tingkat akurasi *Intersection over Union* (IoU).

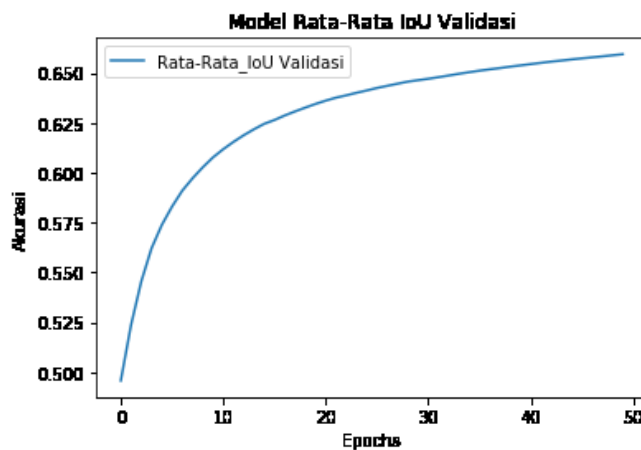
Dalam melakukan proses training ini terdapat data yaitu 11.893 data, tetapi tidak semua data dilibatkan dalam proses training ini, dari 11.893 data yang digunakan untuk training yaitu sebanyak 10.703 data sedangkan sisa dari data tersebut sebanyak 1.190 data digunakan untuk validasi data hasil training agar diketahui tingkat akurasi dari model yang dibentuk.

Untuk nilai *loss validation* menurun seiring dengan peningkatan *epoch* yang dilakukan. Nilai *loss* terus berkurang dari 0,6387 hingga angka 0,23509. Berkurangnya nilai *loss* berarti meningkatnya nilai akurasi dari data yang digunakan untuk validasi. Akurasi dari data validasi yaitu meningkat menuju angka 0,76491. Grafik dari nilai *loss* dapat dilihat pada Gambar 4.13. Grafik *loss validation*.



Gambar 4.13. Grafik model *loss validation*

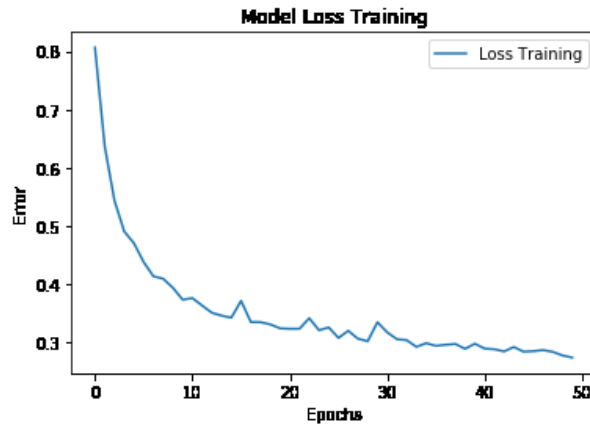
Nilai rata-rata untuk IoU data validasi juga meningkat seiring meningkatnya *epoch* yang dilakukan. Nilai rata IoU data validasi awalnya hanya bernilai 0,4957 kemudian meningkat secara perlahan mengikuti peningkatan *epoch*. Pada saat *epoch* berakhir maka jumlah nilai rata-rata IoU data validasi meningkat hingga 0,6594. Grafik peningkatan nilai rata-rata IoU validasi dapat dilihat pada Gambar 4.14. Grafik model *mean IoU validation*.



Gambar 4.14. Grafik model *mean IoU validation*.

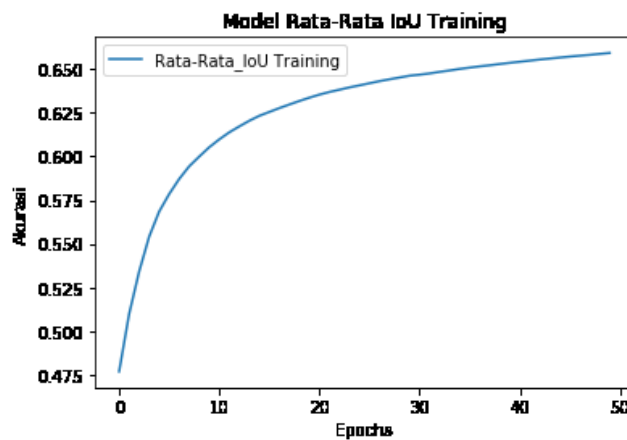
Selain data validasi yang diukur akurasi modelnya, maka hal yang sama juga dilakukan terhadap data training. Data training juga diukur tingkat *loss* yang terbentuk seiring meningkatnya *epoch*. *Loss* awal terbentuk bernilai 0,8065, *loss*

*training* berkurang hingga 0,2732 pada *epoch* ke 50. Gambar 4.15. Grafik *loss training* menunjukkan penurunan dari *loss* model yang dibentuk.



Gambar 4.15. Grafik *loss training*

Nilai rata-rata *IoU training* menunjukkan tingkat kecerdasan dari model dalam mengenali sebuah objek yang ada pada data *training*. Pada awal dilakukan proses *training*, maka nilai rata-rata dari *IoU training* yaitu sebesar 0,4768, nilai ini terus meningkat beriringan dengan peningkatan jumlah *epoch* nilai tertinggi dari *IoU training* pada *epoch* ke 50 yaitu 0,6592. Gambar 4.16. Grafik rata-rata *IoU training* menunjukkan peningkatan nilai rata-rata kecerdasan model dalam mengenali objek pada data training.



Gambar 4.16. Grafik rata-rata *IoU training* menunjukkan penurunan dari *loss* model yang dibentuk.

Dari 50 *epoch* yang dilakukan untuk membuat model, maka hanya 17 kali terjadi peningkatan akurasi (*loss validation*), sedangkan 33 kali tidak terjadi peningkatan akurasi. Adapun informasi peningkatan akurasi tiap *epoch* dapat dilihat pada Tabel 4.23. Perkembangan akurasi tiap *epoch*.

Tabel 4.23. Perkembangan Nilai Akurasi Tiap *Epoch*

<i>Epoch</i>	<b>Peningkatan Akurasi (<i>loss validation</i>)</b>	
	<b>dari</b>	<b>ke</b>
1	tidak diketahui	0.63870
2	0.63870	0.61348
3	0.61348	0.45005
4	0.45005	0.43704
5	0.43704	0.40782
6	0.40782	0.35881
7	0.35881	0.34851
9	0.34851	0.32060
10	0.32060	0.31135
16	0.31135	0.28729
19	0.28729	0.28105
20	0.28105	0.26562
22	0.26562	0.26140
29	0.26140	0.25441
35	0.25441	0.24858
42	0.24858	0.24425

48	0.24425	0.23509
----	---------	---------

Pada awal proses training, karena *loss* pada model yang sangat besar maka akurasi dengan mudahnya dapat ditingkatkan. Proses meningkatnya akurasi terjadi pada *epoch* 1-7 dimana *loss* mengecil dari nilai yang tidak diketahui menjadi 0,34851. Pada *epoch* ke-8 model tidak dapat mengenali objek lebih baik dari sebelumnya sehingga tidak meningkat akurasi pada tahap ini.

pada *epoch* 9-10 model kembali dapat mengenali objek-objek yang ada pada citra sehingga kecerdasannya meningkat dari 0,34851 menjadi 0,31135. Pada *epoch* ke-11, model kembali tidak dapat mengenali objek lebih baik dari sebelumnya sehingga dilakukan proses *backpropagation* ulang hingga *epoch* 15 juga tidak terjadi penurunan nilai *loss* nya.

Peningkatan akurasi baru terjadi pada *epoch* ke-16, *loss* menurun dari 0,31135 menjadi 0,28729. Kemudian pada *epoch* ke 17-18 kembali terjadi hal sebelumnya sehingga tidak dapat mengenali objek lebih baik. *Epoch* ke-19 model kembali dapat menurunkan *loss* setelah melakukan proses *propagation* dan *backpropagation* hingga nilai *loss* menurun dari 0,28729 menjadi 0,28175.

Proses *propagation* dan *backpropagation* terus dilakukan hingga *epoch* ke-50. Dalam proses ini kadang terjadi peningkatan akurasi, kadang juga tidak. Peningkatan akurasi terjadi pada *epoch* antara 20-30 sebanyak 3 kali, yang terjadi pada *epoch* ke 20, 22, dan 29, sehingga *loss* menurun dari 0,28105 menjadi 0,25441. Pada *range epoch* 30-40 terjadi satu kali peningkatan akurasi pada *epoch* ke-35, dimana nilai *loss* menjadi 0,24858. Sedangkan pada *range* 40-50 terjadi dua kali peningkatan akurasi yaitu pada *epoch* 42 dan *epoch* 48.

Hingga berakhirnya *epoch* ke-50 nilai *loss* nya bertahan pada angka 0,23509. Model dengan nilai *loss* 0,23509 ini akan digunakan untuk melakukan klasifikasi citra satelit WorldView-2 level 2A untuk mendapatkan kelas tutupan lahan yang telah ditentukan yang terdiri dari bangunan, jalan, kebun, kolam, ladang, lahan terbuka, sawah, dan sungai.

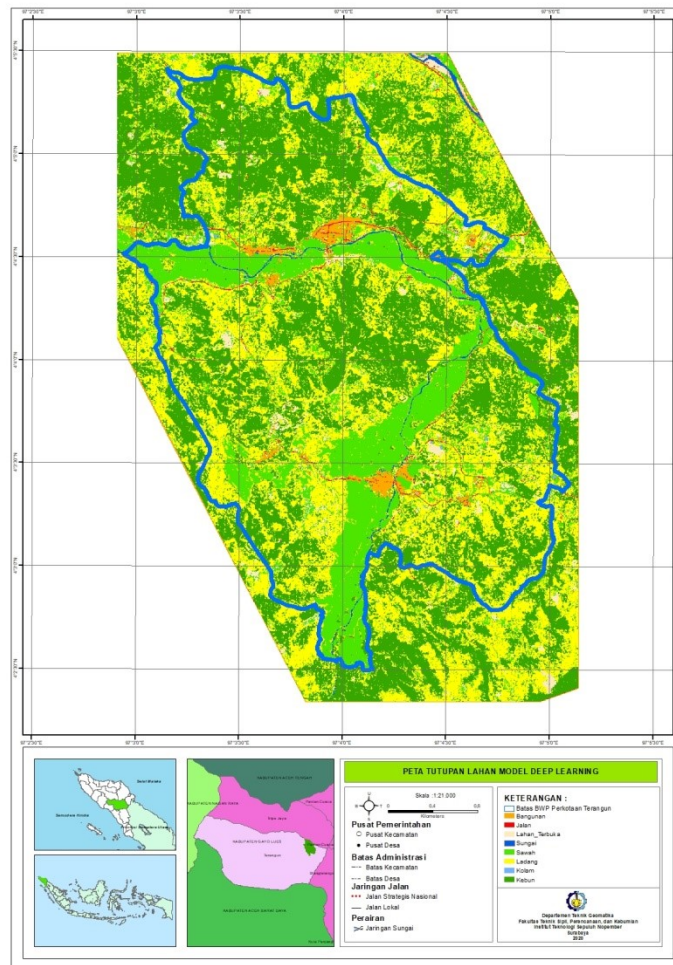


#### 4.18. Hasil Klasifikasi Model *Deep Learning*

Model yang sudah terbentuk dari proses *training* akan digunakan untuk melakukan klasifikasi citra untuk mendapatkan informasi tutupan lahan secara keseluruhan di kawasan Perkotaan Terangun. Klasifikasi yang akan dilakukan berbasis piksel, untuk melakukan klasifikasi maka diperlukan dua data yaitu data citra satelit yang akan diklasifikasi dan data definisi model.

Data definisi model berektensi .EMD (*Esri Model Definition*) dalam data tersebut akan diberikan informasi terkait klasifikasi yang akan dilakukan seperti ukuran file citra pada saat melakukan *training*, tipe dari model yang akan telah dibuat, *inferencefunction*, lokasi file dari model, jumlah band dari citra yang digunakan dan jumlah kelas yang akan diklasifikasikan. Isi dari file definisi model dapat dilihat pada **Lampiran 3 Struktur Data Esri Definisi Model (EMD)**.

Setelah melakukan klasifikasi tutupan lahan dengan model *deep learning*, maka diperoleh peta tutupan lahan di Perkotaan Terangun. Data hasil klasifikasi tutupan lahan tersebut masih dalam format raster, oleh sebab itu perlu dikonversi terlebih dahulu kedalam format vektor. Adapun peta tutupan lahan tersebut dapat dilihat pada Gambar 4.17. Peta tutupan lahan hasil klasifikasi model *deep learning*.



