

**KLASIFIKASI AREA TUTUPAN LAHAN BERBASIS *NEURAL
NETWORK***

THESIS

**Oleh:
GALAN RAMADAN HARYA GALIB
NIM. 230605210014**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**KLASIFIKASI AREA TUTUPAN LAHAN BERBASIS *NEURAL
NETWORK***

THESIS

**Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Magister Komputer (M.Kom)**

**Oleh:
GALAN RAMADAN HARYA GALIB
NIM. 230605210014**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

KLASIFIKASI AREA TUTUPAN LAHAN BERBASIS *NEURAL NETWORK*

THESIS

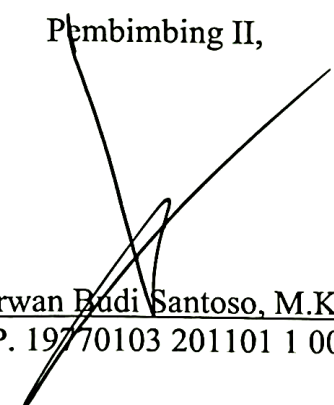
Oleh:
GALAN RAMADAN HARYA GALIB
NIM. 230605210014

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 5 Juni 2025

Pembimbing I,

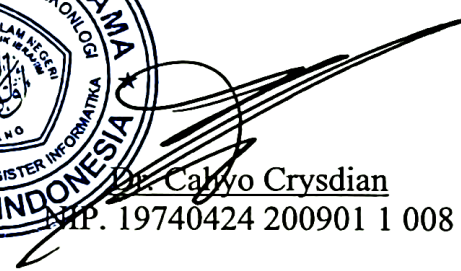

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

Pembimbing II,


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

KLASIFIKASI AREA TUTUPAN LAHAN BERBASIS *NEURAL NETWORK*

THESIS

Oleh:
GALAN RAMADAN HARYA GALIB
NIM. 230605210014

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 5 Juni 2025

Susunan Dewan Penguji

Penguji I	: <u>Dr. Zainal Abidin, M.Kom</u> NIP. 19760613 200501 1 004
Penguji II	: <u>Dr. Ir. Fresy Nugroho, M.T, IPM</u> NIP. 19710722 201101 1 001
Pembimbing I	: <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008
Pembimbing II	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004

()
()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang


Cahyo Crysdian
19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : GALAN RAMADAN HARYA GALIB
NIM : 230605210014
Fakultas / Prodi : Sains dan Teknologi / Magister Informatika
Judul Thesis : KLASIFIKASI AREA TUTUPAN LAHAN BERBASIS
NEURAL NETWORK

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 1 Juni 2025

Yang membuat pernyataan,

A handwritten signature in black ink is written over a yellow revenue stamp. The stamp features the Garuda Pancasila emblem, the text 'REPUBLIK INDONESIA', '1000', 'RUPIAH', and 'MEPERAI TEMPEL'. A unique alphanumeric code '3B5D1AKX425066440' is printed at the bottom of the stamp.

GALAN RAMADAN HARYA GALIB
NIM. 230605210014

MOTTO

“Dunia Hanya Sementara, Akhirat Selamanya”

*“Kumpulkanlah bekal amal kebaikan dari dunia yang sementara
ini untuk kehidupan yang abadi di akhirat
(yakni Surga Allah SWT)”*

HALAMAN PERSEMBAHAN

Dengan segenap rasa syukur *alhamdulillah rabbil 'alamin*, saya persembahkan hasil karya Thesis ini kepada:

1. Kedua orang tua saya, yakni Puji Hartono, M.Pd. dan Siti Komariyah, S.Pd. yang telah membimbing, mendoakan, dan menyayangi saya dari awal saya berada di dunia hingga saat ini dan kelak nanti.
2. Saudara saya, Arifan Qorrib Marto Raharjo yang telah memberi semangat dan dukungan kepada saya hingga sampai pada titik ini.
3. Para guru saya dari kecil hingga sekarang, terutama Ustadz Khozin, Ustadz Sulthon, dan Habib Muhammad bin Anies Shahab yang telah mendoakan dan mengajarkan ilmu yang bermanfaat untuk dunia dan akhirat serta telah mengenalkan saya tentang siapa itu Nabi Muhammad Shalallahu Alaihi Wassalam.
4. Dosen saya, Dr. Cahyo Crysdiyan. yang telah mengajari dan membukakan pintu wawasan yang sangat luas mengenai dunia riset dan akademik kepada saya.
5. Sahabat seperjuangan saya, Abdurrozzaq Ashshiddiqi Zuhri, Achmad Fahreza Alif Pahlevi, Aldian Faizzul Anwar, Aji Bagas Prakasa, Fahrendra Khoirul Ihtada, Revaldi Rahmatmulya, serta sahabat lainnya yang tidak dapat saya sebutkan satu persatu.

Semoga kita semua selalu mendapat petunjuk mengenai jalan yang lurus dari Allah serta semoga kita dapat mewujudkan harapan dan cita – cita yang telah diimpikan. Amin
Allahumma Amin.

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Segala puji hanya milik Allah Subhanahu Wa Ta'ala atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan Thesis yang berjudul “Klasifikasi Area Tutupan Lahan Berbasis *Neural Network*”. Semoga shalawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallallahu ‘Alaihi wa Sallam. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih yang tak terhingga kepada semua pihak-pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk dapat menyelesaikan Thesis ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian. selaku Ketua Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Cahyo Crysdian. selaku dosen pembimbing I dan Dr. Irwan Budi Santoso, M.Kom. selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan Thesis ini.
5. Dr. Zainal Abidin, M.Kom. selaku dosen penguji I dan Dr. Fresy Nugroho, MT. selaku dosen penguji II yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan Thesis dengan baik.

6. Segenap Dosen, Admin, Laboran dan Mahasiswa Program Studi Magister Informatika yang telah memberikan banyak dukungan dan bimbingan selama pengerjaan Thesis ini.
7. Ibu, Ayah, serta saudara saya yang selalu memberikan dukungan dan motivasi untuk terus berusaha, dan doa yang tak putus-putusnya selalu disampaikan agar dapat menuntaskan Thesis ini dengan lancar dan baik.

Akhir kata, penulis mengakui bahwa penulisan pada Thesis ini masih banyak kekurangan. Saya berharap semoga Thesis ini diterima sebagai amal ibadah yang tulus dan bermanfaat di sisi Allah Subhanahu Wa Ta'ala. Semoga karya ini menjadi bagian dari kontribusi yang tak terputus dalam rangka memperkuat dan mengembangkan ilmu pengetahuan, serta melaksanakan tugas sebagai hamba Allah yang berkomitmen.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 1 Juni 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Pernyataan Masalah	8
1.3 Tujuan Penelitian.....	8
1.4 Batasan Masalah	8
1.5 Manfaat Penelitian.....	8
BAB II KAJIAN PUSTAKA	10
2.1 Klasifikasi Area Tutupan Lahan	10
2.2 Kerangka Teori	14
BAB III METODOLOGI PENELITIAN	22
3.1 Kerangka Konseptual.....	22
3.2 Desain Penelitian.....	23
3.2.1 Pengumpulan Data	24
3.2.2 Desain Sistem	26
3.2.2.1 <i>Input Image</i>	27
3.2.2.2 <i>Preprocessing Citra</i>	27
A. Pemilihan ROI	27
B. <i>Patching</i>	28
3.2.2.3 Ekstraksi Fitur Menggunakan GLCM	28
3.2.2.4 Model <i>Neural Network</i>	32
3.2.3 Eksperimen	35
3.2.4 Evaluasi	36
3.3 Instrumen Penelitian	37
BAB IV <i>NEURAL NETWORK SATU HIDDEN LAYER</i>	39
4.1 Desain	39
4.2 Implementasi	46
4.3 Uji Coba NN Model A1.....	46
4.4 Uji Coba NN Model A2.....	50
4.5 Komparasi Performa NN Model A1 dan A2	54
BAB V <i>NEURAL NETWORK DUA HIDDEN LAYER</i>	57

5.1 Desain	57
5.2 Implementasi	63
5.3 Uji Coba NN Model B1	63
5.4 Uji Coba NN Model B2	67
5.5 Komparasi Performa NN Model B1 dan B2	70
BAB VI <i>NEURAL NETWORK</i> TIGA <i>HIDDEN LAYER</i>	73
6.1 Desain	73
6.2 Implementasi	81
6.3 Uji Coba NN Model C1	81
6.4 Uji Coba NN Model C2	85
6.5 Komparasi Performa NN Model C1 dan C2	89
BAB VII <i>NEURAL NETWORK</i> EMPAT <i>HIDDEN LAYER</i>	92
7.1 Desain	92
7.2 Implementasi	100
7.3 Uji Coba NN Model D1	101
7.4 Uji Coba NN Model D2	104
7.5 Komparasi Performa NN Model D1 dan D2	108
BAB VIII <i>NEURAL NETWORK</i> LIMA <i>HIDDEN LAYER</i>	112
8.1 Desain	112
8.2 Implementasi	123
8.3 Uji Coba NN Model E1	124
8.4 Uji Coba NN Model E2	128
8.5 Komparasi Performa NN Model E1 dan E2	131
BAB IX PEMBAHASAN.....	135
BAB X PENUTUP.....	154
10.1Kesimpulan	154
10.2Saran	155
DAFTAR PUSTAKA	156
LAMPIRAN.....	161

DAFTAR TABEL

Tabel 2.1. Kinerja model yang digunakan peneliti sebelumnya	16
Tabel 3.1. Contoh Citra Beresolusi Tinggi dan Ground Truth	25
Tabel 3.2. Komposisi Fitur GLCM	35
Tabel 3.3. Jumlah Neuron Pada Setiap Layer Neural Network	36
Tabel 3.4. Instrumen Penelitian	37
Tabel 4.1. Detail Hyperparameter Model NN A1 dan NN A2	46
Tabel 4.2. Beberapa Hasil Klasifikasi Model NN A1 Terhadap Citra Beresolusi Tinggi	48
Tabel 4.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN A1	49
Tabel 4.4. Performa Hasil Pengujian Model NN A1	50
Tabel 4.5. Beberapa Hasil Klasifikasi Model NN A2 Terhadap Citra Beresolusi Tinggi	52
Tabel 4.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN A2	53
Tabel 4.7. Performa Hasil Pengujian Model NN A2	54
Tabel 5.1. Detail Hyperparameter Model NN B1 dan NN B2	63
Tabel 5.2. Beberapa Hasil Klasifikasi Model NN B1 Terhadap Citra Beresolusi Tinggi	64
Tabel 5.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN B1	65
Tabel 5.4. Performa Hasil Pengujian Model NN B1	67
Tabel 5.5. Beberapa Hasil Klasifikasi Model NN B2 Terhadap Citra Beresolusi Tinggi	68
Tabel 5.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN B2	69
Tabel 5.7. Performa Hasil Pengujian Model NN B2	70
Tabel 6.1. Detail Hyperparameter Model NN C1 dan NN C2	81
Tabel 6.2. Beberapa Hasil Klasifikasi Model NN C1 Terhadap Citra Beresolusi Tinggi	82
Tabel 6.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN C1	84
Tabel 6.4. Performa Hasil Pengujian Model NN C1	85
Tabel 6.5. Beberapa Hasil Klasifikasi Model NN C2 Terhadap Citra Beresolusi Tinggi	87
Tabel 6.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN C2	88
Tabel 6.7. Performa Hasil Pengujian Model NN C2	89
Tabel 7.1. Detail Hyperparameter model NN D1 dan NN D2	100
Tabel 7.2. Beberapa Hasil Klasifikasi Model NN D1 Terhadap Citra Beresolusi Tinggi	102
Tabel 7.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN D1	103
Tabel 7.4. Performa Hasil Pengujian Model NN D1	104
Tabel 7.5. Beberapa Hasil Klasifikasi Model NN D2 Terhadap Citra Beresolusi Tinggi	105
Tabel 7.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN D1	106
Tabel 7.7. Performa Hasil Pengujian Model NN D2	108

Tabel 8.1. Detail Hyperparameter model NN E1 dan NN E2.....	123
Tabel 8.2. Beberapa Hasil Klasifikasi Model NN E1 Terhadap Citra Beresolusi Tinggi.....	125
Tabel 8.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN E1	126
Tabel 8.4. Performa Hasil Pengujian Model NN E1.....	127
Tabel 8.5. Beberapa Hasil Klasifikasi Model NN E2 Terhadap Citra Beresolusi Tinggi.....	129
Tabel 8.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN E2	130
Tabel 8.7. Performa Hasil Pengujian Model NN E2.....	131
Tabel 9.1. Perbandingan Rata-Rata Performa Antara Semua Model.....	137

DAFTAR GAMBAR

Gambar 1.1. Pertumbuhan Penduduk Dunia 1950 - 2023	2
Gambar 2.1. Kerangka Teori.....	15
Gambar 3.1. Kerangka Konseptual	22
Gambar 3.2. Desain Penelitian.....	24
Gambar 3.3. Desain Sistem.....	26
Gambar 3.4. Preprocessing Citra	27
Gambar 3.5. Contoh Hasil Pemilihan ROI Dari Setiap Kelas	28
Gambar 4.1. Arsitektur Neural Network 1 Hidden Layer.....	39
Gambar 4.2. Flowchart Neural Network.....	41
Gambar 4.3. Grafik Loss Training NN Model A1	47
Gambar 4.4. Grafik Loss Training NN Model A2	51
Gambar 4.5. Perbandingan Accuracy Antara Model NN A1 dan A2	55
Gambar 4.6. Perbandingan Precision Antara Model NN A1 dan A2	55
Gambar 4.7. Perbandingan Recall Antara Model NN A1 dan A2	55
Gambar 4.8. Perbandingan F1-Score Antara Model NN A1 dan A2.....	56
Gambar 4.9. Perbandingan IoU Antara Model NN A1 dan A2	56
Gambar 5.1. Arsitektur Neural Network 2 Hidden layer	57
Gambar 5.2. Grafik Loss Training NN Model B1	64
Gambar 5.3. Grafik Loss Training NN Model B2	67
Gambar 5.4. Perbandingan Accuracy Antara Model NN B1 dan B2	71
Gambar 5.5. Perbandingan Precision Antara Model NN B1 dan B2.....	71
Gambar 5.6. Perbandingan Recall Antara Model NN B1 dan B2	71
Gambar 5.7. Perbandingan F1-Score Antara Model NN B1 dan B2.....	72
Gambar 5.8. Perbandingan IoU Antara Model NN B1 dan B2	72
Gambar 6.1. Arsitektur Neural Network 3 Hidden Layer.....	73
Gambar 6.2. Grafik Loss Training NN Model C1	82
Gambar 6.3. Grafik Loss Training NN Model C2	86
Gambar 6.4. Perbandingan Accuracy Antara Model NN C1 dan C2	90
Gambar 6.5. Perbandingan Precision Antara Model NN C1 dan C2.....	90
Gambar 6.6. Perbandingan Recall Antara Model NN C1 dan C2	90
Gambar 6.7. Perbandingan F1-Score Antara Model NN C1 dan C2.....	91
Gambar 6.8. Perbandingan IoU Antara Model NN C1 dan C2	91
Gambar 7.1. Arsitektur Neural Network 4 Hidden Layer.....	92
Gambar 7.2. Grafik Loss Training NN Model D1	101
Gambar 7.3. Grafik Loss Training NN Model D2	105
Gambar 7.4. Perbandingan Accuracy Antara Model NN D1 dan D2.....	108
Gambar 7.5. Perbandingan Precision Antara Model NN D1 dan D2	109
Gambar 7.6. Perbandingan Recall Antara Model NN D1 dan D2.....	109
Gambar 7.7. Perbandingan F1-Score Antara Model NN D1 dan D2.....	109
Gambar 7.8. Perbandingan IoU Antara Model NN D1 dan D2.....	110
Gambar 8.1. Arsitektur Neural Network 5 Hidden Layer.....	112

Gambar 8.2. Grafik Loss Training NN Model E1	124
Gambar 8.3. Grafik Loss Training NN Model E2	128
Gambar 8.4. Perbandingan Accuracy Antara Model NN E1 dan E2.....	132
Gambar 8.5. Perbandingan Precision Antara Model NN E1 dan E2	132
Gambar 8.6. Perbandingan Recall Antara Model NN E1 dan E2.....	132
Gambar 8.7. Perbandingan F1-score Antara Model NN E1 dan E2	133
Gambar 8.8. Perbandingan IoU Antara Model NN E1 dan E2.....	133
Gambar 9.1. Distribusi Data ROI Masing-Masing Kelas	135
Gambar 9.2. Perbandingan Metrik Accuracy Antar Model Berdasarkan Jumlah Fitur GLCM	138
Gambar 9.3. Perbandingan Metrik Precision Antar Model Berdasarkan Jumlah Fitur GLCM	139
Gambar 9.4. Perbandingan Metrik Recall Antar Model Berdasarkan Jumlah Fitur GLCM	139
Gambar 9.5. Perbandingan Metrik F1-Score Antar Model Berdasarkan Jumlah Fitur GLCM	139
Gambar 9.6. Perbandingan Metrik IoU Antar Model Berdasarkan Jumlah Fitur GLCM	140
Gambar 9.7. Perbandingan Metrik Accuracy Antar Model Berdasarkan Jumlah Hidden Layer.....	141
Gambar 9.8. Perbandingan Metrik Precision Antar Model Berdasarkan Jumlah Hidden Layer.....	141
Gambar 9.9. Perbandingan Metrik Recall Antar Model Berdasarkan Jumlah Hidden Layer.....	142
Gambar 9.10. Perbandingan Metrik F1-Score Antar Model Berdasarkan Jumlah Hidden Layer.....	142
Gambar 9.11. Perbandingan Metrik IoU Antar Model Berdasarkan Jumlah Hidden Layer	143
Gambar 9.12. Hasil k-fold cross validation model B2 pada metrik accuracy dengan nilai k=10.....	145
Gambar 9.13. Hasil k-fold cross validation model B2 pada metrik precision dengan nilai k=10.....	146
Gambar 9.14. Hasil k-fold cross validation model B2 pada metrik recall dengan nilai k=10	147
Gambar 9.15. Hasil k-fold cross validation model B2 pada metrik F1-Score dengan nilai k=10.....	147
Gambar 9.16. Hasil k-fold cross validation model B2 pada metrik IoU dengan nilai k=10	148

ABSTRAK

Galib, Galan Ramadan Harya. 2025. **Klasifikasi Area Tutupan Lahan Berbasis *Neural Network***. Thesis. Jurusan Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Cahyo Crysdian, M.Cs. (II) Dr. Iwan Budi Santoso, M.Kom.

Kata kunci: *Klasifikasi, Tutupan Lahan, Neural Network.*

Penelitian ini bertujuan untuk mengembangkan model yang dapat melakukan klasifikasi area tutupan lahan terhadap citra beresolusi tinggi. Data yang digunakan bersumber dari ISPRS (*International Society for Photogrammetry and Remote Sensing*) Vaihingen. Model yang digunakan adalah *Neural Network* (NN). Pengujian dilakukan pada 10 skenario yang membandingkan penggunaan jumlah fitur GLCM sebanyak 5 dan 6 fitur, serta jumlah *hidden layer* pada arsitektur NN sebanyak 1 *hidden layer* hingga 5 *hidden layers*. Hasil menunjukkan bahwa dengan menambahkan jumlah fitur GLCM menjadi 6 fitur dapat meningkatkan performa model NN dalam melakukan klasifikasi area tutupan lahan. Kemudian pada pengujian tentang penggunaan jumlah *hidden layer*, didapatkan bahwa dengan melakukan penambahan *hidden layer* pada arsitektur NN tidak selalu dapat meningkatkan kinerja model, bahkan model dapat mengalami *overfitting* jika menggunakan terlalu banyak *layer*. Kinerja terbaik yang didapatkan oleh NN model B2 yang memanfaatkan 6 fitur GLCM dan 2 *hidden layer*, dengan nilai metrik *accuracy* sebesar 87.60%, *precision* sebesar 63.37%, *recall* sebesar 63.40%, *F1-Score* sebesar 63.39%, dan IoU sebesar 48.22%.

ABSTRACT

Galib, Galan Ramadan Harya. 2025. **Classification of Land Cover Area Based on Neural Network**. Thesis. Master Program of Computer Science, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Cahyo Crysdiyan, M.Cs. (II) Dr. Iwan Budi Santoso M. Kom.

This research aims to develop a model that can classify land cover areas on high-resolution images. The data used is sourced from ISPRS (International Society for Photogrammetry and Remote Sensing) Vaihingen. The model used is a Neural Network (NN). Tests were conducted on 10 scenarios that compared the use of 5 and 6 GLCM features and also compared the amounts of hidden layers on NN architecture from 1 hidden layer to 5 hidden layers. The results show that adding the GLCM feature to the 6 features can improve the performance of NN to do a land cover classification. Then, for the comparison of the amounts of hidden layers that are used on NN, it was found that adding hidden layers to the NN architecture does not always improve the model performance, the models could even face overfitting if NN uses too many layers. The best performance was obtained by the NN B2 model that used 6 GLCM features and 2 hidden layers, with an accuracy value of 87.60%, precision of 63.37%, recall of 63.40%, F1-Score of 63.39%, and IoU of 48.22%.

Keywords: *Classification, Neural Network, Land Cover.*

مستخلص البحث

غالب ، جالان رمضان هاريا. 2025. *تصنيف مساحة الغطاء الأرضي بناءً على الشبكة العصبية الأبطوحة*. برنامج ماجستير علوم الحاسب الآلي، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (I) الدكتور جاهيو كريسديان الماجستير ، (II) الدكتور إيوان بودي سانتوسو الماجستير.

الكلمات المفتاحية: التصنيف، والشبكة العصبية، والغطاء الأرضي.

يهدف هذا البحث إلى تطوير نموذج يمكنه تصنيف مناطق الغطاء الأرضي على صور عالية الدقة. وقد تم الحصول على البيانات المستخدمة من الجمعية الدولية للمسح التصويري والاستشعار عن بُعد *ISPRS* (الجمعية الدولية للمسح التصويري والاستشعار عن بُعد) *Vaihingen*. النموذج المستخدم هو شبكة عصبية (*NN*). أُجريت الاختبارات على 10 سيناريوهات قارنت بين استخدام 5 و 6 ميزات *GLCM*، وقارنت أيضًا بين كميات الطبقات المخفية على بنية *NN* من طبقة مخفية واحدة إلى 5 طبقات مخفية. تُظهر النتائج أن إضافة ميزة *GLCM* إلى الميزات الست يمكن أن يحسن أداء *NN* للقيام بتصنيف الغطاء الأرضي. ثم، بالنسبة لمقارنة كميات الطبقات المخفية المستخدمة في *NN*، وجد أن إضافة طبقات مخفية إلى بنية *NN* لا يحسن دائمًا من أداء النموذج، بل يمكن أن تواجه النماذج الإفراط في التركيب إذا استخدمت *NN* طبقات كثيرة. تم الحصول على أفضل أداء من خلال نموذج *NNB2* الذي استخدم 6 ميزات *GLCM* وطبقتين مخفيتين، حيث بلغت قيمة الدقة 87.60%، والدقة 63.73%، والاستدعاء 63.40%، و *F1-Score* 63.39%، و *IoU* 48.22%.

BAB I

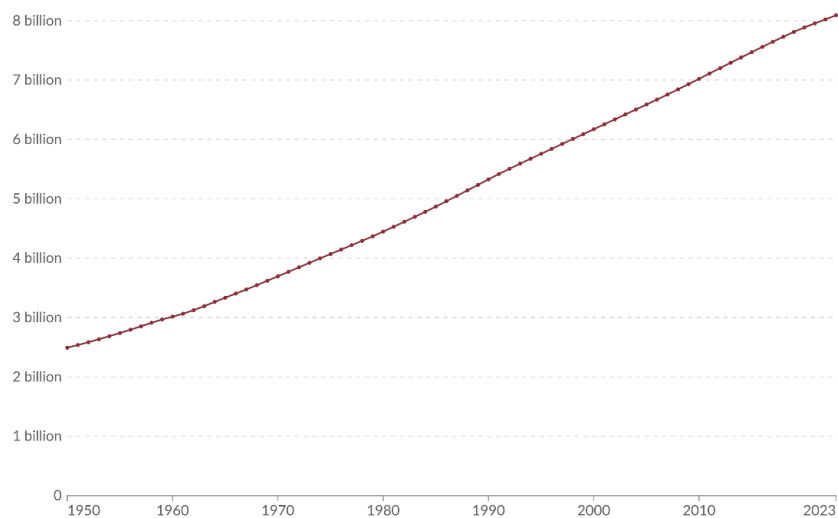
PENDAHULUAN

1.1 Latar Belakang

Meningkatnya jumlah penduduk suatu daerah memiliki dampak terhadap kebutuhan manusia akan lingkungan, khususnya dalam memanfaatkan lahan untuk mendirikan bangunan atau rumah. Pertumbuhan jumlah populasi atau penduduk di suatu tempat memiliki hubungan dengan Kebutuhan akan tempat tinggal atau rumah (Equere *et al.*, 2021). Peningkatan populasi akan mengakibatkan meningkatnya kebutuhan penggunaan lahan untuk membangun rumah, dan pembangunan rumah akan mengakibatkan terjadinya migrasi atau perpindahan masyarakat menuju tempat yang memiliki ketersediaan rumah (Mulder, 1996). Hal ini dibuktikan pada penelitian yang dilakukan oleh Seyam *et al.* (2023) bahwa meningkatnya populasi penduduk mengakibatkan meningkatnya urbanisasi dan industrialisasi, yang mana hal ini berdampak terhadap peningkatan kebutuhan penduduk disana akan tempat tinggal atau rumah. Selanjutnya Prawesti *et al.*, (2024) membandingkan laju pertumbuhan populasi dengan perubahan area tutupan lahan. Didapatkan bahwa peningkatan jumlah populasi berdampak terhadap peningkatan luas area hunian penduduk dan menurunnya luas area vegetasi. Pada penelitian lainnya yang dilakukan oleh Rachman *et al.* (2024) mengungkapkan bahwa terjadinya perubahan penggunaan lahan diakibatkan karena tumbuhnya populasi dengan pesat. Dampak dari pertumbuhan populasi yang pesat ini adalah

meningkatnya kebutuhan rumah dan bangunan, serta mengakibatkan menurunnya lahan vegetasi.

Menurut data yang diterbitkan oleh institusi *Our World in Data* yang bekerjasama dengan Universitas Oxford di Inggris pada tahun 2023 tentang jumlah penduduk dunia dengan rentang tahun 1950 hingga 2023, jumlah penduduk dunia pada tahun 2023 sebanyak 8.09 miliar jiwa (Ritchie *et al.*, 2023). Jumlah ini meningkat sebesar 225% dari tahun 1950 yang mana pada tahun tersebut jumlah penduduk dunia sebanyak 2.49 miliar jiwa. Grafik pertumbuhan penduduk dunia disajikan dalam Gambar 1.1.



Gambar 1.1. Pertumbuhan Penduduk Dunia 1950 - 2023

Beberapa peneliti telah mempelajari tentang fenomena meningkatnya jumlah penduduk dari waktu ke waktu dan dampaknya terhadap lingkungan. Wafiq & Suryanto (2021) melakukan analisa untuk mengetahui korelasi antara pertumbuhan kepadatan penduduk dengan dampaknya terhadap lingkungan. Dari penelitian tersebut didapati bahwa meningkatnya kepadatan penduduk memiliki dampak

negatif terhadap kualitas lingkungan. Selanjutnya, Wargadinata (2021) dalam penelitiannya juga menyatakan bahwa laju pertumbuhan penduduk seiringan dengan laju hilangnya kawasan hutan. Dengan kata lain semakin banyak jumlah manusia, maka hutan akan semakin hilang. Berkurangnya kawasan hutan atau deforestasi dan menurunnya kualitas lingkungan merupakan salah satu penyebab dari terjadinya peningkatan suhu bumi dan perubahan iklim (Lawrence *et al.*, 2022).

Deforestasi merupakan masalah serius yang harus diperhatikan. Pada penelitian Abbass *et al.* (2022) dinyatakan bahwa perubahan iklim yang terjadi dapat memberikan dampak negatif terhadap kesehatan masyarakat, pertanian, kehutanan, sektor pariwisata, serta berkurangnya keanekaragaman hayati dan habitat satwa. Terjadinya deforestasi atau berkurangnya lahan vegetasi juga dapat menyebabkan terjadinya banjir (Ramadhan *et al.*, 2023).

Berdasarkan penjelasan diatas, meningkatnya jumlah penduduk dan deforestasi merupakan dua masalah yang memiliki kesamaan, yakni berkaitan dengan tutupan lahan. Meningkatnya jumlah penduduk berdampak terhadap kebutuhan tanah sebagai lahan untuk mendirikan bangunan atau rumah, yang mana hal ini juga merupakan salah satu faktor terjadinya deforestasi. Untuk memitigasi masalah ini, klasifikasi area tutupan lahan merupakan salah satu kunci dalam analisa tata kelola area perkotaan maupun pedesaan, perencanaan penggunaan lahan, manajemen infrastruktur konstruksi, dan untuk membantu penanggulangan bencana alam (Yuan *et al.*, 2021).

Dalam beberapa dekade terakhir telah banyak peneliti menerapkan metode *machine learning* untuk melakukan klasifikasi area tutupan lahan dengan

menggunakan data citra beresolusi tinggi dan UAV (*Unmanned Aerial Vehicle*) atau *drone*. Civco (1993) mengambil citra satelit Landsat-TM pada daerah Coventry, Inggris. Hasil tangkapan citra tersebut kemudian diproses untuk dilakukan klasifikasi area tutupan lahan dengan membandingkan metode *maximum likelihood* dan NN (*Neural Network*). Selanjutnya Pal (2005) membandingkan kinerja metode SVM (*Support Vector Machine*) dan RF (*Random Forest*) dalam mengklasifikasikan area tutupan lahan di kota Cambridgeshire, UK (United Kingdom) yang diambil menggunakan satelit Landsat-7. Penelitian C. Yang *et al.* (2013) menggunakan citra beresolusi tinggi di dalam bidang agrikultur untuk memantau perkembangan dari tanaman yang ada di sawah. Selanjutnya penelitian oleh Teluguntla *et al.* (2018) mengklasifikasikan area tutupan lahan di China dan Australia dengan memanfaatkan citra dari satelit Landsat-8 dan Landsat-7 menggunakan metode RF. Kemudian Zhang *et al.* (2019) melakukan klasifikasi area tutupan lahan di UK, khususnya pada daerah Manchester dan Southampton menggunakan metode MLP (*Multi-layer Perceptron*), SVM, MRF (*Markov Random Field*), dan CNN (*Convolutional Neural Network*). Pada penelitian oleh Timilsina *et al.* (2020) memanfaatkan hasil citra satelit *Quickbird* dan *Google Earth* untuk melakukan pemetaan perubahan area tutupan pohon yang ada di daerah perkotaan Hobart, Tasmania, Australia dengan menggunakan metode CNN. Lalu pada bidang geologi, Eldosouky *et al.* (2021) memanfaatkan citra satelit ASTER untuk menemukan lokasi mineral tambang. Penelitian oleh (Si *et al.*, 2021) memanfaatkan citra beresolusi tinggi untuk memprediksi pergerakan awan. Dalam penelitian lainnya yang dilakukan oleh Rajkumar *et al.* (2022) memanfaatkan *drone*

untuk pengambilan citra beresolusi tinggi di area Utara Palm Beach, Florida. Hasil tangkapan citra kemudian diklasifikasikan area tutupan lahannya menggunakan metode CNN dengan membandingkan beberapa arsitektur. Selanjutnya, (Mujetahid *et al.*, 2023) memanfaatkan citra satelit dari Sentinel-1 dan Sentinel-2 melalui *Google Earth Engine* untuk memonitor dampak dari *illegal logging* yang mengakibatkan deforestasi pada kawasan hutan Provinsi Sulawesi Selatan.

Dalam konteks perbandingan kinerja metode dalam melakukan klasifikasi area tutupan lahan, Jozdani *et al.* (2019) mengimplementasikan beberapa metode *machine learning* untuk mengklasifikasikan area tutupan lahan dari citra beresolusi tinggi daerah Illionis, Amerika Serikat. Didapatkan kinerja *overall accuracy* metode RF sebesar 92.04%, GB (*Gradient Boosting*) sebesar 92.87%, XGB (*Extreme Gradient Boosting*) sebesar 92.94%, BT (*Bagging Trees*) sebesar 90.78%, SVM sebesar 92.45%, NN sebesar 93.64%, RAE (*Regular Autoencoder*) sebesar 92.45%, SAE (*Sparse Autoencoder*) sebesar 91.89%, VAE (*Variational Autoencoder*) sebesar 92.73%, dan CNN sebesar 91.41%. Berdasarkan hasil perbandingan antara kinerja dari masing-masing metode, metode NN mendapatkan kinerja terbaik dibandingkan metode lainnya.

Pada penelitian-penelitian sebelumnya tidak dilakukan eksplorasi terhadap metode ekstraksi fitur, yang mana secara spesifik para peneliti pada penelitian-penelitian sebelumnya membandingkan kinerja metode *machine learning* satu dengan lainnya, atau membandingkan arsitektur *machine learning* satu dengan lainnya tanpa membandingkan efek dari fitur atau metode ekstraksi fitur yang digunakan. Dalam pemrosesan citra, Kumar dan Bhatia (2014) mengungkapkan

bahwa ekstraksi fitur merupakan proses mendeskripsikan informasi yang relevan dari *pattern* atau pola sebuah citra menjadi dimensi atau ukuran yang lebih kecil. Pada konteks *supervised learning*, fitur disini memiliki informasi atau label yang bisa digunakan untuk membedakan antara satu kelas dengan kelas lainnya. Sehingga akan memudahkan proses klasifikasi yang dilakukan oleh model *machine learning*. Salah satu metode ekstraksi fitur yang digunakan oleh peneliti untuk data citra beresolusi tinggi adalah GLCM (*Gray-Level Co-occurrence Matrix*). Lan dan Liu (2018) mengimplementasikan metode GLCM untuk mengekstrak fitur tekstur dari citra beresolusi tinggi dengan area studi Hainan, China. Diterapkan *windowing* dalam ekstraksi fitur GLCM agar bisa mendapatkan fitur dari area yang kecil. Namun kelemahan dari teknik ini adalah bercampurnya piksel dari kelas satu dengan kelas lainnya di dalam *window* sehingga menimbulkan ambiguitas dan menurunkan kinerja dari model *machine learning*.

Berdasarkan uraian diatas, penelitian ini berfokus pada kinerja metode NN dengan memanfaatkan hasil ekstraksi fitur dari GLCM untuk mengklasifikasikan area tutupan lahan pada citra beresolusi tinggi. Kemudian untuk menghilangkan ambiguitas karena bercampurnya piksel antara satu kelas dengan kelas lainnya dalam satu *window*, penelitian ini menggunakan *Region of Interest* (ROI) pada setiap kelas. Setelah ROI dari setiap kelas ditentukan, kemudian dilakukan *windowing* untuk mendapatkan fitur dari area kecil untuk ROI kelas yang telah disiapkan sebelumnya. Data yang digunakan dalam penelitian ini adalah data citra beresolusi tinggi yang diambil menggunakan UAV atau *drone* yang bersumber dari ISPRS (*International Society for Photogrammetry and Remote Sensing*). Salah satu

alasan dilakukannya penelitian ini adalah untuk membantu memenuhi ikhtiar manusia sebagai umat Muslim dalam memenuhi amanat dari Allah untuk menjaga lingkungan. Allah bersabda dalam Al-Qur'an:

إِنَّا عَرَضْنَا الْأَمَانَةَ عَلَى السَّمَوَاتِ وَالْأَرْضِ وَالْجِبَالِ فَأَبَيْنَ أَنْ يَحْمِلْنَهَا وَأَشْفَقْنَ مِنْهَا وَحَمَلَهَا الْإِنْسَانُ إِنَّهُ كَانَ ظَلُومًا جَهُولًا ﴿٧٧﴾

“Sesungguhnya Kami telah menawarkan amanat kepada langit, bumi, dan gunung-gunung; tetapi semuanya enggan untuk memikul amanat itu dan mereka khawatir tidak akan melaksanakannya. Lalu, dipikullah amanat itu oleh manusia. Sesungguhnya ia (manusia) sangat zalim lagi sangat bodoh.” (QS. Al-Ahzab [33]: 72).

Pada penelitian yang dilakukan oleh Achmad Gusyairi pada tahun 2022, diungkapkan bahwa *At-Tabari* sebagaimana dinukil oleh Aisyah Abdurrahman, menjelaskan bahwa amanat yang dimaksud dalam QS. Al-Ahzab [33]: 72, secara umum diartikan sebagai seluruh amanat-amanat di dalam agama dan dalam kehidupan manusia. Dalam konteks menjaga lingkungan hidup, salah satu amanat dari Allah kepada manusia untuk menjaga lingkungan hidup tertuang di dalam Al-Qur'an sebagai berikut:

وَلَا تُفْسِدُوا فِي الْأَرْضِ بَعْدَ إِصْلَاحِهَا وَادْعُوهُ خَوْفًا وَطَمَعًا إِنَّ رَحْمَتَ اللَّهِ قَرِيبٌ مِّنَ الْمُحْسِنِينَ ﴿٥٦﴾

“Dan janganlah kamu berbuat kerusakan di bumi setelah (diciptakan) dengan baik. Berdoalah kepada-Nya dengan rasa takut dan penuh harap. Sesungguhnya rahmat Allah sangat dekat kepada orang yang berbuat kebaikan.” (QS. Al-A'raf [7]: 56).