Gambar 4.17. Peta tutupan lahan hasil klasifikasi model *deep learning*

#### 4.19. Hasil Klasifikasi Metode *Maximum Likelihood Classification*

Untuk mendapatkan informasi akurasi dari hasil klasifikasi *deep learning*, maka diperlukan data klasifikasi pembanding. Dalam penelitian ini digunakan hasil klasifikasi metode *maximum likelihood classification*. Dalam melakukan klasifikasi *maximum likelihood classification* maka diperlukan data *training sample*.

##### 4.19.1. *Training Sample*

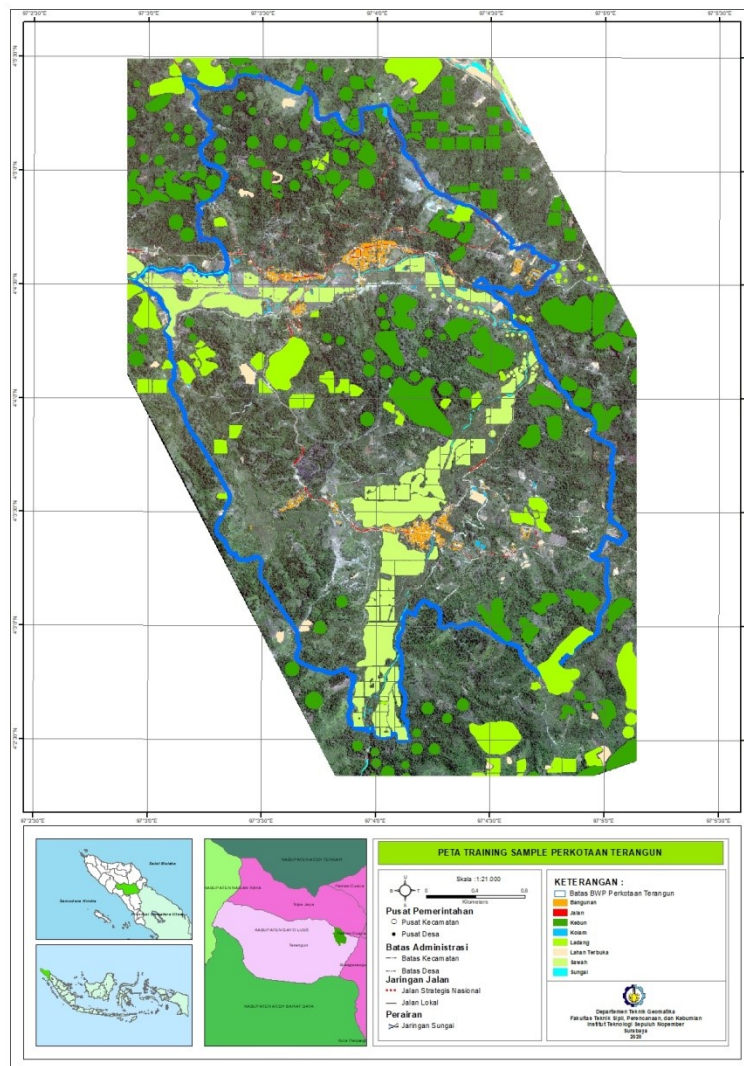
Adapun data *training sample* yang digunakan dalam metode *maximum likelihood classification* ini merupakan data *training sample* yang sama sebagaimana digunakan pada proses klasifikasi menggunakan model *deep*

*learning*. *Training sample* diperlukan untuk memberikan informasi pada aplikasi agar mengenali objek-objek yang ada pada citra satelit. *Training sample* yang digunakan dalam penelitian ini terdiri dari delapan kelas, yaitu bangunan, jalan, lahan terbuka, sungai, sawah, ladang, kolam, dan kebun. Jumlah *training sample* untuk tiap kelas tutupan lahan dapat dilihat pada Tabel 4.24. Jumlah sampel tutupan lahan.

Tabel 4.24. Jumlah Sampel Tutupan Lahan untuk Tiap Kelas Tutupan Lahan

No	Kelas Tutupan Lahan	Jumlah Sampel	Jumlah Piksel
1	Bangunan	981	507.155
2	Jalan	93	204.488
3	Lahan Terbuka	40	376.617
4	Sungai	71	203.716
5	Sawah	130	4.259.372
6	Ladang	68	3.719.385
7	Kolam	41	74.964
8	Kebun	153	7.574.063
<b>Total</b>		<b>1577</b>	<b>16.919.760</b>

Untuk melakukan klasifikasi tutupan lahan dengan metode *maximum likelihood classification*, *training sample* yang sudah dibuat akan digunakan sebagai acuan dalam melakukan klasifikasi tutupan lahan. Citra satelit WorldView-2 level 2A dan *training sample* yang akan digunakan untuk melakukan klasifikasi metode *maximum likelihood classification* dapat dilihat pada Gambar 4.18. Sebaran *training sample*.



Gambar 4.18. Sebaran *training sample*

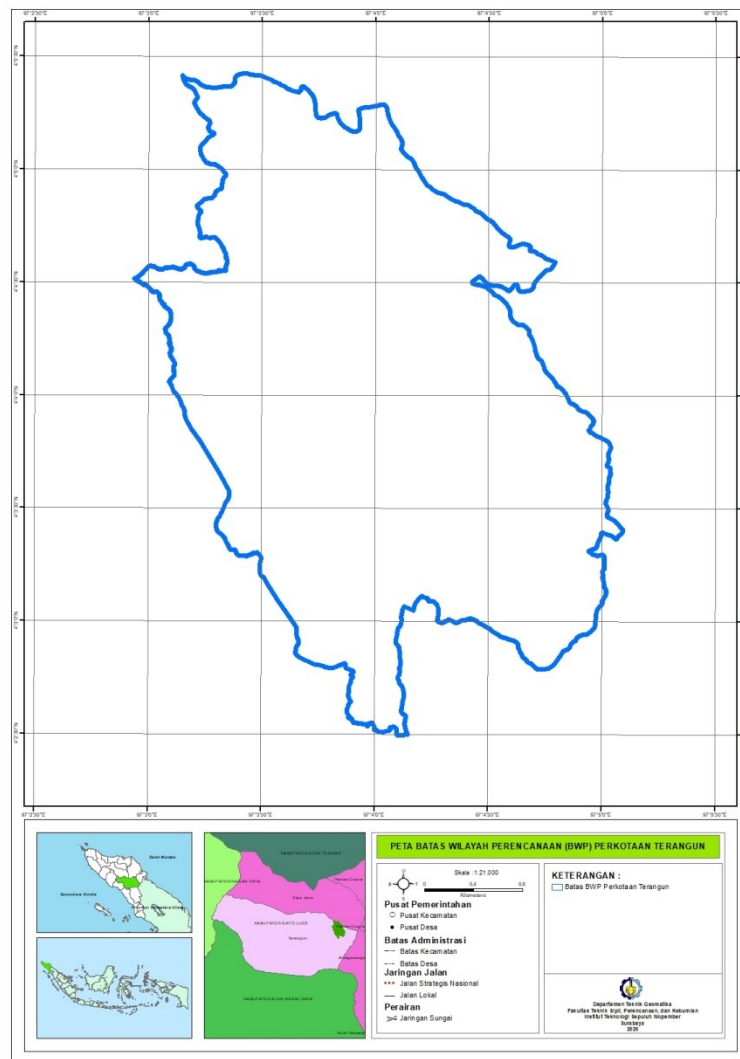
Hasil klasifikasi metode *maximum likelihood classification* setelah dilakukan klasifikasi juga masih berformat data raster, maka diperlukan proses konversi menjadi data vektor seperti yang dilakukan pada data klasifikasi model *deep learning*. Adapun hasil klasifikasi dengan menggunakan metode *maximum likelihood classification* dapat dilihat pada Gambar 4.19. Peta tutupan lahan hasil klasifikasi metode *maximum likelihood classification*.



Gambar 4.19. Peta tutupan lahan hasil klasifikasi metode *maximum likelihood classification*

#### 4.20. Batas Wilayah Perencanaan (BWP) Perkotaan Terangun

Data hasil klasifikasi tutupan lahan yang sudah dikonversi kedalam format vektor masih terdapat informasi diluar area penelitian, oleh sebab itu diperlukan proses *subset* dengan batas wilayah Perkotaan Terangun. Batas wilayah Perkotaan Terangun memiliki luas sebesar 1211,77 hektar. Adapun peta batas wilayah Perkotaan Terangun dapat dilihat pada Gambar 4.20. Peta BWP Perkotaan Terangun.



Gambar 4.20. Peta BPW Perkotaan Terangun

#### 4.21. Luas Tutupan Lahan

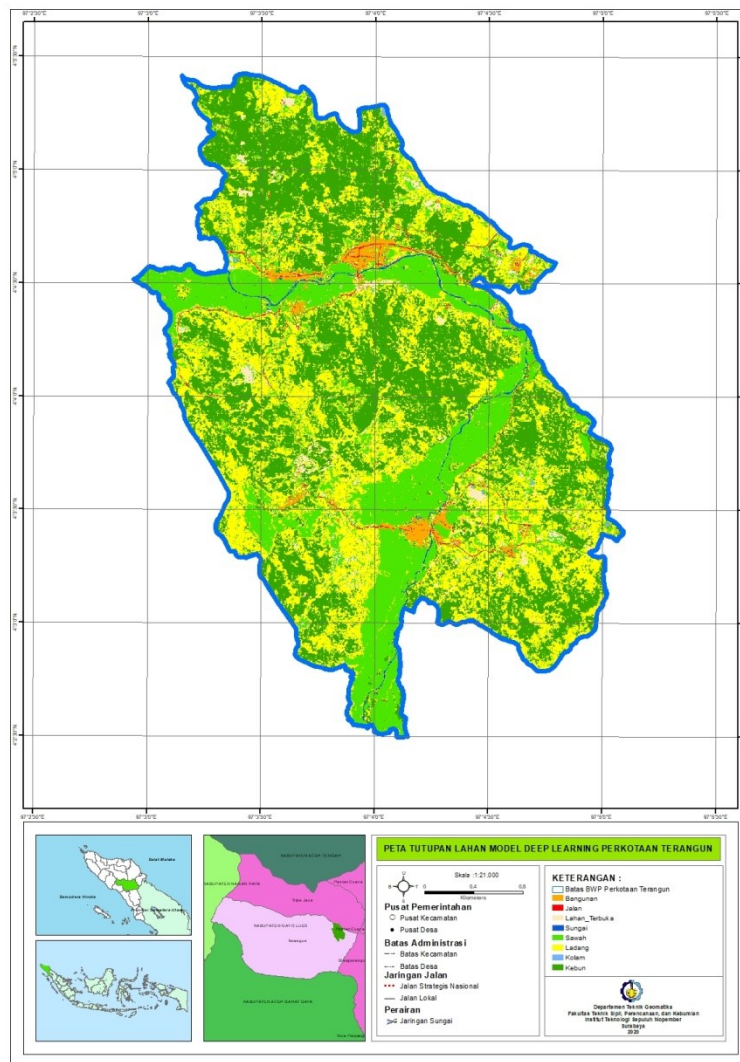
Setelah dilakukan proses *subset* dengan BWP Perkotaan Terangun maka luas tutupan lahan hasil klasifikasi model *deep learning* dihitung luasnya sehingga diketahui luas tiap kelas tutupan lahan sebagaimana pada Tabel 4.25. Luas tutupan lahan model *deep learning*.

Tabel 4.25. Luas Tutupan Lahan Model *Deep Learning*

No	Kelas Tutupan Lahan	Luas (Ha)
1	Bangunan	26,7
2	Jalan	11,42
3	Lahan Terbuka	37,29
4	Sungai	7,65
5	Sawah	264,63
6	Ladang	394,11
7	Kolam	0,85
8	Kebun	469,12
<b>Total</b>		<b>1211,77</b>

Kelas tutupan lahan yang masuk dalam kategori luasnya besar yaitu kebun dengan luas 469,12 ha, ladang 394,11 ha, dan sawah dengan luas 264,63 ha. Sementara yang kelas tutupan lahan lainnya yang terdiri dari lahan terbuka (37,29 ha), bangunan (26,7 ha), jalan (11,42 ha) , sungai (7,65 ha) dan kolam (0,85 ha) masuk dalam kategori luasannya kecil mempunyai total luasan dibawah 50 ha.





Gambar 4.21. Peta tutupan lahan model *deep learning* Perkotaan Terangun

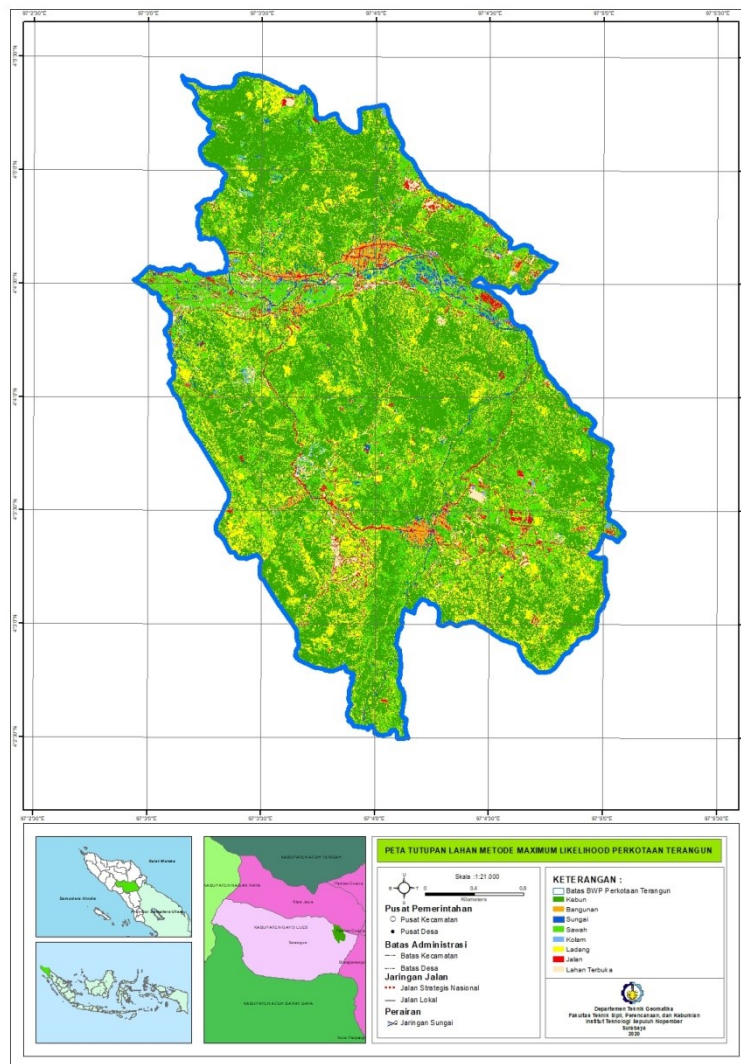
Tutupan lahan metode *maximum likelihood classification* juga dihitung luasan tiap kelas tutupan lahannya. Karna data luasan akan digunakan untuk membandingkan dengan luasan yang dimiliki oleh peta tutupan lahan hasil interpretasi berbasis objek yang sudah mendapatkan rekomendasi dari BIG. Adapun luas tiap kelas tutupan lahan dapat dilihat pada Tabel 4.26. Luas tutupan lahan metode *maximum likelihood classification*



Tabel 4.26. Luas Tutupan Lahan Metode *Maximum Likelihood Classification*

No	Kelas Tutupan Lahan	Luas (Ha)
1	Bangunan	14,62
2	Jalan	42,45
3	Lahan Terbuka	30,87
4	Sungai	22,24
5	Sawah	231,74
6	Ladang	236,75
7	Kolam	6,32
8	Kebun	626,78
<b>Total</b>		<b>1211,77</b>

Dari hasil klasifikasi metode *maximum likelihood classification* maka luasan yang dikategorikan besar terdiri dari kebun (626,78 ha), ladang (236,75 ha) dan sawah (231,74 ha). Sedangkan luasan yang kategori kecil terdiri dari jalan (42,45 ha), lahan terbuka (30,87 ha), sungai (22,24 ha), bangunan (14,62 ha) dan kolam (6,32 ha). Luasan kategori kecil juga sama seperti dalam klasifikasi model *deep learning* luasannya dibawah 50 ha.



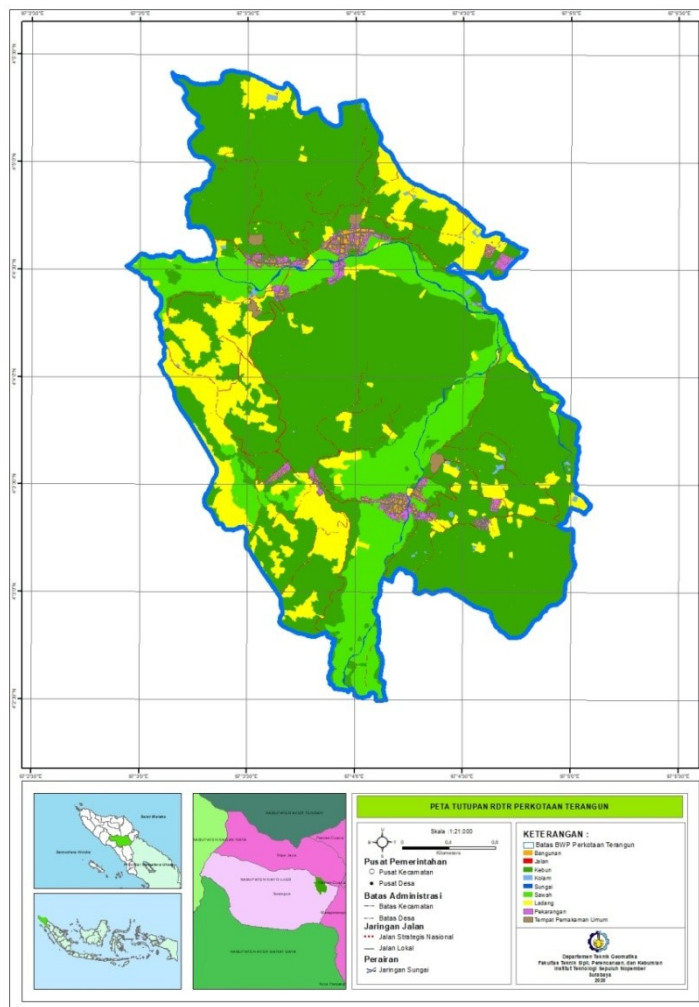
Gambar 4.22. Peta tutupan lahan metode *maximum likelihood classification*

Sebelum melakukan uji akurasi data tutupan lahan hasil klasifikasi menggunakan model *deep learning* dan hasil klasifikasi menggunakan metode *maximum likelihood classification* maka diperlukan data tutupan lahan lainnya yang sudah ada legalitas dari data tersebut. Adapun data yang digunakan adalah data tutupan lahan Perkotaan Terangun skala 1:5.000 yang sudah mendapat rekomendasi dari BIG.

Data tutupan lahan yang sudah mendapatkan legalitas dari BIG merupakan data hasil interpretasi berbasis objek. Data ini sudah dapat digunakan sebagai acuan dalam pemetaan skala 1:5.000. untuk legalitas peta tutupan lahan

Perkotaan Terangun dapat dilihat pada **Lampiran 4 Berita Acara Tutupan Lahan RDTR Terangun.**

Untuk data tutupan lahan hasil interpretasi berbasis objek dapat dilihat pada Gambar 4.23. Tutupan Lahan Perkotaan Terangun.



Gambar 4.23. Peta tutupan lahan hasil interpretasi berbasis objek RDTR Terangun

Tutupan lahan hasil interpretasi berbasis objek terdiri dari sembilan kelas tutupan lahan yaitu bangunan, jaringan jalan, kebun campuran, kolam, pekarangan, sawah, sungai, tegalan, dan tempat pemakaman umum. Untuk tiap luasan dari tutupan lahan hasil interpretasi berbasis objek dapat dilihat pada Tabel 4.28. Luas Tutupan Lahan Interpretasi Berbasis Objek.

Tabel 4.27. Luas Tutupan Lahan Interpretasi Berbasis Objek

No	Kelas Tutupan Lahan	Luas
1	Bangunan	13,1
2	Jalan	19,14
3	Sungai	7,52
4	Sawah	189,39
5	Ladang	175,91
6	Kolam	4,2
7	Kebun	778,69
8	TPU	4,24
9	Pekarangan	19,58
Total		1211,77

Kelas tutupan lahan yang luasannya dikategorikan besar yaitu kebun (778,69 Ha), sawah (189,39 Ha), dan ladang (175,91 Ha). Sedangkan yang termasuk dalam kategori luasannya kecil yaitu pekarangan (19,58 Ha), jalan (19,14 Ha), bangunan (13,1 Ha), sungai (7,52 Ha), TPU (4,24 Ha) dan Kolam (4,2 Ha).

Dalam kelas tutupan lahan RDTR Terangun terdapat 2 kelas yang berbeda dengan kelas tutupan lahan model *deep learning* dan metode *maximum likelihood classification*. Perbedaan tersebut yaitu pada tutupan lahan RDTR Terangun terdapat kelas Taman Pemakaman Umum (TPU) dan pekarangan. Untuk kelas TPU dapat dikategorikan sebagai fasilitas, karena dilihat dari fungsinya bukan tutupan lahannya, jika dilihat secara detil, maka lokasi TPU tersebut terletak di dalam kawasan tutupan lahan ladang.

Sedangkan untuk kelas tutupan lahan pekarangan yang terdapat dalam peta tutupan lahan RDTR Terangun tidak dapat diklasifikasikan dengan model *deep learning* dan metode *maximum likelihood classification* karena pekarangan

terletak diantara celah bangunan dengan jalan. Sedangkan dalam klasifikasi *deep learning* dan metode *maximum likelihood classification* pekarangan tersebut dianggap sebagai bangunan.

#### **4.22. Selisih Luasan Tutupan Lahan**

Setelah dilakukan proses klasifikasi menggunakan model *deep learning* dan *maximum likelihood classification*, maka diperoleh informasi luasan tiap-tiap kelas tutupan lahan. Luas tiap kelas tutupan lahan tidak sama antara hasil klasifikasi model *deep learning* dan hasil klasifikasi metode *maximum likelihood classification*.

Hasil luasan tersebut kemudian dibanding satu sama lain, adapun perbandingan yang dilakukan yaitu:

- a. Luasan klasifikasi model *deep learning* dengan luasan klasifikasi berbasis objek yaitu peta tutupan lahan RDTR Kota Terangun.
- b. Luasan klasifikasi metode *maximum likelihood classification* dengan luasan klasifikasi berbasis objek yaitu peta tutupan lahan RDTR Kota Terangun.
- c. Luasan klasifikasi model *deep learning* dengan luasan klasifikasi metode *maximum likelihood classification*.

##### **4.22.1. Selisih Luasan Klasifikasi Model *Deep Learning* dengan RDTR Kota Terangun**

Selisih luasan hasil klasifikasi model *deep learning* dengan RDTR Kota Terangun totalnya 627,81 ha. Nilai selisih yang diperoleh tidak dihitung luasan lahan terbuka, TPU, dan pekarangan, karena tutupan kelas lahan terbuka tidak ada datanya pada data hasil klasifikasi RDTR Kota Terangun. Sedangkan kelas tutuan lahan TPU dan pekarangan tidak ada datanya pada data hasil klasifikasi menggunakan model *deep learning*.

Selisih yang terdapat dari dua perbandingan tersebut yang paling luas yaitu kebun (309,57 ha), ladang (218,2 ha), sawah (75,24 ha), sedangkan selisih luasan tutupan yang lain dikatergorikan kecil yaitu selisih dari bangunan (13,6 ha), jalan (7,72 ha), kolam (3,35 ha), dan sungai hanya (0,13 ha). Untuk selisih

luasan klasifikasi model *deep learning* dengan luasan RDTR Kota Terangun dapat dilihat pada Tabel 4.29. Selisih luasan model *deep learning* dengan luasan RDTR Kota Terangun.

Tabel 4.29. Selisih Luasan Model *Deep Learning* dengan luasan RDTR Terangun

No	Kelas Tutupan Lahan	Luas Klasifikasi Model Deep Learning (Ha)	Luas Klasifikasi RDTR (Ha)	Luas Selisih (Ha)
1	Bangunan	26,7	13,1	13,6
2	Jalan	11,42	19,14	7,72
3	Lahan Terbuka	37,29		37,29
4	Sungai	7,65	7,52	0,13
5	Sawah	264,63	189,39	75,24
6	Ladang	394,11	175,91	218,2
7	Kolam	0,85	4,2	3,35
8	Kebun	469,12	778,69	309,57
9	TPU		4,24	4,24
10	Pekarangan		19,58	19,58
<b>Total</b>		<b>1211,77</b>	<b>1211,77</b>	<b>627,81</b>

Dari tabel 4.29. Selisih Luasan Model *Deep Learning* dengan luasan RDTR Terangun, maka dapat dilihat bahwa selisih paling luas terdapat pada tutupan lahan kebun dimana dari 778, 69 ha luas kebun hasil klasifikasi berbasis objek dari peta tutupan lahan Perkotaan Terangun, hanya 469,12 ha yang dapat dikenali.