Ayat diatas menjelaskan tentang larangan manusia untuk berbuat kerusakan di bumi, yang mana berbuat kerusakan adalah salah satu bentuk dari perbuatan yang melampaui batas. Larangan membuat kerusakan yang ada di dalam ayat ini mencakup semua bidang, seperti merusak jasmani dan rohani orang lain, merusak

lingkungan hidup, merusak sumber penghidupan (perdagangan, pertanian, dan lain-lain), dan sebagainya. Penjelasan ayat diatas bersumber dari penelitian yang dilakukan oleh Muhammad Mu'tiq Rosyadi (2020) yang menggunakan *Tafsir al-Misbah* karya M. Quraish Shihab.

1.2 Pernyataan Masalah

1. Faktor apa saja yang mempengaruhi kinerja model NN dalam melakukan klasifikasi area tutupan lahan?
2. Strategi apa yang paling optimal untuk melakukan klasifikasi area tutupan lahan menggunakan model NN?

1.3 Tujuan Penelitian

1. Mengetahui faktor apa saja yang mempengaruhi kinerja model NN dalam melakukan klasifikasi area tutupan lahan.
2. Mendapatkan strategi terbaik untuk melakukan klasifikasi area tutupan lahan menggunakan model NN.

1.4 Batasan Masalah

Dataset yang digunakan merupakan citra beresolusi tinggi yang diambil dari ISPRS Vaihingen.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini diharapkan dapat memberikan manfaat di masa depan, yaitu:

1. Bagi Kementrian Lingkungan Hidup dan Kehutanan Indonesia dalam memonitor perubahan tutupan lahan dengan lebih cepat dan efisien.

2. Bagi Kementrian Agraria dan Tata Ruang Indonesia dalam melaksanakan pedoman penyediaan dan pemanfaatan ruang terbuka hijau di kawasan perkotaan.
3. Bagi Badan Pusat Statistik Indonesia dalam menyediakan data akurat terkait perubahan tutupan lahan untuk analisis dan kebijakan terkait lingkungan dan kehutanan.

BAB II

KAJIAN PUSTAKA

2.1 Klasifikasi Area Tutupan Lahan

Penelitian terkait klasifikasi area tutupan lahan telah banyak dilakukan pada penelitian sebelumnya dengan berbagai metode. Penelitian oleh Jozdani *et al.* (2019) membandingkan kinerja metode RF, GB, XGB, BT, SVM, MLP atau NN, RAE, SAE, VAE, dan CNN untuk memetakan atau mengklasifikasi area tutupan lahan pada area perkotaan berbasis citra beresolusi tinggi. Didapatkan bahwa kinerja metode NN merupakan yang terbaik, dengan nilai OA sebesar 93.64%, KC 92.3%, dan *F1-Score* 93.7%.

Selanjutnya Abdalla *et al.* (2019) menyatakan klasifikasi area tutupan lahan pada bidang pertanian bermanfaat untuk berbagai aplikasi seperti deteksi penyakit tanaman, estimasi fraksi vegetasi, pemantauan pertumbuhan tanaman, dan manajemen lokasi. Dari pengujian yang dilakukan, didapatkan hasil dari model *encoder-decoder* berbasis VGG16 Net memiliki kinerja yang bagus dalam klasifikasi area tanaman lobak biji minyak diantara gulma dengan kepadatan tinggi. Didapati juga bahwa augmentasi data dapat meningkatkan akurasi klasifikasi area tutupan lahan dan kinerja keseluruhan model. Arsitektur *encoder-decoder* berbasis VGG16 Net mendapatkan nilai *F1-Score* sebesar 93%, *accuracy* 93%, dan *Intersection over Union* (IoU) 87%.

Hasil penelitian Li *et al.* (2021) mengungkapkan bahwa klasifikasi area tutupan lahan menghadapi tantangan tersendiri dalam gambar penginderaan jarak

jauh resolusi tinggi karena varians intraklas atau variasi dari data di dalam kelompok atau kelas tertentu memiliki keberagaman variasi serta objek tanah yang kompleks. Untuk menyelesaikan permasalahan ini, peneliti mengusulkan arsitektur SCAttNet yang memiliki kompleksitas komputasi lebih kecil, serta model ini praktis dan efisien karena hanya memiliki sejumlah kecil parameter. Dari pengujian yang dilakukan, model menunjukkan kinerja yang baik dengan nilai *F1-Score* sebesar 82.06%, OA 85.47%, IoU 70.2%.

Selanjutnya pada penelitian yang dilakukan Adugna *et al.* (2022) menerapkan metode SVM dengan kernel RBF (*Radial Basis Function*) untuk mengklasifikasikan citra area tutupan lahan pada daerah Afrika yang ditangkap dari satelit FengYun-3C. Adugna *et al.* (2022) melakukan pengujian berdasarkan variasi nilai parameter C dan γ (Gamma) untuk mendapatkan kombinasi dengan kinerja model paling optimal. Didapatkan nilai parameter C paling optimal sebesar 300 dan untuk parameter γ nilai default dari *library* sklearn yang digunakan adalah paling optimal. Dari pengujian yang dilakukan, didapatkan nilai OA sebesar 85%, F1-Score 85%, dan *Kappa Coefficient* (KC) 81%.

Menurut Yu *et al.* (2022) masalah utama dalam melakukan klasifikasi area tutupan lahan berbasis citra beresolusi tinggi adalah kompleksnya tekstur dan pola spasial pada lahan, sehingga ketika diklasifikasikan metode tidak bisa menangkap area tutupan lahan secara detail. Yu *et al.* (2022) pada penelitiannya mengusulkan metode DUA-Net yang merupakan penggabungan antara U-Net dan DenseAPP (*Densely connected Atrous Spatial Pyramid Pooling*). Dari proses pengujian yang

dilakukan, metode DUA-Net mendapatkan nilai OA sebesar 89.71%, F1-Score 86.64%, dan IoU 82.31%.

Selanjutnya Su *et al.* (2022) berupaya meningkatkan kinerja model untuk klasifikasi area tutupan lahan pada citra pemandangan alam. Peneliti mengusulkan arsitektur yang merupakan gabungan dari DenseNet, U-Net, dilated convolution dan DeconvNet yang kemudian dinamai AtrousDenseUDeconvNet. Model yang diusulkan menunjukkan peningkatan kinerja yang signifikan dibandingkan model U-Net. Dari segi kecepatan, model yang diusulkan kecepatannya meningkat sekitar 1.18x. Lalu dari segi jumlah parameter juga lebih sedikit dari U-Net, artinya kompleksitas dari model berkurang. Dari pengujian yang dilakukan, didapatkan nilai *recall* sebesar 90.06%, *precision* 88.3%, *F1-Score* 89.4%, *accuracy* 86.2%, IoU 80.09%. Namun, model yang diusulkan tidak dapat melakukan klasifikasi area tutupan lahan pada objek mobil, dan lebih cocok untuk objek berukuran besar dan luas.

Pada penelitian oleh Basheer *et al.* (2022) diungkapkan klasifikasi area penggunaan tutupan lahan merupakan satu hal vital dalam manajemen sumber daya alam dan untuk mempelajari perubahan alam yang berpengaruh terhadap iklim. Untuk bisa mendapatkan akurasi dan kinerja yang efisien, dibutuhkan dataset dan metode yang kuat untuk menangani tugas klasifikasi tutupan lahan. Dalam penelitiannya, Basheer *et al.* (2022) membandingkan kinerja metode *Random Forest* (RF), SVM, dan *Maximum Likelihood* (ML) untuk melakukan klasifikasi area tutupan lahan pada citra beresolusi tinggi. Dari hasil pengujian dan

perbandingan metode yang dilakukan, didapatkan bahwa metode SVM merupakan metode terbaik dengan nilai OA sebesar 92%, dan KC 83%.

Selanjutnya pada penelitian Yang *et al.* (2022) menyatakan bahwa masalah yang sering dihadapi ketika melakukan klasifikasi area tutupan lahan pada citra beresolusi tinggi adalah kategori atau kelas objek yang tidak seimbang, kompleksnya objek yang ada di dalam citra sehingga sulit untuk diidentifikasi, tidak jelasnya batas antara objek satu atau area satu dengan lainnya, dan lain-lain. Untuk menyelesaikan permasalahan ini, Yang *et al.* (2022) mengusulkan metode AD-HRNet yang merupakan kombinasi dari HRNet dengan *attention mechanism* dan *dilated convolution*. Hasil dari pengujian metode AD-HRNet yang dilakukan pada dataset ISPRS Vaihingen mendapatkan nilai OA sebesar 88.10%, F1-Score 82.52%, dan IoU 71.58%.

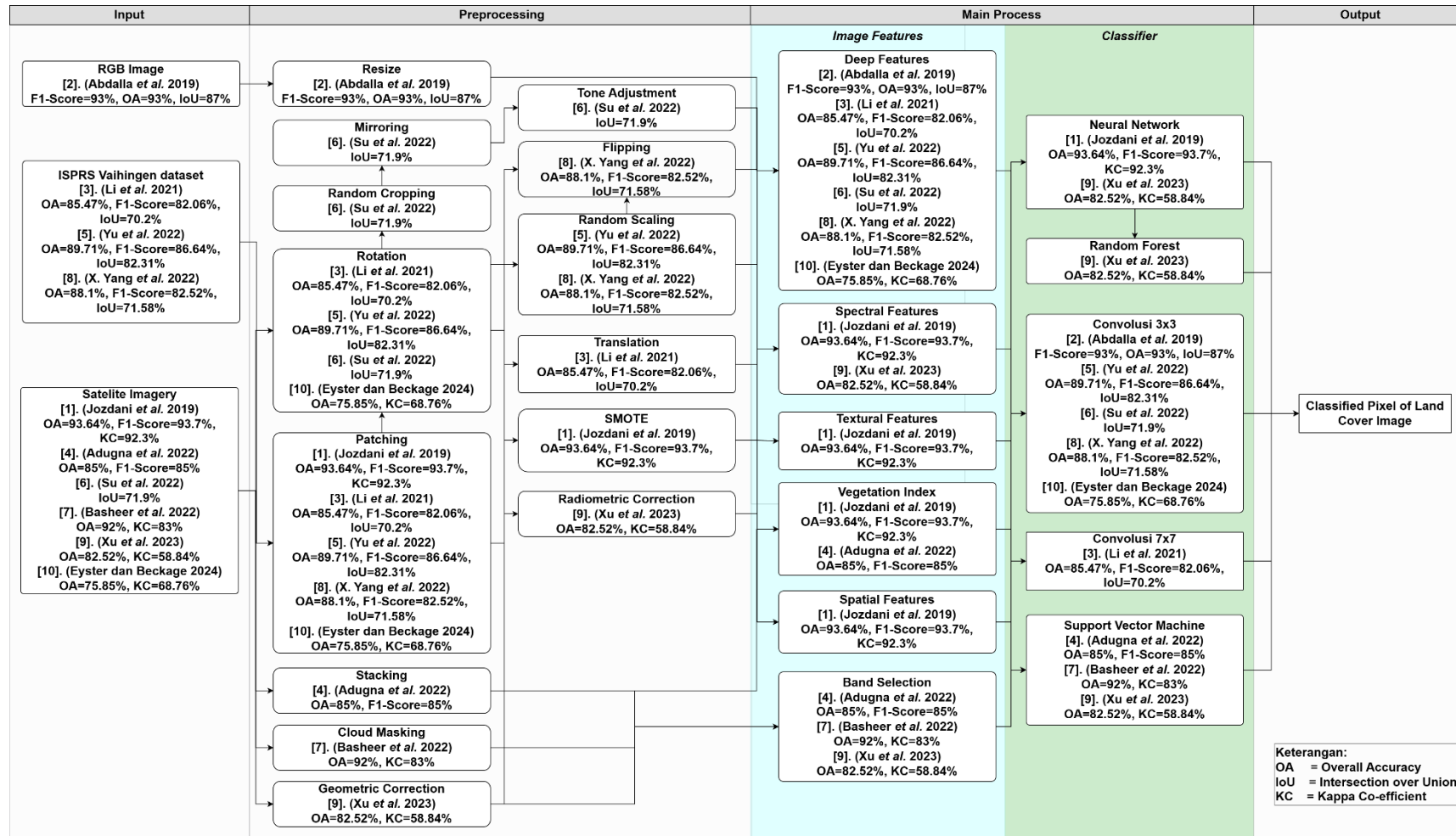
Menurut Xu *et al.* (2023) metode *deep learning* memiliki berbagai kekurangan seperti kompleksitas yang tinggi, biaya yang mahal, dan membutuhkan waktu tunggu yang lama untuk mendapatkan hasil. Karena itu dalam penelitiannya, Xu *et al.* (2023) mengusulkan dan mengembangkan metode *machine learning* untuk meningkatkan performanya dalam tugas klasifikasi area tutupan lahan. Metode yang diusulkan merupakan gabungan dari *Neural Network* (NN) dan RF, kemudian diberi nama ANN-RF. Dalam pengujian metode ANN-RF yang dilakukan pada citra satelit Sentinel-2A-2016 didapatkan nilai OA sebesar 82.52%, dan KC 58.54%.

Selanjutnya pada penelitian Eyster dan Beckage (2024) menyatakan bahwa beberapa tantangan dari klasifikasi area tutupan lahan adalah karena dataset historis

atau dataset citra beresolusi tinggi pada masa lampau memiliki kualitas yang buruk, kurangnya data yang telah dianotasi untuk digunakan *training* model, dan perbedaan bentuk atau area tutupan lahan pada masa lampau dan masa kini. Untuk menyelesaikan permasalahan ini diusulkanlah model DeepLabv3+ dengan memanfaatkan *transfer learning* dan *data augmentation*. Tahap awal dari *training* pada model adalah dengan menggunakan data citra beresolusi tinggi pada masa kini, kemudian dilakukan penyempurnaan pada dataset citra masa lampau yang dianotasi secara manual. Kemudian untuk *data augmentation* yang digunakan adalah rotasi. Hasil dari pengujian menunjukkan bahwa data citra masa lampau dapat diklasifikasi dengan lebih baik. Didapatkan kinerja dari pengujian model berupa OA sebesar 75.85%, dan KC sebesar 68.76%.

2.2 Kerangka Teori

Bagian ini membahas teori pendukung terkait klasifikasi area tutupan lahan untuk menentukan parameter dan variable sebagai data masukan dan menyelidiki penggunaan beberapa pendekatan berbasis *machine learning* dan apa keluaran hasil kerangka teori tersebut. Kerangka teori untuk klasifikasi lahan disajikan pada Gambar 2.1.



Gambar 2.1. Kerangka Teori

Beberapa metrik evaluasi seperti OA, IoU, *F1-Score* (Fs), dan KC dapat memberikan penilaian terhadap kinerja model *machine learning*. Rangkuman hasil evaluasi kinerja model yang digunakan oleh para peneliti sebelumnya disajikan pada Tabel 2.1.

Tabel 2.1. Kinerja model yang digunakan peneliti sebelumnya

No	Strategi	Peneliti	OA	IoU	Fs	KC
1	NN	Jozdani <i>et al.</i> (2019)	93.64%	-	93.7%	92.3%
2	Encoder-Decoder VGG16-Net	Abdalla <i>et al.</i> (2019)	93%	87%	93%	-
3	SCAttNet	Li <i>et al.</i> (2021)	85.47%	70.2%	82.06%	-
4	SVM-RBF	Adugna <i>et al.</i> (2022)	85%	-	85%	81%
5	DUA-Net	Yu <i>et al.</i> (2022)	89.71%	82.31%	86.64%	-
6	AtrousDenseUDEconvNet	Su <i>et al.</i> (2022)	86.2%	80.09%	89.4%	-
7	SVM	Basheer <i>et al.</i> (2022)	92%	-	-	83%
8	AD-HRNet	Yang <i>et al.</i> (2022)	88.10%	71.58%	82.52%	-
9	ANN-RF	Xu <i>et al.</i> (2023)	82.52%	-	-	58.54%
10	DeepLabv3+	Eyster dan Beckage (2024)	75.85%	-	-	68.76%

Tabel 2.1 menunjukkan kinerja prediksi kelas piksel citra digital yang diuji dan diukur menggunakan berbagai pendekatan berdasarkan metode *machine learning* dalam melakukan klasifikasi area tutupan lahan oleh para peneliti. Untuk mendapatkan hasil kinerja tersebut, para peneliti memiliki strategi pelatihan dan pengujian model yang berbeda-beda. Selain itu, meskipun beberapa peneliti ada yang menggunakan jenis dataset sama, namun strategi pembagiannya juga berbeda-beda.

Strategi yang pertama adalah penelitian yang dilakukan oleh Jozdani *et al.* (2019) membandingkan kinerja beberapa metode *machine learning* untuk melakukan klasifikasi area tutupan lahan. Didapatkan metode dengan kinerja

terbaik adalah NN. Dataset citra beresolusi tinggi yang telah didapatkan dibagi atau dilakukan proses *patching* dengan ukuran 224x224. Kemudian mereka menentukan nilai *learning rate* sebesar 0.001 dengan *optimizer* Adam untuk meminimalkan nilai *error*. Kemudian untuk arsitektur dari NN yang digunakan dalam penelitian oleh Jozdani *et al.* (2019) memiliki *hidden layer* sebanyak 3 dengan masing-masing *layer* memiliki 100 *neurons*.

Penelitian selanjutnya yang dilakukan oleh Abdalla *et al.* (2019) memodifikasi model VGG16 Net untuk melakukan klasifikasi piksel terhadap citra RGB. Mereka mengatur *learning rate* sebesar 0.001, iterasi diatur sejumlah 100, dan *loss function* yang digunakan adalah *cross entropy*. Jumlah data yang mereka gunakan adalah 400 citra berdimensi 400x500 (setelah dilakukan *resize*) dengan pembagian 60% untuk *training*, 20% untuk *validation* dan 20% sisanya untuk *testing*. Mereka menambahkan aturan, jika *validation loss* sudah berhenti berkurang dalam empat iterasi beruntun, maka proses *training* akan dihentikan.

Sedangkan dalam penelitian yang dilakukan oleh Li *et al.* (2021) mengusulkan metode SCAttNet untuk melakukan klasifikasi piksel terhadap dataset ISPRS Vaihingen dan Potsdam. Strategi mereka membagi data citra ISPRS Vaihingen menjadi dua, citra dengan label ID 30, 32, 34, dan 37 digunakan untuk mengevaluasi performa dari model, dan dua belas citra lainnya digunakan untuk pelatihan model. Citra ISPRS Vaihingen melalui tahap *preprocessing*, yaitu *cropping* setiap dimensi 256x256, yang kemudian dilakukan rotasi dan operasi translasi, akhirnya didapatkan 12.000 bagian citra yang digunakan untuk pelatihan model. Lalu untuk dataset ISPRS Potsdam juga dibagi menjadi dua bagian, citra

dengan label ID 2_12, 3_12, 4_12, 5_12, 6_12, dan 7_12 digunakan untuk *testing*, dan delapan belas citra lainnya digunakan untuk pelatihan model. Dilakukan *random cropping* dan didapatkan bagian citra sejumlah 27.000 dengan dimensi 256x256 yang digunakan untuk pelatihan model. Untuk pelatihan model, peneliti mengatur nilai *learning rate* pada dataset Vaihingen dan Potsdam masing-masing sebesar 0.003 dan 0.004. *Optimizers* yang digunakan adalah Adam, untuk *loss function* adalah *cross entropy*, dan jumlah epochnya sebanyak 50.

Pada penelitian lainnya oleh Adugna *et al.* (2022) membandingkan kinerja RF dan SVM untuk melakukan klasifikasi area tutupan lahan, didapatkan hasil bahwa metode SVM memiliki kinerja yang lebih baik dibandingkan dengan RF. Dilakukan pembagian untuk data *training* sebesar 75% dan untuk *testing* sebesar 25% dari total dataset. Mereka menggunakan kernel RBF (*Radial Basis Function*) untuk SVM dalam mempelajari pola data. Nilai parameter C paling optimal yang didapatkan adalah sebesar 300 dengan total durasi waktu *training* selama 56 jam.

Selanjutnya Yu *et al.* (2022) menyatakan bahwa klasifikasi area tutupan lahan di wilayah perkotaan merupakan salah satu kunci untuk membantu pekerjaan dan aplikasi perencanaan wilayah dan administrasi kota. Diusulkan arsitektur DUA-Net yang merupakan penggabungan dari U-Net dan DenseASPP (*Densely connected Atrous Spatial Pyramid Pooling*). Mereka membagi citra beresolusi tinggi dengan *patching* sehingga didapatkan masing-masing citra *patch* memiliki ukuran 256x256. Hasil *patching* tersebut kemudian dibagi 60% untuk data *training*, 20% untuk data *validation*, dan 20% sisanya untuk data *testing*. Digunakan *optimizer* SGD (*Stochastic Gradient Descent*) dengan nilai *learning rate* sebesar 0.001.

Karena keterbatasan memori atau *computation resources*, mereka menetapkan nilai *batch size* sebesar 8. Kemudian untuk *loss function* yang digunakan adalah *Cross-Entropy*.

Pada penelitian yang dilakukan oleh Su *et al.* (2022) mengusung arsitektur U-Net yang ditingkatkan dengan mengkombinasikan DenseNet, *dilated convolution*, dan DeconvNet yang kemudian diberi nama AtrousDenseUDEconvNet. Dataset yang mereka gunakan adalah ISPRS Potsdam. Untuk menghindari overfitting pada model, data citra tersebut dilakukan augmentasi seperti *random cropping*, *rotation*, *mirroring*, penyesuaian pencahayaan, penyesuaian kontras, penyesuaian chroma, dan menambahkan *noise*. Didapatkan data citra sejumlah 120.000 dengan tiga channel warna yang berukuran 256x256. Pembagian data menjadi tiga, yaitu *training set*, *validation set*, dan *test set* yang secara berurutan rasionya adalah 4:1:1. Pada proses pelatihan, Su *et al.* (2022) menggunakan *cross entropy* sebagai *loss function* dengan Adam sebagai *optimizer*. Nilai *learning rate* diatur menjadi 0.001. *Early stopping* dan penyesuaian *learning rate* dilakukan seiringan dengan proses *training model*.

Penelitian oleh Basheer *et al.* (2022) menyatakan bahwa klasifikasi area tutupan lahan merupakan hal yang penting dan vital untuk manajemen sumber daya alam dan juga untuk mempelajari bagaimana perubahan alam yang terjadi sehingga mengakibatkan perubahan iklim. Untuk menyelesaikan permasalahan ini, diusulkan perbandingan antara metode SVM, RF, dan ML. Dari hasil pengujian didapatkan SVM merupakan metode dengan kinerja yang terbaik. Basheer *et al.*

(2022) menggunakan kernel *linear* pada SVM untuk melakukan klasifikasi area tutupan lahan.

Selanjutnya pada penelitian yang dilakukan oleh Yang *et al.* (2022) mengusulkan model AD-HRNet, yang mana merupakan kombinasi dari HRNet dengan *attention mechanism* dan *dilated convolution*. Dalam penelitiannya, mereka menggunakan dataset ISPRS Potsdam dan ISPRS Vaihingen. Dataset ISPRS Potsdam mereka bagi menjadi dua, yaitu 24 citra untuk digunakan *training* dan 14 citra digunakan untuk *testing*. Karena ukuran citra yang besar (6000x6000 pixel), maka citra dibagi menjadi *patch* berukuran 512x512, sehingga didapatkan 5400 citra untuk *training* dan 3150 citra untuk *testing*. Selanjutnya untuk dataset ISPRS Vaihingen berjumlah 33 citra *remote sensing* yang ukurannya berbeda, yaitu diantara 1996x1995 hingga 3816x2550 dibagi menjadi dua bagian, 16 citra untuk *training* dan 17 citra untuk *testing*. Citra dibagi menjadi path berukuran 512x512, sehingga didapatkan 479 citra untuk *training* dan 555 citra untuk *testing*. Pada proses *training*, peneliti menggunakan *optimizer* SGD (*Stochastic Gradient Descent*) dengan nilai *learning rate* 0.01. Untuk *loss function* yang digunakan adalah *weighted cross-entropy*. Terakhir untuk jumlah epoch pada Dataset ISPRS Potsdam sejumlah 484, dan untuk ISPRS Vaihingen sejumlah 100.

Pandangan berbeda dikemukakan oleh Xu *et al.* (2023), dalam penelitiannya dinyatakan bahwa perkembangan *deep learning* memiliki banyak sekali variasi. Tidak sedikit dari variasi model atau arsitektur *deep learning* memiliki kompleksitas yang terus meningkat, membutuhkan *computational resources* yang tinggi, dan membutuhkan waktu yang lama untuk mendapatkan hasil. Untuk itu

diusulkan gabungan dari metode NN dan RF yang memiliki parameter sedikit, sehingga bisa menghemat waktu dan *computational resources*. Arsitektur NN yang diusulkan memiliki 5 *neuron* pada *input layer*, 3 *neuron* pada *hidden layer*, dan 1 *neuron* pada *output layer*. Metode RF akan menerima nilai *output* dari NN untuk kemudian meningkatkan performa hasil prediksi dari NN.

Terakhir Eyster dan Beckage (2024) dalam penelitiannya menyatakan bahwa klasifikasi area tutupan lahan dan area perairan memberikan gambaran dan membantu memahami fenomena perubahan temperature di wilayah perkotaan dan perubahan iklim. Citra beresolusi tinggi dibagi terlebih dahulu menjadi *patch* berukuran 512x512, sehingga didapatkan citra dengan *channel* RGB (*Red Green Blue*) dan *ground truth* dari masing-masing *patch* citra berjumlah 5320 citra. Model DeepLabv3+ dengan *backbone* model *pre-trained* ResNet50 digunakan untuk melakukan klasifikasi area tutupan lahan. Eyster dan Beckage (2024) menentukan nilai *epoch* sejumlah 25 dan *batch size* sebesar 4. Kemudian *optimizer* yang digunakan adalah SGD (*Stochastic Gradient Descent*), dan untuk *loss function* menggunakan *sparse categorical crossentropy*.

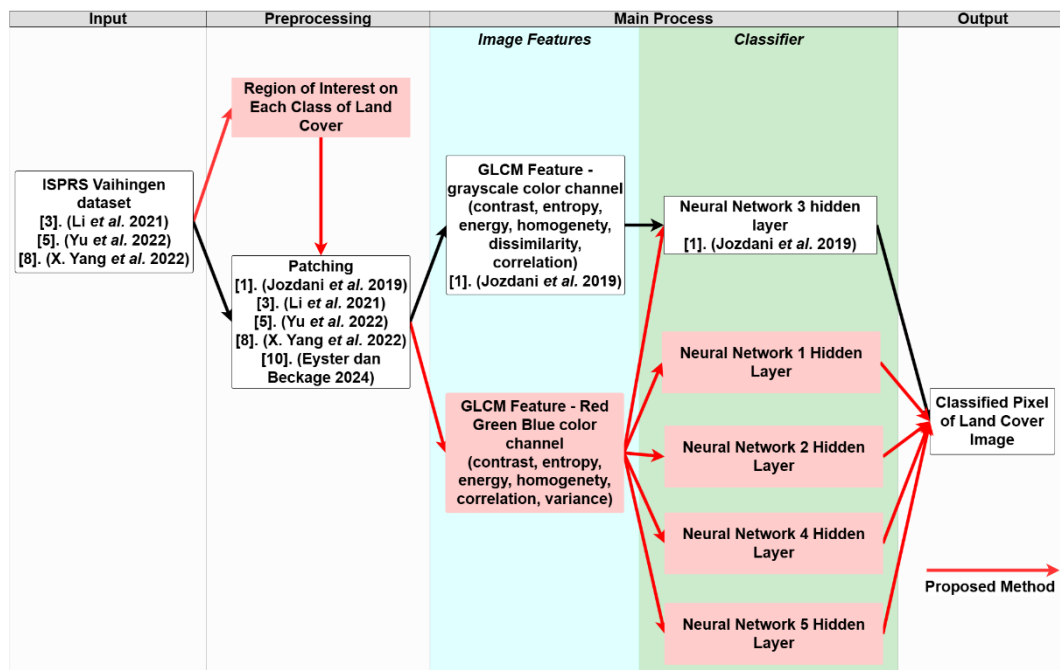
Berdasarkan penjelasan ini, dapat diambil kesimpulan bahwa model NN yang diusung oleh Jozdani *et al.* (2019) memiliki kinerja terbaik untuk melakukan klasifikasi area tutupan lahan dengan nilai *overall accuracy* sebesar 93.64%, *F1-Score* sebesar 93.7%, dan *kappa co-efficient* sebesar 92.3%

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Konseptual

Kerangka konseptual menjelaskan alur kerja dan hubungan antara komponen dan proses yang digunakan dalam penelitian ini. Kerangka ini dapat memberikan gambaran bagaimana data *input*, *preprocessing*, *main process* (ekstraksi fitur dan klasifikasi), hingga *output* saling berkaitan hingga akhir dengan tujuan klasifikasi area tutupan lahan. Kerangka konsep disajikan dalam Gambar 3.1.



Gambar 3.1. Kerangka Konseptual

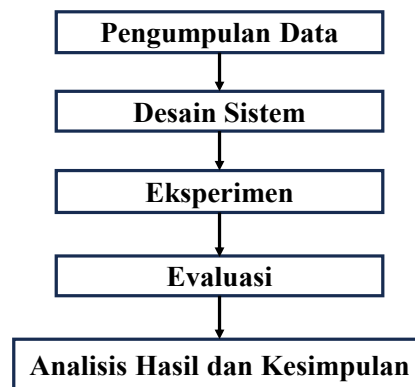
Gambar 3.1 menjelaskan kerangka konsep yang dijalankan pada penelitian ini, dimana citra *input* adalah citra ISPRS Vaihingen. Kemudian Langkah selanjutnya adalah memilih citra *input* ROI (*Region of Interest*) dari setiap kelas area tutupan lahan. Hasil ROI dari setiap kelas area tutupan lahan kemudian

dilakukan *patching* untuk membaginya menjadi bagian-bagian yang lebih kecil, hal ini bertujuan agar ekstraksi fitur yang dilakukan oleh GLCM menjadi lebih detail. Setelah melewati proses *patching*, hasil dari *patching* kemudian diekstrak fiturnya menggunakan GLCM. Ekstraksi fitur dilakukan dengan memanfaatkan masing-masing *channel* yang terdiri dari *infrared*, *red*, dan *green* sebagai *single channel*. Setelah dilakukan ekstraksi fitur menggunakan GLCM, maka masing-masing *channel* warna dari setiap *patch* akan direpresentasikan dengan fitur GLCM. Masing-masing fitur dari setiap *channel* warna *infrared*, *red*, dan *green* tersebut kemudian digabungkan dalam satu *array* sehingga jumlah *data input* untuk NN adalah sesuai dengan jumlah fitur GLCM. Seluruh fitur dari masing-masing *patch* kemudian dipelajari oleh NN. Model NN yang sudah melewati proses *training* dan mendapatkan parameter optimal kemudian digunakan untuk mengklasifikasikan area tutupan lahan.

3.2 Desain Penelitian

Desain penelitian merupakan urutan langkah-langkah yang dilakukan oleh peneliti. Langkah tersebut dimulai dari pengumpulan data yang kemudian dilanjutkan dengan membuat desain sistem yang merupakan rangkaian proses yang terjadi di dalam sistem. Rangkaian proses yang ada di dalam desain sistem kemudian diimplementasikan kedalam sistem, tujuan utamanya adalah agar model mempelajari data yang diberikan. Untuk dapat mengetahui kinerja terbaik model dan faktor apa saja yang mempengaruhi kinerjanya maka diperlukan beberapa skenario dalam melakukan eksperimen. Setiap dari skenario eksperimen akan menghasilkan kinerja dari model, sehingga akan diketahui skenario mana yang

menghasilkan kinerja terbaik dari model. Langkah terakhir dari desain penelitian adalah melakukan evaluasi dan menganalisis hasil yang telah didapatkan dari eksperimen. Desain penelitian ditunjukkan dalam Gambar 3.2.

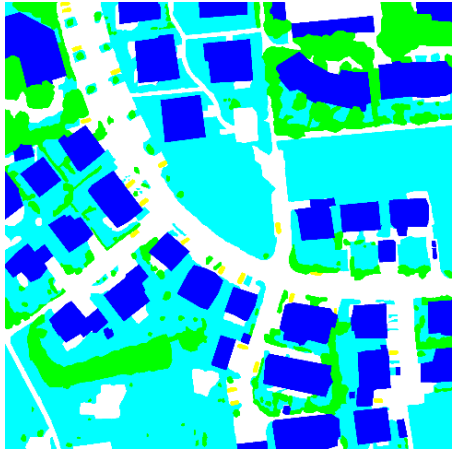
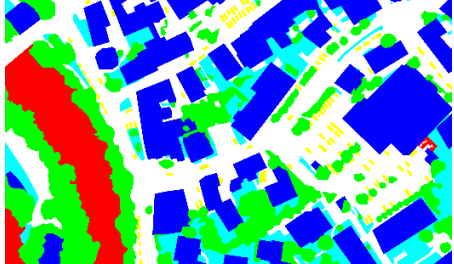
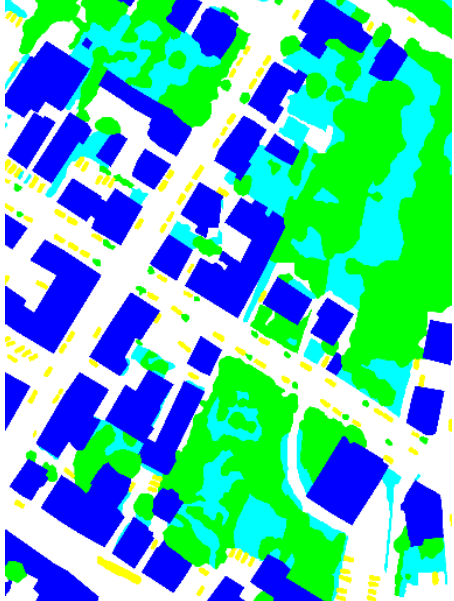


Gambar 3.2. Desain Penelitian

3.2.1 Pengumpulan Data

Pada penelitian ini, untuk melakukan klasifikasi area tutupan lahan vegetasi berdasarkan citra digital dibutuhkan dataset citra beresolusi tinggi dan *ground truth* dari citra itu sendiri. Dataset yang digunakan adalah data primer yang bersumber dari website ISPRS (<https://www.isprs.org>), yang diakses dan diunduh pada 10 November 2023. Data citra beresolusi tinggi yang digunakan dalam penelitian ini diambil menggunakan UAV dengan jumlah data sebanyak 9 citra beresolusi tinggi. Jumlah total kelas dari dataset yang digunakan adalah sebanyak 6 kelas, yakni permukaan kedap air, *background*, pepohonan, bangunan, *low vegetation*, dan mobil. Beberapa Contoh citra beresolusi tinggi dan *ground truth* dari citra disajikan dalam Tabel 3.1 berikut ini.

Tabel 3.1. Contoh Citra Beresolusi Tinggi dan *Ground Truth*

No	Citra Beresolusi Tinggi	<i>Ground truth</i>
1		
2		
3		

Setiap data citra Vaihingen pada Tabel 3.1 memiliki kode area yang berbeda.

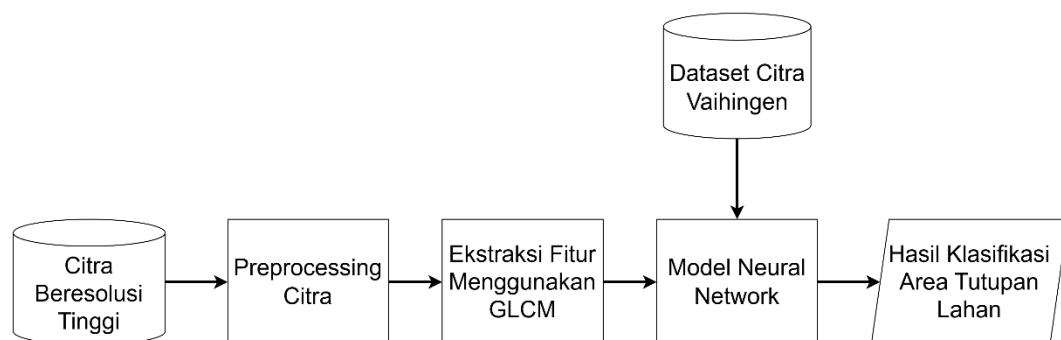
Pada data Citra beresolusi tinggi dan *ground truth* bernomor 1 memiliki kode area

37 dengan ukuran dimensi 1996×1995 piksel. Sedangkan pada data citra beresolusi tinggi dan *ground truth* bernomor 2 memiliki kode area 26 dengan dimensi 2996×1783 piksel. Lalu untuk data citra beresolusi tinggi dan *ground truth* pada nomor 3 memiliki kode area 7 dengan ukuran dimensi 1887×2557 piksel.

Pada bagian *ground truth*, warna yang berbeda bermakna kelas yang berbeda. Warna putih dengan kode warna (255, 255, 255) mewakili kelas permukaan kedap air atau *impervious surfaces*, warna merah dengan kode warna (255, 0, 0) merupakan *background*, warna hijau dengan kode warna (0, 255, 0) adalah pepohonan, warna biru tua dengan kode warna (0, 0, 255) mewakili bangunan, warna biru muda dengan kode warna (0, 255, 255) adalah *low vegetation*, dan warna kuning dengan kode warna (255, 255, 0) merupakan mobil.

3.2.2 Desain Sistem

Desain sistem dibuat untuk menggambarkan secara umum bagaimana sistem melakukan klasifikasi area tutupan lahan vegetasi menggunakan model *Neural Network*. Desain sistem untuk penelitian ini ditunjukkan pada Gambar 3.3.