Setelah dianalisis ternyata hal ini disebabkan bahwa banyak terdapat tutupan lahan ladang yang digolongkan sebagai kebun pada tutupan lahan berbasis objek sehingga terjadi peningkatan luasan dari tutupan lahan kebun tersebut yang sangat signifikan. Maka dari hasil analisis tersebut ternyata masih banyak ketidaksesuaian tutupan lahan hasil klasifikasi berbasis objek dengan kondisi eksisting yang terdapat pada citra satelit WorldView-2 level 2A.

#### 4.22.2. Selisih Luasan Klasifikasi Metode *Maximum Likelihood Classification* dengan RDTR Kota Terangun

Selisih luasan hasil klasifikasi metode *maximum likelihood* dengan RDTR Kota Terangun yaitu totalnya 296,77 ha. Selisih kedua tutupan lahan tersebut meliputi kebun (151,91 ha), ladang (60,84 ha), sawah (42,35 ha), jalan (23,31 ha) dan sungai (14,72 ha). Sedangkan selisih luasan bangunan dan kolam sangat kecil perbedaannya yaitu kolam (2,12 ha) dan bangunan (1,52 ha).

Selisih luasan metode *maximum likelihood* dengan RDTR Kota Terangun dapat dilihat pada Tabel 4.30. Selisih luasan metode *maximum likelihood* dengan luasan RDTR Kota Terangun.

Tabel 4.30. Selisih Luasan Metode *Maximum Likelihood Classification* dengan luasan RDTR Terangun

No	Kelas Tutupan Lahan	Luas Klasifikasi Maximum Likelihood (Ha)	Luas Klasifikasi RDTR (Ha)	Luas Selisih (Ha)
1	Bangunan	14,62	13,1	1,52
2	Jalan	42,45	19,14	23,31
3	Lahan Terbuka	30,87		30,87
4	Sungai	22,24	7,52	14,72
5	Sawah	231,74	189,39	42,35
6	Ladang	236,75	175,91	60,84
7	Kolam	6,32	4,2	2,12
8	Kebun	626,78	778,69	151,91
9	TPU		4,24	4,24
10	Pekarangan		19,58	19,58
<b>Total</b>		<b>1211,77</b>	<b>1211,77</b>	<b>296,77</b>

Dari Tabel 4.30 Selisih Luasan Model *Deep Learning* dengan luasan RDTR Terangun juga terdapat selisih luasan yang sangat luas pada kelas kebun, tetapi tidak terlalu banyak dibandingkan dengan model *deep learning*. Dari hasil klasifikasi metode *maximum likelihood classification* perbedaannya hanya (151,91 ha).

Jika dilihat dari luasan, maka akurasinya lebih bagus hasil klasifikasi metode *maximum likelihood classification* dengan hasil klasifikasi model *deep learning*, karena hanya selisih 151,91 ha, sedangkan hasil klasifikasi model *deep learning* 309,57 ha, tapi secara visual metode *maximum likelihood classification* masih jauh akurasinya dalam mengelompokkan tiap kelas tutupan lahan dibandingkan dengan hasil pengelompokan yang dilakukan oleh model *deep learning*.

#### 4.22.3. Selisih Luasan Klasifikasi Model *Deep Learning* dengan Metode *Maximum Likelihood Classification*

Dari hasil klasifikasi model *deep learning* dan metode *maximum likelihood classification*, maka terdapat selisih luasan dari tiap kelas tutupan lahan. Selisih diantara dua metode tersebut (417,5 ha) yaitu pada kelas kebun sebesar (157,66 ha), ladang (157,36 ha), sawah (32,89 ha), jalan (31,03 ha), sungai (14,59 ha), bangunan (12,08 ha), lahan terbuka (6,42 ha), dan kolam (5,47 ha). Untuk selisih tiap kelas tutupan lahan dapat dilihat pada Tabel 4.31. Selisih luasan kelas tutupan lahan.

Tabel 4.31. Selisih Luasan Klasifikasi Model *Deep Learning* dengan Metode *Maximum Likelihood Classification*

No	Kelas Tutupan Lahan	Luas Klasifikasi <i>Maximum Likelihood</i> (Ha)	Luas Klasifikasi Model <i>Deep Learning</i> (Ha)	Luas Selisih (Ha)
1	Bangunan	14,62	26,7	12,08
2	Jalan	42,45	11,42	31,03
3	Lahan Terbuka	30,87	37,29	6,42
4	Sungai	22,24	7,65	14,59
5	Sawah	231,74	264,63	32,89
6	Ladang	236,75	394,11	157,36
7	Kolam	6,32	0,85	5,47
8	Kebun	626,78	469,12	157,66
<b>Total</b>		<b>1211,77</b>	<b>1211,77</b>	<b>417,5</b>



Dari Tabel 4.31. Selisih luasan kelas tutupan lahan maka model *deep learning* dalam melakukan klasifikasinya lebih cenderung mendeteksi piksel kebun, jalan, sungai dan kolam sedangkan metode *maximum likelihood classification* lebih cenderung mendeteksi piksel ladang, sawah, bangunan, dan lahan terbuka.

Kecenderungan dari model *deep learning* dalam mendeteksi piksel rata-rata yang mempunyai sampelnya banyak. Karena, sampel kebun, jalan, sungai, dan kolam jumlah sampelnya lebih dari 1500 sampel untuk tiap kelasnya. Dari hasil ini dapat dilihat bahwa semakin banyak sampel yang digunakan untuk proses *training* maka model akan semakin mengenali objek dari sampel tersebut.

#### **4.23. Uji Akurasi**

Uji akurasi diperlukan untuk mengetahui tingkat akurasi yang dihasilkan dari tiap hasil klasifikasi. Uji akurasi menggunakan metode *Confusion Matrix* untuk mendapatkan nilai *Overall Accuracy* dan nilai *Kappa Accuracy*. Metode *Confusion Matrix* yang digunakan untuk menguji akurasi klasifikasi model *deep learning* dengan tutupan lahan RDTR Terangun dan akurasi klasifikasi metode *maximum likelihood classification* dengan tutupan lahan RDTR Terangun.

Titik uji akurasi berjumlah 100 piksel untuk tiap pengujian. Titik uji akurasi ditentukan berdasarkan persentase luasan tiap tutupan lahan. Tutupan lahan yang memiliki luasan sedikit maka jumlah titik uji juga sedikit, jika tutupan lahan memiliki luasan besar maka titik uji akan bertambah sesuai dengan persentase masing-masing. Semua titik uji disebar secara acak.

##### **4.23.1. Uji Akurasi Model *Deep Learning* dengan Tutupan Lahan RDTR Terangun**

Dari hasil uji akurasi kelas tutupan lahan model *deep learning* dengan kelas tutupan lahan RDTR Terangun maka titik ujinya dipilih secara acak sebanyak 100 piksel yang terdiri dari bangunan berjumlah 1 piksel, jalan 2 piksel, lahan terbuka 2 piksel, sungai 1 piksel, sawah 21 piksel, ladang 29 piksel, kolam 1 piksel, dan kebun 43 piksel. Setelah dilakukan uji akurasi maka didapatkan

informasi seperti dalam Tabel 4.32. matrik *confusion* tutupan lahan model *deep learning*.

Tabel 4.32. Matrik *Confusion* Tutupan Lahan Model *Deep Learning*

No	Landcover	Bangunan	Jalan	Lahan Terbuka	Sungai	Sawah	Ladang	Kolam	Kebun	Total Baris	User's Accuracy
1	Bangunan	1	0	0	0	2	0	0	0	3	33
2	Jalan	0	2	0	1	1	1	0	0	5	40
3	Lahan Terbuka	0	0	0	0	0	0	0	0	0	0
4	Sungai	0	0	0	0	0	0	0	0	0	0
5	Sawah	0	0	0	0	17	2	0	0	19	89
6	Ladang	0	0	1	0	1	5	0	4	11	45
7	Kolam	0	0	0	0	0	0	1	0	1	100
8	Kebun	0	0	1	0	0	21	0	39	61	64
Total Kolom		1	2	2	1	21	29	1	43	100	
Producer's Accuracy		100	100	0	0	81	17	100	91		

Dari hasil uji akurasi maka tutupan lahan bangunan, jalan, dan kolam dapat diklasifikasi dengan sempurna oleh model *deep learning*, sedangkan lahan terbuka dan sungai hasil klasifikasinya tidak sesuai. Lahan terbuka diklasifikasikan sebagai ladang dan kebun, hal ini disebabkan banyak lahan terbuka terdapat diantara ladang dan kebun karena banyak terjadi pembukaan lahan baru pada lokasi tersebut untuk bercocok tanam serah wangi.

Tutupan lahan sawah secara garis besar klasifikasinya sudah sesuai, terdapat sekitar 81% kesesuaian klasifikasi, sedangkan 19% tidak sesuai. Klasifikasi yang tidak sesuai ini sebagian ada yang diklasifikasi sebagai bangunan, jalan dan ladang. Ketidaksesuaian ini diakibatkan sebagian besar bangunan, jalan dan ladang terletak bersebelahan dengan sawah.

Hasil klasifikasi tutupan lahan ladang hanya sesuai 17%, sekitar 83% klasifikasi tutupan lahan ladang dikategorikan sebagai jalan, sawah dan kebun. Kesalahan klasifikasi tutupan lahan ladang yang terjadi untuk jalan hanya terdapat satu lokasi, sedangkan untuk sawah sebanyak dua lokasi, kesalahan paling banyak

terjadi terhadap kebun yaitu 21 lokasi. kesalahan ini diakibatkan warna dari kelas tutupan lahan ladang hampir mirip dengan kelas tutupan lahan kebun.

Klasifikasi tutupan lahan kebun menunjukkan kesesuaian pada angka 91%, hanya 9% yang tidak sesuai. Ketidaksesuaian ini akibat kelas ladang dianggap sebagai kebun, hal ini terjadi akibat kemiripan dari warna dua kelas tutupan lahan tersebut. Dari sebaran 100 titik uji tersebut maka didapatkan nilai *Overall Accuracy* sebesar 65%, dan *Kappa Accuracy* sebesar 48,09% . Nilai akurasi dari *Overall Accuracy* dan *Kappa Accuracy* masih sangat rendah sehingga *training sample* dan *epoch* perlu diperbanyak untuk meningkatkan akurasi tersebut.

#### 4.23.2. Uji Akurasi Metode *Maximum Likelihood Classification* dengan Tutupan Lahan RDTR Terangun

Untuk membandingkan akurasi model *deep learning* dengan akurasi metode *maximum likelihood classification*, maka perlu dilakukan uji akurasi metode *maximum likelihood classification* dengan tutupan lahan RDTR Terangun, titik uji yang digunakan juga terdiri dari 100 titik piksel yang disebar secara acak meliputi bangunan berjumlah 1 piksel, jalan 1 piksel, lahan terbuka 1 piksel, sungai 1 piksel, sawah 11 piksel, ladang 15 piksel, kolam 1 piksel, dan kebun 69 piksel. Setelah dilakukan uji akurasi maka didapatkan informasi seperti dalam Tabel 4.33. matrik *confusion* tutupan lahan metode *maximum likelihood*.

Tabel 4.33. Matrik *Confusion* Tutupan Lahan Metode *Maximum Likelihood Classification*

No	Landcover	Bangunan	Jalan	Lahan Terbuka	Sungai	Sawah	Ladang	Kolam	Kebun	Total Baris	User's Accuracy
1	Bangunan	0	0	0	0	0	1	0	0	1	0
2	Jalan	0	0	0	0	0	3	0	1	4	0
3	Lahan Terbuka	0	0	0	0	0	2	0	1	3	0
4	Sungai	1	0	0	1	0	0	0	0	2	50
5	Sawah	0	1	0	0	6	3	0	9	19	32
6	Ladang	0	0	1	0	2	0	0	17	20	0
7	Kolam	0	0	0	0	0	0	1	0	1	100

8	Kebun	0	0	0	0	3	6	0	41	50	82
Total Kolom		1	1	1	1	11	15	1	69	100	
Producer's Accuracy		0	0	0	100	55	0	100	59		

Dari Tabel 4.30. Matrik *confusion* tutupan lahan metode *maximum likelihood classification* dapat dilihat bahwa hanya kelas tutupan lahan sungai dan kolam saja yang dapat diklasifikasi secara tepat, sedangkan bangunan, jalan, lahan terbuka, dan ladang tidak sesuai hasil klasifikasinya. Klasifikasi yang seharusnya bangunan tapi waktu diuji akurasi ternyata sungai, sedangkan klasifikasi yang seharusnya jalan waktu diuji ternyata sawah, sedangkan lahan terbuka menjadi ladang.

Sedangkan ladang sendiri yang titik ujinya berjumlah 15 piksel tidak satupun yang sesuai dari 15 piksel titik uji yang seharusnya ladang menjadi bangunan sebanyak satu piksel, jalan tiga piksel, lahan terbuka dua piksel, sawah tiga piksel dan kebun enam piksel.

Klasifikasi tutupan lahan sawah sebanyak 55% klasifikasinya sesuai dan 45% lainnya tidak sesuai klasifikasi yang seharusnya sawah menjadi ladang dan kebun, hal ini disebabkan oleh warna dari piksel yang berdekatan karena warnanya identik hijau. Sedangkan kebun menunjukkan 59% kesesuaiannya 41% tidak sesuai karena waktu diuji piksel yang diklasifikasi ternyata jalan satu piksel, lahan terbuka satu piksel, sawah sembilan piksel dan ladang 17 piksel.

Dari sebaran titik uji tersebut maka didapatkan nilai *Overall Accuracy* sebesar 49%, dan *Kappa Accuracy* sebesar 25%. Dalam penelitian ini nilai akurasi metode *maximum likelihood classification* lebih rendah dibandingkan dengan nilai akurasi model *deep learning*. Hal ini disebabkan karena dalam metode *Deep Learning* banyak sekali data yang di *training* karena adanya proses augmentasi data, sedangkan dalam metode *maximum likelihood classification* hanya mengandalkan data *training sample* yang diberikan oleh *user*.

*Halaman ini sengaja dikosongkan*

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Dari penelitian yang telah dilakukan, maka didapatkan kesimpulan yang akan dijelaskan sebagai berikut:

1. Model deep learning berhasil mengklasifikasi kelas tutupan lahan dengan baik, adapun ukuran gambar yang digunakan 128 piksel x 128 piksel, dan nilai *stride* 64. Terjadi 17 kali peningkatan akurasi dari jumlah 50 *epoch* yang dijalankan. Hasil klasifikasi model *deep learning* lebih baik dibandingkan dengan hasil klasifikasi metode *maximum likelihood classification*. Nilai akurasi klasifikasi *deep learning* yaitu *overall accuracy* (65%) dan *kappa accuracy* (48,09%). Sedangkan nilai akurasi klasifikasi metode *maximum likelihood classification* yaitu *overall accuracy* (49%) dan *kappa accuracy* (25%).

2. Adapun perbedaan luas klasifikasi *deep learning* dengan peta tutupan lahan RDTR yaitu 627,81 ha, sedangkan perbedaan luas klasifikasi *maximum likelihood classification* dengan peta tutupan lahan RDTR yaitu 296,77 ha.

3. Model *deep learning* dalam melakukan klasifikasi lebih mengenal objek yang telah dilakukan proses augmentasi, dan hasil klasifikasi secara visual lebih bagus dibandingkan dengan model *maximum likelihood classification*.

4. Metode *maximum likelihood classification* lebih cepat dalam melakukan klasifikasi, tetapi secara visual hasil klasifikasinya kurang bagus dibandingkan dengan hasil klasifikasi model *deep learning*.

5. Dalam melakukan klasifikasi tutupan lahan dengan model *deep learning* memerlukan spesifikasi komputer yang tinggi agar cepat dalam melakukan proses pembuatan model. Minimal harus menggunakan prosesor core i7, RAM 16GB, VGA 2 GB. Karena satu *epoch* memerlukan waktu 30 menit – 60 menit proses *training* untuk jumlah sampel 11.893.

6. Klasifikasi dengan model *deep learning* membutuhkan waktu yang lama dibandingkan dengan klasifikasi menggunakan metode *maximum likelihood classification*. Dalam melakukan klasifikasi dengan menggunakan model yang

telah dibuat dengan 50 *epoch* maka membutuhkan waktu 1 jam untuk luas lokasi 4,1 km x 5,8 km. Sedangkan dengan menggunakan metode *maximum likelihood* hanya memerlukan waktu 10 menit.

7. Klasifikasi dengan model *deep learning* tidak cocok digunakan untuk keperluan penyusunan RDTR atau citra resolusi tinggi, karena dalam pemetaan untuk kebutuhan RDTR bangunan harus tergambar setiap persilnya. Sedangkan dalam model dalam penelitian ini tidak dapat tergambar garis tepi bangunan.

8. Jika ingin menggunakan hasil klasifikasi model *deep learning* untuk keperluan RDTR, maka harus dipadukan dengan model *deep learning* untuk klasifikasi objek, agar persil bangunan dapat tergambar dengan baik.

## **5.2. Saran**

Dari penelitian yang telah dilakukan, maka dapat disarankan beberapa saran sebagai berikut;

1. Perlunya dilakukan penelitian lebih lanjut untuk mengetahui hasil klasifikasi yang lebih baik lagi.
2. Supaya dikembangkan model *deep learning* yang dapat mengenali objek tepi bangunan.
3. Perlunya dilakukan penelitian lanjutan untuk membuat model *deep learning* yang dapat memetakan jaringan (*line*) baik jaringan jalan, jaringan sungai dan lainnya.

## DAFTAR PUSTAKA

- Abrahams, S., Hafner, D., Erwitte, E., Scarpinelli, A., (2016), *Tensorflow for Machine Intelligence : A Hands-On Introduction to Learning Algorithms*, Bleeding Edge Press, Santa Roca.
- Amalisana, B., Rokhmatullah, Hernina, R., (2017), "IOP Conference Series : Earth and Environmental Science", *Land Cover Analysis by Using Pixel-Based and Object-Based Image Classification Method in Bogor*, Sci 98 012005
- Asperti, A., Mastronardo, C., (2017), *The Effectiveness of Data Augmentation for Detection of Gastrointestinal Diseases from Endoscopic Images*, Department of Informatics, Universitas of Bologna, Italy.
- Badan Informasi Geospasial, (2017), *Modul Validasi Peta Rencana Tata Ruang*, BIG, Bogor.
- Bhanu, B., Kumar, A., (2000), *Deep Learning for Biometrics*, Springer, Jerman.
- Chollet, F., (2006), *Deep Learning with Python*, Manning Publication Co., New York.
- Danoedoro, P., (1996), *Pengolahan Citra Digital : Teori dan Aplikasinya dalam Bidang Penginderaan Jauh*, Fakultas Geografi Universitas Gadjah Mada, Yogyakarta.
- DigitalGlobe, (2016), *Data Sheet*, DigitalGlobe, Amerika.
- Dev, D., (2017), *Deep Learning with Hadoop*, Packt Publishing, Inggris.
- ERDAS, (1999), *ERDAS Field Guide*, ERDAS, Inc., Georgia.
- Gulli, A., Pal, S., (2017), *Deep Learning with Keras*, Packt Publishing, Birmingham.
- Gulli,A., (2015), *A Collection of Data Science Interview Questions Solved in Python and Spark*, *Hands-on Big Data and Machine Learning*, Packt Publishing, Birmingham-Mumbai.
- Gulli, A., Kapoor, A., (2017), *TensorFlow 1.x : Deep Learning Cookbook*, Packt Publishing, Birmingham-Mumbai.



- He, L., Huang, S., (2017), *Modified Firefly Algorithm Based Multilevel Thresholding for Color Image Segmentation*, Neurocomputing, ELSEVIER, Belanda.
- Heaton, J., (2015), *Artificial Intelligence for Humans, Volume 3 : Deep Learning and Neural Networks*, Heaton Research, Inc, London
- Hope, T., Resheff, Y. S., dan Lieder I., (2017), *Learning TensorFlow : A Guide to Building Deep Learning Systems*, O’Rielly Media, Amerika.
- Kete, S.C.R., Suprihatin, Tarigan, S.D, Efendi, H., (2019), “IOP Conference Series : Earth and Environmental Science”, *Land Use Classification Based on Object and Pixel Using Landsat OLI in Kendary City, Southeast Sulawesi Province, Indonesia*, Sci 284 012019.
- Kukreja, H., (2016), An Introduction to Artificial Neural Network, *International Journal of Advance Research and Innovative Ideas in Education*, Vol.1, No.5, hal. 27-30.
- Kursini, S.Kom, (2006), *Sistem Pakar : Teori dan Aplikasi*, CV. Andi Offset, Yogyakarta.
- Lantz, B., (2015), *Machine Learning with R*, Packt Publishing, Birmingham.
- Lillesand, T. ., Kiefer, R.W., (1999), *Penginderaan Jauh dan Interpretasi Citra, Terjemahan*, Gadjah Mada University Press, Yogyakarta.
- Lowell, K., Jatón, A., (1999), *Spatial Accuracy Assessment : Land Information Uncertainty in Natural Resources*, Sleeping Bear Press, Inc., USA.
- Lunetta, R.,S., and Lyon, J., G., (2005), *Remote Sensing and GIS Accuracy Assessment*, CRC Press, Washington, D.C.
- Martin, S., (2004), *An Introduction to Ocean Remote Sensing*, University of Cambridge, United Kingdom.
- Osinga, D., (2018), *Deep Learning Cookbook : Practical Recipes to Get Started Quickly*, O’Reilly, Amerika.
- Patterson, J., Gibson, A., (2017), *Deep Learning : A Practitioner’s Approach*, O’Rielly Media, Amerika.
- Prahasta, E., (2008), *Remote Sensing: Praktis Penginderaan Jauh & Pengolahan Citra Digital Dengan Perangkat Lunak ER Mapper*, Informatika, Bandung.

- Rahman, M. A., Wang, Y., (2016), *Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation*, Department of Computer Science, University of Manitoba, Canada.
- Ramadhani, M. F., (2015), *Pembangunan Aplikasi Informasi, Pengaduan, Kritik, dan Saran Seputar Kota Cimahi pada Platform Android*, Skripsi, Universitas Komputer Indonesia, Bandung.
- Ronneberger, O., Fischer, P., Brox, T., (2015), *U-Net : Convolutional Network for Biomedical Image Segmentation*, Computer Science Department and BIOS Centre for Biological Signalling Studies, University of Freiburg, Germany.
- Ruder, S., (2017), *An overview of gradient descent optimization algorithm*, Insight Centre for Data Analytics, NUI Galway Aylien Ltd., Dublin.
- Rungta, K., (2019), *TensorFlow : Learn in 1 Day*. Guru99, India.
- S. Kuldeep, G.S. Anitha, (2015), *An Introduction to Artificial Neural Network*, International Journal of Power Electronics Controllers and Converters, Vol.1, No.1, hal. 21-28.
- Santoso, A., Ariyanto, G., (2018), Implementasi Deep Learning Berbasis Keras untuk Pengenalan Wajah, Emitor : Jurnal Teknik Elektro, Vol. 18. No. 01:15-21.
- Setti, S., Wanto, A., (2018), Analysis of Backpropagation Algorithm in Predicting the Most Number of Internet Users in the World, *Jurnal Online Informatika*, Vol.3, No.2, hal. 110-115.
- Shanmugamani, R, (2018), *Deep Learning for Computer Vision*, Packt Publishing, Birmingham - Mumbai.
- Skansi, S., (2017), *Introduction to Deep Learning : From Logical Calculus to Artificial Intelligence*, Springer, Jerman.
- Supardi, Y., (2017), *Python Basic : Semua Bisa Menjadi Programmer*, PT. Elex Media Komputindo, Jakarta.
- Sugomori, Y., Kaluza, B., Saores, F.M., Souza, A.M.F, (2017), *Deep Learning : Practical Neural Networks with Java*, Packt Publishing, Birmingham.
- Sutanto, (1986), *Penginderaan Jauh I*, Gadjah Mada University Press, Yogyakarta

- Tayyebi, A., (2019), *High-Resolution Land Cover Mapping using Deep Learning*, <https://medium.com/geoai/high-resolution-land-cover-mapping-using-deep-learning-7126fee571dd>
- Utoyo, B., (2007), *Geografi : Membuka Cakrawala Dunia*, PT. Setia Purna Inves, Bandung.
- Widyatmanti, W. Natalia, D., (2006), *Geografi untuk SMP dan MTs*, Grasindo, Jakarta
- Wiley, J.F., (2016), *D Deep Learning Essentials*, Pack Publishing, Birmingham
- Withanawasam, J., (2015), *Apache Mahout Essentials : Implement Top-Notch Machine Learning Algorithms for Classification, Clustering, and Recommendations with Apache Mahout*, Packt Publishing, Birmingham-Mumbai.