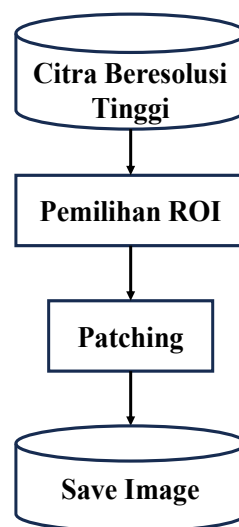
Gambar 3.3. Desain Sistem

3.2.2.1 Input Image

Citra beresolusi tinggi beserta *ground truth* yang sebelumnya telah didapatkan akan digunakan sebagai masukan pada sistem yang akan dibuat. Data citra beserta *ground truth* memiliki format *Tagged Image Format* (*.tif). Jumlah total data input adalah 9 data citra beresolusi tinggi dan 9 *ground truth*.

3.2.2.2 Preprocessing Citra

Preprocessing merupakan tahapan untuk mempersiapkan citra yang akan dimasukkan kedalam model. Ada dua tahap dalam *preprocessing* citra pada penelitian ini yang disajikan pada Gambar 3.4.



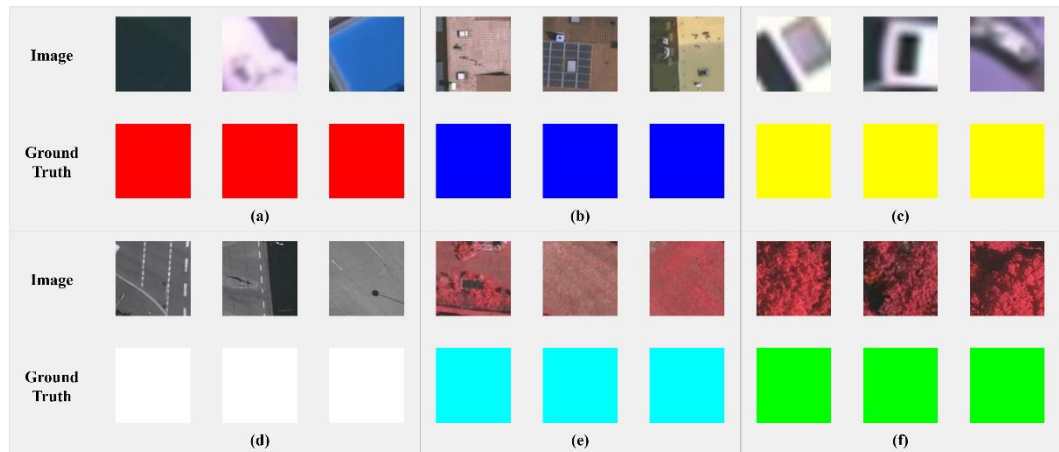
Gambar 3.4. *Preprocessing* Citra

A. Pemilihan ROI

Proses pertama yang terjadi dalam tahap *preprocessing* citra adalah pemilihan ROI. Tujuan dari dilakukannya pemilihan ROI terhadap citra adalah untuk mendetailkan area dari setiap kelas tutupan lahan tanpa terjadi percampuran antara satu kelas dengan kelas lainnya dalam setiap ROI. Pemilihan ROI dilakukan dengan

cara memilih area dari setiap kelas secara manual pada citra beresolusi tinggi.

Contoh hasil pemilihan ROI untuk setiap kelas ditampilkan pada Gambar 3.5.



Gambar 3.5. Contoh Hasil Pemilihan ROI Dari Setiap Kelas

Pada Gambar 3.5, secara berurutan hasil ROI dari setiap kelas ditandai dengan huruf yang berbeda. Huruf (a) merepresentasikan hasil ROI dari kelas *background*, huruf (b) merepresentasikan hasil ROI dari kelas bangunan, huruf (c) merepresentasikan hasil ROI dari kelas mobil, huruf (d) merepresentasikan hasil ROI dari kelas permukaan kedap air, huruf (e) merepresentasikan hasil ROI dari kelas *low vegetation*, huruf (f) merepresentasikan hasil ROI dari kelas pepohonan.

B. *Patching*

Proses berikutnya adalah melakukan pembagian atau *patching* hasil ROI setiap kelas dari citra beresolusi tinggi beserta *ground truth* menjadi bagian-bagian kecil, yaitu dibagi menjadi ukuran *window* 5x5 piksel. Hal ini dilakukan agar model dapat melakukan klasifikasi secara pada setiap area dengan lebih mendetail.

3.2.2.3 Ekstraksi Fitur Menggunakan GLCM

Langkah selanjutnya adalah mengekstrak fitur dari citra ROI yang telah dilakukan *patching* dengan menggunakan GLCM. GLCM atau *Gray Level Co-*

ocurence Matrix merupakan salah satu metode ekstraksi fitur tekstur citra yang bekerja dengan cara menghitung nilai matriks *co-ocurence* dari hubungan antara dua pixel pada jarak dan 4 orientasi arah sudut yang berbeda, yakni sudut 0° , 45° , 90° , 135° . Metode GLCM merupakan metode ekstraksi fitur tekstur yang memiliki efektifitas akurasi dan waktu komputasi lebih efisien dibandingkan metode ekstraksi fitur tekstur lainnya (Praseptiyana *et al.*, 2019). Beberapa peneliti menggunakan fitur GLCM seperti *contrast*, *entropy*, *energy*, *homogeneity*, *correlation*, dan *variance* untuk mengekstrak fitur dari data citra beresolusi tinggi (Liu *et al.*, 2023; Wang & Gu, 2024; R. Zhang *et al.*, 2020). Untuk itu dalam penelitian ini mengadopsi atau menggunakan fitur *contrast*, *entropy*, *energy*, *homogeneity*, *correlation*, dan *variance*.

1. *Contrast* merupakan perhitungan perbedaan intensitas atau perubahan antara suatu piksel dengan piksel-piksel yang berdekatan dalam citra. Semakin tinggi nilainya bermakna citra memiliki tekstur yang kasar, dan semakin rendah nilainya bermakna citra memiliki tekstur yang halus. Rumus perhitungan fitur *contrast* disajikan dalam Persamaan 3.1.

$$contrast = \sum_{i,j} (i - j)^2 p(i, j) \quad (3.1)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLM.
- i = Indeks baris pada matriks GLCM.
- j = Indeks kolom pada matriks GLCM.
- $p(i, j)$ = Nilai probabilitas kemunculan pasangan piksel.

2. *Entropy* menggambarkan seberapa banyak informasi yang dimiliki oleh sebuah objek atau citra dan seberapa kompleks hal tersebut. Semakin tinggi nilainya bermakna tekstur semakin kompleks, dan semakin rendah nilainya bermakna tekstur semakin sederhana. Semakin tinggi nilainya bermakna citra

memiliki tekstur yang kompleks, dan semakin rendah nilainya bermakna citra memiliki tekstur yang seragam atau sederhana. Rumus perhitungan fitur *entropy* disajikan dalam Persamaan 3.2.

$$entropy = -\sum_{i,j} p(i,j) \log(p(i,j)) \quad (3.2)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLM.
- i = Indeks baris pada matriks GLCM.
- j = Indeks kolom pada matriks GLCM.
- $p(i,j)$ = Nilai probabilitas kemunculan pasangan piksel.
- $\log$ = Fungsi logaritma.

3. *Energy* menunjukkan tingkat pengulangan pola atau tingkat keseragaman intensitas piksel pada citra. Semakin tinggi nilainya bermakna citra memiliki pola yang berulang, dan semakin rendah nilainya bermakna citra tidak memiliki pola yang berulang atau acak. Rumus perhitungan fitur *energy* disajikan dalam Persamaan 3.3.

$$energy = \sum_{i,j} p(i,j)^2 \quad (3.3)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLM.
- i = Indeks baris pada matriks GLCM.
- j = Indeks kolom pada matriks GLCM.
- $P(i,j)$ = Nilai probabilitas kemunculan pasangan piksel.

4. *Homogeneity* mengukur kesamaan variasi intensitas citra atau dengan kata lain merepresentasikan pasangan piksel dengan nilai intensitas yang mirip. Semakin tinggi nilainya bermakna intensitas piksel satu dengan lainnya serupa, dan semakin rendah nilainya bermakna intensitas piksel satu dengan lainnya berbeda. Rumus perhitungan *homogeneity* disajikan dalam Persamaan 3.4.

$$homogeneity = \sum_{i,j} \frac{p(i,j)}{1+(i-j)^2} \quad (3.4)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLCM.
 i = Indeks baris pada matriks GLCM.
 j = Indeks kolom pada matriks GLCM.
 $P(i,j)$ = Nilai probabilitas kemunculan pasangan piksel.

5. *Correlation* menggambarkan tinggi atau rendahnya intensitas hubungan atau korelasi piksel satu dengan piksel lainnya. Semakin tinggi nilainya bermakna hubungan antara satu piksel dengan piksel lainnya kuat, dan semakin rendah nilainya bermakna hubungan antara satu piksel dengan lainnya lemah. Rumus perhitungan *correlation* disajikan dalam Persamaan 3.5.

$$correlation = \frac{\sum_{i,j}(ij)p(i,j) - \mu_i\mu_j}{\sigma_i\sigma_j} \quad (3.5)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLCM.
 i = Indeks baris pada matriks GLCM.
 j = Indeks kolom pada matriks GLCM.
 $P(i,j)$ = Nilai probabilitas kemunculan pasangan piksel.
 $\mu_i\mu_j$ = Rata-rata dari intensitas piksel pada baris (i) dan kolom (j).
 $\sigma_i\sigma_j$ = Nilai standar deviasi dari intensitas piksel pada baris (i) dan kolom (j).

6. *Variance* menunjukkan tingkat heterogen atau keberagaman tekstur antar piksel dari citra. Semakin tinggi nilai *variance* bermakna intensitas atau variasi antar piksel dalam citra beragam atau signifikan, dan semakin rendah nilai *variance* bermakna intensitas atau variasi antar piksel dalam citra memiliki perbedaan yang kecil atau tidak signifikan. Rumus perhitungan *variance* disajikan dalam Persamaan 3.6.

$$variance = \sum_{i,j}(i - \mu)^2p(i,j) \quad (3.6)$$

Keterangan:

- $\sum_{i,j}$ = Notasi untuk menjumlahkan semua nilai matriks GLCM.
 i = Indeks baris pada matriks GLCM.
 j = Indeks kolom pada matriks GLCM.
 μ = Rata-rata dari keseluruhan intensitas pada matriks GLCM.
 $P(i,j)$ = Nilai probabilitas kemunculan pasangan piksel.

3.2.2.4 Model *Neural Network*

Pada model *neural network*, hal pertama yang dilakukan adalah menentukan variabel-variabel yang mempengaruhi perhitungan prediksi. Dalam hal ini variabel yang digunakan sebagai *input* pada jaringan *neural network* adalah hasil ekstraksi fitur GLCM. Terdapat lima model *neural network* yang digunakan dalam penelitian ini, yaitu *neural network 1 hidden layer*, *neural network 2 hidden layers*, *neural network 3 hidden layers*, *neural network 4 hidden layers*, dan *neural network 5 hidden layers*. Jumlah *neuron* pada *input layer* didasarkan pada jumlah fitur GLCM yang digunakan sebagai *input*. Dalam hal ini ada 2 variasi fitur GLCM yang digunakan sebagai input, yakni 5 fitur (*contrast*, *entropy*, *energy*, *homogeneity*, *correlation*), dan 6 fitur (*contrast*, *entropy*, *energy*, *homogeneity*, *correlation*, *variance*). Masing-masing fitur GLCM melakukan ekstraksi terhadap setiap *channel* warna dari data gambar, sehingga akan didapatkan jumlah *neuron* pada *input layer* sebanyak 15 *neuron* jika menggunakan GLCM 5 fitur, dan jumlah *neuron input layer* sebanyak 18 *neuron* jika menggunakan GLCM 6 fitur. Kemudian untuk jumlah *neuron* pada *hidden layer* ditentukan dengan cara mengalikan jumlah *neuron* pada *layer* sebelumnya dengan nilai 3 (Alkhasawneh 2021). Selanjutnya Santoso *et al.* (2025) dalam penelitiannya menemukan bahwa dengan menambah *hidden layer* pada arsitektur NN dapat meningkatkan performa model dalam melakukan klasifikasi, sehingga dalam penelitian ini dilakukan uji coba dengan menambahkan *hidden layer* pada arsitektur NN.

Masing-masing model *neural network* yang diusulkan dalam penelitian ini memiliki *hyperparameter learning rate*, *optimizer*, dan fungsi aktivasi yang sama.

Nilai *learning rate* yang digunakan pada penelitian ini adalah 0.001 (Meng *et al.*, 2021). Untuk *optimizer* yang digunakan adalah SGD, kemudian untuk fungsi aktivasi adalah ReLU (*Rectified Linear Unit*) sehingga nilai keluaran dari fungsi ini akan berada di rentang nilai 0 sampai nilai positif tak hingga (Koch *et al.*, 2021). Proses *training* model NN akan terus berjalan hingga mencapai titik *convergence* atau titik dimana nilai *loss* dari model sudah tidak bisa lebih optimal lagi. Untuk mendapatkan titik *convergence* ini diterapkan teknik *early stopping* (Ying, 2019).

Pada penelitian ini memanfaatkan 5 jenis arsitektur NN, pertama adalah arsitektur NN 1 *hidden layer* yang terdiri dari *input layer*, *hidden layer*, dan *output layer*. Terdapat 2 variasi arsitektur NN 1 *hidden layer*, variasi arsitektur pertama menggunakan 5 fitur GLCM sehingga memiliki 15 *neuron* pada *input layer*, 45 *neuron* pada *hidden layer*, dan 6 *neuron* pada *output layer*. Kemudian untuk variasi arsitektur kedua menggunakan 6 fitur GLCM sehingga memiliki 18 *neuron* pada *input layer*, 54 *neuron* pada *hidden layer*, dan 6 *neuron* pada *output layer*.

Selanjutnya adalah arsitektur NN 2 *hidden layer* yang terdiri dari *input layer*, *hidden layer 1*, *hidden layer 2*, dan *output layer*. Terdapat 2 variasi arsitektur NN 2 *hidden layer*, variasi arsitektur pertama menggunakan 5 fitur GLCM sehingga memiliki 15 *neuron* pada *input layer*, 45 *neuron* pada *hidden layer 1*, 135 *neuron* pada *hidden layer 2*, dan 6 *neuron* pada *output layer*. Kemudian untuk variasi arsitektur kedua menggunakan 6 fitur GLCM sehingga memiliki 18 *neuron* pada *input layer*, 54 *neuron* pada *hidden layer 1*, 162 *neuron* pada *hidden layer 2*, dan 6 *neuron* pada *output layer*.

Berikutnya adalah arsitektur NN 3 *hidden layer* yang terdiri dari *input layer*, *hidden layer 1*, *hidden layer 2*, *hidden layer 3*, dan *output layer*. Terdapat 2 variasi arsitektur NN 3 *hidden layer*, variasi arsitektur pertama menggunakan 5 fitur GLCM sehingga memiliki 15 *neuron* pada *input layer*, 45 *neuron* pada *hidden layer 1*, 135 *neuron* pada *hidden layer 2*, 405 *neuron* pada *hidden layer 3*, dan 6 *neuron* pada *output layer*. Kemudian untuk variasi arsitektur kedua menggunakan 6 fitur GLCM sehingga memiliki 18 *neuron* pada *input layer*, 54 *neuron* pada *hidden layer 1*, 162 *neuron* pada *hidden layer 2*, 486 *neuron* pada *hidden layer 3*, dan 6 *neuron* pada *output layer*.

Arsitektur berikutnya adalah arsitektur NN 4 *hidden layer* yang terdiri dari *input layer*, *hidden layer 1*, *hidden layer 2*, *hidden layer 3*, *hidden layer 4*, dan *output layer*. Terdapat 2 variasi arsitektur NN 4 *hidden layer*, arsitektur pertama menggunakan 5 fitur GLCM sehingga memiliki 15 *neuron* pada *input layer*, 45 *neuron* pada *hidden layer 1*, 135 *neuron* pada *hidden layer 2*, 405 *neuron* pada *hidden layer 3*, 1215 *neuron* pada *hidden layer 4*, dan 6 *neuron* pada *output layer*. Kemudian untuk variasi arsitektur kedua menggunakan 6 fitur GLCM sehingga memiliki 18 *neuron* pada *input layer*, 54 *neuron* pada *hidden layer 1*, 162 *neuron* pada *hidden layer 2*, 486 *neuron* pada *hidden layer 3*, 1458 *neuron* pada *hidden layer 4*, dan 6 *neuron* pada *output layer*.

Terakhir adalah arsitektur NN 5 *hidden layer* terdiri dari *input layer*, *hidden layer 1*, *hidden layer 2*, *hidden layer 3*, *hidden layer 4*, *hidden layer 5*, dan *output layer*. Terdapat 2 variasi arsitektur NN 5 *hidden layer*, variasi arsitektur pertama menggunakan 5 fitur GLCM sehingga memiliki 15 *neuron* pada *input layer*, 45

neuron pada *hidden layer* 1, 135 *neuron* pada *hidden layer* 2, 405 *neuron* pada *hidden layer* 3, 1215 *neuron* pada *hidden layer* 4, 3645 *neuron* pada *hidden layer* 5, dan 6 *neuron* pada *output layer*. Kemudian untuk variasi arsitektur kedua menggunakan 6 fitur GLCM sehingga memiliki 18 *neuron* pada *input layer*, 54 *neuron* pada *hidden layer* 1, 162 *neuron* pada *hidden layer* 2, 486 *neuron* pada *hidden layer* 3, 1458 *neuron* pada *hidden layer* 4, 4374 *neuron* pada *hidden layer* 5, dan 6 *neuron* pada *output layer*.

3.2.3 Eksperimen

Tahap eksperimen dilakukan dengan memanfaatkan *Kaggle notebook* untuk mengklasifikasikan area tutupan lahan pada dataset citra Vaihingen menggunakan *neural network*. *Kaggle notebook* yang digunakan untuk eksperimen pada penelitian ini memiliki spesifikasi CPU 4 *cores* dengan RAM 30 GB. Kemudian GPU yang digunakan adalah NVIDIA TESLA P100 dengan RAM GPU sejumlah 16 GB. Kemudian untuk mengetahui model *neural network* yang memiliki kinerja paling optimal dibutuhkan beberapa skenario pengujian.

Terdapat 2 komposisi fitur GLCM yang terdiri dari 5 fitur GLCM, dan 6 fitur GLCM yang akan digunakan sebagai *input* ke *neural network*, sehingga model dapat melakukan klasifikasi area tutupan lahan. Lebih detail, komposisi fitur GLCM disajikan dalam Tabel 3.2.

Tabel 3.2. Komposisi Fitur GLCM

Komposisi	Fitur GLCM
1	<i>contrast, entropy, energy, homogeneity, correlation</i>
2	<i>contrast, entropy, energy, homogeneity, correlation, variance</i>

Setelah membagi komposisi fitur GLCM, dalam penelitian ini juga melakukan perbandingan arsitektur yang memiliki jumlah *neuron* dan *hidden layer* berbeda pada jaringan *neural network*. Lebih detail disajikan dalam Tabel 3.3.

Tabel 3.3. Jumlah *Neuron* Pada Setiap *Layer Neural Network*

Model	Jumlah <i>Hidden Layer</i>	Jumlah <i>Neuron</i> Pada Setiap <i>Layer</i>
A1	1	15 – 45 – 6
A2	1	18 – 54 – 6
B1	2	15 – 45 – 135 – 6
B2	2	18 – 54 – 162 – 6
C1	3	15 – 45 – 135 – 405 – 6
C2	3	18 – 54 – 162 – 486 – 6
D1	4	15 – 45 – 135 – 405 – 1215 – 6
D2	4	18 – 54 – 162 – 486 – 1458 – 6
E1	5	15 – 45 – 135 – 405 – 1215 – 3645 – 6
E2	5	18 – 54 – 162 – 486 – 1458 – 4374 – 6

Pada setiap jumlah *hidden layer*, model *neural network* diujikan dengan komposisi fitur GLCM pertama atau kedua. Sebagai contoh model A1 dengan jumlah *hidden layer* 1 akan memiliki *neuron* pada *input layer* sejumlah 15, karena pada model A1 memanfaatkan fitur GLCM pertama yang berjumlah 5 fitur. Contoh berikutnya adalah pada model A2, meskipun memiliki jumlah *hidden layer* yang sama dengan model A1, jumlah *neuron* pada *input layer* model A2 adalah sejumlah 18, karena pada model A2 memanfaatkan fitur GLCM kedua yang berjumlah 6 fitur. Setiap skenario pengujian memanfaatkan pembagian data *training* dan *testing* dengan rasio 80:20 (Joseph, 2022).

3.2.4 Evaluasi

Pada penelitian ini, hasil kinerja dari model *neural network* yang telah melalui proses *training* dievaluasi dan dianalisis dengan memanfaatkan *confusion matrix*. Analisis ini digunakan untuk menunjukkan performa atau kinerja dari model *neural*

network dalam melakukan klasifikasi area tutupan lahan. Berikut adalah persamaan dari metrik yang digunakan untuk mengevaluasi model:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.7)$$

$$Precision = \frac{TP}{TP+FP} \quad (3.8)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.9)$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.10)$$

$$IoU = \frac{TP}{TP+FP+FN} \quad (3.11)$$

Keterangan:

- TP = *True Positif*. Kelas sebenarnya positif, dan diprediksi positif.
Misal area sebenarnya “bangunan”, dan diprediksi sebagai “bangunan”.
- FP = *False Positif*. Kelas sebenarnya negatif, tetapi diprediksi positif.
Misal area sebenarnya “pepohonan”, tetapi diprediksi sebagai “bangunan”.
- FN = *False Negatif*. Kelas sebenarnya positif, tetapi diprediksi negatif.
Misal area sebenarnya “bangunan”, tetapi diprediksi sebagai “pepohonan”.
- TN = *True Negatif*, kelas sebenarnya negatif dan diprediksi negatif.
Misal area bukan “bangunan”, dan diprediksi sebagai bukan “bangunan”

3.3 Instrumen Penelitian

Instrumen penelitian menggambarkan variabel apa saja yang akan digunakan dalam proses penelitian. Variabel yang digunakan dalam penelitian ini berupa variabel bebas (*independent*), variabel terikat (*dependen*), dan variabel antara (*intervening*). Detail mengenai instrument penelitian disajikan dalam Tabel 3.4.

Tabel 3.4. Instrumen Penelitian

Variabel <i>independent</i>	Main process	Variabel <i>intervening</i>	Variabel <i>dependen</i>
Citra Beresolusi Tinggi (ISPRS Vaihingen)	<i>Neural Network</i>	Fitur GLCM	Kinerja model (<i>Accuracy, Precision, Recall, F1-Score, IoU</i>)

Variabel *independent* atau variabel pengaruh yang digunakan dalam penelitian ini adalah data citra beresolusi tinggi (ISPRS Vaihingen). Dataset citra

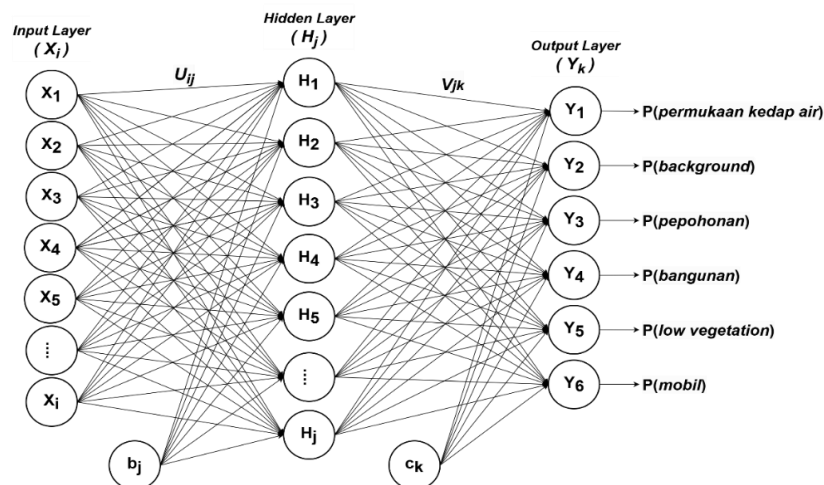
ISPRS Vaihingen ini memiliki informasi area tutupan lahan yang kemudian digunakan fiturnya untuk mengklasifikasikan area tutupan lahan. Tanpa data ini, proses ekstraksi fitur dan klasifikasi tidak dapat dilakukan. Selanjutnya *main process* dalam penelitian ini menggunakan model *neural network* untuk memproses fitur yang telah diekstrak dari citra. Model *neural network* akan mempelajari pola hubungan antara fitur tekstur dari GLCM untuk menghasilkan klasifikasi area tutupan lahan. Kemudian variabel *intervening* atau variabel antara dalam penelitian ini merupakan fitur-fitur tekstur yang dihasilkan dari metode GLCM dari setiap *patch* citra. Terakhir, variabel *dependent* atau variabel yang dipengaruhi oleh variabel *independent* dalam penelitian ini berupa hasil matriks evaluasi dari kinerja model berupa *accuracy*, *precision*, *recall*, *F1-Score*, *IoU*.

BAB IV

NEURAL NETWORK SATU HIDDEN LAYER

4.1 Desain

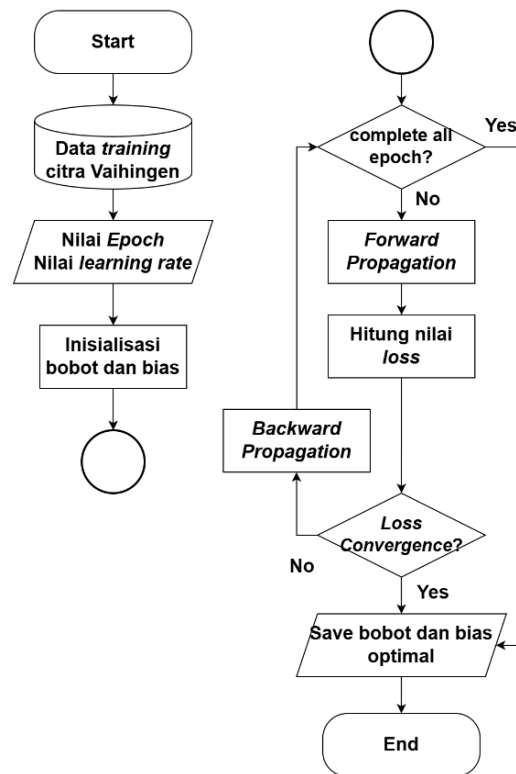
Desain membahas mengenai tahapan proses yang dilakukan oleh peneliti dengan tujuan mendapatkan hasil yang ditujukan untuk menyelesaikan permasalahan dari penelitian yang dilakukan. Dalam penelitian ini model NN digunakan untuk melakukan klasifikasi area tutupan lahan. Arsitektur pada model NN A1 dan model NN A2 menggunakan satu *hidden layer*. Secara lebih detail arsitektur NN 1 *hidden layer* ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Arsitektur *Neural Network 1 Hidden Layer*

Proses *training* dari NN dimulai dengan memanggil data *training* citra Vaihingen yang telah disiapkan. Selanjutnya adalah menentukan nilai *hyperparameter epoch* dan *learning rate* serta menginisialisasi nilai parameter bobot dan bias yang akan terus diperbarui saat proses *training* berjalan.

Proses selanjutnya adalah dengan mengecek apakah nilai *epoch* maksimal sudah tercapai atau belum, jika belum maka proses *forward propagation* dijalankan untuk menghitung nilai prediksi model terhadap data citra Vaihingen untuk mengklasifikasikan kelas area tutupan lahan. Setelah itu dilakukan perhitungan nilai *loss* menggunakan *sparse categorical cross-entropy*. Proses *training* akan terus berjalan hingga mencapai titik *convergence*, jika model belum mencapai titik *convergence* maka selanjutnya dilakukan pembaruan nilai bobot dan bias melalui mekanisme *backward propagation*. Selanjutnya dilakukan pengecekan lagi apakah nilai *epoch* maksimal telah tercapai atau belum, jika belum maka proses *training* akan terus berjalan. Proses *training* akan terhenti jika nilai *epoch* maksimal telah dicapai atau model telah mencapai titik *convergence*. Setelah proses *training* terhenti maka nilai bobot dan bias optimal akan disimpan untuk kemudian digunakan dalam proses *testing*. *Flowchart* proses *training* dijelaskan dalam Gambar 4.2.



Gambar 4.2. Flowchart Neural Network

Langkah-langkah perhitungan *neural network* 1 *hidden layer* sebagai berikut:

1. Inisialisasi data *input*, parameter, dan *hyperparameter* model.
2. Fase *forward propagation*:
 - Mengkalikan setiap data *input* pada *input layer* (X_i) dengan bobot (U_{ij}) yang menyambungkan *input layer* dengan *hidden layer* (H_j). Kemudian jumlahkan hasil perkalian pada masing-masing *neuron* di *hidden layer*. Kemudian dilanjutkan dengan menambahkan nilai *bias* (b_j) pada setiap hasil penjumlahan pada masing-masing *neuron* di *hidden layer*. Dilanjutkan dengan memasukkan nilai masing-masing *neuron* di *hidden layer* hasil penjumlahan *bias* kedalam fungsi aktivasi ReLU.

Rumus propagasi maju di *hidden layer* disajikan dalam Persamaan 4.1 dan 4.2.

$$Z_j = \sum_{i=1}^n (X_i U_{ij}) + b_j \quad (4.1)$$

Keterangan:

Z_j = Hasil perkalian antara *input* dengan bobot yang kemudian ditambah bias.

X_i = Data *input* pada *input layer*

U_{ij} = Bobot yang menyambungkan *input layer* dengan *hidden layer*

b_j = Nilai bias dari *neuron* pada *hidden layer*

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$H_j = \text{ReLU}(Z_j) = \max(0, Z_j) \quad (4.2)$$

Keterangan:

H_j = Nilai pada *hidden layer* setelah diaktivasi menggunakan fungsi ReLU.

- Selanjutnya adalah menghitung nilai prediksi pada *output layer* (Y_k). Cara perhitungannya sama seperti pada perhitungan pada *hidden layer* (H_j), yang membedakan adalah bobot (V_{jk}) yang menyambungkan *hidden layer* ke *output layer*, dan bias (c_k) pada *output layer*. Rumus propagasi maju di *output layer* disajikan dalam Persamaan 4.3.

$$Z_k = \sum_{j=1}^n (H_j V_{jk}) + c_k \quad (4.3)$$

Keterangan:

Z_k = Hasil perkalian antara H_j dengan V_{jk} yang kemudian ditambah c_k

H_j = Nilai hasil perhitungan pada *hidden layer*

V_{jk} = Bobot yang menyambungkan *hidden layer* dengan *output layer*

c_k = Nilai bias dari *neuron* pada *output layer*

Setelah didapatkan nilai Z_k , langkah berikutnya adalah menghitung nilai probabilitas dari setiap kelas menggunakan fungsi *softmax*. Total dari nilai semua probabilitas setiap kelas adalah 1, kelas dengan nilai

probabilitas tertinggi merupakan keluaran hasil prediksi model ($P(Y_k)$).

Rumus perhitungan *softmax* disajikan dalam Persamaan 4.4.

$$P(Y_k) = \frac{\exp(Z_k)}{\sum_{m=1}^6 \exp(Z_m)} \quad (4.4)$$

Keterangan:

$P(Y_k)$ = Nilai probabilitas kelas ke- k yang terprediksi benar.

$\exp(Z_k)$ = Nilai eksponensial dari kelas k di *output layer*.

$\sum_{m=1}^6 \exp(Z_m)$ = Jumlah dari semua nilai eksponensial dari setiap kelas.

- Hitung nilai *loss* untuk mengetahui seberapa jauh jarak prediksi model terhadap kelas asli atau *ground truth*. Nilai *loss* yang dihasilkan dapat memberikan gambaran konvergensi terhadap proses *training* model. Pada penelitian ini *loss function* yang digunakan adalah *sparse categorical cross-entropy*. Rumusnya disajikan dalam Persamaan 4.5.

$$L = -\frac{1}{n} \sum_{i=1}^n \log(P(Y_{i,true})) \quad (4.5)$$

Keterangan:

L = Nilai *loss* hasil perhitungan *loss function*.

$P(Y_{i,true})$ = Nilai probabilitas kelas yang diprediksi oleh model.

n = Jumlah data atau sampel dalam dataset.

3. Fase *backward propagation*:

- Menghitung nilai *error* pada *output layer* dengan turunan dari *loss function* ($\frac{\delta L}{\delta Y_k}$) dengan cara menghitung jarak nilai atau perbedaan antara nilai probabilitas kelas terprediksi dengan kelas *ground truth*. Rumus perhitungan nilai *error* pada *output layer* terdapat pada Persamaan 4.6.

$$\frac{\delta L}{\delta Y_k} = P(Y_k) - y_k \quad (4.6)$$

Keterangan:

$\frac{\delta L}{\delta Y_k}$ = Nilai *error* pada *output layer*.

$P(Y_k)$ = Nilai probabilitas kelas k yang terprediksi benar.

y_k = Kelas asli atau *ground truth*.

- Setelah mendapatkan nilai *error* pada *output layer* sebelum dilakukan aktivasi ($\frac{\delta L}{\delta Z_k}$), selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer* dan *output layer* ($\frac{\delta L}{\delta V_{jk}}$). Rumus perhitungannya ditampilkan pada Persamaan 4.7.

$$\frac{\delta L}{\delta V_{jk}} = \frac{\delta L}{\delta Z_k} \cdot H_j \quad (4.7)$$

Keterangan:

H_j = *Output* atau hasil fungsi aktivasi dari *neuron* ke- j pada *hidden layer*.

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *output layer* sebelum diaktivasi.

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

- Kemudian untuk nilai *error* dari bias pada *output layer* ($\frac{\delta L}{\delta c_k}$) memiliki nilai yang sama dengan nilai *error* pada *output layer* sebelum dilakukan aktivasi ($\frac{\delta L}{\delta Z_k}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* dengan menghitung turunan dari hasil perhitungan di *hidden layer* sebelum dilakukan aktivasi. Rumus perhitungan dari proses ini disajikan dalam Persamaan 4.8.

$$\frac{\delta L}{\delta Z_j} = \sum_{k=1}^n \frac{\delta L}{\delta Z_k} \cdot V_{jk} \cdot ReLU'(Z_j) \quad (4.8)$$

Keterangan:

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* sebelum diaktivasi.

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *output layer* sebelum diaktivasi.

V_{jk} = Bobot yang menyambungkan *hidden layer* dengan *output layer*.

$ReLU'(Z_j)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *input layer* dengan *hidden layer* ($\frac{\delta L}{\delta U_{ij}}$). Rumus dari proses ini disajikan dalam Persamaan 4.9.

$$\frac{\delta L}{\delta U_{ij}} = \frac{\delta L}{\delta Z_j} \cdot X_i \quad (4.9)$$

Keterangan:

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari hasil penambahan bias pada perkalian *input* dan bobot.

X_i = Data *input*.

- Kemudian untuk nilai *error* dari bias pada *hidden layer* ($\frac{\delta L}{\delta b_j}$) memiliki nilai yang sama dengan nilai *error* pada *hidden layer* ($\frac{\delta L}{\delta Z_j}$).
4. Melakukan *update* terhadap nilai bobot yang menyambungkan *input layer* dengan *hidden layer*, dan juga *update* nilai bias pada *hidden layer*. Secara berurutan rumus untuk proses ini ditampilkan dalam Persamaan 4.10 hingga 4.13.

- *Update* nilai bobot pada *hidden layer* (U_{ij})

$$U_{ij} = U_{ij} - \eta \cdot \frac{\delta L}{\delta U_{ij}} \quad (4.10)$$

- *Update* nilai bias pada *hidden layer* (b_j)

$$b_j = b_j - \eta \cdot \frac{\delta L}{\delta b_j} \quad (4.11)$$

- *Update* nilai bobot pada *output layer* (V_{jk})

$$V_{jk} = V_{jk} - \eta \cdot \frac{\delta L}{\delta V_{jk}} \quad (4.12)$$

- *Update* nilai bias pada *output layer* (c_k)

$$c_k = c_k - \eta \cdot \frac{\delta L}{\delta c_k} \quad (4.13)$$

Keterangan:

η = Nilai parameter *learning rate*.

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta b_j}$ = Nilai *error* bias pada *hidden layer*

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta c_k}$ = Nilai *error* bias pada *output layer*

4.2 Implementasi

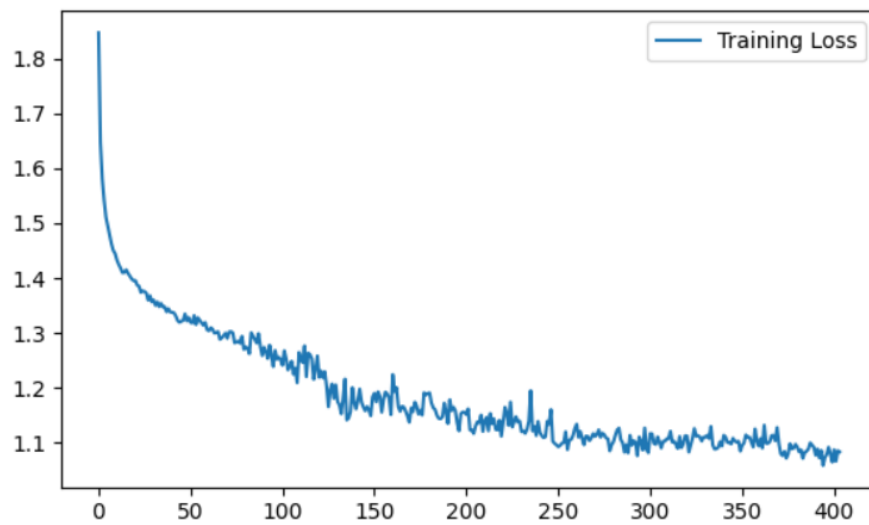
Proses *training* model NN dijalankan menggunakan *Kaggle Notebook*. Pada proses *training*, model NN A1 dan NN A2 memiliki konfigurasi *hyperparameter* yang berbeda. Lebih detail dapat dilihat dalam Tabel 4.1.