# LAMPIRAN

## Lampiran 1 : Berita Acara Spesifikasi Citra Satelit WorldView-2 level 2A



PUSAT PEMETAAN TATA RUANG DAN ATLAS  
BADAN INFORMASI GEOSPASIAL (BIG)  
Jl. Raya Jakarta-Bogor, Km. 46 Cibinong 16911  
Telp./Fax. : (021) 8764613

---

**BERITA ACARA  
SUPERVISI PETA RDTR BWP TERANGUN KABUPATEN GAYO LUES  
PROVINSI NANGGROE ACEH DARUSSALAM  
(Ke-1)**

Pada hari ini, Selasa, tanggal 01 Agustus 2017, bertempat di Badan Informasi Geospasial, telah dilakukan asistensi dan supervisi dalam rangka Penyusunan Peta RDTR BWP Terangun, Kabupaten Gayo Lues, Provinsi Nanggroe Aceh Darussalam.

Pada supervisi ini membahas tentang:

1. Sumber Data
2. Persebaran GCP dan ICP

Dari hasil supervisi tersebut, dinyatakan sebagai berikut:

**1. Sumber Data**

- Sumber data yang digunakan dalam penyusunan Peta RDTR BWP Terangun, Kabupaten Gayo Lues, Provinsi Nanggroe Aceh Darussalam adalah satu *scene* citra dengan spesifikasi sebagai berikut :

a. Nama Folder = 056784125010\_01\_P001  
Jenis Citra = WorldView-2  
Sudut Kemiringan = 7,2°  
Resolusi spasial = 0,5  
Waktu Perekaman = 02-08-2016

Tutupan awan kurang dari 10 %, sehingga dinyatakan sumber data dapat digunakan untuk pemetaan peta RDTR skala 1: 5.000.

- Data Model Ketinggian (DEM) yang digunakan adalah IFSAR yang diperoleh dari Badan Informasi Geospasial.

Berdasarkan keterangan citra diatas dapat disimpulkan bahwa citra satelit WorldView-2 tersebut dapat digunakan sebagai sumber data pembuatan peta dasar untuk Peta RDTR BWP Terangun, Kabupaten Gayo Lues, Provinsi Nanggroe Aceh Darussalam.

**2. Persebaran GCP dan ICP**

- Ketiga luasan citra satelit yang digunakan adalah  $\pm 32,364$  km<sup>2</sup>.
- Jumlah titik kontrol sebanyak 23 titik.
- Penempatan atau pemilihan objek GCP dan ICP serta metode pengamatan, disesuaikan dengan supervisi dari Tim asistensi BIG.
- Pengukuran GCP dan ICP harus menggunakan metode Diferensial dengan menggunakan paling sedikit dua *receiver* yang mana salah satunya diletakan pada titik yang telah diketahui koordinatnya/ stasiun referensi titik BIG orde 0 atau orde 1
- Ketelitian hasil pengukuran GPS untuk GCP dan ICP per titik sesuai spesifikasi adalah <20 cm.

Penentuan titik-titik tersebut sudah sesuai dengan kriteria penempatan titik control dan dapat segera dilakukan pengukuran. Namun titik-titik tersebut perlu disesuaikan kembali dengan kondisi di lapangan, apabila sudah berbeda konsisinya dengan yang terlihat pada citra maka sebaiknya dipindah/ dicari titik lain.

Berdasarkan hal tersebut diatas maka data-data yang perlu diserahkan kepada BIG setelah pengukuran GCP dan ICP dilakukan yaitu :

1. Formulir lapangan sesuai standar BIG
2. Raw data perekaman alat GPS dan hasil pengolahan koordinat GCP dan ICP
3. Spesifikasi alat
4. Laporan dan hasil pengukuran GCP dan ICP

Demikian Berita Acara ini dibuat, untuk digunakan sebagaimana mestinya.

Dinas Pekerjaan Umum dan Penataan Ruang  
Kabupaten Gayo Lues



Ilyas

Badan Informasi Geospasial (BIG)  
Pusat Pemetaan Tata Ruang dan Atlas  
Bidang Pemetaan Tata Ruang



## Lampiran 2 : Berita Acara Hasil Orthorektifikasi



PUSAT PEMETAAN TATA RUANG DAN ATLAS  
BADAN INFORMASI GEOSPASIAL (BIG)  
Jl. Raya Jakarta-Bogor, Km. 46 Cibinong 16911  
Telp./Fax. : (021) 8764613

### BERITA ACARA SUPERVISI PETA RDTR BWP TERANGUN KABUPATEN GAYO LUES PROVINSI NANGGROE ACEH DARUSSALAM (Ke-2)

Pada hari ini Rabu, tanggal 06 September 2017, bertempat di Badan Informasi Geospasial, telah dilakukan asistensi dan supervisi dalam rangka Penyusunan Peta RDTR BWP Terangun Kabupaten Gayo Lues, Provinsi Nanggroe Aceh Darussalam.

Pada supervisi ini membahas tentang:

1. Kelengkapan Data Pengukuran
2. Orthorektifikasi dan Uji Akurasi

Dari hasil supervisi tersebut, dinyatakan sebagai berikut:

#### 1. Kelengkapan Data Pengukuran

Adapun rincian hasil pengukuran titik kontrol tanah sebagai berikut :

- a. Luas area citra satelit yaitu  $\pm 32,364 \text{ km}^2$
- b. Jumlah total titik kontrol tanah yang sudah dilakukan pengukuran yaitu 23 titik. Setelah dilakukan pengecekan objek titik kontrol tersebut (berdasarkan deskripsi titik pengukuran dan kenampakan objek pada citra satelit) dapat disimpulkan keseluruhan titik dapat digunakan.
- c. Alat yang digunakan adalah GPS Geodetic South S86S.
- d. Metode yang digunakan dalam pengamatan GPS adalah metode differensial statik (*post processing*) menggunakan titik referensi yaitu SRGI 2013 (S392A) dan 2 titik perapatan, yaitu TR09A dan TR16. Hasil pengukuran titik perapatan sudah memenuhi syarat.
- e. Sudah melampirkan deskripsi stasiun referensi BIG.
- f. Menggunakan metode radial, dimana jarak baselinenya berkisar antara 1 km – 3 km dengan pengukuran rata-rata 1-2 jam. Sudah melampirkan *baseline processing report* sehingga diketahui akurasi horizontal dan akurasi vertikal pada tiap pengukuran titik. Semua titik hasil pengukuran sudah memenuhi persyaratan akurasi horizontal  $\leq 20 \text{ cm}$  dan akurasi vertikal  $\leq 40 \text{ cm}$ .
- g. Pengolahan dilakukan dengan menggunakan *software* pengolahan PCI *Geomatic Systems*.
- h. Sudah menyertakan data *shapefile* (.shp) sebaran titik GCP dan ICP serta formulir pengukuran titik kontrol (*logsheet*).
- i. Data pengamatan satelit GPS dalam bentuk *rinex* dan *raw* sudah dilampirkan.

#### 2. Orthorektifikasi dan Uji Akurasi

Uji akurasi menggunakan 13 titik ICP dihasilkan akurasi horizontal sebesar 2 meter. Berdasarkan nilai uji akurasi tersebut dapat dinyatakan bahwa sumber data sudah dapat digunakan untuk pemetaan RDTR skala 1:5.000.

Hal-hal seperti disebutkan diatas perlu ditindaklanjuti dan dikonsultasikan kembali oleh Dinas Pekerjaan Umum dan Penataan Ruang Kabupaten Gayo Lues kepada Badan Informasi Geospasial. Tahapan selanjutnya adalah pemeriksaan terhadap peta dasar. Demikian Berita Acara ini dibuat, untuk digunakan sebagaimana mestinya.

Dinas Pekerjaan Umum dan Penataan Ruang  
Kabupaten Gayo Lues

Ilyas

Badan Informasi Geospasial (BIG)  
Pusat Pemetaan Tata Ruang dan Atlas  
Bidang Pemetaan Tata Ruang



*Halaman ini sengaja dikosongkan*

Lampiran 3 : Struktur Data *Esri Model Definition* (EMD) File

```
{  
  
  "Framework" : "Keras",  
  "InferenceFunction" : "D:\\tesis_ilyas\\proses_data\\ImageClassifier.py",  
  "ModelConfiguration": "DeepLab",  
  "ModelFile" : " D:\\tesis_ilyas\\proses_data\\model\\model_epochs_50.h5",  
  "ModelType" : "ImageClassification",  
  "ExtractBands" : "[0,1,2,3]",  
  "ImageWidth": 128,  
  "ImageHeight": 128,  
  "Classes" : [  
    {  
      "Value": 1,  
      "Name": "Bangunan",  
      "Color": [100, 100, 100]  
    },  
    {  
      "Value": 2,  
      "Name": "Jalan",  
      "Color": [204, 204, 204]  
    },  
    {  
      "Value": 3,  
      "Name": "Lahan_Terbuka",  
      "Color": [255, 190, 232]  
    },  
    {  
      "Value": 4,  
      "Name": "Sungai",  
      "Color": [0, 92, 230]  
    }  
  ]  
}
```



```
},
{
  "Value": 5,
  "Name": "Sawah",
  "Color": [76, 230, 0]
},
{
  "Value": 6,
  "Name": "Ladang",
  "Color": [255, 255, 115]
},
{
  "Value": 7,
  "Name": "Kolam",
  "Color": [115, 178, 255]
},
{
  "Value": 8,
  "Name": "Kebun",
  "Color": [56, 168, 0]
}
]
}
```

## Lampiran 4 : Berita Acara Tutupan Lahan RDTR Terangun



**BADAN INFORMASI GEOSPASIAL (BIG)**  
**PUSAT PEMETAAN TATA RUANG DAN ATLAS**  
Jl. Raya Jakarta-Bogor, Km. 46 Cibinong 16911  
Telp./Fax. : (021) 8764613

---

### **BERITA ACARA SUPERVISI PETA RDTR BWP TERANGUN KABUPATEN GAYO LUES PROVINSI ACEH KE - 3**

Pada hari ini, Rabu, tanggal 22 November 2017, bertempat di Badan Informasi Geospasial, Cibinong, telah dilakukan asistensi dan supervisi dalam rangka Penyusunan Peta RDTR BWP Terangun Kabupaten Gayo Lues Provinsi Aceh.

Pada supervisi ini membahas tentang:

1. Peta Dasar
2. Peta Tematik
3. Peta Rencana
4. Album Peta

Catatan penting berdasarkan hasil review tersebut :

#### **1. Peta Dasar**

##### **(1) Batas Administrasi**

- Batas administrasi yang dilampirkan berupa batas desa dan batas kecamatan. Kedua data batas administrasi tersebut sudah dilengkapi dengan keterangan status batas (definitif/indikatif), sumber data dan tahun.
- Batas desa dan batas kecamatan bersifat indikatif dan sudah menyesuaikan dengan unsur fisik berupa sungai dan jalan.

##### **(2) Deliniasi BWP**

- Deliniasi batas BWP sudah menyesuaikan dengan batas fisik berupa sungai dan jalan.
- Sudah melampirkan sub BWP dan blok BWP.

##### **(3) Bangunan**

- Digitasi bangunan sudah sesuai dengan kenampakan pada citra dan sudah mencakup seluruh bangunan dengan ukuran > 2,5 m x 2,5 m.
- Data atribut bangunan sudah dilengkapi informasi tipe penutup lahan bangunan, nama bangunan, sumber data dan tahun.

##### **(4) Jaringan jalan**

- Sudah melampirkan jaringan jalan dalam bentuk polygon untuk jalan dengan lebar > 2,5 m.
- Sudah melampirkan jaringan jalan dalam bentuk polyline untuk jalan dengan lebar <2,5 m dan sebagai as jalan polygon.
- Data atribut jalan sudah dilengkapi dengan informasi nama ruas jalan utama, fungsi jalan, status kewenangan jalan, sumber data dan tahun.

##### **(5) Perairan**

- Sudah melampirkan perairan dalam bentuk polygon untuk sungai dengan lebar > 2,5 m.
- Sudah melampirkan perairan dalam bentuk polyline untuk sungai dengan lebar <2,5 m dan sebagai as sungai polygon.
- Data atribut sungai sudah melampirkan informasi nama sungai, tipe sungai, sumber data dan tahun.

##### **(6) Kontur**

- Data kontur yang dilampirkan sudah dilengkapi dengan nilai ketinggian, sumber data dan tahun.

##### **(7) Toponimi**

- Sudah diletakkan tepat di atas bangunan dan sudah mengacu kepada nama obyek secara spesifik.
- Data atribut toponimi sudah dilengkapi dengan sumber data dan tahun.

##### **(8) Tutupan Lahan**

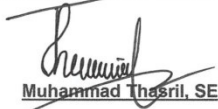
- Klasifikasi tutupan lahan sudah menyesuaikan dengan pedoman klasifikasi tutupan lahan skala 1:5000 yang dikeluarkan oleh BIG.