Tabel 4.1. Detail *Hyperparameter* Model NN A1 dan NN A2

<i>Hyperparameter</i>	Nilai	
Tipe jaringan	A1	A2
Fungsi aktivasi	ReLU	ReLU
<i>Optimizer</i>	SGD	SGD
<i>Input Layer</i>	15	18
<i>Hidden layer</i>	45	54
<i>Output layer</i>	6	6
<i>Learning rate</i>	0.001	0.001

4.3 Uji Coba NN Model A1

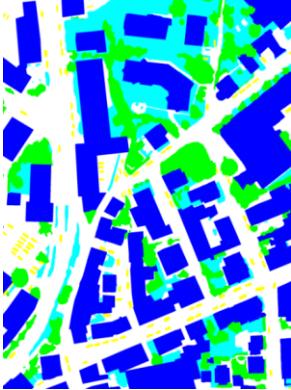
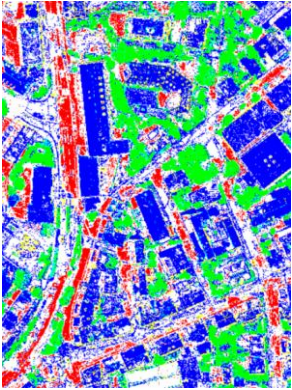
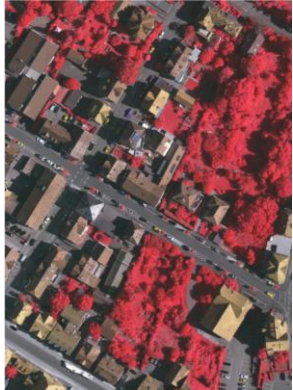
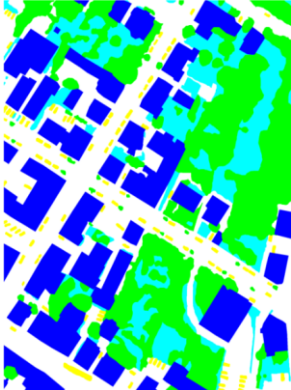
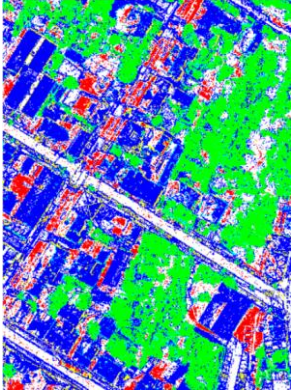
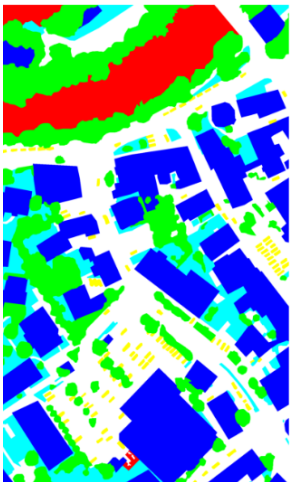
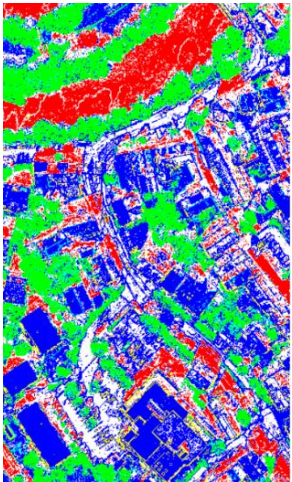
Pada tahap uji coba, *training* model NN A1 menggunakan 5 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-404 dengan nilai *loss* sebesar 1.0830. Gambar 4.3 menunjukkan grafik nilai *loss* dari proses *training* model NN A1.



Gambar 4.3. Grafik *Loss Training* NN Model A1

Pada Gambar 4.3 menunjukkan nilai *loss* pada proses *training* model A1 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Tahap selanjutnya dilakukan pengujian performa model terhadap data *testing*. Didapatkan nilai *accuracy* sebesar 61.40%, *precision* sebesar 63.43%, *recall* sebesar 61.40%, dan *F1-Score* sebesar 59.76%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Beberapa hasil pengujian model A1 dalam melakukan klasifikasi area tutupan lahan ditampilkan dalam Tabel 4.2.

Tabel 4.2. Beberapa Hasil Klasifikasi Model NN A1 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN A1 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Tabel 4.2, model NN A1 dalam melakukan klasifikasi area tutupan lahan masih banyak menghasilkan kesalahan klasifikasi. Untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*,

dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 4.3.

Tabel 4.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN A1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1131049	2572090	505056	721716
	Permukaan kedap air	958228	2338920	837828	794935
	<i>Low vegetation</i>	66858	4152834	155542	554677
	Pepohonan	454912	4058815	244973	171211
	<i>Background</i>	0	4505344	424565	2
	Mobil	13375	4716063	137525	62948
Area 3	Bangunan	1262397	3070202	1329580	369863
	Permukaan kedap air	783294	3409233	563441	1276074
	<i>Low vegetation</i>	153452	4603282	271603	1003705
	Pepohonan	765622	4501005	403643	361772
	<i>Background</i>	0	5700157	331885	0
	Mobil	15521	5824575	151604	40342
Area 7	Bangunan	886260	2663782	856185	418832
	Permukaan kedap air	589537	2781317	551348	902857
	<i>Low vegetation</i>	74256	3978330	209499	562974
	Pepohonan	903240	3261171	281185	379463
	<i>Background</i>	0	4509945	315114	0
	Mobil	22973	4581957	135462	84667
Area 11	Bangunan	675022	2788660	1164381	229375
	Permukaan kedap air	660628	2872219	376487	948104
	<i>Low vegetation</i>	120372	3778288	270998	687780
	Pepohonan	1071301	3117635	270774	397728
	<i>Background</i>	0	4729843	127595	0
	Mobil	16342	4686772	103538	50786
Area 15	Bangunan	883819	2881944	965121	191351
	Permukaan kedap air	295626	3607379	459744	559486
	<i>Low vegetation</i>	212458	3235221	377117	1097439
	Pepohonan	1052273	2827189	451857	590916
	<i>Background</i>	175	4801852	113200	7008
	Mobil	10053	4789759	100792	21631
Area 23	Bangunan	711590	3103674	856080	173694
	Permukaan kedap air	272845	3424868	618518	528807
	<i>Low vegetation</i>	182355	3258345	240965	1163373
	Pepohonan	1207594	2619731	436136	581577
	<i>Background</i>	1404	4612566	224716	6352
	Mobil	5332	4742088	87503	10115
Area 26	Bangunan	954276	2953012	828609	604188
	Permukaan kedap air	647621	3004273	685149	1003042
	<i>Low vegetation</i>	58637	4622856	215273	443319
	Pepohonan	780154	4034384	220076	305471
	<i>Background</i>	338977	4458838	434113	108157
	Mobil	16668	5083310	160532	79575
Area 32	Bangunan	833280	2416941	1371045	437634
	Permukaan kedap air	1284405	2085042	357170	1332283
	<i>Low vegetation</i>	66204	4570333	125971	296392
	Pepohonan	366086	4365837	96789	230188
	<i>Background</i>	495	4755618	247680	55107

Data	Kelas	TP	TN	FP	FN
Area 32	Mobil	40041	4632340	269734	116785
Area 37	Bangunan	644067	1800193	1259248	278512
	Permukaan kedap air	359810	2604999	455905	561306
	<i>Low vegetation</i>	257070	2277198	150275	1297477
	Pepohonan	307719	3111125	305761	257415
	<i>Background</i>	0	3898420	83600	0
	Mobil	2561	3807372	156004	16083

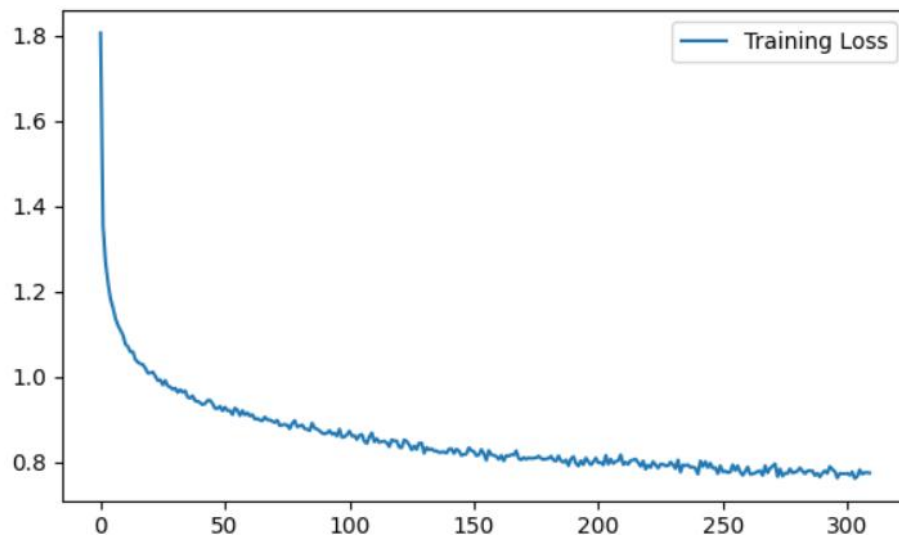
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN A1 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN A1 disajikan dalam Tabel 4.4.

Tabel 4.4. Performa Hasil Pengujian Model NN A1

Kode Area Citra Beresolusi Tinggi	<i>Accuracy</i> (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F1-Score</i> (%)	IoU (%)
Area 1	78.36	53.15	53.18	53.17	36.19
Area 3	82.96	51.02	51.09	51.05	34.24
Area 7	77.39	51.39	51.39	51.39	34.57
Area 11	77.50	52.49	52.39	52.44	35.49
Area 15	83.27	49.84	49.74	49.79	33.18
Area 23	83.00	49.17	49.11	49.14	32.58
Area 26	83.97	52.42	52.00	52.21	35.34
Area 32	83.73	51.28	51.10	51.19	34.39
Area 37	79.49	39.61	39.64	39.62	24.71

4.4 Uji Coba NN Model A2

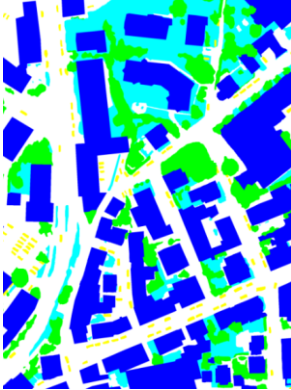
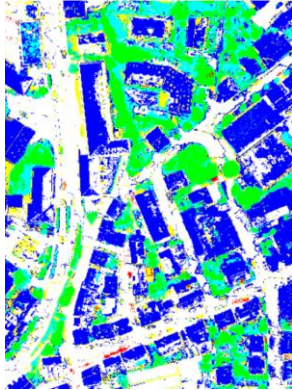
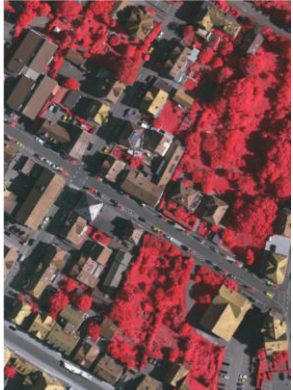
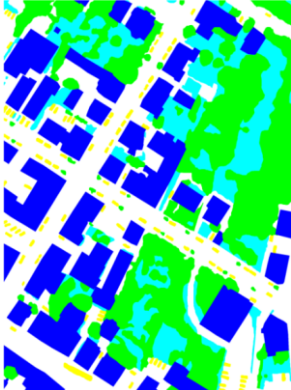
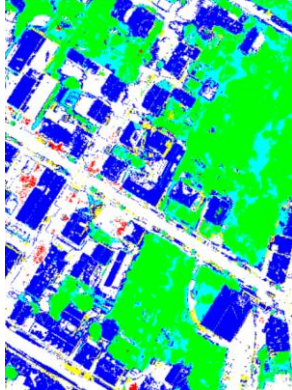
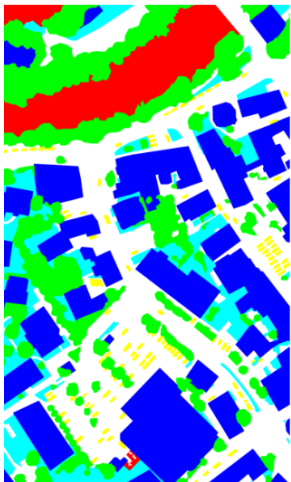
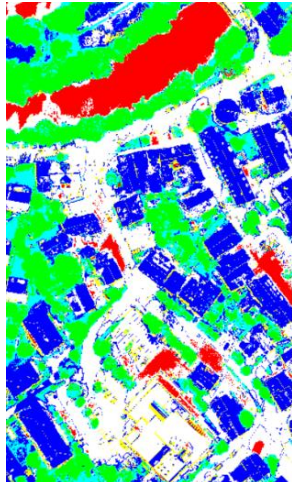
Pada tahap uji coba berikutnya, *training* model NN A2 menggunakan 6 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-310 dengan nilai *loss* sebesar 0.7003. Gambar 4.4 menunjukkan grafik nilai *loss* dari proses *training* model NN A2.



Gambar 4.4. Grafik *Loss Training* NN Model A2

Pada Gambar 4.4 menunjukkan nilai *loss* pada proses *training* model A2 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Tahap selanjutnya dilakukan pengujian performa model terhadap data *testing*. Didapatkan nilai *accuracy* sebesar 74.38%, *precision* sebesar 74.91%, *recall* sebesar 74.38%, dan *F1-Score* sebesar 74.04%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model A2 dalam melakukan klasifikasi area tutupan lahan disajikan dalam Tabel 4.5.

Tabel 4.5. Beberapa Hasil Klasifikasi Model NN A2 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN A2 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Tabel 4.5, model NN A2 memiliki kesalahan klasifikasi yang lebih sedikit dibandingkan dengan model NN A1. Untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan

nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 4.6.

Tabel 4.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN A2

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1212833	2556163	520983	639932
	Permukaan kedap air	1203782	2610720	566028	549381
	<i>Low vegetation</i>	316122	3980348	328028	305413
	Pepohonan	428792	4155925	147863	197331
	<i>Background</i>	0	4916514	13395	2
	Mobil	19182	4680685	172903	57141
Area 3	Bangunan	1189607	3338534	1061248	442653
	Permukaan kedap air	1151457	3628324	344350	907911
	<i>Low vegetation</i>	525482	4294527	580358	631675
	Pepohonan	729475	4630313	274335	397919
	<i>Background</i>	0	5993067	38975	0
	Mobil	14710	5854134	122045	41153
Area 7	Bangunan	897196	3178698	341269	407896
	Permukaan kedap air	1053566	2994037	338628	438828
	<i>Low vegetation</i>	237748	3853887	333942	399482
	Pepohonan	1067863	3197069	345287	214840
	<i>Background</i>	0	4794174	30885	0
	Mobil	30771	4569515	147904	76869
Area 11	Bangunan	570474	3406190	546851	333923
	Permukaan kedap air	1001895	2996528	252178	606837
	<i>Low vegetation</i>	442998	3372689	676597	365154
	Pepohonan	1045907	3214551	173858	423122
	<i>Background</i>	0	4836318	21120	0
	Mobil	21093	4685843	104467	46035
Area 15	Bangunan	707187	3492867	354198	367983
	Permukaan kedap air	440772	3765945	301178	414340
	<i>Low vegetation</i>	677608	2962556	649782	632289
	Pepohonan	1145323	2878989	400057	497866
	<i>Background</i>	2575	4891907	23145	4608
	Mobil	14663	4684804	205747	17021
Area 23	Bangunan	538612	3702281	257473	346672
	Permukaan kedap air	319625	3834803	208583	482027
	<i>Low vegetation</i>	912214	2637484	861826	433514
	Pepohonan	1239188	2727880	327987	549983
	<i>Background</i>	200	4817572	19710	7556
	Mobil	6454	4676425	153166	8993
Area 26	Bangunan	958895	3441396	340225	599569
Area 26	Permukaan kedap air	1130724	3128556	560866	519939
	<i>Low vegetation</i>	254616	4493905	344224	247340
	Pepohonan	901571	4086646	167814	184054
	<i>Background</i>	339902	4770088	122863	107232
	Mobil	22955	5048412	195430	73288
Area 32	Bangunan	544276	2882887	905099	726638
	Permukaan kedap air	1746112	2055399	386813	870576
	<i>Low vegetation</i>	199698	4384252	312052	162898
	Pepohonan	271408	4422959	39667	324866
	<i>Background</i>	837	4831435	171863	53765

Data	Kelas	TP	TN	FP	FN
Area 32	Mobil	35892	4456891	445183	120934
Area 37	Bangunan	711818	2207654	851787	210761
	Permukaan kedap air	616808	2760382	300522	304308
	<i>Low vegetation</i>	669075	2156343	271130	885472
	Pepohonan	287930	3277566	139320	277204
	<i>Background</i>	0	3968870	13150	0
	Mobil	2229	3845125	118251	16415

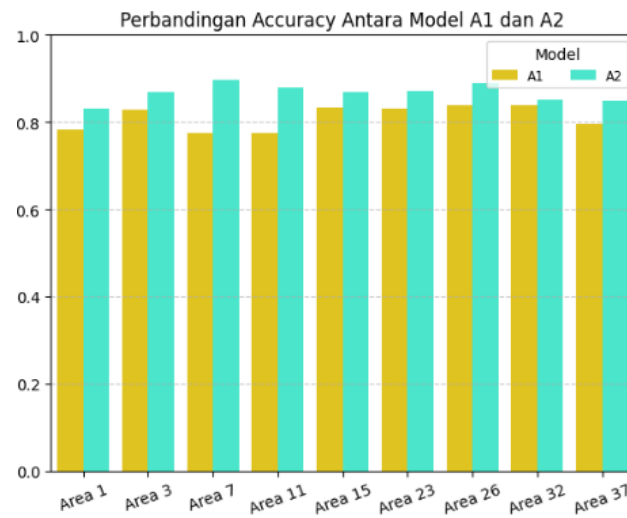
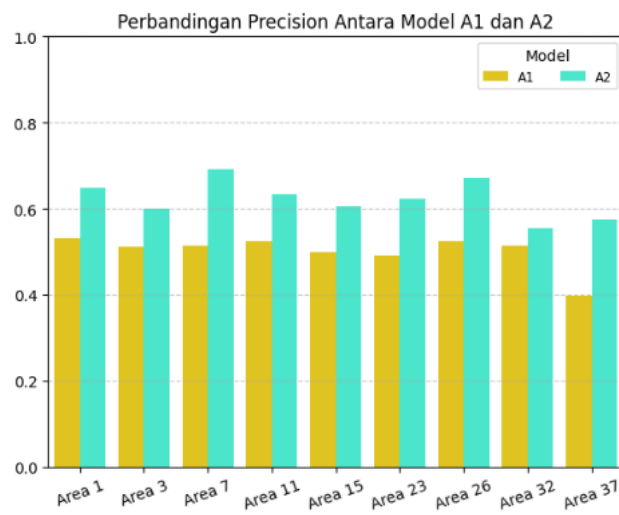
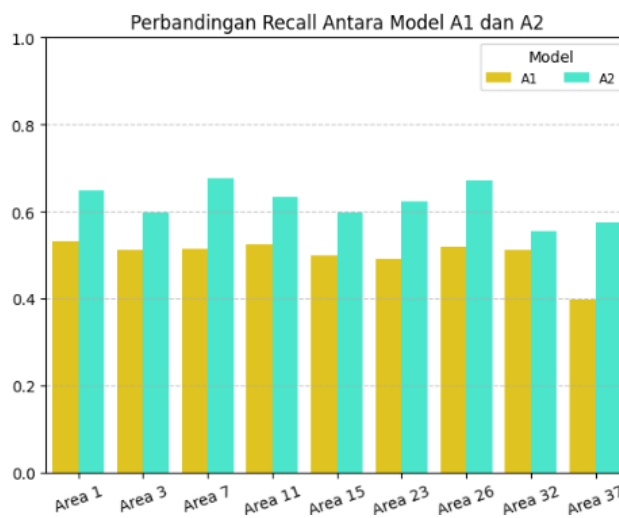
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN A2 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN A2 disajikan dalam Tabel 4.7.

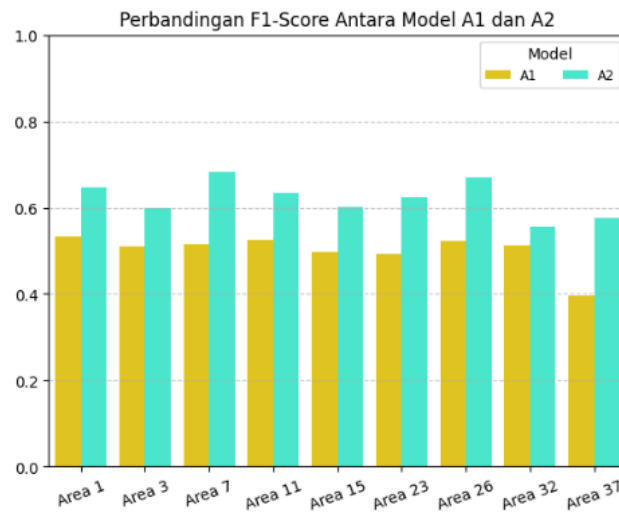
Tabel 4.7. Performa Hasil Pengujian Model NN A2

Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	83.12	64.76	64.73	64.74	47.89
Area 3	86.79	60.12	59.72	59.92	42.77
Area 7	89.80	69.09	67.63	68.35	51.89
Area 11	87.89	63.37	63.37	63.37	46.39
Area 15	86.91	60.60	59.63	60.12	43.06
Area 23	87.20	62.21	62.34	62.30	45.23
Area 26	88.90	67.02	67.02	67.02	50.39
Area 32	85.11	55.50	55.50	55.50	38.41
Area 37	84.76	57.46	57.48	57.47	40.31

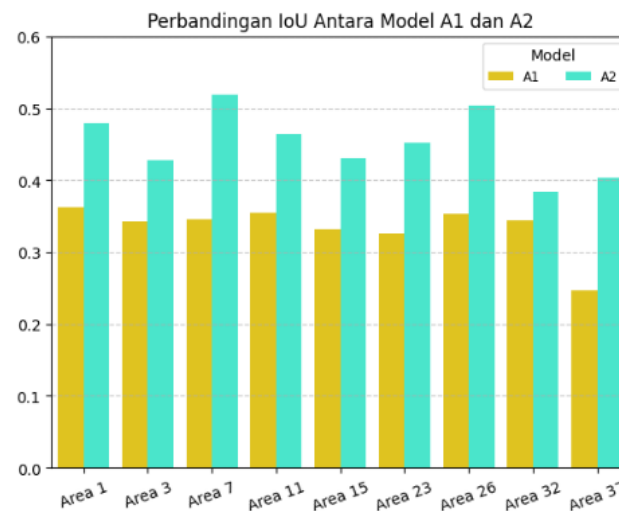
4.5 Komparasi Performa NN Model A1 dan A2

Selanjutnya dilakukan komparasi antara NN model A1 dan A2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 4.5 hingga Gambar 4.9.

Gambar 4.5. Perbandingan *Accuracy* Antara Model NN A1 dan A2Gambar 4.6. Perbandingan *Precision* Antara Model NN A1 dan A2Gambar 4.7. Perbandingan *Recall* Antara Model NN A1 dan A2



Gambar 4.8. Perbandingan F1-Score Antara Model NN A1 dan A2



Gambar 4.9. Perbandingan IoU Antara Model NN A1 dan A2

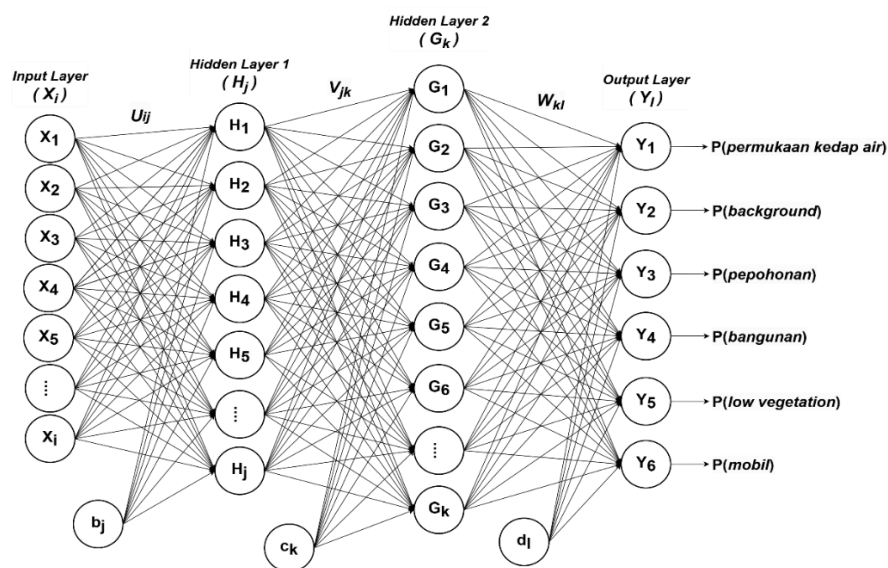
Dari perbandingan yang dilakukan pada Gambar 4.5 hingga Gambar 4.9, ditemukan bahwa model performa model NN A2 yang memanfaatkan 6 fitur GLCM memiliki performa yang lebih tinggi dibandingkan dengan model NN A1 yang memanfaatkan 5 fitur GLCM dari semua metrik, baik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU pada semua area citra beresolusi tinggi.

BAB V

NEURAL NETWORK DUA HIDDEN LAYER

5.1 Desain

Arsitektur pada model NN B1 dan B2 menggunakan dua *hidden layer* untuk melakukan klasifikasi pada area tutupan lahan citra beresolusi tinggi. Secara keseluruhan arsitektur jaringan NN 2 *hidden layer* ditunjukkan dalam Gambar 5.1.



Gambar 5.1. Arsitektur Neural Network 2 Hidden layer

Terjadi proses perhitungan nilai prediksi kelas pada *forward propagation* dan *update* nilai bobot pada *backpropagation*. Lebih detail, berikut adalah proses yang terjadi di dalam *neural network 2 hidden layers*:

1. Inisialisasi data *input*, parameter, dan *hyperparameter* model.
2. Fase *forward propagation*:
 - Mengkalikan setiap data *input* pada *input layer* (X_i) dengan bobot (U_{ij}) yang menyambungkan *input layer* dengan *hidden layer 1* (H_j). Kemudian jumlahkan hasil perkalian pada masing-masing *neuron* di *hidden layer 1*. Kemudian dilanjutkan dengan menambahkan nilai *bias* (b_j) pada setiap

hasil penjumlahan pada masing-masing *neuron* di *hidden layer* 1. Dilanjutnya dengan memasukkan nilai masing-masing *neuron* di *hidden layer* 1 hasil penjumlahan *bias* kedalam fungsi aktivasi ReLU. Rumus propagasi maju di *hidden layer* 1 disajikan dalam Persamaan 5.1 dan 5.2.

$$Z_j = \sum_{i=1}^n (X_i U_{ij}) + b_j \quad (5.1)$$

Keterangan:

Z_j = Hasil perkalian antara *input* dengan bobot yang kemudian ditambah bias.

X_i = Data *input* pada *input layer*.

U_{ij} = Bobot yang menyambungkan *input layer* dengan *hidden layer*.

b_j = Nilai bias dari *neuron* pada *hidden layer* 1.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$H_j = \text{ReLU}(Z_j) = \max(0, Z_j) \quad (5.2)$$

Keterangan:

H_j = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 2 (G_k) dengan cara menambahkan bias *hidden layer* 2 (c_k) pada hasil perkalian antara nilai prediksi *hidden layer* 1 dengan nilai bobot (V_{jk}) yang menyambungkan *hidden layer* 1 ke *hidden layer* 2 (Z_k). Rumus propagasi maju di *hidden layer* 2 ditampilkan dalam Persamaan 5.3 dan 5.4.

$$Z_k = \sum_{j=1}^n (H_j V_{jk}) + c_k \quad (5.3)$$

Keterangan:

Z_k = Hasil perkalian antara *hidden layer* 1 dengan bobot, lalu ditambah bias.

H_j = Nilai prediksi pada *hidden layer* 1.

V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

c_k = Nilai bias dari *neuron* pada *hidden layer* 2.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$G_k = \text{ReLU}(Z_k) = \max(0, Z_k) \quad (5.4)$$

Keterangan:

G_k = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Selanjutnya adalah menghitung nilai prediksi pada *output layer* (Y_l). Cara perhitungannya sama seperti pada perhitungan pada *hidden layer* 2 (G_k), yang membedakan adalah bobot (W_{kl}) yang menyambungkan *hidden layer* 2 ke *output layer*, dan bias (d_l) pada *output layer*. Rumus propagasi maju di *output layer* disajikan dalam Persamaan 5.5.

$$Z_l = \sum_{k=1}^n (G_k W_{kl}) + d_l \quad (5.5)$$

Keterangan:

Z_l = Hasil perkalian antara G_k dengan W_{kl} lalu ditambah dengan bias d_l .

G_k = Nilai hasil perhitungan pada *hidden layer* 2.

W_{kl} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

d_l = Nilai bias dari *neuron* pada *output layer*.

- Langkah berikutnya adalah menghitung nilai probabilitas dari setiap kelas menggunakan fungsi *softmax*. Total dari nilai semua probabilitas setiap kelas adalah 1, kelas dengan nilai probabilitas tertinggi merupakan keluaran hasil prediksi model ($P(Y_l)$). Rumus perhitungan *softmax* disajikan dalam Persamaan 5.6.

$$P(Y_l) = \frac{\exp(Z_l)}{\sum_{m=1}^6 \exp(Z_m)} \quad (5.6)$$

Keterangan:

$P(Y_l)$ = Nilai probabilitas kelas ke- l yang terprediksi benar.

$\exp(Z_l)$ = Nilai eksponensial dari kelas l di *output layer*.

$\sum_{m=1}^6 \exp(Z_m)$ = Jumlah dari semua nilai eksponensial dari setiap kelas.

- Hitung nilai *loss* untuk mengetahui seberapa jauh jarak prediksi model terhadap kelas asli atau *ground truth*. Nilai *loss* yang dihasilkan dapat memberikan gambaran konvergensi terhadap proses *training* model. Pada penelitian ini *loss function* yang digunakan adalah *sparse categorical cross-entropy*. Rumusnya disajikan dalam Persamaan 5.7.

$$L = -\frac{1}{n} \sum_{i=1}^n \log(P(Y_{i,true})) \quad (5.7)$$

Keterangan:

- L = Nilai *loss* hasil perhitungan *loss function*.
 $P(Y_{i,true})$ = Nilai probabilitas kelas yang diprediksi oleh model.
 n = Jumlah data atau sampel dalam dataset.

3. Fase *backward propagation*:

- Menghitung nilai *error* pada *layer* dengan turunan dari *loss function* ($\frac{\delta L}{\delta Z_l}$) dengan cara menghitung jarak nilai atau perbedaan antara nilai probabilitas kelas terprediksi dengan kelas *ground truth*. Rumus perhitungan nilai *error* pada *output layer* terdapat pada Persamaan 5.8.

$$\frac{\delta L}{\delta Z_l} = P(Y_l) - y_l \quad (5.8)$$

Keterangan:

- $\frac{\delta L}{\delta Z_l}$ = Nilai *error* pada *output layer*.
 $P(Y_l)$ = Nilai probabilitas kelas k yang terprediksi benar.
 y_l = Kelas asli atau *ground truth*.

- Setelah mendapatkan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$), selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer* dan *output layer* ($\frac{\delta L}{\delta W_{kl}}$). Rumus perhitungannya ditampilkan pada Persamaan 5.9.

$$\frac{\delta L}{\delta W_{kl}} = \frac{\delta L}{\delta Z_l} \cdot G_k \quad (5.9)$$

Keterangan:

- G_k = *Output* atau hasil fungsi aktivasi dari *neuron* ke- k pada *hidden layer* 2.
 $\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer*.
 $\frac{\delta L}{\delta W_{kl}}$ = Nilai *error* pada bobot W_{kl} .

- Kemudian untuk nilai *error* dari bias pada *output layer* ($\frac{\delta L}{\delta d_l}$) memiliki nilai yang sama dengan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 2. Rumus perhitungan dari proses ini disajikan dalam Persamaan 5.10.

$$\frac{\delta L}{\delta Z_k} = \sum_{l=1}^n \frac{\delta L}{\delta Z_l} \cdot W_{kl} \cdot ReLU'(Z_k) \quad (5.10)$$

Keterangan:

- $\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.
- $\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer* sebelum diaktivasi.
- W_{kl} = Bobot yang menyambungkan *hidden layer* 2 dengan *output layer*.
- $ReLU'(Z_k)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2 ($\frac{\delta L}{\delta V_{jk}}$). Rumus dari proses ini disajikan dalam Persamaan 5.11.

$$\frac{\delta L}{\delta V_{jk}} = \frac{\delta L}{\delta Z_k} \cdot H_j \quad (5.11)$$

Keterangan:

- $\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .
 - $\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.
 - H_j = *Output* atau hasil fungsi aktivasi dari *neuron* ke- j pada *hidden layer* 2.
- Kemudian untuk nilai *error* dari bias *hidden layer* 2 ($\frac{\delta L}{\delta C_k}$) memiliki nilai yang sama dengan nilai *error hidden layer* 2 sebelum diaktivasi ($\frac{\delta L}{\delta Z_k}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 1. Rumus perhitungan dari proses ini disajikan dalam Persamaan 5.12.

$$\frac{\delta L}{\delta Z_j} = \sum_{k=1}^n \frac{\delta L}{\delta Z_k} \cdot V_{jk} \cdot ReLU'(Z_j) \quad (5.12)$$

Keterangan:

- $\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* 1 sebelum diaktivasi.
- $\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.
- V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.
- $ReLU'(Z_j)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2 ($\frac{\delta L}{\delta U_{ij}}$). Rumus dari proses ini disajikan dalam Persamaan 5.13.

$$\frac{\delta L}{\delta U_{ij}} = \frac{\delta L}{\delta Z_j} \cdot X_i \quad (5.13)$$

Keterangan:

$$\begin{aligned}\frac{\delta L}{\delta U_{ij}} &= \text{Nilai error pada bobot } U_{ij}. \\ \frac{\delta L}{\delta Z_j} &= \text{Nilai error dari hidden layer 1 sebelum diaktivasi.} \\ X_i &= \text{Data input.}\end{aligned}$$

- Kemudian untuk nilai error dari bias hidden layer 1 ($\frac{\delta L}{\delta b_j}$) memiliki nilai yang sama dengan nilai error hidden layer 1 sebelum diaktivasi ($\frac{\delta L}{\delta Z_j}$).

4. Melakukan *update* terhadap nilai bobot yang menyambungkan *input layer* dengan *hidden layer 1*, *hidden layer 1* dengan *hidden layer 2*, dan *hidden layer 2* ke *output layer*. Secara berurutan rumus untuk proses ini ditampilkan dalam Persamaan 5.14 hingga 5.19.

- *Update* nilai bobot pada hidden layer 1 (U_{ij})

$$U_{ij} = U_{ij} - \eta \cdot \frac{\delta L}{\delta U_{ij}} \quad (5.14)$$

- *Update* nilai bias pada hidden layer 1 (b_j)

$$b_j = b_j - \eta \cdot \frac{\delta L}{\delta b_j} \quad (5.15)$$

- *Update* nilai bobot pada hidden layer 2 (V_{jk})

$$V_{jk} = V_{jk} - \eta \cdot \frac{\delta L}{\delta V_{jk}} \quad (5.16)$$

- *Update* nilai bias pada hidden layer 2 (c_k)

$$c_k = c_k - \eta \cdot \frac{\delta L}{\delta c_k} \quad (5.17)$$

- *Update* nilai bobot pada output layer (W_{kl})

$$W_{kl} = W_{kl} - \eta \cdot \frac{\delta L}{\delta W_{kl}} \quad (5.18)$$

- *Update* nilai bias pada output layer (d_l)

$$d_l = d_l - \eta \cdot \frac{\delta L}{\delta d_l} \quad (5.19)$$

Keterangan:

η = Nilai parameter *learning rate*.

$$\begin{aligned} \frac{\delta L}{\delta U_{ij}} &= \text{Nilai error pada bobot } U_{ij}. \\ \frac{\delta L}{\delta b_j} &= \text{Nilai error bias pada hidden layer 2.} \\ \frac{\delta L}{\delta V_{jk}} &= \text{Nilai error pada bobot } V_{jk}. \\ \frac{\delta L}{\delta c_k} &= \text{Nilai error bias pada hidden layer 1.} \\ \frac{\delta L}{\delta V_{jk}} &= \text{Nilai error pada bobot } V_{jk}. \\ \frac{\delta L}{\delta d_l} &= \text{Nilai error bias pada output layer.} \end{aligned}$$

5.2 Implementasi

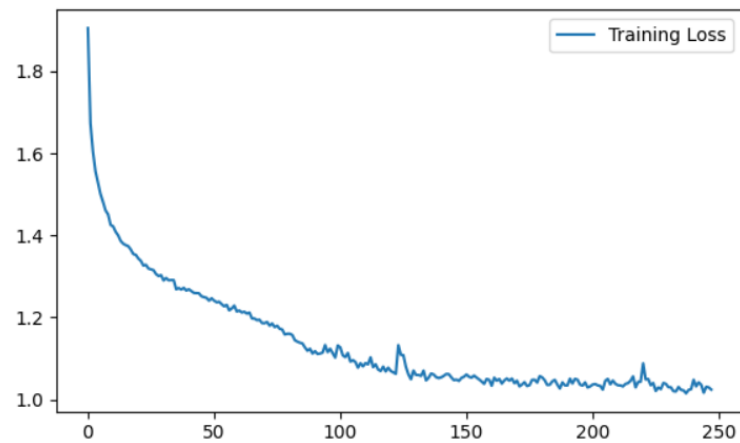
Proses *training* model NN dua *hidden layer* memanfaatkan beberapa parameter, model NN B1 dan NN B2 memiliki parameter yang berbeda. Lebih detail dapat dilihat dalam Tabel 5.1.