- Data atribut tutupan lahan sudah menampilkan informasi klasifikasi tutupan lahan, toponimi, sumber data dan tahun.
- (9) Topologi
- Sudah tidak ditemukan kesalahan topologi pada hasil digitasi peta dasar.
- 2. Peta Tematik**
- Data peta tematik yang sudah dilampirkan antara lain peta tematik curah hujan, daerah aliran sungai, fasilitas, geologi, hidrologi, jaringan jalan eksisting, kepadatan penduduk, ketinggian, lereng, morfologi, topologi, tutupan lahan eksisting.
  - Seluruh peta tematik terlampir sudah sesuai dengan deliniasi batas BWP yang digunakan. Selain itu juga sudah mencantumkan sumber data dan tahun.
  - Untuk klasifikasi tutupan lahan perlu dibuat lebih detil lagi dan lebih spesifik mengacu kepada fungsi penggunaan lahannya.
  - Perlu melengkapi peta tematik pertambangan, rawan bencana, cekungan air tanah, jenis tanah, kawasan kehutanan, kawasan pertahanan, kesesuaian lahan, pertanian dan perkebunan, peternakan, jaringan eksisting (jaringan telekomunikasi, jaringan energi kelistrikan, jaringan persampahan, jaringan pembuangan limbah, jaringan drainase, jaringan air minum dsb) bila ada.
- 3. Peta Rencana**
- (1) Pola Ruang
- Peta pola ruang yang dilampirkan sudah mengacu kepada Permen PU No. 20 Tahun 2011.
  - Sungai dan jalan diklasifikasikan sebagai satu kesatuan yang terpisah dari pola ruang budidaya dan lindung.
- (2) Struktur Ruang
- Peta struktur ruang yang dilampirkan antara lain : rencana jaringan transportasi, rencana jaringan telekomunikasi, rencana jaringan persampahan, rencana jaringan energi kelistrikan, rencana jaringan IPAL, rencana jaringan drainase, rencana jaringan air bersih, rencana intensitas bangunan.
  - Peta jaringan rencana dan eksisting belum dijadikan ke dalam satu shp yang sama dan belum dilengkapi dengan atribut keterangan status jaringan eksisting/rencana/pengembangan.
  - Perlu memperbaiki atribut peta jaringan rencana agar menyesuaikan dengan kondisi perencanaan.
- (3) Sub BWP Prioritas
- Belum melampirkan sub BWP yang diprioritaskan.
- 4. Album Peta**
- Album peta yang dilampirkan antara lain: peta administrasi kawasan perencanaan, peta tematik, peta pola ruang secara keseluruhan, peta pola ruang per NLP, peta struktur jaringan secara keseluruhan dan peta rencana struktur jaringan terpisah.
  - Album peta yang dibuat perlu menyesuaikan dengan RSNI tentang Spesifikasi Penyajian Peta RDTR.
  - Simbolisasi dan pewarnaan perlu menyesuaikan dengan RSNI dan PP No 8 Tahun 2013.
  - Pewarnaan peta Pola Ruang perlu menyesuaikan dengan Permen PU No. 20 Tahun 2011.
  - Belum melampirkan peta sub BWP yang diprioritaskan.

Hal-hal seperti disebutkan diatas perlu ditindaklanjuti dan dikonsultasikan kembali oleh Dinas Pekerjaan Umum dan Penataan Ruang Kabupaten Gayo Lues kepada Badan Informasi Geospasial. Tahap selanjutnya adalah pemeriksaan terhadap perbaikan catatan tersebut diatas.

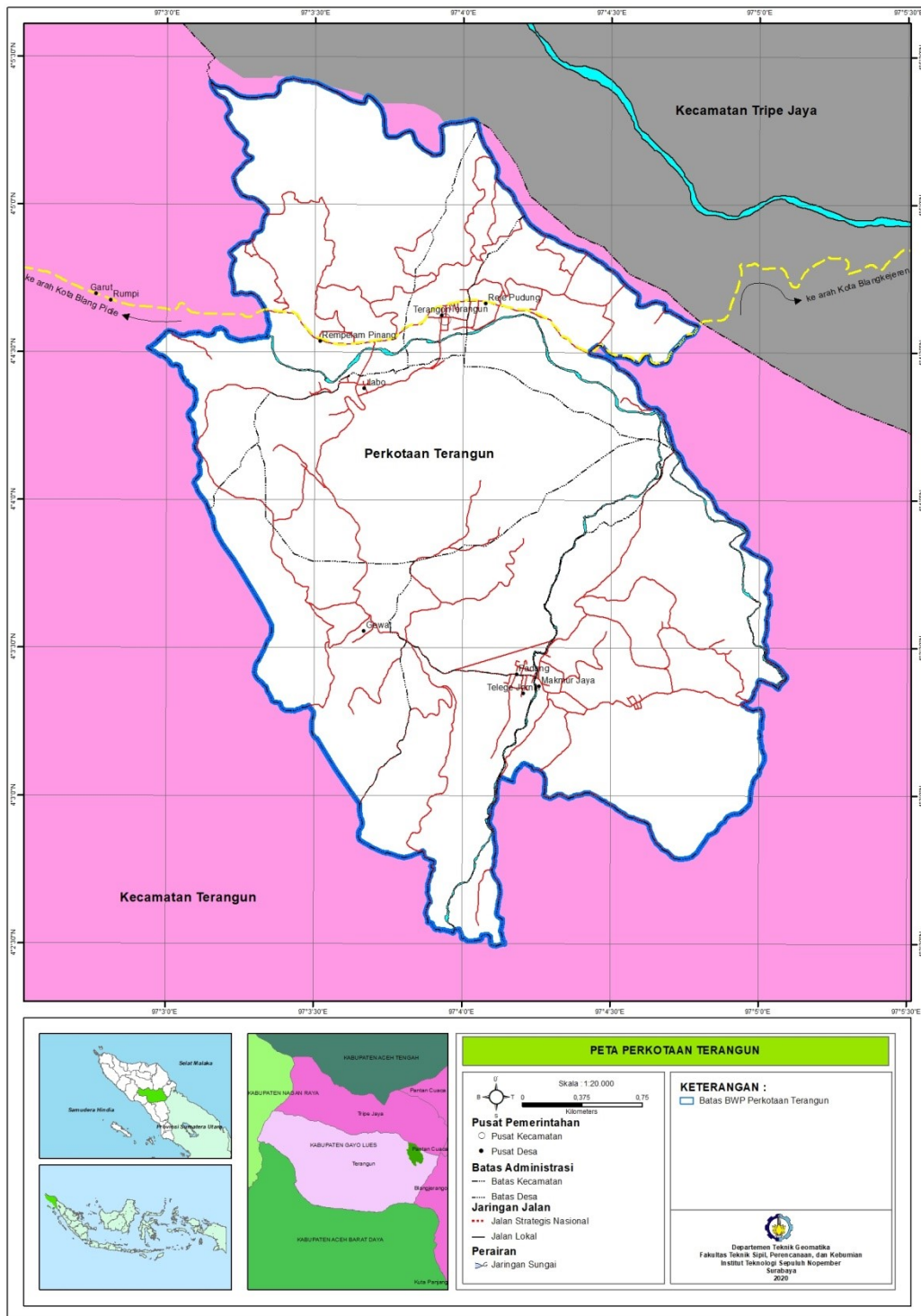
Demikian Berita Acara ini dibuat, untuk digunakan sebagaimana mestinya.