Tabel 5.1. Detail *Hyperparameter* Model NN B1 dan NN B2

<i>Hyperparameter</i>	<i>Nilai</i>	
Tipe jaringan	B1	B2
Fungsi aktivasi	ReLU	ReLU
<i>Optimizer</i>	SGD	SGD
<i>Input Layer</i>	15	18
<i>Hidden layer 1</i>	45	54
<i>Hidden layer 2</i>	135	162
<i>Output layer</i>	6	6
<i>Learning rate</i>	0.001	0.001

5.3 Uji Coba NN Model B1

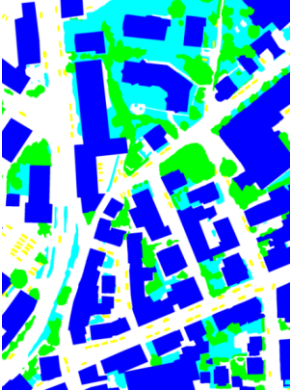
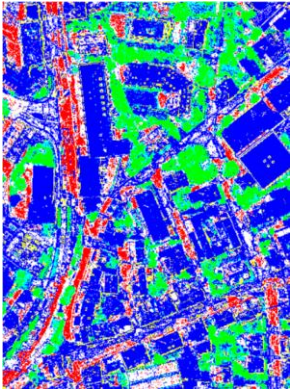
Pada tahap uji coba, *training* model NN B1 menggunakan 5 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-248 dengan nilai *loss* sebesar 1.0245. Gambar 5.2 menunjukkan grafik nilai *loss* dari proses *training* model NN B1.

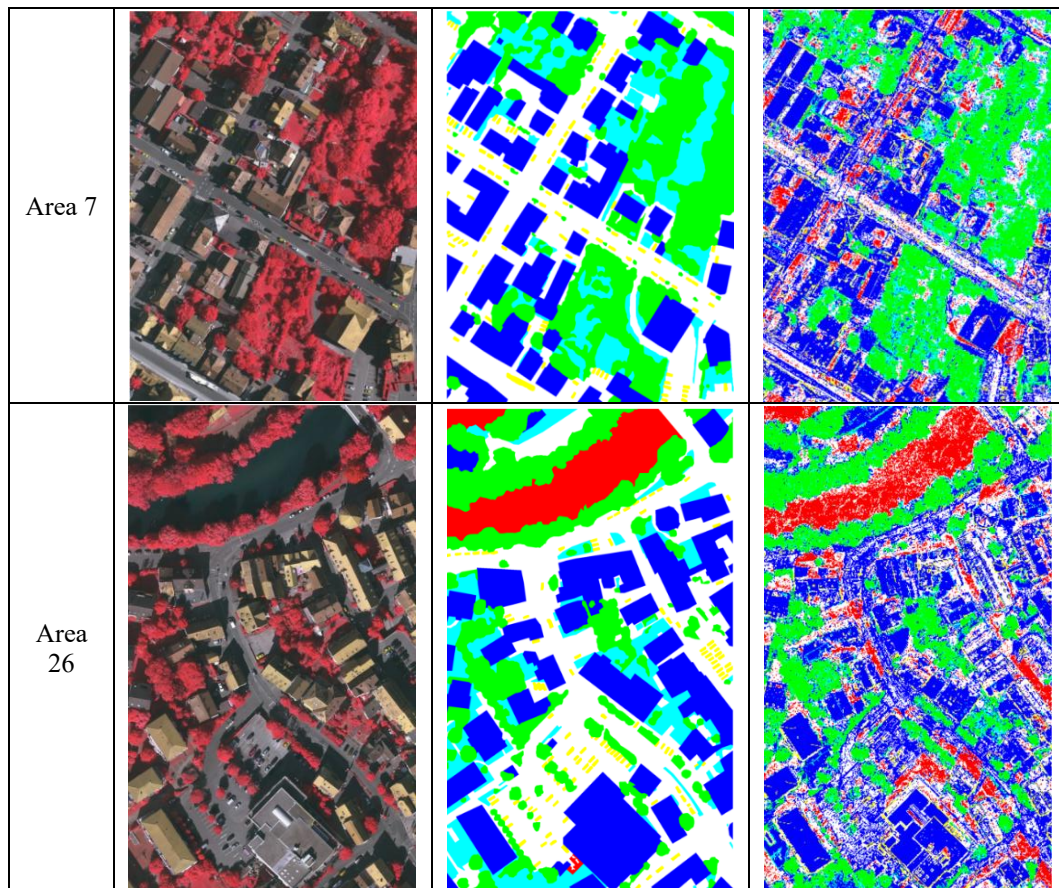


Gambar 5.2. Grafik *Loss Training* NN Model B1

Setelah proses *training* dilakukan dan model sudah optimal, tahap selanjutnya dilakukan pengujian performa model terhadap data *testing*. Didapatkan nilai *accuracy* sebesar 60.17%, *precision* sebesar 60.86%, *recall* sebesar 60.17%, dan F1-Score sebesar 59.41%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model B1 dalam melakukan klasifikasi area tutupan lahan disajikan dalam Tabel 5.2.

Tabel 5.2. Beberapa Hasil Klasifikasi Model NN B1 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			



Hasil klasifikasi model NN B1 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Tabel 5.2, model NN B1 dalam melakukan klasifikasi area tutupan lahan masih banyak menghasilkan kesalahan klasifikasi. Untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 5.3.

Tabel 5.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN B1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1377695	2168331	908815	475070
	Permukaan kedap air	598087	2705225	471523	1155076
	<i>Low vegetation</i>	157800	4061226	247150	463735
	Pepohonan	401891	4137909	165879	224232
	<i>Background</i>	0	4542504	387405	2
	Mobil	18290	4658212	195376	58033
Area 3	Bangunan	1283065	3028430	1371352	349195
	Permukaan kedap air	663418	3504577	468097	1395950
	<i>Low vegetation</i>	316424	4436049	438836	840733
	Pepohonan	688152	4596005	308643	439242
	<i>Background</i>	0	5751512	280530	0

Data	Kelas	TP	TN	FP	FN
Area 3	Mobil	15970	5778624	197555	39893
Area 7	Bangunan	917251	2707878	812089	387841
	Permukaan kedap air	573431	2847316	485349	918963
	<i>Low vegetation</i>	164595	3866144	321685	472635
	Pepohonan	863853	33011634	240722	418850
	<i>Background</i>	0	4560005	265054	0
	Mobil	23395	4559784	157635	84245
Area 11	Bangunan	709240	2775893	1177148	195157
	Permukaan kedap air	563287	2915573	333133	1045445
	<i>Low vegetation</i>	236517	3632833	416453	571635
	Pepohonan	992380	3182679	205730	476649
	<i>Background</i>	0	4754098	103340	0
	Mobil	16839	4686939	103371	50289
Area 15	Bangunan	871629	2787409	1059656	203541
	Permukaan kedap air	254628	3555956	511167	600484
	<i>Low vegetation</i>	268545	3162123	450215	1041352
	Pepohonan	953667	2883393	394653	689522
	<i>Background</i>	300	4848837	66215	6883
	Mobil	8269	4807260	83291	23415
Area 23	Bangunan	716539	2989955	969799	168745
	Permukaan kedap air	262041	3424957	618429	539611
	<i>Low vegetation</i>	240311	3253461	245849	1105417
	Pepohonan	1209403	2665680	390187	579768
	<i>Background</i>	1198	4718470	118812	6558
	Mobil	4733	4761854	67737	10714
Area 26	Bangunan	959033	2824819	956802	599431
	Permukaan kedap air	606803	2943805	745617	1043860
	<i>Low vegetation</i>	75121	4612945	225184	426835
	Pepohonan	799104	4053319	201141	286521
	<i>Background</i>	314607	4588793	304158	132527
	Mobil	15752	5107079	136763	80491
Area 32	Bangunan	711633	1959619	1828367	559281
	Permukaan kedap air	905566	1778328	663884	1711122
	<i>Low vegetation</i>	84378	4531732	164572	278218
	Pepohonan	332966	4385542	77084	263308
	<i>Background</i>	198	4840421	162877	55404
	Mobil	31929	4806628	95446	124897
Area 37	Bangunan	691269	1682440	1377001	231310
	Permukaan kedap air	313793	2681222	379682	607323
	<i>Low vegetation</i>	326811	2254404	173069	1227736
	Pepohonan	288554	3163930	252956	276580
	<i>Background</i>	0	3941890	40130	0
	Mobil	2350	3826971	136405	16294

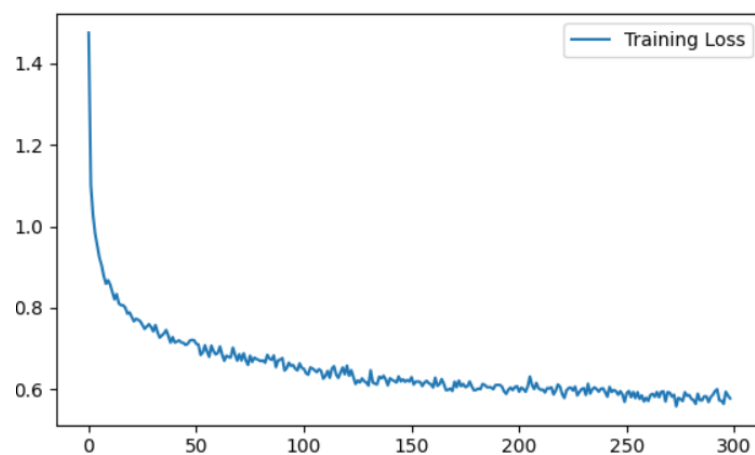
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN B1 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN B1 disajikan dalam Tabel 5.4.

Tabel 5.4. Performa Hasil Pengujian Model NN B1

Kode Area Citra Beresolusi Tinggi	<i>Accuracy</i> (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F1-Score</i> (%)	IoU (%)
Area 1	83.94	51.80	51.80	51.80	34.96
Area 3	83.14	49.20	49.10	49.15	32.59
Area 7	84.23	56.28	56.28	56.28	35.76
Area 11	84.26	52.97	51.81	52.38	35.49
Area 15	82.47	47.91	47.17	47.53	31.19
Area 23	82.95	50.6	49.04	49.84	44.19
Area 26	84.14	51.88	52.63	52.25	35.36
Area 32	80.36	40.88	41.03	40.95	25.75
Area 37	80.27	40.68	40.77	40.73	25.26

5.4 Uji Coba NN Model B2

Pada tahap uji coba, *training* model NN B2 menggunakan 6 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-299 dengan nilai *loss* sebesar 0.5778. Gambar 5.3 menunjukkan grafik nilai *loss* dari proses *training* model NN B2.

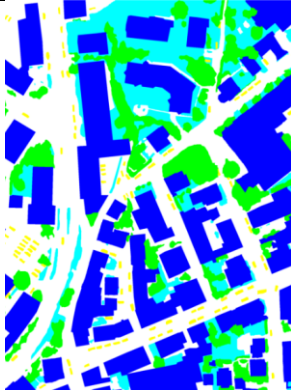
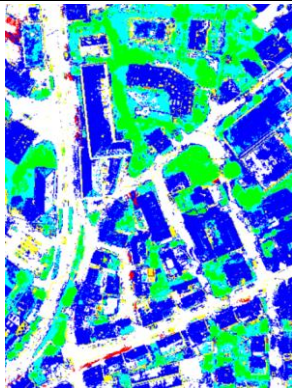
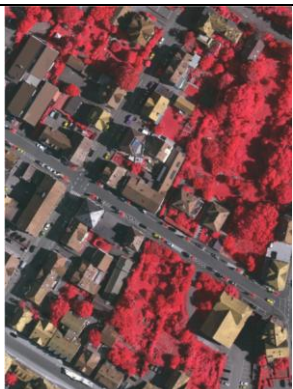
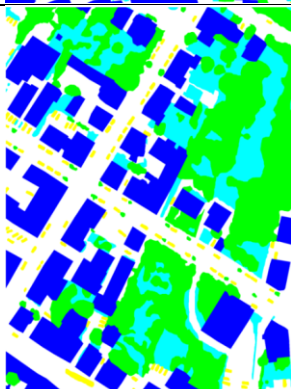
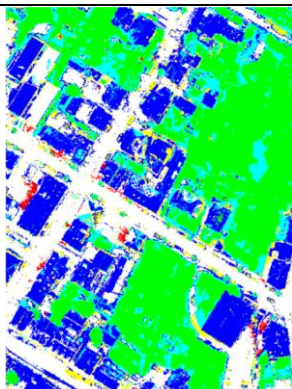
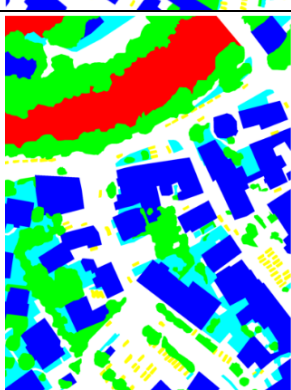
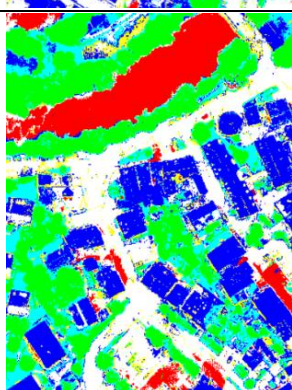


Gambar 5.3. Grafik Loss Training NN Model B2

Pada Gambar 5.3 menunjukkan nilai *loss* pada proses *training* model B2 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Tahap selanjutnya dilakukan pengujian performa model terhadap data *testing*. Didapatkan nilai *accuracy* sebesar 79.92%, *precision* sebesar 80.43%, *recall* sebesar 79.92%, dan F1-Score sebesar 79.81%. Model optimal yang

telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model B2 dalam melakukan klasifikasi area tutupan lahan lebih detail disajikan dalam ditampilkan dalam Tabel 5.5.

Tabel 5.5. Beberapa Hasil Klasifikasi Model NN B2 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN B2 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Gambar 5.3, model NN B2 memiliki kesalahan klasifikasi yang lebih sedikit dibandingkan dengan model NN B1. Untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 5.6.

Tabel 5.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN B2

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1258594	2609259	467887	594171
	Permukaan kedap air	1075910	2770823	405925	677253
	<i>Low vegetation</i>	369360	3866386	441990	252175
	Pepohonan	454193	4148356	155432	171930
	<i>Background</i>	0	4892479	37430	2
	Mobil	25778	4616176	237412	50545
Area 3	Bangunan	1040030	3591477	808305	592230
	Permukaan kedap air	1071828	3685400	287274	987540
	<i>Low vegetation</i>	721725	4015700	859185	435432
	Pepohonan	745366	4636634	268014	382028
	<i>Background</i>	0	5997242	34800	0
	Mobil	19333	5799997	176182	36530
Area 7	Bangunan	881590	3214702	305265	423502
	Permukaan kedap air	997004	3066750	265915	495390
	<i>Low vegetation</i>	236998	3831737	356092	400232
	Pepohonan	1137903	3175749	366607	144800
	<i>Background</i>	0	4790279	34780	0
	Mobil	38962	4513476	203943	68678
Area 11	Bangunan	539385	3546338	406703	365012
	Permukaan kedap air	949008	3060574	188132	659724
	<i>Low vegetation</i>	436960	3376526	672760	371192
	Pepohonan	1199920	3123399	265010	269109
	<i>Background</i>	0	4833198	24240	0
	Mobil	26640	4641630	148680	40488
Area 15	Bangunan	706724	3589954	257111	368446
	Permukaan kedap air	409414	3865087	202036	445698
	<i>Low vegetation</i>	611482	3036830	575508	698415
	Pepohonan	1321658	2734784	544262	321531
	<i>Background</i>	1175	4884232	30820	6008
	Mobil	15755	4644261	246290	15929
Area 23	Bangunan	521014	3787993	171761	364270
	Permukaan kedap air	292789	3908212	135174	508863
	<i>Low vegetation</i>	852150	2746240	753070	493578
	Pepohonan	1438383	2616050	439817	350788
	<i>Background</i>	50	4821352	15930	7706
	Mobil	8240	4612931	216660	7207
Area 26	Bangunan	964126	3530602	251019	594338

Data	Kelas	TP	TN	FP	FN
Area 26	Permukaan kedap air	1084365	3249912	439510	566298
	<i>Low vegetation</i>	316202	4431536	406593	185754
	Pepohonan	923278	4098708	155752	162347
	<i>Background</i>	371674	4769960	122991	75460
	Mobil	30652	4969919	273923	65591
Area 32	Bangunan	518124	2985135	802851	752790
	Permukaan kedap air	1706475	2091462	350750	910213
	<i>Low vegetation</i>	205377	4295381	400923	157219
	Pepohonan	337636	4384687	77939	258638
	<i>Background</i>	903	4830526	172772	54699
	Mobil	41382	4458306	443768	115444
Area 37	Bangunan	694153	2527339	532102	228426
	Permukaan kedap air	601377	2825316	235588	319739
	<i>Low vegetation</i>	961172	2081120	346353	593375
	Pepohonan	289991	3284172	132714	275143
	<i>Background</i>	0	3969345	12675	0
	Mobil	3492	3790973	172403	15152

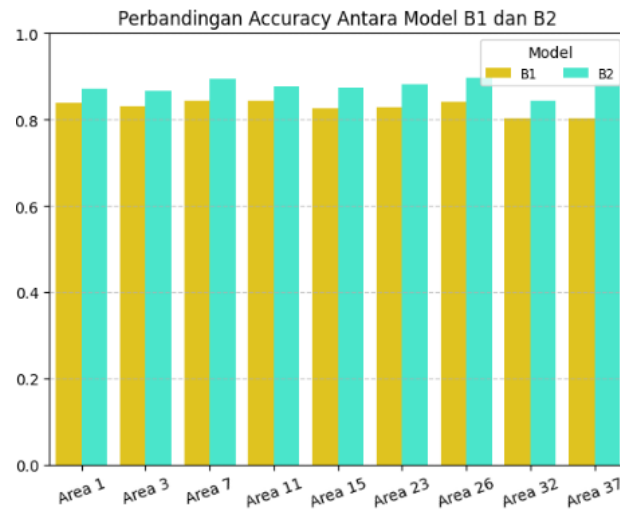
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN B2 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN B2 disajikan dalam Tabel 5.5.

Tabel 5.7. Performa Hasil Pengujian Model NN B2

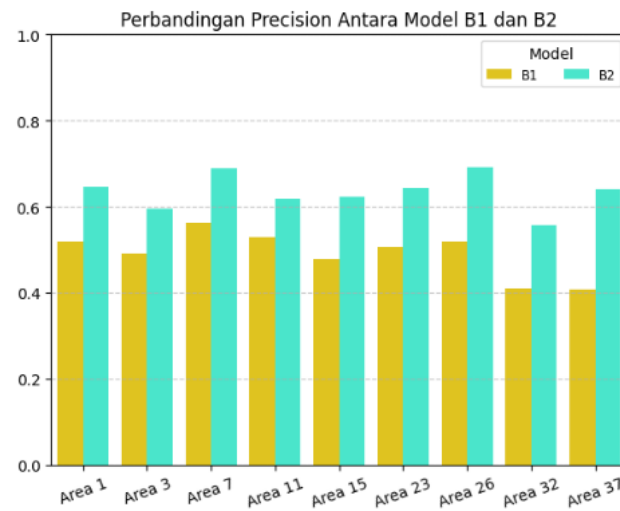
Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	87.24	64.68	64.79	64.73	52.58
Area 3	86.59	59.60	59.67	59.64	45.36
Area 7	89.42	68.80	68.89	68.85	53.46
Area 11	87.53	61.79	61.85	61.82	46.52
Area 15	87.49	62.34	62.29	62.32	46.94
Area 23	88.07	64.21	64.21	64.21	47.82
Area 26	89.68	69.08	69.08	69.08	52.65
Area 32	84.35	55.71	55.71	55.71	41.32
Area 37	88.05	64.12	64.12	64.12	47.34

5.5 Komparasi Performa NN Model B1 dan B2

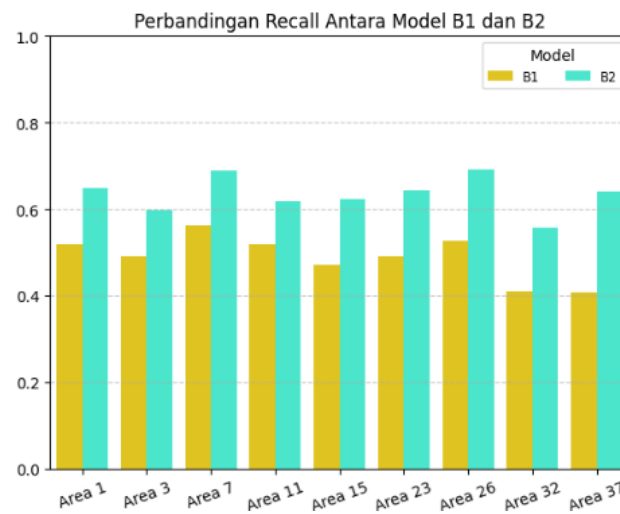
Selanjutnya dilakukan komparasi antara NN model B1 dan B2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 5.4 hingga Gambar 5.8.



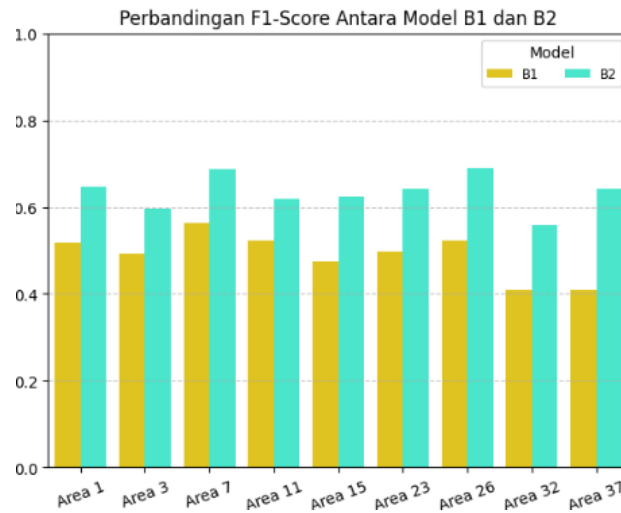
Gambar 5.4. Perbandingan Accuracy Antara Model NN B1 dan B2



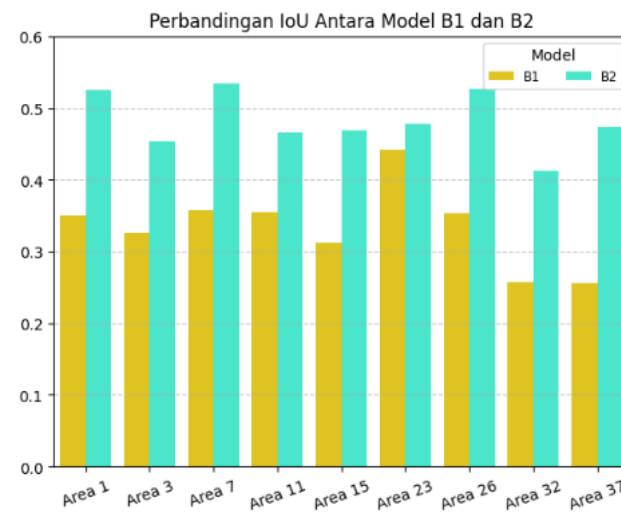
Gambar 5.5. Perbandingan Precision Antara Model NN B1 dan B2



Gambar 5.6. Perbandingan Recall Antara Model NN B1 dan B2



Gambar 5.7. Perbandingan F1-Score Antara Model NN B1 dan B2



Gambar 5.8. Perbandingan IoU Antara Model NN B1 dan B2

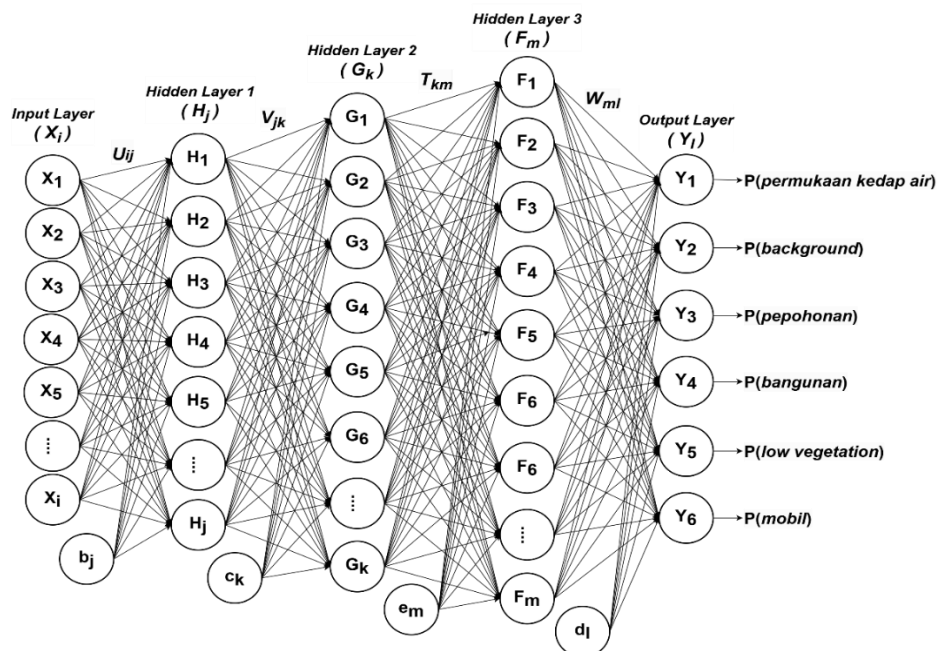
Dari perbandingan yang dilakukan pada Gambar 5.4 hingga Gambar 5.8, ditemukan bahwa model performa model NN B2 yang memanfaatkan 6 fitur GLCM memiliki performa yang lebih tinggi dibandingkan dengan model NN B1 yang memanfaatkan 5 fitur GLCM dari semua metrik, baik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU.

BAB VI

NEURAL NETWORK TIGA HIDDEN LAYER

6.1 Desain

Arsitektur pada model NN C1 dan C2 menggunakan tiga *hidden layer* untuk melakukan klasifikasi pada area tutupan lahan citra beresolusi tinggi. Detail arsitektur dari NN tiga *hidden layer* terdapat pada Gambar 6.1.



Gambar 6.1. Arsitektur Neural Network 3 Hidden Layer

Terjadi proses perhitungan nilai prediksi kelas pada *forward propagation* dan *update* nilai bobot pada *backpropagation*. Lebih detail, berikut adalah proses yang terjadi di dalam *neural network 3 hidden layers*:

1. Inisialisasi data *input*, parameter, dan *hyperparameter* model.
2. Fase *forward propagation*:

- Mengkalikan setiap data *input* pada *input layer* (X_i) dengan bobot (U_{ij}) yang menyambungkan *input layer* dengan *hidden layer* 1 (H_j). Kemudian jumlahkan hasil perkalian pada masing-masing *neuron* di *hidden layer* 1. Kemudian dilanjutkan dengan menambahkan nilai *bias* (b_j) pada setiap hasil penjumlahan pada masing-masing *neuron* di *hidden layer* 1. Dilanjutnya dengan memasukkan nilai masing-masing *neuron* di *hidden layer* 1 hasil penjumlahan *bias* kedalam fungsi aktivasi ReLU. Rumus propagasi maju di *hidden layer* 1 disajikan dalam Persamaan 6.1 dan 6.2.

$$Z_j = \sum_{i=1}^n (X_i U_{ij}) + b_j \quad (6.1)$$

Keterangan:

Z_j = Hasil perkalian antara *input* dengan bobot yang kemudian ditambah bias.

X_i = Data *input* pada *input layer*.

U_{ij} = Bobot yang menyambungkan *input layer* dengan *hidden layer*.

b_j = Nilai bias dari *neuron* pada *hidden layer* 1.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$H_j = \text{ReLU}(Z_j) = \max(0, Z_j) \quad (6.2)$$

Keterangan:

H_j = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 2 (G_k) dengan cara menambahkan bias *hidden layer* 2 (c_k) pada hasil perkalian antara nilai prediksi *hidden layer* 1 dengan nilai bobot (V_{jk}) yang menyambungkan *hidden layer* 1 ke *hidden layer* 2 (Z_k). Rumus propagasi maju di *hidden layer* 2 ditampilkan dalam Persamaan 6.3 dan 6.4.

$$Z_k = \sum_{j=1}^n (H_j V_{jk}) + c_k \quad (6.3)$$

Keterangan:

Z_k = Hasil perkalian antara *hidden layer* 1 dengan bobot, lalu ditambah bias.

H_j = Nilai prediksi pada *hidden layer* 1.

V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

c_k = Nilai bias dari *neuron* pada *hidden layer* 2.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$G_k = ReLU(Z_k) = \max(0, Z_k) \quad (6.4)$$

Keterangan:

G_k = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 3 (F_m) dengan cara menambahkan bias *hidden layer* 3 (e_m) pada hasil perkalian antara nilai prediksi *hidden layer* 2 dengan nilai bobot (T_{km}) yang menyambungkan *hidden layer* 2 ke *hidden layer* 3 (Z_m). Rumus propagasi maju di *hidden layer* 3 ditampilkan dalam Persamaan 6.5 dan 6.6.

$$Z_m = \sum_{k=1}^n (G_k T_{km}) + e_m \quad (6.5)$$

Keterangan:

Z_m = Hasil perkalian antara *hidden layer* 2 dengan bobot, lalu ditambah bias.

G_k = Nilai prediksi pada *hidden layer* 2.

T_{km} = Bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3.

e_m = Nilai bias dari *neuron* pada *hidden layer* 3.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$F_m = ReLU(Z_m) = \max(0, Z_m) \quad (6.6)$$

Keterangan:

F_m = Nilai pada *hidden layer* 3 setelah diaktivasi menggunakan fungsi ReLU.

- Selanjutnya adalah menghitung nilai prediksi pada *output layer* (Y_l). Cara perhitungannya sama seperti pada perhitungan pada *hidden layer* 3 (F_m),

yang membedakan adalah bobot (W_{ml}) yang menyambungkan *hidden layer* 3 ke *output layer*, dan bias (d_l) pada *output layer*. Rumus propagasi maju di *output layer* disajikan dalam Persamaan 6.7.

$$Z_l = \sum_{m=1}^n (F_m W_{ml}) + d_l \quad (6.7)$$

Keterangan:

Z_l = Hasil perkalian antara F_m dengan W_{ml} lalu ditambah dengan bias d_l .
 F_m = Nilai hasil perhitungan pada *hidden layer* 3.
 W_{ml} = Bobot yang menyambungkan *hidden layer* 3 dengan *output layer*.
 d_l = Nilai bias dari *neuron* pada *output layer*.

- Langkah berikutnya adalah menghitung nilai probabilitas dari setiap kelas menggunakan fungsi *softmax*. Total dari nilai semua probabilitas setiap kelas adalah 1, kelas dengan nilai probabilitas tertinggi merupakan keluaran hasil prediksi model ($P(Y_l)$). Rumus perhitungan *softmax* disajikan dalam Persamaan 6.8.

$$P(Y_l) = \frac{\exp(Z_l)}{\sum_{m=1}^6 \exp(Z_m)} \quad (6.8)$$

Keterangan:

$P(Y_l)$ = Nilai probabilitas kelas ke- l yang terprediksi benar.
 $\exp(Z_l)$ = Nilai eksponensial dari kelas l di *output layer*.
 $\sum_{m=1}^6 \exp(Z_m)$ = Jumlah dari semua nilai eksponensial dari setiap kelas.

- Hitung nilai *loss* untuk mengetahui seberapa jauh jarak prediksi model terhadap kelas asli atau *ground truth*. Nilai *loss* yang dihasilkan dapat memberikan gambaran konvergensi terhadap proses *training* model. Pada penelitian ini *loss function* yang digunakan adalah *sparse categorical cross-entropy*. Rumusnya disajikan dalam Persamaan 6.9.

$$L = -\frac{1}{n} \sum_{i=1}^n \log(P(Y_{i,true})) \quad (6.9)$$

Keterangan:

L = Nilai *loss* hasil perhitungan *loss function*.
 $P(Y_{i,true})$ = Nilai probabilitas kelas yang diprediksi oleh model.
 n = Jumlah data atau sampel dalam dataset.

5. Fase *backward propagation*:

- Menghitung nilai *error* pada *output layer* dengan turunan dari *loss function* ($\frac{\delta L}{\delta Z_l}$) dengan cara menghitung jarak nilai atau perbedaan antara nilai probabilitas kelas terprediksi dengan kelas *ground truth*. Rumus perhitungan nilai *error* pada *output layer* terdapat pada Persamaan 6.10.

$$\frac{\delta L}{\delta Z_l} = P(Y_l) - y_l \quad (6.10)$$

Keterangan:

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* pada *output layer*.

$P(Y_l)$ = Nilai probabilitas kelas k yang terprediksi benar.

y_l = Kelas asli atau *ground truth*.

- Setelah mendapatkan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$), selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer* 3 dan *output layer* ($\frac{\delta L}{\delta W_{ml}}$). Rumus perhitungannya ditampilkan pada Persamaan 6.11.

$$\frac{\delta L}{\delta W_{ml}} = \frac{\delta L}{\delta Z_l} \cdot F_m \quad (6.11)$$

Keterangan:

F_m = *Output* atau hasil fungsi aktivasi pada *hidden layer* 3.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer*.

$\frac{\delta L}{\delta W_{ml}}$ = Nilai *error* pada bobot W_{kl} .

- Kemudian untuk nilai *error* dari bias pada *output layer* ($\frac{\delta L}{\delta d_l}$) memiliki nilai yang sama dengan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 3. Rumus perhitungan dari proses ini disajikan dalam Persamaan 6.12.

$$\frac{\delta L}{\delta Z_m} = \sum_{l=1}^6 \frac{\delta L}{\delta Z_l} \cdot W_{ml} \cdot ReLU'(Z_m) \quad (6.12)$$

Keterangan:

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer* sebelum diaktivasi.

W_{ml} = Bobot yang menyambungkan *hidden layer* 3 dengan *output layer*.

$ReLU'(Z_m)$ = Turunan dari fungsi aktivasi ReLU.

- Selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer* 2 dan *hidden layer* 3 ($\frac{\delta L}{\delta T_{km}}$). Rumus perhitungannya ditampilkan pada Persamaan 6.13.

$$\frac{\delta L}{\delta T_{km}} = \frac{\delta L}{\delta Z_m} \cdot G_k \quad (6.13)$$

Keterangan:

G_k = *Output* atau hasil fungsi aktivasi pada *hidden layer* 2.

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

$\frac{\delta L}{\delta T_{km}}$ = Nilai *error* pada bobot T_{km} .

- Kemudian untuk nilai *error* dari bias pada *hidden layer* 3 ($\frac{\delta L}{\delta e_m}$) memiliki nilai yang sama dengan nilai *error* pada *hidden layer* 3 ($\frac{\delta L}{\delta Z_m}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 2. Rumus perhitungan dari proses ini disajikan dalam Persamaan 6.14.

$$\frac{\delta L}{\delta Z_k} = \sum_{m=1}^n \frac{\delta L}{\delta Z_m} \cdot T_{km} \cdot ReLU'(Z_k) \quad (6.14)$$

Keterangan:

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

T_{km} = Bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3.

$ReLU'(Z_k)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 1 ($\frac{\delta L}{\delta V_{jk}}$). Rumus dari proses ini disajikan dalam Persamaan 6.15.

$$\frac{\delta L}{\delta V_{jk}} = \frac{\delta L}{\delta Z_k} \cdot H_j \quad (6.15)$$

Keterangan:

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

H_j = *Output* atau hasil fungsi aktivasi pada *hidden layer* 2.

- Kemudian untuk nilai *error* dari bias *hidden layer* 2 ($\frac{\delta L}{\delta C_k}$) memiliki nilai yang sama dengan nilai *error hidden layer* 2 sebelum diaktivasi ($\frac{\delta L}{\delta Z_k}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 1.

Rumus perhitungan dari proses ini disajikan dalam Persamaan 6.16.

$$\frac{\delta L}{\delta Z_j} = \sum_{k=1}^n \frac{\delta L}{\delta Z_k} \cdot V_{jk} \cdot ReLU'(Z_j) \quad (6.16)$$

Keterangan:

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* 1 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

$ReLU'(Z_j)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 1 dengan *input layer* ($\frac{\delta L}{\delta V_{jk}}$). Rumus dari proses ini disajikan dalam Persamaan 6.17.

$$\frac{\delta L}{\delta U_{ij}} = \frac{\delta L}{\delta Z_j} \cdot X_i \quad (6.17)$$

Keterangan:

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* 1 sebelum diaktivasi.

X_i = Data *input*.

- Kemudian untuk nilai *error* dari bias *hidden layer* 1 ($\frac{\delta L}{\delta b_j}$) memiliki nilai yang sama dengan nilai *error hidden layer* 1 sebelum diaktivasi ($\frac{\delta L}{\delta Z_j}$).

6. Melakukan *update* terhadap nilai bobot yang menyambungkan *input layer* dengan *hidden layer 1*, *hidden layer 1* dengan *hidden layer 2*, *hidden layer 2* dengan *hidden layer 3*, dan *hidden layer 3* dengan *output layer*. Secara berurutan rumus untuk proses ini ditampilkan dalam Persamaan 6.18 hingga 6.25.

- *Update* nilai bobot pada *hidden layer 1* (U_{ij})

$$U_{ij} = U_{ij} - \eta \cdot \frac{\delta L}{\delta U_{ij}} \quad (6.18)$$

- *Update* nilai bias pada *hidden layer 1* (b_j)

$$b_j = b_j - \eta \cdot \frac{\delta L}{\delta b_j} \quad (6.19)$$

- *Update* nilai bobot pada *hidden layer 2* (V_{jk})

$$V_{jk} = V_{jk} - \eta \cdot \frac{\delta L}{\delta V_{jk}} \quad (6.20)$$

- *Update* nilai bias pada *hidden layer 2* (c_k)

$$c_k = c_k - \eta \cdot \frac{\delta L}{\delta c_k} \quad (6.21)$$

- *Update* nilai bobot pada *hidden layer 3* (T_{km})

$$T_{km} = T_{km} - \eta \cdot \frac{\delta L}{\delta T_{km}} \quad (6.22)$$

- *Update* nilai bias pada *hidden layer 3* (e_m)

$$e_m = e_m - \eta \cdot \frac{\delta L}{\delta e_m} \quad (6.23)$$

- *Update* nilai bobot pada *output layer* (W_{ml})

$$W_{ml} = W_{ml} - \eta \cdot \frac{\delta L}{\delta W_{ml}} \quad (6.24)$$

- *Update* nilai bias pada *output layer* (d_l)

$$d_l = d_l - \eta \cdot \frac{\delta L}{\delta d_l} \quad (6.25)$$

Keterangan:

η = Nilai parameter *learning rate*.

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta b_j}$ = Nilai *error* bias pada *hidden layer* 1.

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta c_k}$ = Nilai *error* bias pada *hidden layer* 2.

$\frac{\delta L}{\delta T_{km}}$ = Nilai *error* pada bobot T_{km} .

$\frac{\delta L}{\delta e_m}$ = Nilai *error* bias pada *hidden layer* 3.

$\frac{\delta L}{\delta W_{ml}}$ = Nilai *error* pada bobot W_{ml} .

$\frac{\delta L}{\delta d_l}$ = Nilai *error* bias pada *output layer*.

6.2 Implementasi

Proses *training* model NN tiga *hidden layer* memanfaatkan beberapa parameter, model NN C1 dan NN C2 memiliki parameter yang berbeda. Lebih detail dapat dilihat dalam Tabel 6.1.

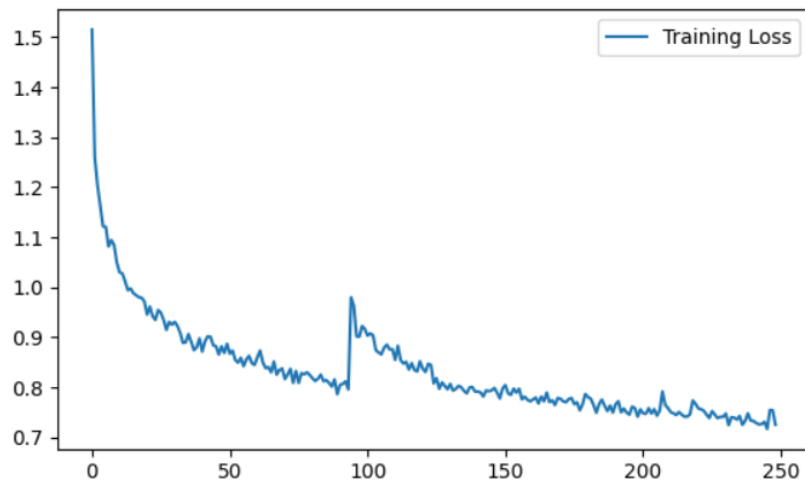
Tabel 6.1. Detail *Hyperparameter* Model NN C1 dan NN C2

Hyperparameter	Nilai	
	C1	C2
Tipe jaringan	C1	C2
Fungsi aktivasi	ReLU	ReLU
Optimizer	SGD	SGD
Input Layer	15	18
Hidden layer 1	45	54
Hidden layer 2	135	162
Hidden layer 3	405	486
Output layer	6	6
Learning rate	0.001	0.001

6.3 Uji Coba NN Model C1

Pada tahap uji coba, *training* model NN C1 menggunakan 5 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk

data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-249 dengan nilai *loss* sebesar 0.7252. Gambar 6.2 menunjukkan grafik nilai *loss* dari proses *training* model NN C1.

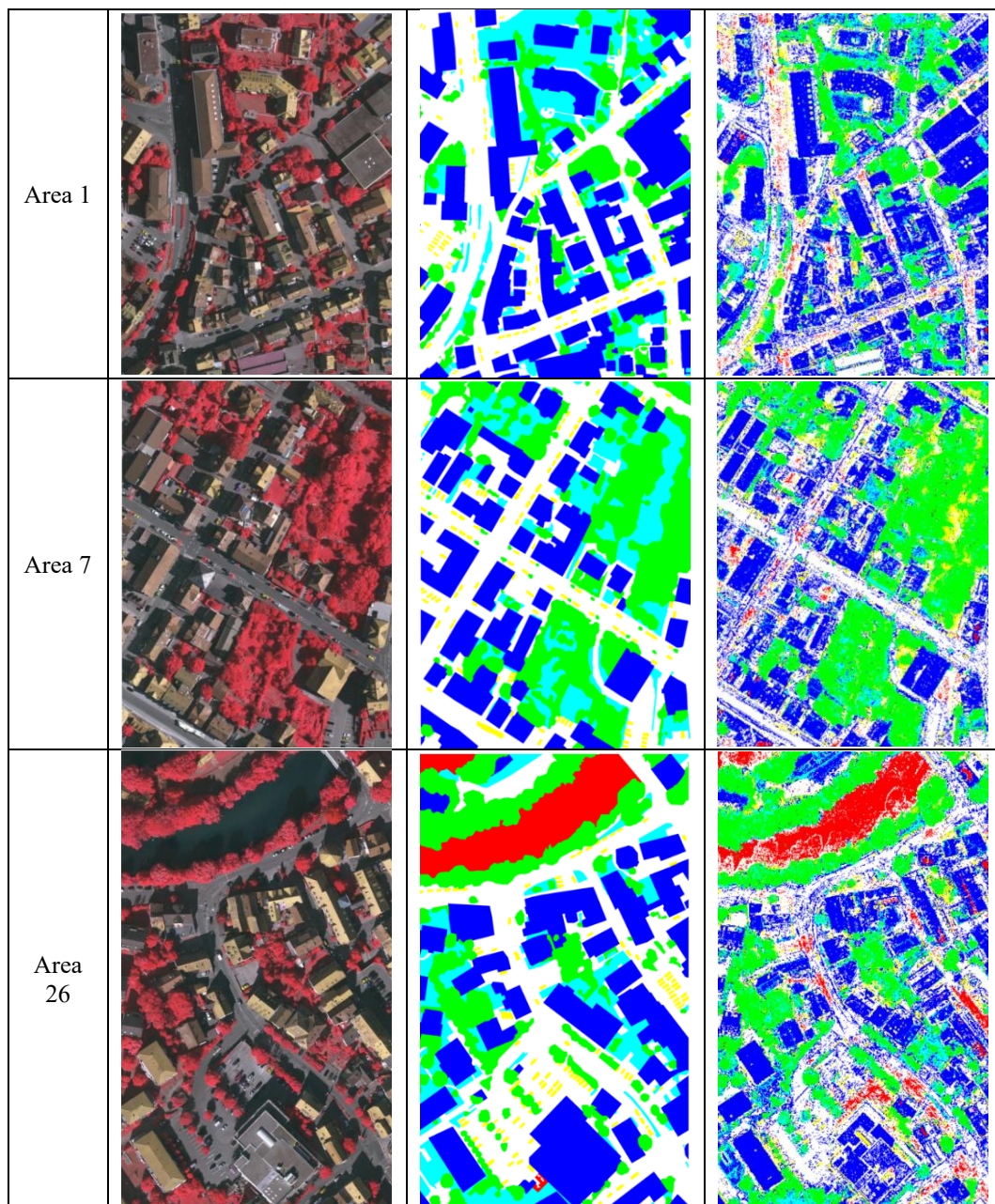


Gambar 6.2. Grafik *Loss Training* NN Model C1

Pada Gambar 6.2 menunjukkan nilai *loss* pada proses *training* model C1 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Tahap selanjutnya dilakukan pengujian performa model terhadap data *testing*. Didapatkan nilai *accuracy* sebesar 69.75%, *precision* sebesar 70.65%, *recall* sebesar 69.75%, dan F1-Score sebesar 69.38%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model C1 dalam melakukan klasifikasi area tutupan lahan disajikan pada Tabel 6.2.

Tabel 6.2. Beberapa Hasil Klasifikasi Model NN C1 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
-----------	-------------------------	--------------	---------------------------------



Hasil klasifikasi model NN C1 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Tabel 6.2, model NN C1 dalam melakukan klasifikasi area tutupan lahan memiliki hasil visual yang lebih baik dibandingkan model NN B1 dan A1, meskipun dari ketiga model tersebut menggunakan GLCM 5 fitur. Untuk menghitung nilai metrik *accuracy*, *precision*,

recall, *F1-Score*, dan *IoU* dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi

tutupan lahan yang secara detail disajikan dalam Tabel 6.3.

Tabel 6.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN C1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1306943	2501009	576137	545822
	Permukaan kedap air	1063477	2629999	546749	689686
	<i>Low vegetation</i>	192984	4022920	285456	428551
	Pepohonan	421265	4108283	195505	204858
	<i>Background</i>	0	4869524	60385	2
	Mobil	23951	4596529	257059	52372
Area 3	Bangunan	1150414	3391099	1008683	481846
	Permukaan kedap air	1110567	3413606	559068	948801
	<i>Low vegetation</i>	433827	4300752	574133	723330
	Pepohonan	670281	4599489	305159	457113
	<i>Background</i>	0	6012882	19160	0
	Mobil	17844	5793273	182906	38019
Area 7	Bangunan	873512	3049954	470013	431580
	Permukaan kedap air	938789	2903250	429415	553605
	<i>Low vegetation</i>	159767	3932191	255638	477463
	Pepohonan	1010064	3233130	309226	272639
	<i>Background</i>	0	4774814	50245.0	0
	Mobil	33103	4422132	295287	74537
Area 11	Bangunan	599150	3259003	694038	305247
	Permukaan kedap air	886158	2945854	302852	722574
	<i>Low vegetation</i>	303692	3557763	491523	504460
	Pepohonan	1043602	3165181	223228	425427
	<i>Background</i>	0	4854973	2465	0
	Mobil	23242	4502822	287488	43886
Area 15	Bangunan	768379	3319929	527136	306791
	Permukaan kedap air	375842	3744800	322323	479270
	<i>Low vegetation</i>	486263	2799121	813217	823634
	Pepohonan	800435	2919871	359175	842754
	<i>Background</i>	1475	4913502	1550	5708
	Mobil	13383	4437494	453057	18301
Area 23	Bangunan	647349	3590478	369276	237935
	Permukaan kedap air	401836	3650619	392767	399816
	<i>Low vegetation</i>	491712	3070797	428513	854016
	Pepohonan	1234330	2689507	366360	554841
	<i>Background</i>	0	4835192	2090	7756
	Mobil	6665	4325451	504140	8782
Area 26	Bangunan	964287	3260233	521388	594177
	Permukaan kedap air	976807	3005994	683428	673856
	<i>Low vegetation</i>	145808	4526682	311447	356148
	Pepohonan	828204	4029809	224651	257421
	<i>Background</i>	293217	4775128	117823	153917
	Mobil	28922	4999739	244103	67321
Area 32	Bangunan	588491	2898302	889684	682423
	Permukaan kedap air	1741306	1762868	679344	875382
	<i>Low vegetation</i>	180964	4362468	333836	181632
	Pepohonan	231360	4410311	52315	364914
	<i>Background</i>	29	4982377	20921	55573

Data	Kelas	TP	TN	FP	FN
Area 32	Mobil	52020	4613444	288630	104806
Area 37	Bangunan	645011	2201437	858004	277568
	Permukaan kedap air	497559	2663228	397676	423557
	<i>Low vegetation</i>	684872	2120565	306908	869675
	Pepohonan	237574	3254710	162176	327560
	<i>Background</i>	0	3968095	13925	0
	Mobil	3916	3788977	174399	14728

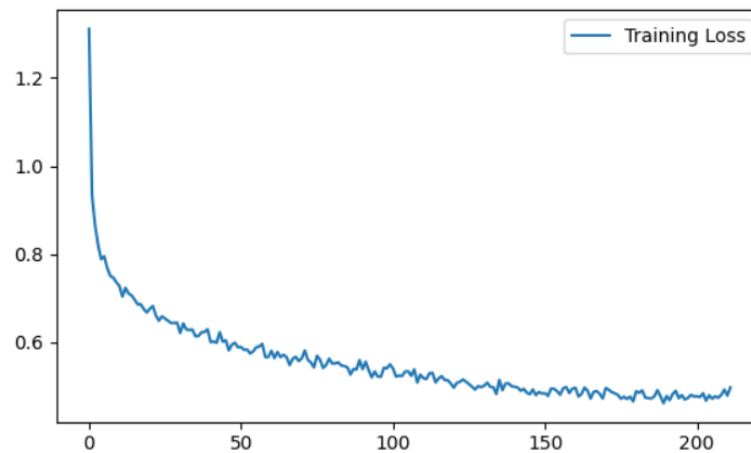
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN C1 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN C1 disajikan dalam Tabel 6.4.

Tabel 6.4. Performa Hasil Pengujian Model NN C1

Kode Area Citra Beresolusi Tinggi	<i>Accuracy</i> (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F1-Score</i> (%)	IoU (%)
Area 1	87.00	61.05	61.04	61.05	43.97
Area 3	85.48	56.50	56.58	56.54	39.39
Area 7	87.65	62.74	62.64	62.69	45.66
Area 11	86.48	58.82	58.82	58.82	41.65
Area 15	83.28	49.67	49.88	49.75	33.15
Area 23	85.79	57.53	57.53	57.53	40.38
Area 26	89.67	60.67	60.67	60.67	43.54
Area 32	85.09	55.24	55.24	55.24	38.15
Area 37	83.97	51.96	51.96	51.96	35.12

6.4 Uji Coba NN Model C2

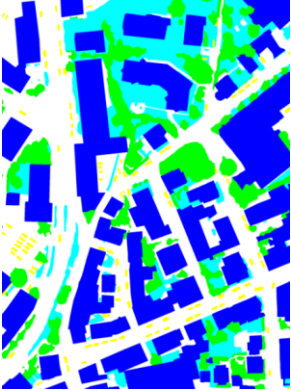
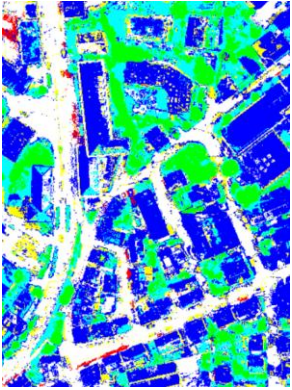
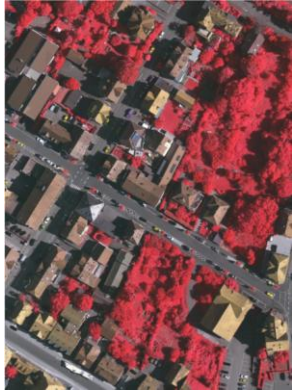
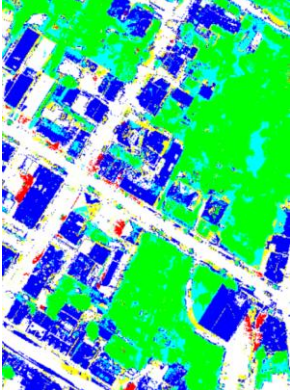
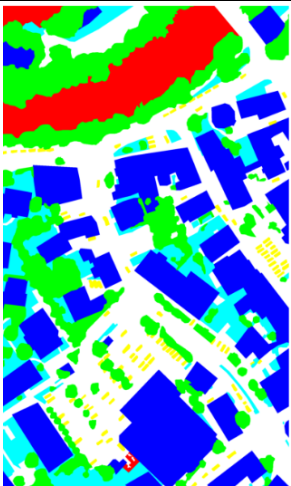
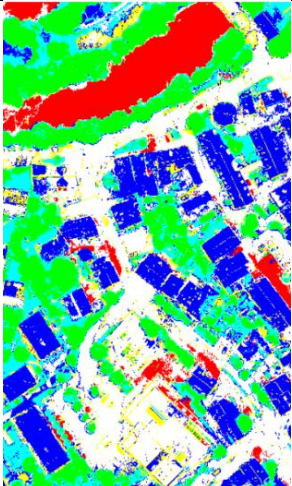
Pada tahap uji coba, *training* model NN C2 menggunakan 6 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-212 dengan nilai *loss* sebesar 0.4974. Gambar 6.3 menunjukkan grafik nilai *loss* dari proses *training* model NN C2



Gambar 6.3. Grafik *Loss Training* NN Model C2

Pada Gambar 6.3 menunjukkan nilai *loss* pada proses *training* model C2 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Didapatkan nilai *accuracy* sebesar 83.06%, *precision* sebesar 83.26%, *recall* sebesar 83.06%, dan F1-Score sebesar 83.04%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model C2 dalam melakukan klasifikasi area tutupan lahan lebih detail disajikan dalam ditampilkan dalam Tabel 6.5.

Tabel 6.5. Beberapa Hasil Klasifikasi Model NN C2 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN C2 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Terlihat dalam Tabel 6.5, model NN C2 menghasilkan hasil klasifikasi tutupan lahan yang lebih baik dibandingkan model NN C1. Selanjutnya untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*,

dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 6.6.

Tabel 6.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN C2

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1183366	2690261	386885	669399
	Permukaan kedap air	1113485	2693463	483285	639678
	<i>Low vegetation</i>	367335	3863911	444465	254200
	Pepohonan	429728	4167406	136382	196395
	<i>Background</i>	0	4856354	73555	2
	Mobil	26688	4568851	284737	49635
Area 3	Bangunan	937828	3696090	703692	694432
	Permukaan kedap air	1159554	3633421	339253	899814
	<i>Low vegetation</i>	721200	3915905	958980	435957
	Pepohonan	682263	4667341	237307	445131
	<i>Background</i>	0	5992342	39700	0
	Mobil	21247	5745161	231018	34616
Area 7	Bangunan	831240	3245047	274920	473852
	Permukaan kedap air	1006251	3020017	312648	486143
	<i>Low vegetation</i>	247835	3823309	364520	389395
	Pepohonan	1105818	3194104	348252	176885
	<i>Background</i>	0	4772529	52530	0
	Mobil	39378	4475752	241667	68262
Area 11	Bangunan	519929	3584642	368399	384468
	Permukaan kedap air	968417	3026048	222658	640315
	<i>Low vegetation</i>	443025	3330596	718690	365127
	Pepohonan	1134495	3155454	232955	334534
	<i>Background</i>	0	4825843	31595	0
	Mobil	28557	4601592	188718	38571
Area 15	Bangunan	657787	3595482	251583	417383
	Permukaan kedap air	412864	3806632	260491	442248
	<i>Low vegetation</i>	623758	2994506	617832	686139
	Pepohonan	1269618	2772279	506767	373571
	<i>Background</i>	0	4873557	41495	7183
	Mobil	16022	4626533	264018	15662
Area 23	Bangunan	485654	3801993	157761	399630
	Permukaan kedap air	298939	3872937	170449	502713
	<i>Low vegetation</i>	898423	2673563	825747	447305
	Pepohonan	1354519	2678271	377596	434652
	<i>Background</i>	50	4820257	17025	7706
	Mobil	7824	4578540	251051	7623
Area 26	Bangunan	913711	3567592	214029	644753
	Permukaan kedap air	1096175	3209522	479900	554488
	<i>Low vegetation</i>	316179	4429538	408591	185777
	Pepohonan	899382	4112702	141758	186243
	<i>Background</i>	375264	4760530	132421	71870
	Mobil	31566	4912733	331109	64677
Area 32	Bangunan	546344	2995555	792431	724570
	Permukaan kedap air	1700344	2066331	375881	916344
	<i>Low vegetation</i>	206410	4261839	434465	156186
	Pepohonan	303669	4401170	61456	292605
	<i>Background</i>	1002	4818600	184698	54600

Data	Kelas	TP	TN	FP	FN
Area 32	Mobil	43055	4492929	409145	113771
Area 37	Bangunan	665328	2506564	552877	257251
	Permukaan kedap air	628456	2775140	285764	292660
	<i>Low vegetation</i>	926258	2099226	328247	628289
	Pepohonan	271230	3293906	122980	293904
	<i>Background</i>	0	3968970	13050	0
	Mobil	3576	3779122	184254	15068

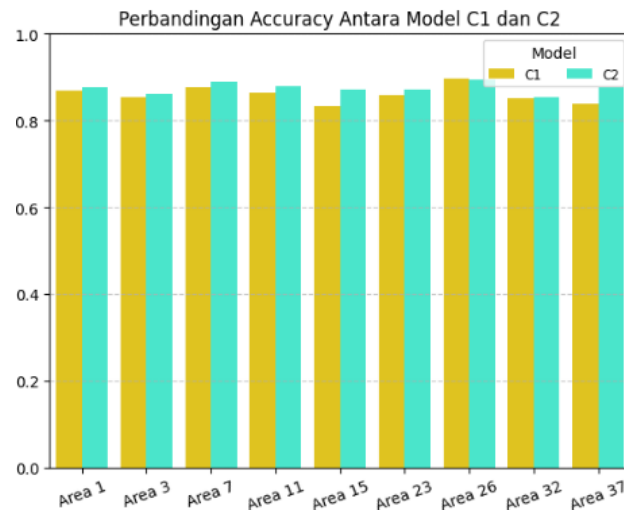
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN C2 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN C2 disajikan dalam Tabel 6.7.

Tabel 6.7. Performa Hasil Pengujian Model NN C2

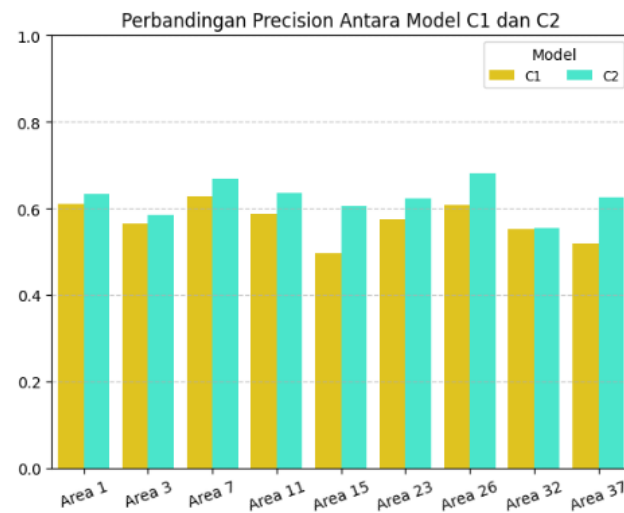
Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	87.67	63.37	63.38	63.38	46.40
Area 3	86.10	58.37	58.39	58.38	41.23
Area 7	88.83	66.92	66.86	66.89	50.26
Area 11	88.03	63.66	63.68	63.67	46.71
Area 15	87.02	60.45	60.56	60.50	43.39
Area 23	87.15	62.18	62.02	62.10	45.05
Area 26	89.37	68.00	67.92	67.96	51.45
Area 32	85.33	55.50	55.46	55.48	38.39
Area 37	87.78	62.63	62.70	62.66	45.63

6.5 Komparasi Performa NN Model C1 dan C2

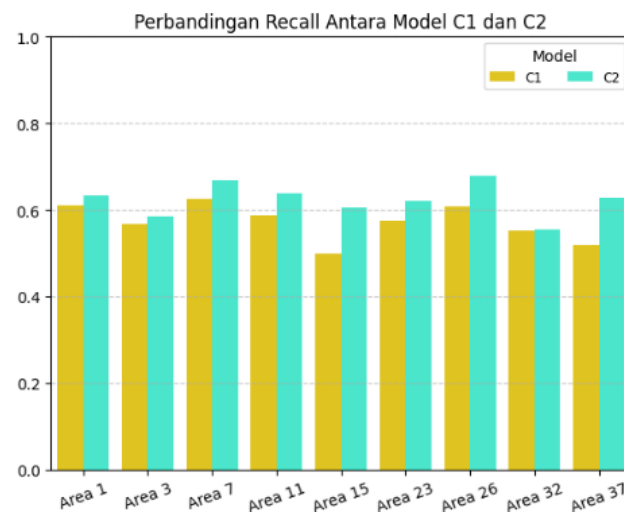
Selanjutnya dilakukan komparasi antara NN model C1 dan C2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 6.4 hingga Gambar 6.8.



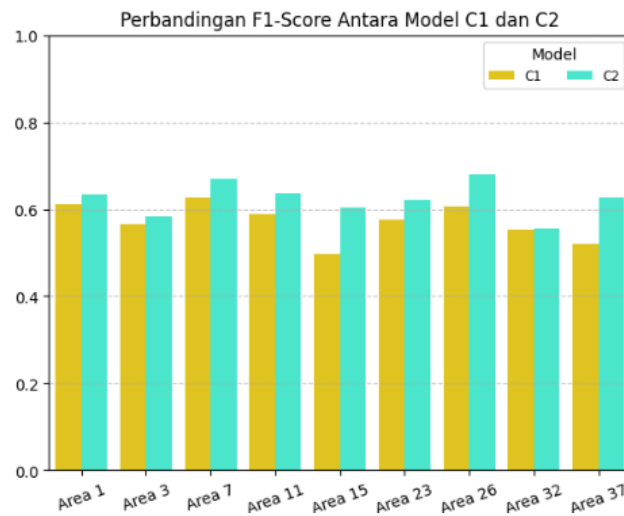
Gambar 6.4. Perbandingan *Accuracy* Antara Model NN C1 dan C2



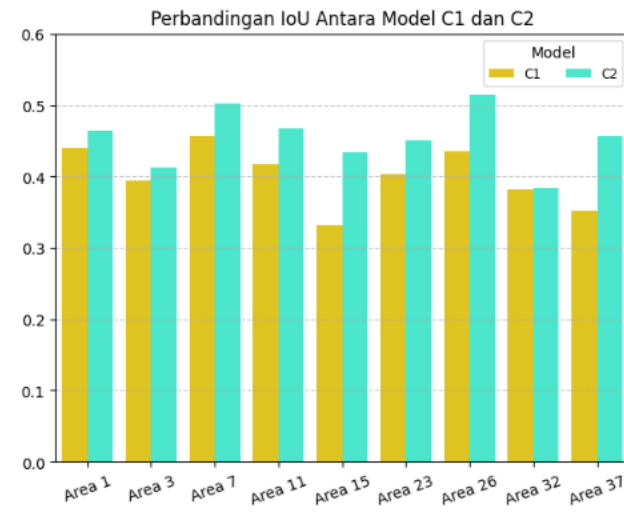
Gambar 6.5. Perbandingan *Precision* Antara Model NN C1 dan C2



Gambar 6.6. Perbandingan *Recall* Antara Model NN C1 dan C2



Gambar 6.7. Perbandingan *F1-Score* Antara Model NN C1 dan C2



Gambar 6.8. Perbandingan IoU Antara Model NN C1 dan C2

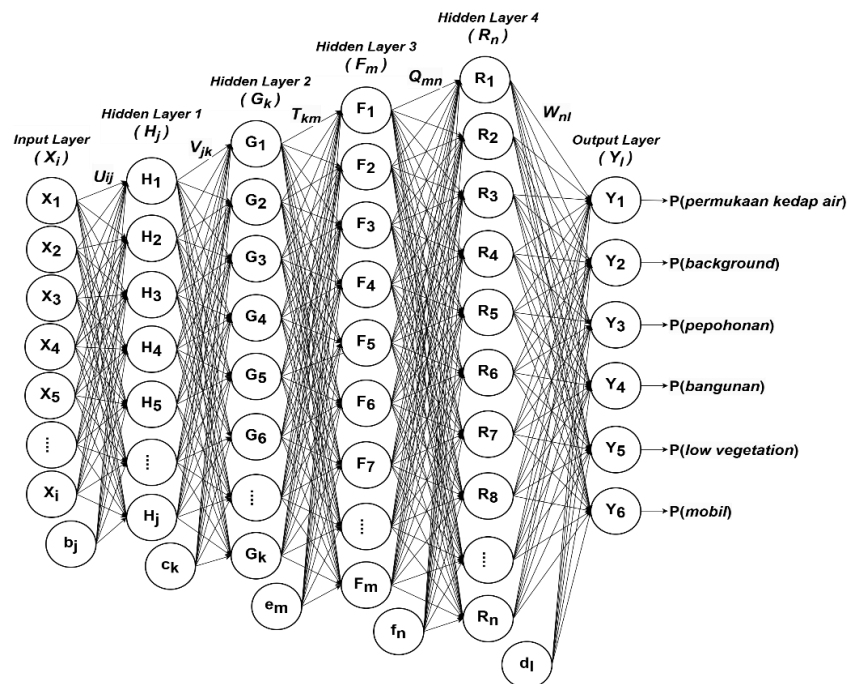
Dari perbandingan yang dilakukan pada Gambar 6.6 hingga Gambar 6.10, ditemukan bahwa model performa model NN C2 yang memanfaatkan 6 fitur GLCM memiliki performa yang lebih tinggi dibandingkan dengan model NN B1 yang memanfaatkan 5 fitur GLCM dari semua metrik, baik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU.

BAB VII

NEURAL NETWORK EMPAT HIDDEN LAYER

7.1 Desain

Arsitektur pada model NN D1 dan D2 menggunakan empat *hidden layer* untuk melakukan klasifikasi pada area tutupan lahan citra beresolusi tinggi. Detail arsitektur dari NN empat *hidden layer* ditunjukkan dalam Gambar 7.1.



Gambar 7.1. Arsitektur Neural Network 4 Hidden Layer

Terjadi proses perhitungan nilai prediksi kelas pada *forward propagation* dan *update* nilai bobot pada *backpropagation*. Lebih detail, berikut adalah proses yang terjadi di dalam *neural network 4 hidden layers*:

1. Inisialisasi data *input*, parameter, dan *hyperparameter* model.
2. Fase *forward propagation*:

- Mengkalikan setiap data *input* pada *input layer* (X_i) dengan bobot (U_{ij}) yang menyambungkan *input layer* dengan *hidden layer* 1 (H_j). Kemudian jumlahkan hasil perkalian pada masing-masing *neuron* di *hidden layer* 1. Kemudian dilanjutkan dengan menambahkan nilai *bias* (b_j) pada setiap hasil penjumlahan pada masing-masing *neuron* di *hidden layer* 1. Dilanjutnya dengan memasukkan nilai masing-masing *neuron* di *hidden layer* 1 hasil penjumlahan *bias* kedalam fungsi aktivasi ReLU. Rumus propagasi maju di *hidden layer* 1 disajikan dalam Persamaan 7.1 dan 7.2.

$$Z_j = \sum_{i=1}^n (X_i U_{ij}) + b_j \quad (7.1)$$

Keterangan:

Z_j = Hasil perkalian antara *input* dengan bobot yang kemudian ditambah bias.

X_i = Data *input* pada *input layer*.

U_{ij} = Bobot yang menyambungkan *input layer* dengan *hidden layer*.

b_j = Nilai bias dari *neuron* pada *hidden layer* 1.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$H_j = \text{ReLU}(Z_j) = \max(0, Z_j) \quad (7.2)$$

Keterangan:

H_j = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 2 (G_k) dengan cara menambahkan bias *hidden layer* 2 (c_k) pada hasil perkalian antara nilai prediksi *hidden layer* 1 dengan nilai bobot (V_{jk}) yang menyambungkan *hidden layer* 1 ke *hidden layer* 2 (Z_k). Rumus propagasi maju di *hidden layer* 2 ditampilkan dalam Persamaan 7.3 dan 7.4.

$$Z_k = \sum_{j=1}^n (H_j V_{jk}) + c_k \quad (7.3)$$

Keterangan:

Z_k = Hasil perkalian antara *hidden layer* 1 dengan bobot, lalu ditambah bias.

H_j = Nilai prediksi pada *hidden layer* 1.

V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

c_k = Nilai bias dari *neuron* pada *hidden layer* 2.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$G_k = ReLU(Z_k) = \max(0, Z_k) \quad (7.4)$$

Keterangan:

G_k = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 3 (F_m) dengan cara menambahkan bias *hidden layer* 3 (e_m) pada hasil perkalian antara nilai prediksi *hidden layer* 2 dengan nilai bobot (T_{km}) yang menyambungkan *hidden layer* 2 ke *hidden layer* 3 (Z_m). Rumus propagasi maju di *hidden layer* 3 ditampilkan dalam Persamaan 7.5 dan 7.6.

$$Z_m = \sum_{k=1}^n (G_k T_{km}) + e_m \quad (7.5)$$

Keterangan:

Z_m = Hasil perkalian antara *hidden layer* 2 dengan bobot, lalu ditambah bias.

G_k = Nilai prediksi pada *hidden layer* 2.

T_{km} = Bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3.

e_m = Nilai bias dari *neuron* pada *hidden layer* 3.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$F_m = ReLU(Z_m) = \max(0, Z_m) \quad (7.6)$$

Keterangan:

F_m = Nilai pada *hidden layer* 3 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 4 (R_n) dengan cara menambahkan bias *hidden layer* 4 (f_n) pada hasil perkalian antara nilai

prediksi *hidden layer* 3 dengan nilai bobot (Q_{mn}) yang menyambungkan *hidden layer* 3 ke *hidden layer* 4 (Z_n). Rumus propagasi maju di *hidden layer* 4 ditampilkan dalam Persamaan 7.7 dan 7.8.

$$Z_n = \sum_{m=1}^n (F_m Q_{mn}) + f_n \quad (7.7)$$

Keterangan:

Z_n = Hasil perkalian antara *hidden layer* 3 dengan bobot, lalu ditambah bias.

F_m = Nilai prediksi pada *hidden layer* 3.

Q_{mn} = Bobot yang menyambungkan *hidden layer* 3 dengan *hidden layer* 4.

f_n = Nilai bias dari *neuron* pada *hidden layer* 4.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$R_n = \text{ReLU}(Z_n) = \max(0, Z_n) \quad (7.8)$$

Keterangan:

R_n = Nilai pada *hidden layer* 4 setelah diaktivasi menggunakan fungsi ReLU.

- Selanjutnya adalah menghitung nilai prediksi pada *output layer* (Y_l). Cara perhitungannya sama seperti pada perhitungan pada *hidden layer* 3 (F_m), yang membedakan adalah bobot (W_{nl}) yang menyambungkan *hidden layer* 3 ke *output layer*, dan bias (d_l) pada *output layer*. Rumus propagasi maju di *output layer* disajikan dalam Persamaan 7.9.

$$Z_l = \sum_{n=1}^n (R_n W_{nl}) + d_l \quad (7.9)$$

Keterangan:

Z_l = Hasil perkalian antara R_n dengan W_{nl} lalu ditambah dengan bias d_l .

R_n = Nilai hasil perhitungan pada *hidden layer* 4.

W_{nl} = Bobot yang menyambungkan *hidden layer* 4 dengan *output layer*.

d_l = Nilai bias dari *neuron* pada *output layer*.

- Langkah berikutnya adalah menghitung nilai probabilitas dari setiap kelas menggunakan fungsi *softmax*. Total dari nilai semua probabilitas setiap kelas adalah 1, kelas dengan nilai probabilitas tertinggi merupakan

keluaran hasil prediksi model ($P(Y_l)$). Rumus perhitungan *softmax* disajikan dalam Persamaan 7.10.

$$P(Y_l) = \frac{\exp(Z_l)}{\sum_{m=1}^6 \exp(Z_m)} \quad (7.10)$$

Keterangan:

$P(Y_l)$ = Nilai probabilitas kelas ke- l yang terprediksi benar.

$\exp(Z_l)$ = Nilai eksponensial dari kelas l di *output layer*.

$\sum_{m=1}^6 \exp(Z_m)$ = Jumlah dari semua nilai eksponensial dari setiap kelas.

- Hitung nilai *loss* untuk mengetahui seberapa jauh jarak prediksi model terhadap kelas asli atau *ground truth*. Nilai *loss* yang dihasilkan dapat memberikan gambaran konvergensi terhadap proses *training* model. Pada penelitian ini *loss function* yang digunakan adalah *sparse categorical cross-entropy*. Rumusnya disajikan dalam Persamaan 7.11.

$$L = -\frac{1}{n} \sum_{i=1}^n \log(P(Y_{i,true})) \quad (7.11)$$

Keterangan:

L = Nilai *loss* hasil perhitungan *loss function*.

$P(Y_{i,true})$ = Nilai probabilitas kelas yang diprediksi oleh model.

n = Jumlah data atau sampel dalam dataset.