Dinas Pekerjaan Umum dan Penataan  
Ruang Kabupaten Gayo Lues

  
Muhammad Thasril, SE

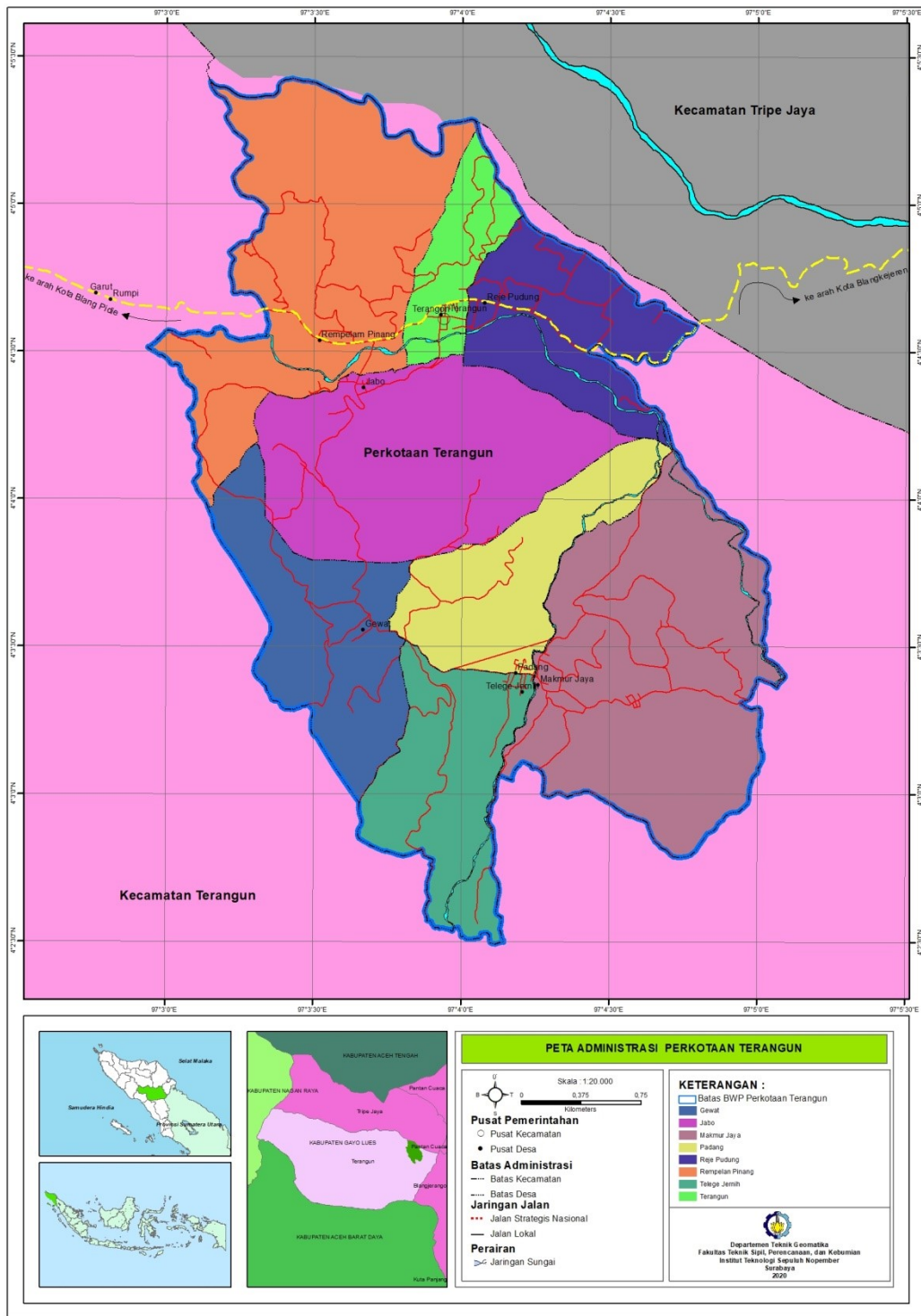


Lampiran 5 : Peta Batas Wilayah Perencanaan (BWP) Perkotaan Terangun



*Halaman ini sengaja dikosongkan*

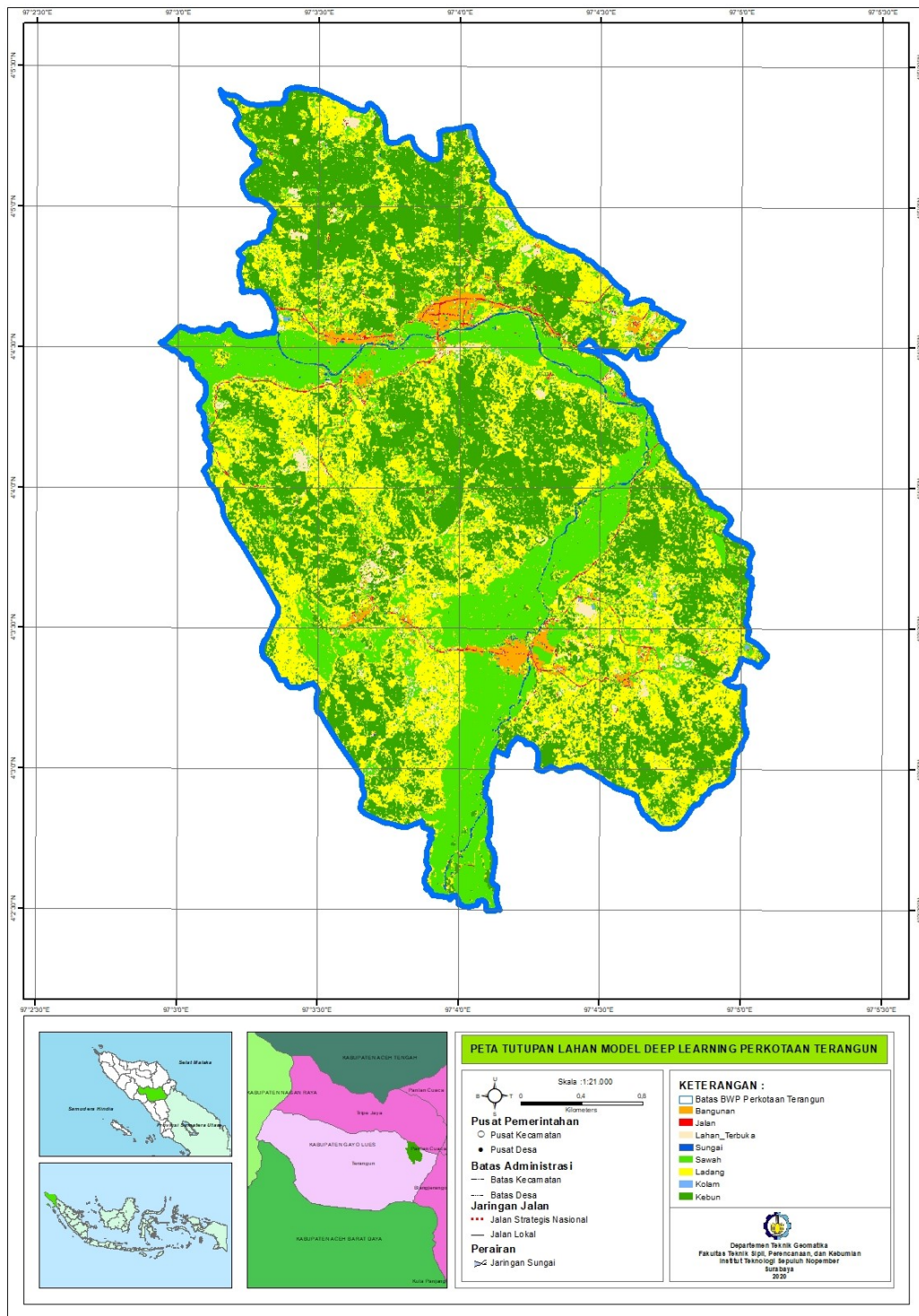
Lampiran 6 : Peta Batas Administrasi Perkotaan Terangun



*Halaman ini sengaja dikosongkan*



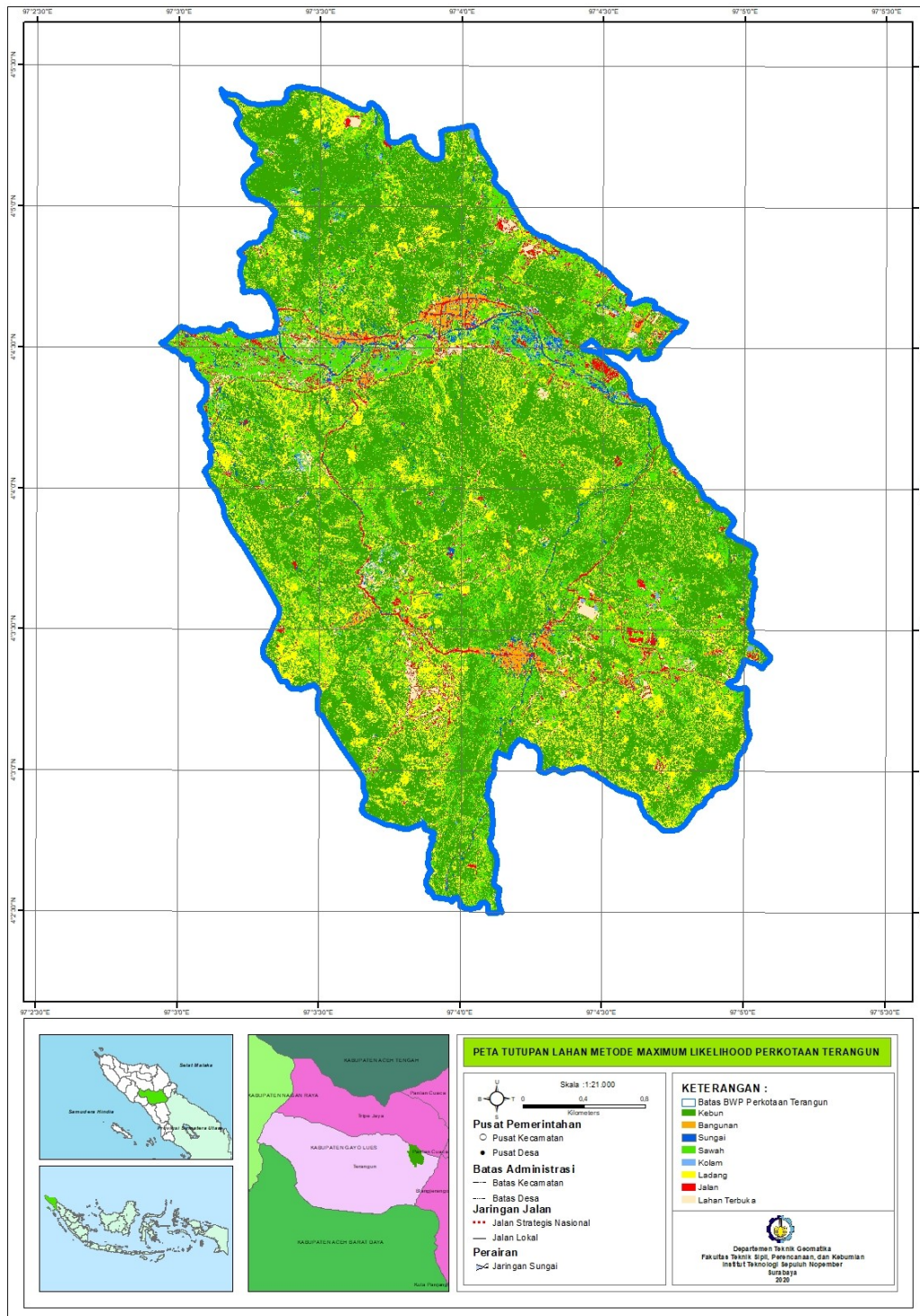
Lampiran 7 : Peta Klasifikasi Tutupan Lahan Model *Deep Learning*





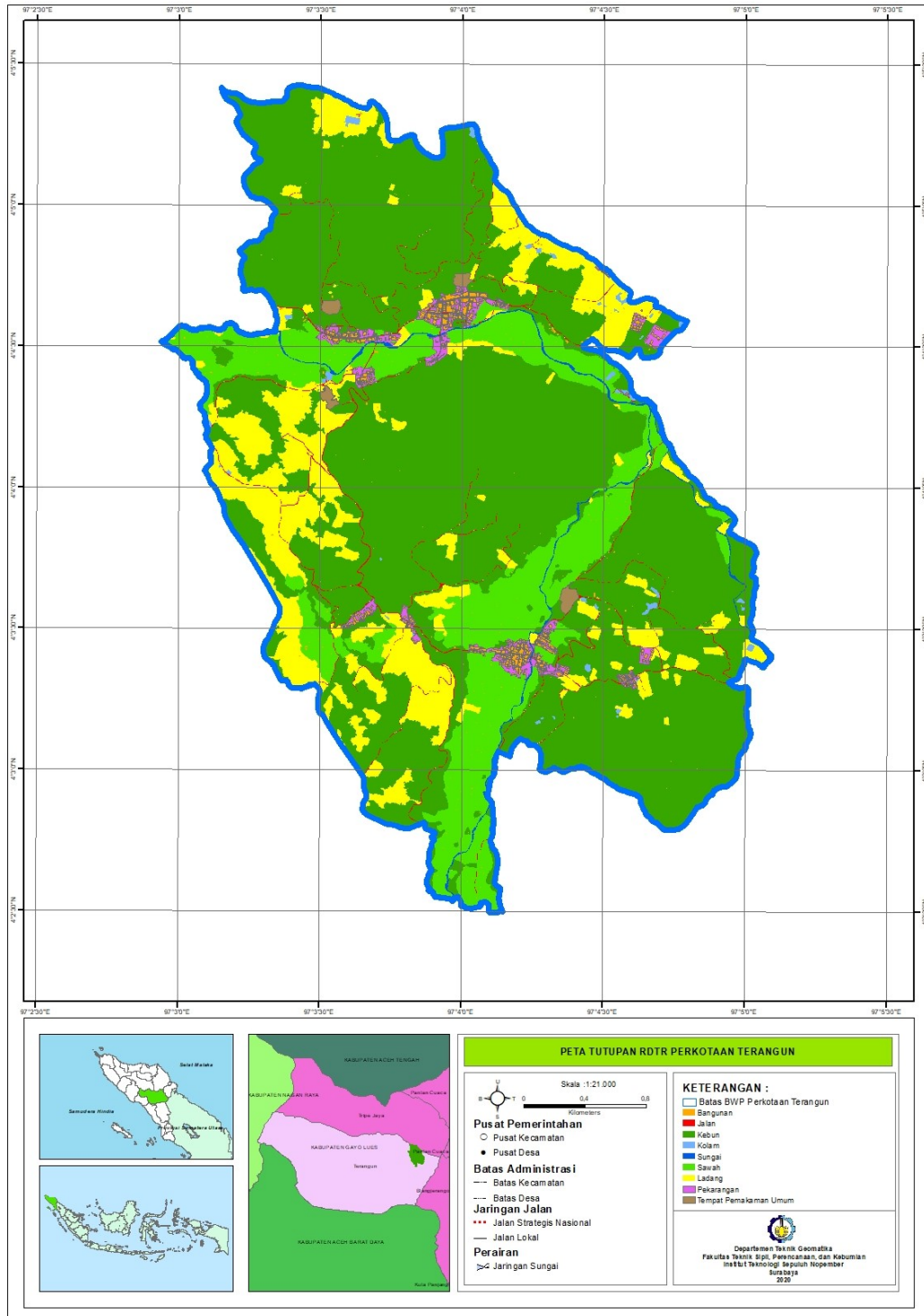
*Halaman ini sengaja dikosongkan*

Lampiran 8 : Peta Klasifikasi Tutupan Lahan Metode *Maximum Likelihood Classification*



*Halaman ini sengaja dikosongkan*

Lampiran 9 : Peta Tutupan Lahan RDTR Perkotaan Terangun



*Halaman ini sengaja dikosongkan*

## BIOGRAFI PENULIS



Penulis, Ilyas, dilahirkan di Paloh, pada tanggal 18 Juli 1985 merupakan anak pertama dari dua bersaudara. Penulis menempuh pendidikan dasar di SD Negeri Mangki, Kecamatan Simpang Tiga, Kabupaten Pidie, SMP Negeri 2 Simpang Tiga, Kabupaten Pidie, SMK Lilawangsa Sigli, (1 tahun), SMK Negeri 2 Sigli Prodi Mesin Produksi. Untuk strata pendidikan sarjana (S1) penulis menempuh pendidikan di Fakultas MIPA, Departemen Matematika, Bidang Matematika Komputasi di Universitas Syiah Kuala Banda Aceh. Setelah selesai S1 penulis banyak menghabiskan waktu di dunia kerja menjadi staf ahli di Bappeda Provinsi Aceh, PUPR Provinsi Aceh, Dosen di Sekolah Tinggi Ilmu Kehutanan (STIK) Pante Kulu Banda Aceh, staf ahli di Konsultan Tata Ruang, Konsultan Amdal dan juga sebagai Surveyor. Sementara untuk strata pendidikan magister (S2) penulis baru melanjutkan kembali pada tahun 2018 memilih untuk memperdalam ilmu pada bidang SIG dan Penginderaan Jauh dan memanfaatkan kemajuan teknologi *Artificial Intelligence* untuk mendukung kedua bidang tersebut dengan judul tesis “Analisis Klasifikasi Citra Satelit WorldView-2 Menggunakan Model *Deep Learning* (Studi Kasus: Peta tutupan Lahan Perkotaan Terangun, Kabupaten Gayo Lues, Provinsi Aceh”. Penulis dapat dihubungi melalui email [ilyaskaoy@gmail.com](mailto:ilyaskaoy@gmail.com)

*Halaman ini sengaja dikosongkan*