7. Fase *backward propagation*:

- Menghitung nilai *error* pada *output layer* dengan turunan dari *loss function* ($\frac{\delta L}{\delta Z_l}$) dengan cara menghitung jarak nilai atau perbedaan antara nilai probabilitas kelas terprediksi dengan kelas *ground truth*. Rumus perhitungan nilai *error* pada *output layer* terdapat pada Persamaan 7.12.

$$\frac{\delta L}{\delta Z_l} = P(Y_l) - y_l \quad (7.12)$$

Keterangan:

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* pada *output layer*.

$P(Y_l)$ = Nilai probabilitas kelas k yang terprediksi benar.

y_l = Kelas asli atau *ground truth*.

- Setelah mendapatkan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$), selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer* 4 dan *output layer* ($\frac{\delta L}{\delta W_{nl}}$). Rumus perhitungannya ditampilkan pada Persamaan 7.13.

$$\frac{\delta L}{\delta W_{nl}} = \frac{\delta L}{\delta Z_l} \cdot R_n \quad (7.13)$$

Keterangan:

R_n = Output atau hasil fungsi aktivasi dari *hidden layer* 4.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer*.

$\frac{\delta L}{\delta W_{nl}}$ = Nilai *error* pada bobot W_{nl} .

- Kemudian untuk nilai *error* dari bias pada *output layer* ($\frac{\delta L}{\delta d_l}$) memiliki nilai yang sama dengan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 4. Rumus perhitungan dari proses ini disajikan dalam Persamaan 7.14.

$$\frac{\delta L}{\delta Z_n} = \sum_{m=1}^6 \left(\frac{\delta L}{\delta Z_l} \cdot W_{nl} \right) \cdot ReLU'(Z_n) \quad (7.14)$$

Keterangan:

$\frac{\delta L}{\delta Z_n}$ = Nilai *error* dari *hidden layer* 4 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer* sebelum diaktivasi.

W_{nl} = Bobot yang menyambungkan *hidden layer* 4 dengan *output layer*.

$ReLU'(Z_n)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 3 dengan *hidden layer* 4 ($\frac{\delta L}{\delta Q_{mn}}$). Rumus dari proses ini disajikan dalam Persamaan 7.15.

$$\frac{\delta L}{\delta Q_{mn}} = \frac{\delta L}{\delta Z_n} \cdot F_m \quad (7.15)$$

Keterangan:

$$\begin{aligned}\frac{\delta L}{\delta Q_{mn}} &= \text{Nilai error pada bobot } Q_{mn}. \\ \frac{\delta L}{\delta Z_n} &= \text{Nilai error dari hidden layer 3 sebelum diaktivasi.} \\ F_m &= \text{Output atau hasil fungsi aktivasi dari hidden layer 3.}\end{aligned}$$

- Kemudian untuk nilai error dari bias hidden layer 2 ($\frac{\delta L}{\delta C_k}$) memiliki nilai yang sama dengan nilai error hidden layer 2 sebelum diaktivasi ($\frac{\delta L}{\delta Z_k}$).
- Langkah selanjutnya adalah mendapatkan nilai error pada hidden layer 1. Rumus perhitungan dari proses ini disajikan dalam Persamaan 7.16.

$$\frac{\delta L}{\delta Z_j} = \sum_{k=1}^n \frac{\delta L}{\delta Z_k} \cdot V_{jk} \cdot \text{ReLU}'(Z_j) \quad (7.16)$$

Keterangan:

$$\begin{aligned}\frac{\delta L}{\delta Z_j} &= \text{Nilai error dari hidden layer 1 sebelum diaktivasi.} \\ \frac{\delta L}{\delta Z_k} &= \text{Nilai error dari hidden layer 2 sebelum diaktivasi.} \\ V_{jk} &= \text{Bobot yang menyambungkan hidden layer 1 dengan hidden layer 2.} \\ \text{ReLU}'(Z_j) &= \text{Turunan dari fungsi aktivasi ReLU.}\end{aligned}$$

- Proses berikutnya adalah menghitung nilai error pada bobot yang menyambungkan hidden layer 1 dengan hidden layer 2 ($\frac{\delta L}{\delta V_{jk}}$). Rumus dari proses ini disajikan dalam Persamaan 7.17.

$$\frac{\delta L}{\delta U_{ij}} = \frac{\delta L}{\delta Z_j} \cdot X_i \quad (7.17)$$

Keterangan:

$$\begin{aligned}\frac{\delta L}{\delta U_{ij}} &= \text{Nilai error pada bobot } U_{ij}. \\ \frac{\delta L}{\delta Z_j} &= \text{Nilai error dari hidden layer 1 sebelum diaktivasi.} \\ X_i &= \text{Data input.}\end{aligned}$$

- Kemudian untuk nilai error dari bias hidden layer 1 ($\frac{\delta L}{\delta b_j}$) memiliki nilai yang sama dengan nilai error hidden layer 1 sebelum diaktivasi ($\frac{\delta L}{\delta Z_j}$).

8. Melakukan *update* terhadap nilai bobot yang menyambungkan *input layer* dengan *hidden layer 1*, *hidden layer 1* dengan *hidden layer 2*, dan *hidden layer 2* ke *hidden layer 3*, *hidden layer 3* ke *hidden layer 4*, dan *hidden layer 4* ke *output layer*. Secara berurutan rumus untuk proses ini ditampilkan dalam Persamaan 7.18 hingga 7.27.

- *Update* nilai bobot pada *hidden layer 1* (U_{ij})

$$U_{ij} = U_{ij} - \eta \cdot \frac{\delta L}{\delta U_{ij}} \quad (7.18)$$

- *Update* nilai bias pada *hidden layer 1* (b_j)

$$b_j = b_j - \eta \cdot \frac{\delta L}{\delta b_j} \quad (7.19)$$

- *Update* nilai bobot pada *hidden layer 2* (V_{jk})

$$V_{jk} = V_{jk} - \eta \cdot \frac{\delta L}{\delta V_{jk}} \quad (7.20)$$

- *Update* nilai bias pada *hidden layer 2* (c_k)

$$c_k = c_k - \eta \cdot \frac{\delta L}{\delta c_k} \quad (7.21)$$

- *Update* nilai bobot pada *hidden layer 3* (T_{km})

$$T_{km} = T_{km} - \eta \cdot \frac{\delta L}{\delta T_{km}} \quad (7.22)$$

- *Update* nilai bias pada *hidden layer 3* (e_m)

$$e_m = e_m - \eta \cdot \frac{\delta L}{\delta e_m} \quad (7.23)$$

- *Update* nilai bobot pada *hidden layer 4* (Q_{mn})

$$Q_{mn} = Q_{mn} - \eta \cdot \frac{\delta L}{\delta Q_{mn}} \quad (7.24)$$

- *Update* nilai bias pada *hidden layer 4* (f_n)

$$f_n = f_n - \eta \cdot \frac{\delta L}{\delta f_n} \quad (7.25)$$

- Update nilai bobot pada *output layer* (W_{nl})

$$W_{nl} = W_{nl} - \eta \cdot \frac{\delta L}{\delta W_{nl}} \quad (7.26)$$

- Update nilai bias pada *output layer* (d_l)

$$d_l = d_l - \eta \cdot \frac{\delta L}{\delta d_l} \quad (7.27)$$

Keterangan:

η = Nilai parameter *learning rate*.

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta b_j}$ = Nilai *error* bias pada *hidden layer* 1.

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta c_k}$ = Nilai *error* bias pada *hidden layer* 2.

$\frac{\delta L}{\delta T_{km}}$ = Nilai *error* pada bobot T_{km} .

$\frac{\delta L}{\delta e_m}$ = Nilai *error* bias pada *hidden layer* 3.

$\frac{\delta L}{\delta W_{mn}}$ = Nilai *error* pada bobot W_{mn} .

$\frac{\delta L}{\delta f_n}$ = Nilai *error* bias pada *hidden layer* 4.

$\frac{\delta L}{\delta W_{nl}}$ = Nilai *error* pada bobot W_{nl} .

$\frac{\delta L}{\delta d_l}$ = Nilai *error* bias pada *output layer*.

7.2 Implementasi

Proses *training* model NN empat *hidden layer* memanfaatkan beberapa parameter, model NN D1 dan NN D2 memiliki parameter yang berbeda. Lebih detail dapat dilihat dalam Tabel 7.1.

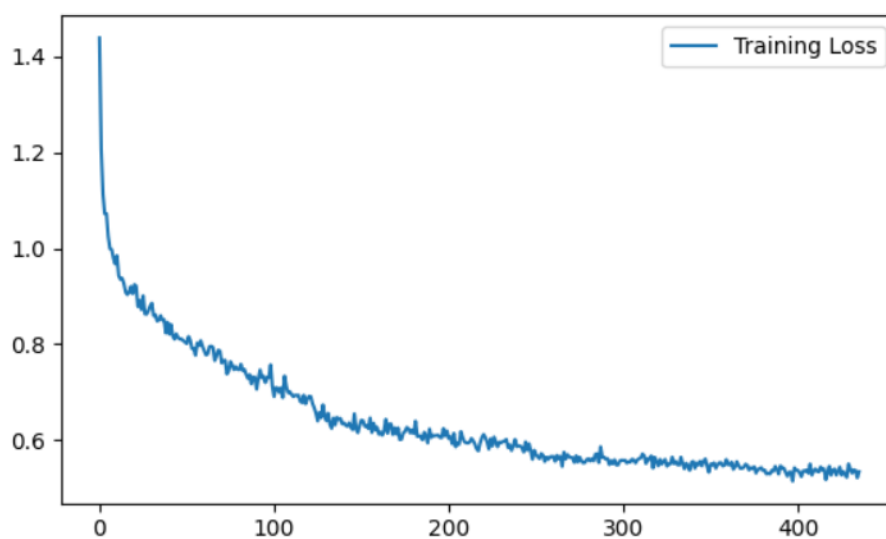
Tabel 7.1. Detail *Hyperparameter* model NN D1 dan NN D2

Hyperparameter	Nilai	
Tipe jaringan	D1	D2
Fungsi aktivasi	ReLU	ReLU
Optimizer	SGD	SGD
Input Layer	15	18
Hidden layer 1	45	54
Hidden layer 2	135	162
Hidden layer 3	405	486
Hidden layer 4	1215	1458

Hyperparameter	Nilai	
<i>Output layer</i>	6	6
<i>Learning rate</i>	0.001	0.001

7.3 Uji Coba NN Model D1

Pada tahap uji coba, *training* model NN D1 menggunakan 5 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-436 dengan nilai *loss* sebesar 0.5347. Gambar 7.2 menunjukkan grafik nilai *loss* dari proses *training* NN empat *hidden layer*.

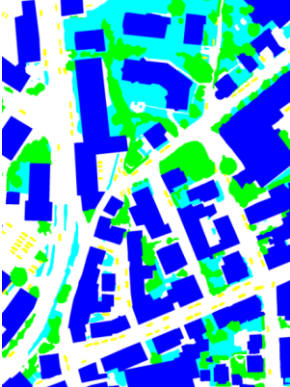
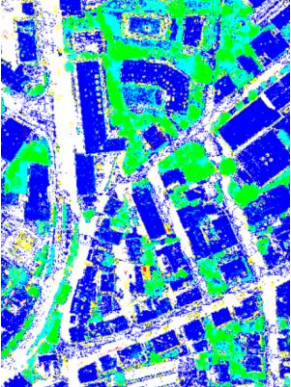
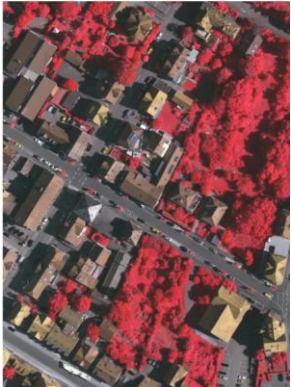
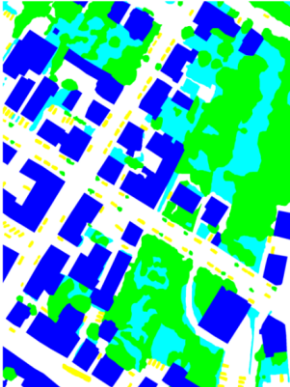
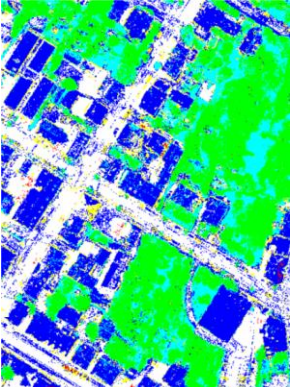
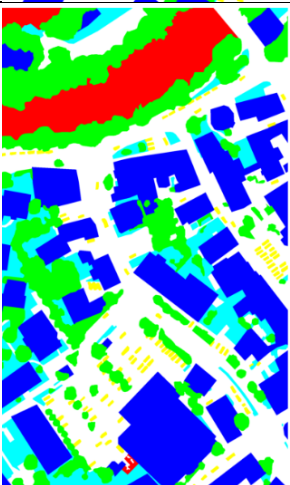
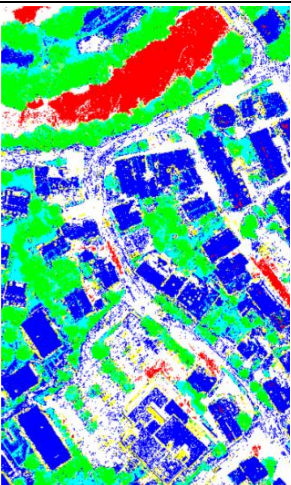


Gambar 7.2. Grafik Loss Training NN Model D1

Pada Gambar 7.2 menunjukkan nilai *loss* pada proses *training* model D1 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Didapatkan nilai *accuracy* sebesar 75.12%, *precision* sebesar 75.67%, *recall* sebesar 75.12%, dan F1-Score sebesar 75.08%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi

tinggi. Hasil pengujian model D1 dalam melakukan klasifikasi area tutupan lahan disajikan dalam Tabel 7.2.

Tabel 7.2. Beberapa Hasil Klasifikasi Model NN D1 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN D1 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Dalam Tabel 7.2, terlihat salah klasifikasi pada beberapa

tutupan lahan oleh model NN D1. Lebih detail, untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan secara detail disajikan dalam Tabel 7.3.

Tabel 7.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN D1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1331895	2533881	543265	520870
	Permukaan kedap air	1109935	2780492	396256	643228
	<i>Low vegetation</i>	319378	3935369	373007	302157
	Pepohonan	395910	4169998	133790	230213
	<i>Background</i>	0	4925214	4695	2
	Mobil	27286	4559094	294494	49037
Area 3	Bangunan	1110672	3247337	1152445	521588
	Permukaan kedap air	916983	3631392	341282	1142385
	<i>Low vegetation</i>	537209	4191869	683016	619948
	Pepohonan	684747	4655030	249618	442647
	<i>Background</i>	0	6026932	5110	0
	Mobil	23579	5648798	327381	32284
Area 7	Bangunan	907623	3103795	416172	397469
	Permukaan kedap air	970860	3037611	295054	521534
	<i>Low vegetation</i>	289930	3784044	403785	347300
	Pepohonan	988768	3278444	263912	293935
	<i>Background</i>	0	4812064	12995	0
	Mobil	35718	4477177	240242	71922
Area 11	Bangunan	535285	3256993	696048	369112
	Permukaan kedap air	771000	3064751	183955	837732
	<i>Low vegetation</i>	449082	3293633	755653	359070
	Pepohonan	998808	3217387	171022	470221
	<i>Background</i>	0	4850558	6880	0
	Mobil	28563	4529168	261142	38565
Area 15	Bangunan	713816	3325001	522064	361354
	Permukaan kedap air	329040	3856968	210155	526072
	<i>Low vegetation</i>	725282	2573990	1038348	584615
	Pepohonan	740552	3038983	240063	902637
	<i>Background</i>	1525	4910077	4975	5658
	Mobil	16063	4510199	380352	15621
Area 23	Bangunan	588441	3556952	402802	296843
	Permukaan kedap air	280426	3918282	125104	521226
	<i>Low vegetation</i>	838091	2606971	892339	507637
	Pepohonan	1117822	2803019	252848	671349
	<i>Background</i>	0	4835632	1650	7756
	Mobil	7979	4492055	337536	7468
Area 26	Bangunan	1029138	3331904	449717	529326
	Permukaan kedap air	1046957	3188784	500638	603706
	<i>Low vegetation</i>	260045	4423804	414325	241911
	Pepohonan	803725	4101455	153005	281900
	<i>Background</i>	303492	4840473	52478	143642
	Mobil	30618	4947895	295947	65625
Area 32	Bangunan	568389	2689350	1098636	702525
	Permukaan kedap air	1455100	1825562	616650	1161588

Data	Kelas	TP	TN	FP	FN
Area 32	<i>Low vegetation</i>	216430	4327559	368745	146166
	Pepohonan	232485	4427861	34765	363789
	<i>Background</i>	125	4935898	67400	55477
	Mobil	55736	4557635	344439	101090
Area 37	Bangunan	682717	2270318	789123	239862
	Permukaan kedap air	558188	2819087	241817	362928
	<i>Low vegetation</i>	760943	2136801	290672	793604
	Pepohonan	243116	3288927	127959	322018
	<i>Background</i>	0	3976070	5950	0
	Mobil	5162	3687003	276373	13482

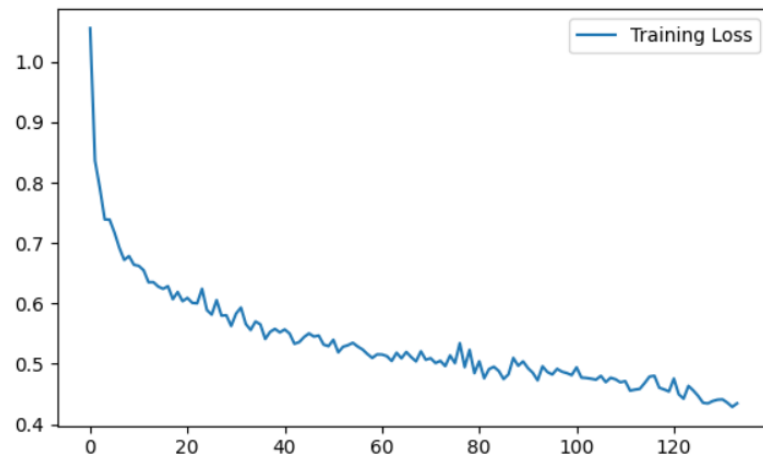
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN D1 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN D1 disajikan dalam Tabel 7.4.

Tabel 7.4. Performa Hasil Pengujian Model NN D1

Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	88.27	64.78	65.06	64.92	48.04
Area 3	87.16	55.72	55.62	55.67	38.58
Area 7	87.01	65.40	60.68	62.95	45.93
Area 11	85.84	57.35	57.64	57.49	40.34
Area 15	87.03	53.99	51.21	52.56	35.66
Area 23	87.68	59.18	65.91	62.38	45.31
Area 26	88.31	65.17	65.21	65.19	48.35
Area 32	83.24	53.25	49.43	51.27	34.48
Area 37	85.92	63.48	63.33	63.40	46.41

7.4 Uji Coba NN Model D2

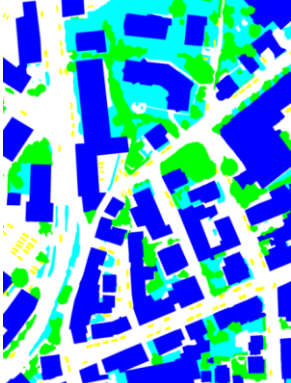
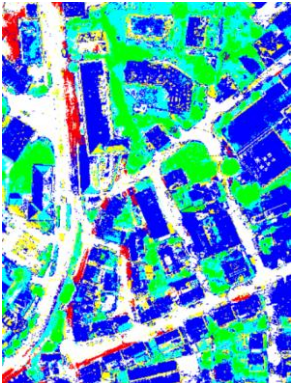
Pada tahap uji coba, *training* model NN D2 menggunakan 6 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-134 dengan nilai *loss* sebesar 0.4346. Gambar 7.3 menunjukkan grafik nilai *loss* dari proses *training* NN empat *hidden layer*.

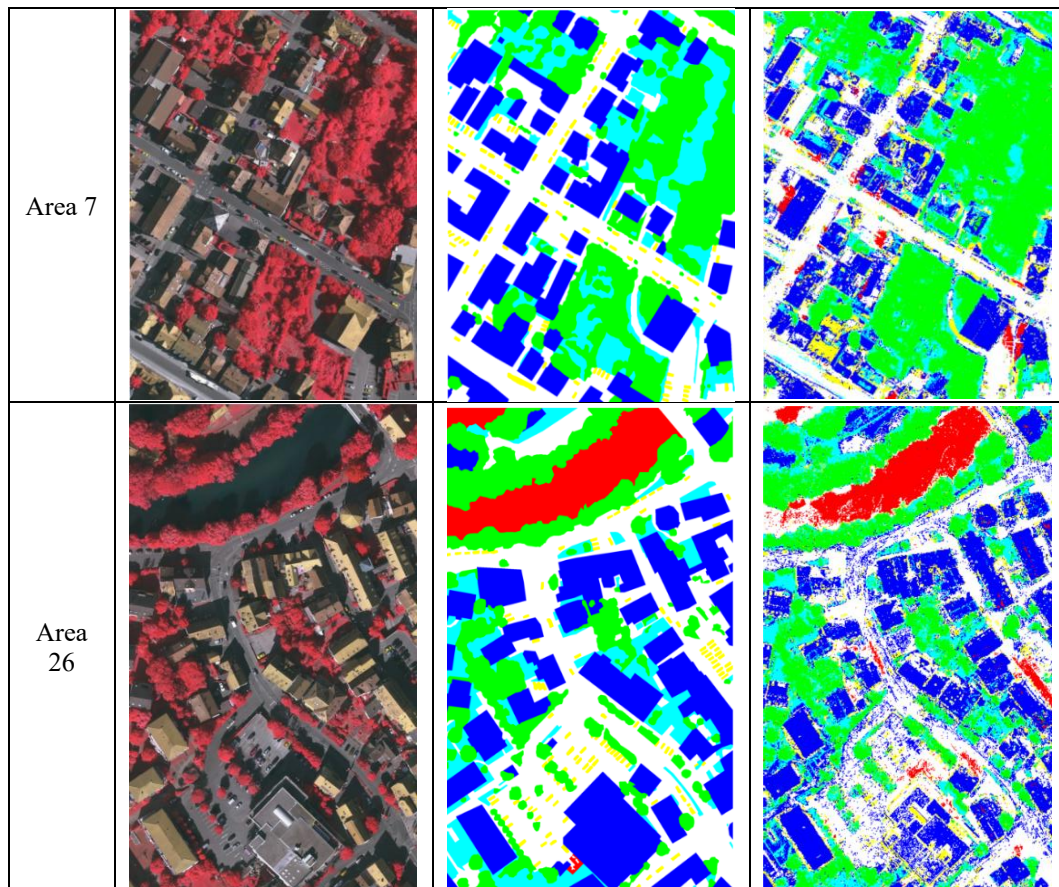


Gambar 7.3. Grafik Loss Training NN Model D2

Pada Gambar 7.3 menunjukkan nilai *loss* pada proses *training* model D2 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Didapatkan nilai *accuracy* sebesar 84.30%, *precision* sebesar 84.48%, *recall* sebesar 84.30%, dan F1-Score sebesar 84.29%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model D2 dalam melakukan klasifikasi area tutupan lahan lebih detail disajikan dalam ditampilkan dalam Tabel 7.5.

Tabel 7.5. Beberapa Hasil Klasifikasi Model NN D2 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			



Hasil klasifikasi model NN D2 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Dalam Tabel 7.5, terlihat model D2 dapat melakukan pada jenis tutupan lahan permukaan kedap air dengan lebih baik dibandingkan model D1. Lebih detail, untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 7.6.

Tabel 7.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN D1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1259482	2643293	433853	593283
	Permukaan kedap air	972341	2849648	327100	780822
	<i>Low vegetation</i>	361164	3838840	469536	260371
	Pepohonan	436275	4152838	150950	189848
	<i>Background</i>	0	4800469	129440	2
	Mobil	33286	4497104	356484	43037
Area 3	Bangunan	893026	3621523	778259	739234

Data	Kelas	TP	TN	FP	FN
Area 3	Permukaan kedap air	960360	3750172	222502	1099008
	<i>Low vegetation</i>	767238	3785253	1089632	389919
	Pepohonan	713405	4664943	239705	413989
	<i>Background</i>	0	6001642	30400	0
	Mobil	24821	5663485	312694	31042
Area 7	Bangunan	799255	3223012	296955	505837
	Permukaan kedap air	928258	3089729	242936	564136
	<i>Low vegetation</i>	244280	3797634	390195	392950
	Pepohonan	1135012	3190793	351563	147691
	<i>Background</i>	0	4759204	65855	0
Area 11	Mobil	45244	4391913	325506	62396
	Bangunan	500084	3529812	423229	404313
	Permukaan kedap air	888763	3089999	158707	719969
	<i>Low vegetation</i>	413663	3370824	678462	394489
	Pepohonan	1186959	3069278	319131	282070
Area 15	<i>Background</i>	0	4829138	28300	0
	Mobil	31471	4561641	228669	35657
	Bangunan	614242	3557937	289128	460928
	Permukaan kedap air	356909	3847062	220061	498203
	<i>Low vegetation</i>	525279	3059247	553091	784618
Area 23	Pepohonan	1312160	2647461	631585	331029
	<i>Background</i>	31	4873318	41734	7152
	Mobil	17443	4529979	360572	14241
	Bangunan	482628	3784954	174800	402656
	Permukaan kedap air	273731	3892567	150819	527921
Area 26	<i>Low vegetation</i>	771124	2760349	738961	574604
	Pepohonan	1419572	2535309	520558	369599
	<i>Background</i>	25	4821832	15450	7731
	Mobil	9956	4542177	287414	5491
	Bangunan	949403	3542789	238832	609061
Area 32	Permukaan kedap air	1002763	3296560	392862	647900
	<i>Low vegetation</i>	329860	4384974	453155	172096
	Pepohonan	878922	4122312	132148	206703
	<i>Background</i>	371667	4725623	167328	75467
	Mobil	37163	4857860	385982	59080
Area 37	Bangunan	444912	2865798	922188	826002
	Permukaan kedap air	1483329	2160391	281821	1133359
	<i>Low vegetation</i>	192531	4185435	510869	170065
	Pepohonan	339358	4371684	90942	256916
	<i>Background</i>	537	4831260	172038	55065
Area 37	Mobil	48676	4330375	571699	108150
	Bangunan	570804	2517445	541996	351775
	Permukaan kedap air	561062	2849791	211113	360054
	<i>Low vegetation</i>	996071	2001704	425769	558476
	Pepohonan	275477	3286613	130273	289657
Area 37	<i>Background</i>	0	3969795	12225	0
	Mobil	4266	3710412	252964	14378

Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik

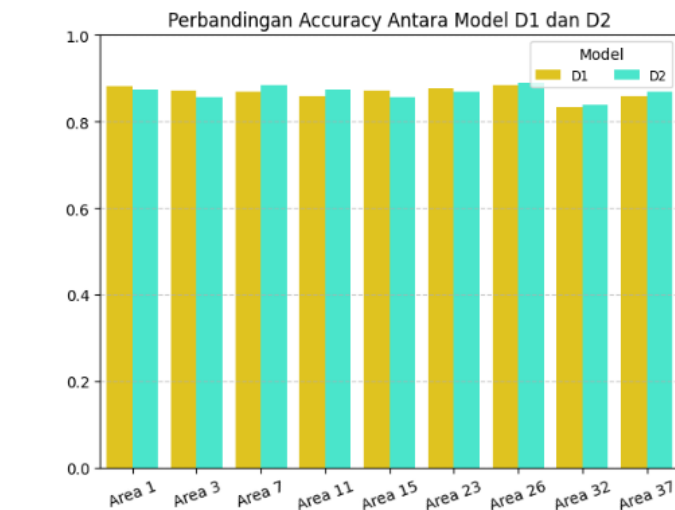
evaluasi yang digunakan untuk mengukur performa model NN D2 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN D2 disajikan dalam Tabel 7.7.

Tabel 7.7. Performa Hasil Pengujian Model NN D2

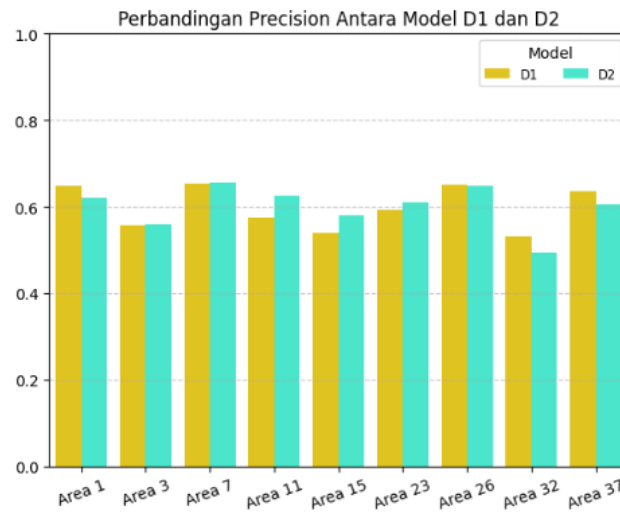
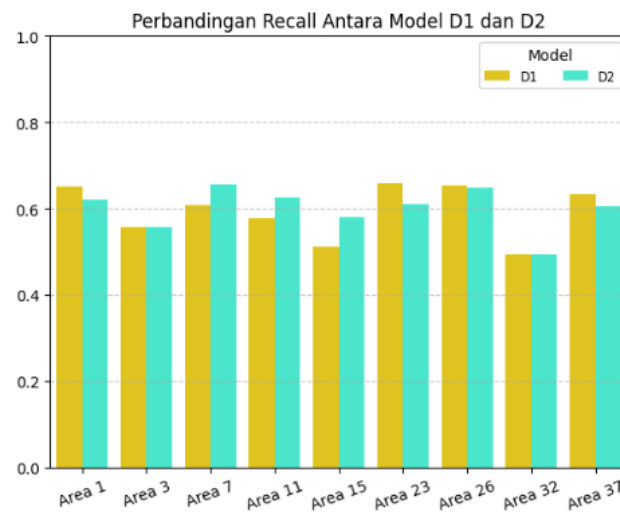
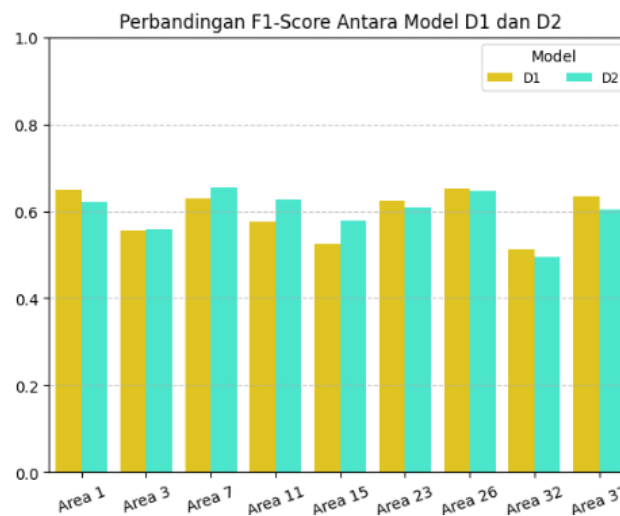
Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	87.45	62.08	62.10	62.09	45.05
Area 3	85.63	55.91	55.73	55.82	38.71
Area 7	88.54	65.51	65.48	65.50	48.69
Area 11	87.38	62.59	62.50	62.55	45.50
Area 15	85.71	57.87	57.93	57.90	40.74
Area 23	86.88	61.01	61.01	61.01	43.91
Area 26	88.81	64.72	64.72	64.72	47.88
Area 32	83.73	49.45	49.45	49.45	32.87
Area 37	86.95	60.48	60.48	60.48	43.35

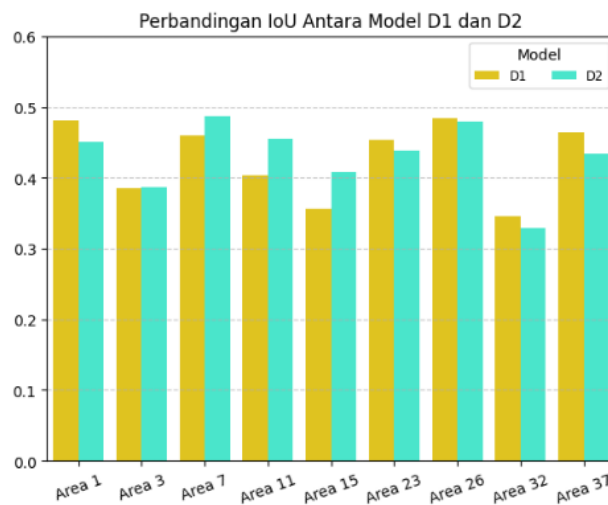
7.5 Komparasi Performa NN Model D1 dan D2

Selanjutnya dilakukan komparasi antara NN model D1 dan D2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 7.4 hingga Gambar 7.8.



Gambar 7.4. Perbandingan *Accuracy* Antara Model NN D1 dan D2

Gambar 7.5. Perbandingan *Precision* Antara Model NN D1 dan D2Gambar 7.6. Perbandingan *Recall* Antara Model NN D1 dan D2Gambar 7.7. Perbandingan *F1-Score* Antara Model NN D1 dan D2



Gambar 7.8. Perbandingan IoU Antara Model NN D1 dan D2

Berbeda dengan perbandingan pada beberapa bab sebelumnya yang mana model dengan 6 fitur GLCM selalu memiliki performa yang lebih tinggi dibandingkan dengan model dengan 5 fitur GLCM, pada perbandingan NN empat *hidden layer* dengan kode model D1 yang menggunakan 5 fitur GLCM dan D2 yang menggunakan 6 fitur GLCM menunjukkan hasil yang berbeda.

Dari segi metrik *accuracy*, meskipun pada beberapa Area model D2 memiliki akurasi yang lebih rendah dibanding model D1, namun secara keseluruhan model D2 memiliki keunggulan dibandingkan dengan model D1. Dari perbandingan yang dilakukan, model D2 lebih unggul dalam 5 Area citra beresolusi tinggi, yakni pada Area 7, Area 11, Area 26, Area 32, dan Area 37.

Lalu berikutnya dari segi metrik *precision*, meskipun pada beberapa Area model D2 memiliki akurasi yang lebih rendah dibanding model D1, namun secara keseluruhan model D2 memiliki keunggulan dibandingkan dengan model D1. Dari perbandingan yang dilakukan, model D2 unggul dalam 5 Area dibandingkan dengan model D1, yakni pada Area 3, Area 7, Area 11, Area 15, dan Area 23.

Selanjutnya dari metrik *recall*, meskipun secara spesifik nilai metrik *recall* dari model D1 dan D2 pada Area 3 dan Area 32 memiliki selisih yang sangat kecil, namun model D2 memiliki performa yang lebih tinggi dibandingkan dengan model D1. Sehingga secara keseluruhan D2 juga lebih unggul dalam 5 Area, yakni pada Area 3, Area 7, Area 11, Area 15, dan Area 32.

Berikutnya pada metrik *F1-Score* model D2 memiliki performa yang lebih rendah dibandingkan dengan model D1. Dari perbandingan yang dilakukan model D1 menunjukkan nilai *F1-Score* yang lebih tinggi juga dalam 5 Area, yakni pada Area 1, Area 23, Area 26, Area 32, dan Area 37.

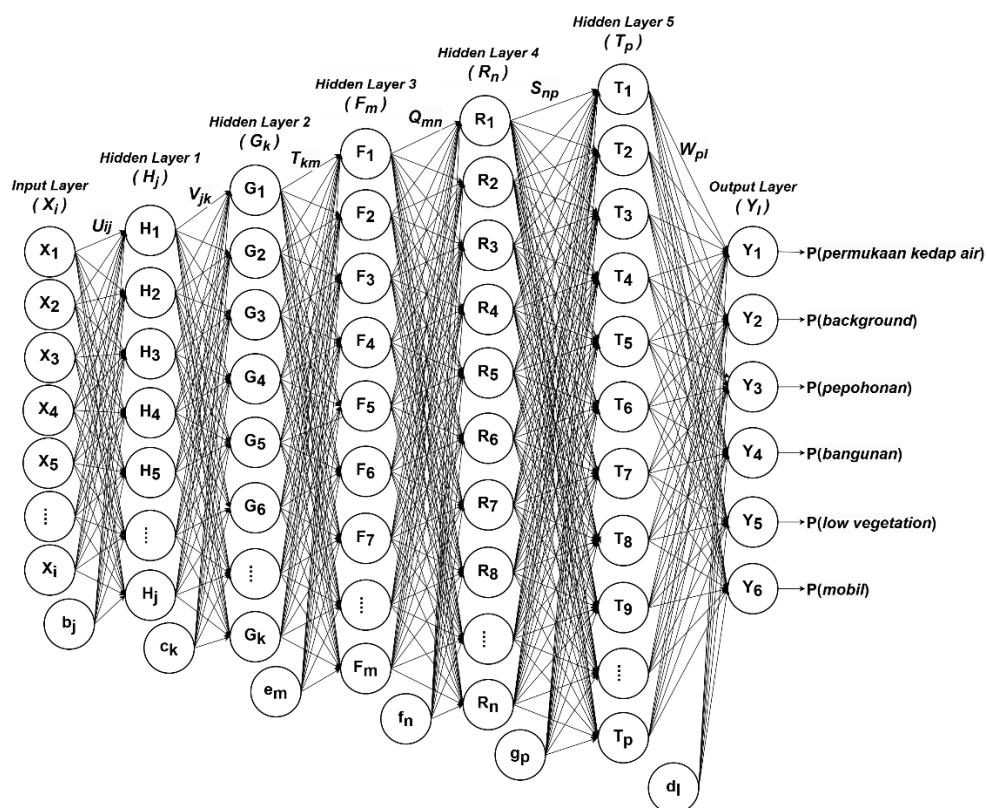
Terakhir adalah perbandingan pada metrik IoU. Dari perbandingan yang dilakukan, didapatkan hasil bahwa model D1 juga memiliki nilai yang lebih unggul dibandingkan dengan model D2. Secara keseluruhan model D1 memiliki nilai IoU yang lebih unggul dari model D2 pada Area 1, Area 23, Area 26, Area 32, dan Area 37.

BAB VIII

NEURAL NETWORK LIMA HIDDEN LAYER

8.1 Desain

Arsitektur pada model NN E1 dan E2 menggunakan lima *hidden layer* untuk melakukan klasifikasi pada area tutupan lahan citra beresolusi tinggi. Detail arsitektur dari NN lima *hidden layer* terdapat pada Gambar 8.1.



Gambar 8.1. Arsitektur *Neural Network 5 Hidden Layer*

Terjadi proses perhitungan nilai prediksi kelas pada *forward propagation* dan *update* nilai bobot pada *backpropagation*. Lebih detail, berikut adalah proses yang terjadi di dalam *neural network 5 hidden layers*:

1. Inisialisasi data *input*, parameter, dan *hyperparameter* model.

2. Fase *forward propagation*:

- Mengkalikan setiap data *input* pada *input layer* (X_i) dengan bobot (U_{ij}) yang menyambungkan *input layer* dengan *hidden layer* 1 (H_j). Kemudian jumlahkan hasil perkalian pada masing-masing *neuron* di *hidden layer* 1. Kemudian dilanjutkan dengan menambahkan nilai *bias* (b_j) pada setiap hasil penjumlahan pada masing-masing *neuron* di *hidden layer* 1. Dilanjutnya dengan memasukkan nilai masing-masing *neuron* di *hidden layer* 1 hasil penjumlahan *bias* kedalam fungsi aktivasi ReLU. Rumus propagasi maju di *hidden layer* 1 disajikan dalam Persamaan 8.1 dan 8.2.

$$Z_j = \sum_{i=1}^n (X_i U_{ij}) + b_j \quad (8.1)$$

Keterangan:

Z_j = Hasil perkalian antara *input* dengan bobot yang kemudian ditambah bias.

X_i = Data *input* pada *input layer*.

U_{ij} = Bobot yang menyambungkan *input layer* dengan *hidden layer*.

b_j = Nilai bias dari *neuron* pada *hidden layer* 1.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$H_j = \text{ReLU}(Z_j) = \max(0, Z_j) \quad (8.2)$$

Keterangan:

H_j = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 2 (G_k) dengan cara menambahkan bias *hidden layer* 2 (c_k) pada hasil perkalian antara nilai prediksi *hidden layer* 1 dengan nilai bobot (V_{jk}) yang menyambungkan *hidden layer* 1 ke *hidden layer* 2 (Z_k). Rumus propagasi maju di *hidden layer* 2 ditampilkan dalam Persamaan 8.3 dan 8.4.

$$Z_k = \sum_{j=1}^n (H_j V_{jk}) + c_k \quad (8.3)$$

Keterangan:

- Z_k = Hasil perkalian antara *hidden layer* 1 dengan bobot, lalu ditambah bias.
- H_j = Nilai prediksi pada *hidden layer* 1.
- V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.
- c_k = Nilai bias dari *neuron* pada *hidden layer* 2.

Hasil pada Z_j kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$G_k = ReLU(Z_k) = \max(0, Z_k) \quad (8.4)$$

Keterangan:

- G_k = Nilai pada *hidden layer* 2 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 3 (F_m) dengan cara menambahkan bias *hidden layer* 3 (e_m) pada hasil perkalian antara nilai prediksi *hidden layer* 2 dengan nilai bobot (T_{km}) yang menyambungkan *hidden layer* 2 ke *hidden layer* 3 (Z_m). Rumus propagasi maju di *hidden layer* 3 ditampilkan dalam Persamaan 8.5 dan 8.6.

$$Z_m = \sum_{k=1}^n (G_k T_{km}) + e_m \quad (8.5)$$

Keterangan:

- Z_m = Hasil perkalian antara *hidden layer* 2 dengan bobot, lalu ditambah bias.
- G_k = Nilai prediksi pada *hidden layer* 2.
- T_{km} = Bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3.
- e_m = Nilai bias dari *neuron* pada *hidden layer* 3.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$F_m = ReLU(Z_m) = \max(0, Z_m) \quad (8.6)$$

Keterangan:

- F_m = Nilai pada *hidden layer* 3 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 4 (R_n) dengan cara menambahkan bias *hidden layer* 4 (f_n) pada hasil perkalian antara nilai prediksi *hidden layer* 3 dengan nilai bobot (Q_{mn}) yang menyambungkan *hidden layer* 3 ke *hidden layer* 4 (Z_m). Rumus propagasi maju di *hidden layer* 4 ditampilkan dalam Persamaan 8.7 dan 8.8.

$$Z_n = \sum_{m=1}^n (F_m Q_{mn}) + f_n \quad (8.7)$$

Keterangan:

Z_n = Hasil perkalian antara *hidden layer* 3 dengan bobot, lalu ditambah bias.

F_m = Nilai prediksi pada *hidden layer* 3.

Q_{mn} = Bobot yang menyambungkan *hidden layer* 3 dengan *hidden layer* 4.

f_n = Nilai bias dari *neuron* pada *hidden layer* 4.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$R_n = ReLU(Z_n) = \max(0, Z_n) \quad (8.8)$$

Keterangan:

R_n = Nilai pada *hidden layer* 4 setelah diaktivasi menggunakan fungsi ReLU.

- Menghitung nilai prediksi pada *hidden layer* 5 (T_p) dengan cara menambahkan bias *hidden layer* 5 (g_p) pada hasil perkalian antara nilai prediksi *hidden layer* 4 dengan nilai bobot (S_{np}) yang menyambungkan *hidden layer* 4 ke *hidden layer* 5 (Z_p). Rumus propagasi maju di *hidden layer* 4 ditampilkan dalam Persamaan 8.9 dan 8.10.

$$Z_p = \sum_{n=1}^n (R_n S_{np}) + g_p \quad (8.9)$$

Keterangan:

Z_p = Hasil perkalian antara *hidden layer* 4 dengan bobot, lalu ditambah bias.

R_n = Nilai prediksi pada *hidden layer* 4.

S_{np} = Bobot yang menyambungkan *hidden layer* 4 dengan *hidden layer* 5.

g_p = Nilai bias dari *neuron* pada *hidden layer* 5.

Hasil pada Z_m kemudian masuk kedalam fungsi aktivasi ReLU. Berikut persamaannya:

$$T_p = \text{ReLU}(Z_p) = \max(0, Z_p) \quad (8.10)$$

Keterangan:

T_p = Nilai pada *hidden layer* 5 setelah diaktivasi menggunakan fungsi ReLU.

- Selanjutnya adalah menghitung nilai prediksi pada *output layer* (Y_l). Cara perhitungannya sama seperti pada perhitungan pada *hidden layer* 3 (F_m), yang membedakan adalah bobot (W_{ml}) yang menyambungkan *hidden layer* 3 ke *output layer*, dan bias (d_l) pada *output layer*. Rumus propagasi maju di *output layer* disajikan dalam Persamaan 8.11.

$$Z_l = \sum_{n=1}^n (T_p W_{pl}) + d_l \quad (8.11)$$

Keterangan:

Z_l = Hasil perkalian antara T_p dengan W_{pl} lalu ditambah dengan bias d_l .

T_p = Nilai hasil perhitungan pada *hidden layer* 5.

W_{pl} = Bobot yang menyambungkan *hidden layer* 5 dengan *output layer*.

d_l = Nilai bias dari *neuron* pada *output layer*.

- Langkah berikutnya adalah menghitung nilai probabilitas dari setiap kelas menggunakan fungsi *softmax*. Total dari nilai semua probabilitas setiap kelas adalah 1, kelas dengan nilai probabilitas tertinggi merupakan keluaran hasil prediksi model ($P(Y_l)$). Rumus perhitungan *softmax* disajikan dalam Persamaan 8.12.

$$P(Y_l) = \frac{\exp(Z_l)}{\sum_{m=1}^6 \exp(Z_m)} \quad (8.12)$$

Keterangan:

$P(Y_l)$ = Nilai probabilitas kelas ke- l yang terprediksi benar.

$\exp(Z_l)$ = Nilai eksponensial dari kelas l di *output layer*.

$\sum_{m=1}^6 \exp(Z_m)$ = Jumlah dari semua nilai eksponensial dari setiap kelas.

- Hitung nilai *loss* untuk mengetahui seberapa jauh jarak prediksi model terhadap kelas asli atau *ground truth*. Nilai *loss* yang dihasilkan dapat

memberikan gambaran konvergensi terhadap proses *training* model. Pada penelitian ini *loss function* yang digunakan adalah *sparse categorical cross-entropy*. Rumusnya disajikan dalam Persamaan 8.13.

$$L = -\frac{1}{n} \sum_{i=1}^n \log(P(Y_{i,true})) \quad (8.13)$$

Keterangan:

L = Nilai *loss* hasil perhitungan *loss function*.

$P(Y_{i,true})$ = Nilai probabilitas kelas yang diprediksi oleh model.

n = Jumlah data atau sampel dalam dataset.

9. Fase *backward propagation*:

- Menghitung nilai *error* pada *output layer* dengan turunan dari *loss function* ($\frac{\delta L}{\delta Z_l}$) dengan cara menghitung jarak nilai atau perbedaan antara nilai probabilitas kelas terprediksi dengan kelas *ground truth*. Rumus perhitungan nilai *error* pada *output layer* terdapat pada Persamaan 8.14.

$$\frac{\delta L}{\delta Z_l} = P(Y_l) - y_l \quad (8.14)$$

Keterangan:

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* pada *output layer*.

$P(Y_l)$ = Nilai probabilitas kelas k yang terprediksi benar.

y_l = Kelas asli atau *ground truth*.

- Setelah mendapatkan nilai *error* pada *output layer* ($\frac{\delta L}{\delta Z_l}$), selanjutnya adalah menghitung nilai *error* dari bobot yang menyambungkan *hidden layer 5* dan *output layer* ($\frac{\delta L}{\delta W_{pl}}$). Rumus perhitungannya ditampilkan pada Persamaan 8.15.

$$\frac{\delta L}{\delta W_{pl}} = \frac{\delta L}{\delta Z_l} \cdot T_p \quad (8.15)$$

Keterangan:

T_p = *Output* atau hasil fungsi aktivasi dari *hidden layer 5*.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer*.

$\frac{\delta L}{\delta W_{pl}}$ = Nilai *error* pada bobot W_{pl} .

- Kemudian untuk nilai *error* dari bias pada *output layer* ($\frac{\delta L}{\delta d_l}$) memiliki nilai yang sama dengan nilai *error* pada *output layer* ($\frac{\delta L}{\delta z_l}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 5.

Rumus perhitungan dari proses ini disajikan dalam Persamaan 8.16.

$$\frac{\delta L}{\delta Z_p} = \sum_{l=1}^6 \frac{\delta L}{\delta Z_l} W_{pl} \cdot ReLU'(Z_p) \quad (8.16)$$

Keterangan:

$\frac{\delta L}{\delta Z_p}$ = Nilai *error* dari *hidden layer* 5 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_l}$ = Nilai *error* dari *output layer* sebelum diaktivasi.

W_{pl} = Bobot yang menyambungkan *hidden layer* 5 dengan *output layer*.

$ReLU'(Z_p)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 4 dengan *hidden layer* 5 ($\frac{\delta L}{\delta S_{np}}$). Rumus dari proses ini disajikan dalam Persamaan 8.17.

$$\frac{\delta L}{\delta S_{np}} = \frac{\delta L}{\delta Z_p} \cdot R_n \quad (8.17)$$

Keterangan:

$\frac{\delta L}{\delta S_{np}}$ = Nilai *error* pada bobot S_{np} .

$\frac{\delta L}{\delta Z_p}$ = Nilai *error* dari *hidden layer* 5 sebelum diaktivasi.

R_n = *Output* atau hasil fungsi aktivasi dari *hidden layer* 4.

- Kemudian untuk nilai *error* dari bias *hidden layer* 5 ($\frac{\delta L}{\delta g_p}$) memiliki nilai yang sama dengan nilai *error* *hidden layer* 5 sebelum diaktivasi ($\frac{\delta L}{\delta Z_p}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 4.

Rumus perhitungan dari proses ini disajikan dalam Persamaan 8.16.

$$\frac{\delta L}{\delta Z_n} = \sum_{p=1}^n \frac{\delta L}{\delta Z_p} S_{np} \cdot ReLU'(Z_n) \quad (8.16)$$

Keterangan:

$\frac{\delta L}{\delta Z_n}$ = Nilai *error* dari *hidden layer* 4 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_p}$ = Nilai *error* dari *hidden layer* 5 sebelum diaktivasi.

S_{np} = Bobot yang menyambungkan *hidden layer* 4 dengan *hidden layer* 5.

$ReLU'(Z_n)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 3 dengan *hidden layer* 4 ($\frac{\delta L}{\delta Q_{mn}}$). Rumus dari proses ini disajikan dalam Persamaan 8.17.

$$\frac{\delta L}{\delta Q_{mn}} = \frac{\delta L}{\delta Z_n} \cdot F_m \quad (8.17)$$

Keterangan:

$\frac{\delta L}{\delta Q_{mn}}$ = Nilai *error* pada bobot Q_{mn} .

$\frac{\delta L}{\delta Z_n}$ = Nilai *error* dari *hidden layer* 4 sebelum diaktivasi.

F_m = *Output* atau hasil fungsi aktivasi dari *hidden layer* 3.

- Kemudian untuk nilai *error* dari bias *hidden layer* 4 ($\frac{\delta L}{\delta F_n}$) memiliki nilai yang sama dengan nilai *error hidden layer* 4 sebelum diaktivasi ($\frac{\delta L}{\delta Z_n}$).
- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 3. Rumus perhitungan dari proses ini disajikan dalam Persamaan 8.18.

$$\frac{\delta L}{\delta Z_m} = \sum_{n=1}^n \frac{\delta L}{\delta Z_n} Q_{mn} \cdot ReLU'(Z_m) \quad (8.18)$$

Keterangan:

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_n}$ = Nilai *error* dari *hidden layer* 4 sebelum diaktivasi.

Q_{mn} = Bobot yang menyambungkan *hidden layer* 3 dengan *hidden layer* 4.

$ReLU'(Z_m)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3 ($\frac{\delta L}{\delta T_{km}}$). Rumus dari proses ini disajikan dalam Persamaan 8.19.

$$\frac{\delta L}{\delta T_{km}} = \frac{\delta L}{\delta Z_m} \cdot G_k \quad (8.19)$$

Keterangan:

$\frac{\delta L}{\delta T_{km}}$ = Nilai *error* pada bobot T_{km} .

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

G_k = *Output* atau hasil fungsi aktivasi dari *hidden layer* 2.

- Kemudian untuk nilai *error* dari bias *hidden layer* 3 ($\frac{\delta L}{\delta e_m}$) memiliki nilai yang sama dengan nilai *error hidden layer* 3 sebelum diaktivasi ($\frac{\delta L}{\delta Z_m}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 2.

Rumus perhitungan dari proses ini disajikan dalam Persamaan 8.20.

$$\frac{\delta L}{\delta Z_k} = \sum_{m=1}^n \frac{\delta L}{\delta Z_m} T_{km} \cdot ReLU'(Z_k) \quad (8.20)$$

Keterangan:

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_m}$ = Nilai *error* dari *hidden layer* 3 sebelum diaktivasi.

T_{km} = Bobot yang menyambungkan *hidden layer* 2 dengan *hidden layer* 3.

$ReLU'(Z_k)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2 ($\frac{\delta L}{\delta V_{jk}}$). Rumus dari proses ini disajikan dalam Persamaan 8.21.

$$\frac{\delta L}{\delta V_{jk}} = \frac{\delta L}{\delta Z_k} \cdot H_j \quad (8.21)$$

Keterangan:

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

H_j = *Output* atau hasil fungsi aktivasi dari *hidden layer* 1.

- Kemudian untuk nilai *error* dari bias *hidden layer* 2 ($\frac{\delta L}{\delta c_k}$) memiliki nilai yang sama dengan nilai *error hidden layer* 2 sebelum diaktivasi ($\frac{\delta L}{\delta Z_k}$).

- Langkah selanjutnya adalah mendapatkan nilai *error* pada *hidden layer* 1.

Rumus perhitungan dari proses ini disajikan dalam Persamaan 8.22.

$$\frac{\delta L}{\delta Z_j} = \sum_{k=1}^n \frac{\delta L}{\delta Z_k} V_{jk} \cdot \text{ReLU}'(Z_j) \quad (8.22)$$

Keterangan:

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* 1 sebelum diaktivasi.

$\frac{\delta L}{\delta Z_k}$ = Nilai *error* dari *hidden layer* 2 sebelum diaktivasi.

V_{jk} = Bobot yang menyambungkan *hidden layer* 1 dengan *hidden layer* 2.

$\text{ReLU}'(Z_j)$ = Turunan dari fungsi aktivasi ReLU.

- Proses berikutnya adalah menghitung nilai *error* pada bobot yang menyambungkan *input layer* dengan *hidden layer* 1 ($\frac{\delta L}{\delta U_{ij}}$). Rumus dari proses ini disajikan dalam Persamaan 8.23.

$$\frac{\delta L}{\delta U_{ij}} = \frac{\delta L}{\delta Z_j} \cdot X_i \quad (8.23)$$

Keterangan:

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta Z_j}$ = Nilai *error* dari *hidden layer* 1 sebelum diaktivasi.

X_i = Data *input*.

- Kemudian untuk nilai *error* dari bias *hidden layer* 1 ($\frac{\delta L}{\delta b_j}$) memiliki nilai yang sama dengan nilai *error hidden layer* 1 sebelum diaktivasi ($\frac{\delta L}{\delta Z_j}$).

10. Melakukan *update* terhadap nilai bobot yang menyambungkan *input layer* dengan *hidden layer* 1, *hidden layer* 1 dengan *hidden layer* 2, *hidden layer* 2 dengan *hidden layer* 3, *hidden layer* 3 dengan *hidden layer* 4, dan *hidden layer* 4 dengan *hidden layer* 5, dan *hidden layer* 5 ke *output layer*. Secara berurutan rumus untuk proses ini ditampilkan dalam Persamaan 8.24 hingga 8.35.

- *Update* nilai bobot pada *hidden layer* 1 (U_{ij})

$$U_{ij} = U_{ij} - \eta \cdot \frac{\delta L}{\delta U_{ij}} \quad (8.24)$$

- *Update* nilai bias pada *hidden layer* 1 (b_j)

$$b_j = b_j - \eta \cdot \frac{\delta L}{\delta b_j} \quad (8.25)$$

- *Update* nilai bobot pada *hidden layer* 2 (V_{jk})

$$V_{jk} = V_{jk} - \eta \cdot \frac{\delta L}{\delta V_{jk}} \quad (8.26)$$

- *Update* nilai bias pada *hidden layer* 2 (c_k)

$$c_k = c_k - \eta \cdot \frac{\delta L}{\delta c_k} \quad (8.27)$$

- *Update* nilai bobot pada *hidden layer* 3 (T_{km})

$$T_{km} = T_{km} - \eta \cdot \frac{\delta L}{\delta T_{km}} \quad (8.28)$$

- *Update* nilai bias pada *hidden layer* 3 (e_m)

$$e_m = e_m - \eta \cdot \frac{\delta L}{\delta e_m} \quad (8.29)$$

- *Update* nilai bobot pada *hidden layer* 4 (Q_{mn})

$$Q_{mn} = Q_{mn} - \eta \cdot \frac{\delta L}{\delta Q_{mn}} \quad (8.30)$$

- *Update* nilai bias pada *hidden layer* 4 (f_n)

$$f_n = f_n - \eta \cdot \frac{\delta L}{\delta f_n} \quad (8.31)$$

- *Update* nilai bobot pada *hidden layer* 5 (S_{np})

$$S_{np} = S_{np} - \eta \cdot \frac{\delta L}{\delta S_{np}} \quad (8.32)$$

- *Update* nilai bias pada *hidden layer* 5 (g_p)

$$g_p = g_p - \eta \cdot \frac{\delta L}{\delta g_p} \quad (8.33)$$

- Update nilai bobot pada *output layer* (W_{pl})

$$W_{pl} = W_{pl} - \eta \cdot \frac{\delta L}{\delta W_{pl}} \quad (8.34)$$

- Update nilai bias pada *output layer* (d_l)

$$d_l = d_l - \eta \cdot \frac{\delta L}{\delta d_l} \quad (8.35)$$

Keterangan:

η = Nilai parameter *learning rate*.

$\frac{\delta L}{\delta U_{ij}}$ = Nilai *error* pada bobot U_{ij} .

$\frac{\delta L}{\delta b_j}$ = Nilai *error* bias pada *hidden layer* 1.

$\frac{\delta L}{\delta V_{jk}}$ = Nilai *error* pada bobot V_{jk} .

$\frac{\delta L}{\delta c_k}$ = Nilai *error* bias pada *hidden layer* 2.

$\frac{\delta L}{\delta T_{km}}$ = Nilai *error* pada bobot T_{km} .

$\frac{\delta L}{\delta e_m}$ = Nilai *error* bias pada *hidden layer* 3.

$\frac{\delta L}{\delta Q_{mn}}$ = Nilai *error* pada bobot Q_{mn} .

$\frac{\delta L}{\delta f_n}$ = Nilai *error* bias pada *hidden layer* 4.

$\frac{\delta L}{\delta S_{np}}$ = Nilai *error* pada bobot S_{np} .

$\frac{\delta L}{\delta g_p}$ = Nilai *error* bias pada *hidden layer* 5.

$\frac{\delta L}{\delta W_{pl}}$ = Nilai *error* pada bobot W_{pl} .

$\frac{\delta L}{\delta d_l}$ = Nilai *error* bias pada *output layer*.

8.2 Implementasi

Proses *training* model NN lima *hidden layer* memanfaatkan beberapa parameter, model NN E1 dan NN E2 memiliki parameter yang berbeda. Lebih detail dapat dilihat dalam Tabel 8.1.

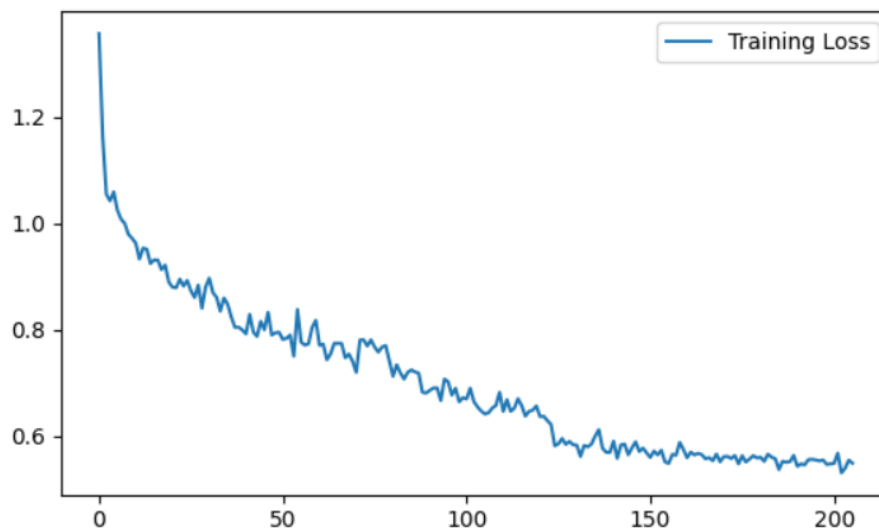
Tabel 8.1. Detail *Hyperparameter* model NN E1 dan NN E2

Hyperparameter	Nilai	
Tipe jaringan	E1	E2

Fungsi aktivasi	ReLU	ReLU
<i>Optimizer</i>	SGD	SGD
<i>Input Layer</i>	15	18
<i>Hidden layer 1</i>	45	54
<i>Hidden layer 2</i>	135	162
<i>Hidden layer 3</i>	405	486
<i>Hidden layer 4</i>	1215	1458
<i>Hidden layer 5</i>	3645	4374
<i>Output layer</i>	6	6
<i>Learning rate</i>	0.001	0.001

8.3 Uji Coba NN Model E1

Pada tahap uji coba, *training* model NN E1 menggunakan 5 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-206 dengan nilai *loss* sebesar 0.5496. Gambar 8.2 menunjukkan grafik nilai *loss* dari proses *training* NN lima *hidden layer*.

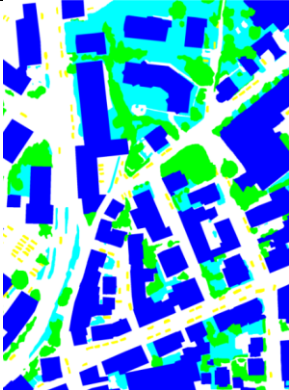
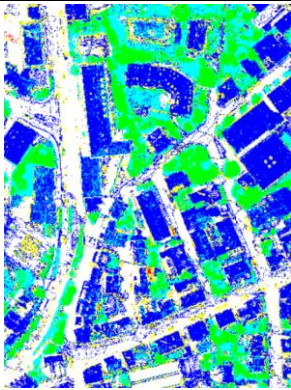
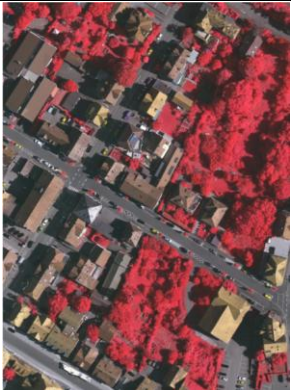
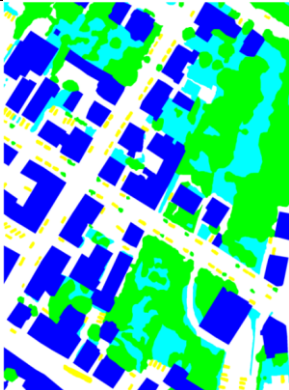
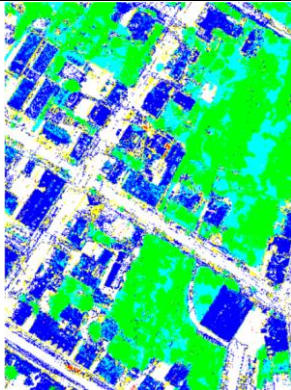


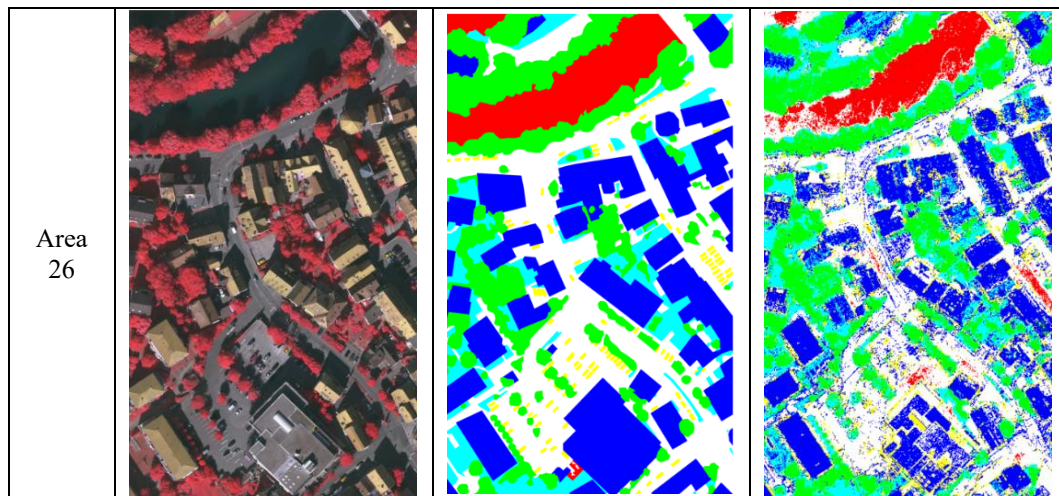
Gambar 8.2. Grafik Loss Training NN Model E1

Pada Gambar 8.2 menunjukkan nilai *loss* pada proses *training* model C1 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Didapatkan nilai *accuracy* sebesar 75.87%, *precision*

sebesar 76.22%, *recall* sebesar 75.87%, dan F1-Score sebesar 75.90%. Model optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model E1 dalam melakukan klasifikasi area tutupan lahan disajikan dalam Tabel 8.2.

Tabel 8.2. Beberapa Hasil Klasifikasi Model NN E1 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			



Hasil klasifikasi model NN E1 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Dalam Gambar 8.3, terlihat salah klasifikasi pada beberapa tutupan lahan oleh model NN E1. Lebih detail, untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan secara detail disajikan dalam Tabel 8.3.

Tabel 8.3. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN E1

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1220610	2663946	413200	632155
	Permukaan kedap air	1114662	2800935	375813	638501
	<i>Low vegetation</i>	339854	3838390	469986	281681
	Pepohonan	422582	4144895	158893	203541
	<i>Background</i>	0	4925509	4400	2
	Mobil	32744	4476421	377167	43579
Area 3	Bangunan	931862	3595984	803798	700398
	Permukaan kedap air	1028553	3653220	319454	1030815
	<i>Low vegetation</i>	599101	4014186	860699	558056
	Pepohonan	732238	4617886	286762	395156
	<i>Background</i>	0	6029092	2950	0
	Mobil	28474	5538028	438151	27389
Area 7	Bangunan	808945	3177132	342835	496147
	Permukaan kedap air	959053	3061329	271336	533341
	<i>Low vegetation</i>	311911	3684285	503544	325319
	Pepohonan	991631	3270707	271649	291072
	<i>Background</i>	0	4820449	4610	0
	Mobil	43975	4401849	315570	63665
Area 11	Bangunan	440012	3455125	497916	464385
	Permukaan kedap air	836783	3103439	145267	771949
	<i>Low vegetation</i>	499299	3210275	839011	308853
	Pepohonan	1045518	3197322	191087	423511

Data	Kelas	TP	TN	FP	FN
Area 11	<i>Background</i>	0	4850643	6795	0
	Mobil	33570	4468130	322180	33558
Area 15	Bangunan	650567	3439447	407618	424603
	Permukaan kedap air	357226	3903434	163689	497886
	<i>Low vegetation</i>	757607	2572180	1040158	552290
	Pepohonan	873527	2976233	302813	769662
	<i>Background</i>	25	4909677	5375	7158
	Mobil	16960	4543881	346670	14724
Area 23	Bangunan	487207	3679596	280158	398077
	Permukaan kedap air	309349	3925797	117589	492303
	<i>Low vegetation</i>	883457	2550677	948633	462271
	Pepohonan	1177033	2758125	297742	612138
	<i>Background</i>	0	4836207	1075	7756
	Mobil	8863	4495659	333932	6584
Area 26	Bangunan	981738	3438704	342917	576726
	Permukaan kedap air	1054626	3202423	486999	596037
	<i>Low vegetation</i>	280117	4361761	476368	221839
	Pepohonan	821526	4077511	176949	264099
	<i>Background</i>	292777	4866368	26583	154357
	Mobil	37234	4881591	362251	59009
Area 32	Bangunan	584939	2998625	789361	685975
	Permukaan kedap air	1675036	1996698	445514	941652
	<i>Low vegetation</i>	241448	4329177	367127	121148
	Pepohonan	285016	4415792	46834	311258
	<i>Background</i>	43	4949041	54257	55559
	Mobil	69056	4401805	500269	87770
Area 37	Bangunan	657448	2455984	603457	265131
	Permukaan kedap air	520129	2837143	223761	400987
	<i>Low vegetation</i>	899386	2113374	314099	655161
	Pepohonan	274563	3259324	157562	290571
	<i>Background</i>	0	3979545	2475	0
	Mobil	6410	3640646	322730	12234

Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN E1 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN E1 disajikan dalam Tabel 8.4.

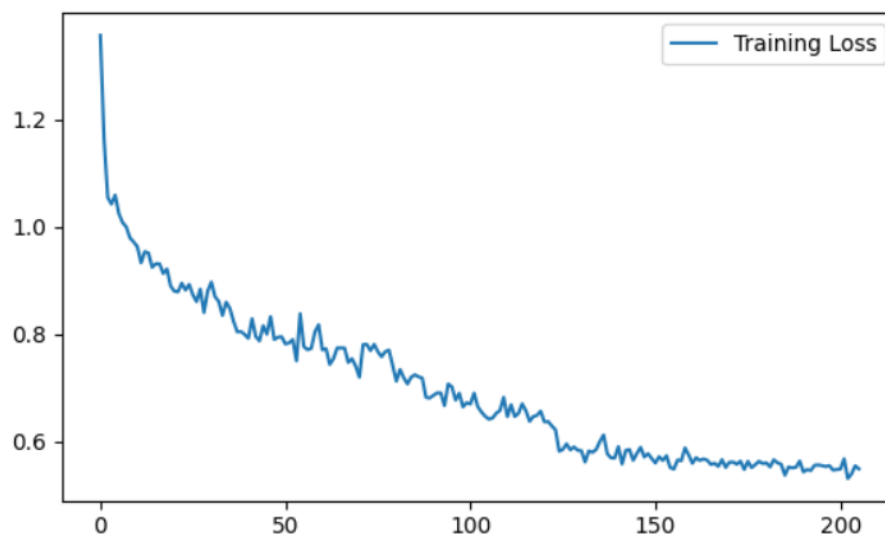
Tabel 8.4. Performa Hasil Pengujian Model NN E1

Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	88.12	63.73	63.75	63.74	46.77
Area 3	85.04	51.01	51.97	51.46	34.66
Area 7	88.14	64.63	64.66	64.64	47.76
Area 11	86.03	58.73	58.72	58.73	41.57

Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 15	85.14	54.00	53.03	53.53	36.56
Area 23	86.39	59.23	59.17	59.19	42.06
Area 26	88.30	64.88	64.78	64.79	47.99
Area 32	86.02	56.62	56.67	56.64	39.49
Area 37	86.02	59.22	59.10	59.16	42.02

8.4 Uji Coba NN Model E2

Pada tahap uji coba, *training* model NN E2 menggunakan 6 fitur GLCM, dan untuk rasio pembagian datanya adalah sebesar 80% atau sebanyak 4838 citra untuk data *training* dan 20% atau sebanyak 1210 citra untuk data *testing*. Proses *training* berhenti dan mencapai titik *convergence* pada *epoch* ke-206 dengan nilai *loss* sebesar 0.5496. Gambar 8.3 menunjukkan grafik nilai *loss* dari proses *training* NN lima *hidden layer*.

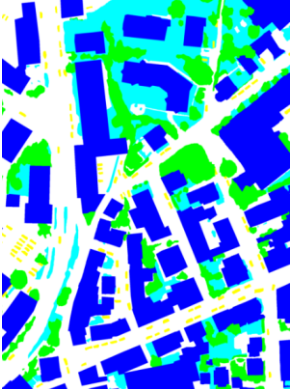
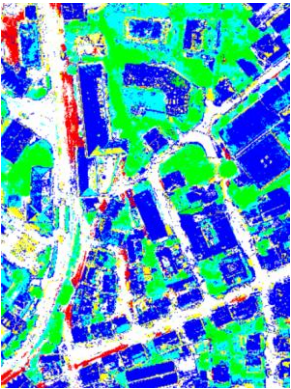
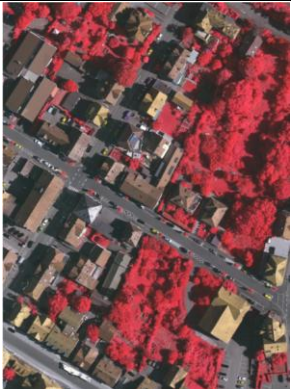
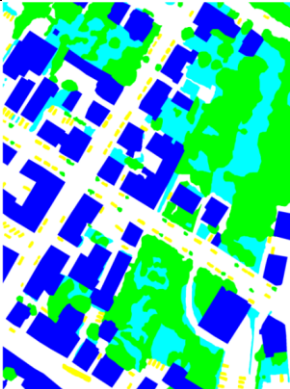
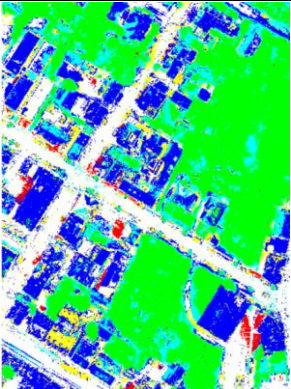
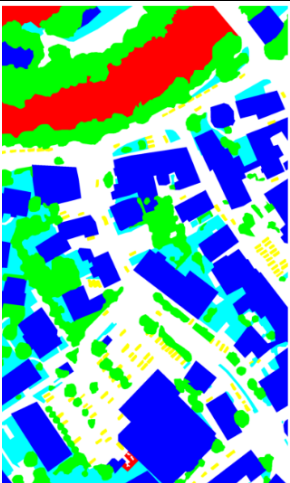
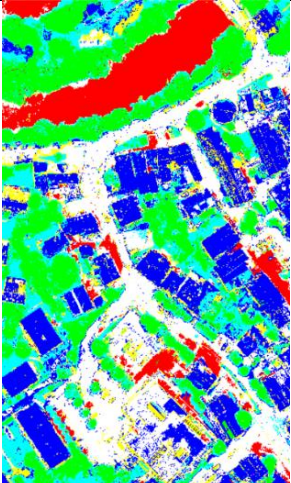


Gambar 8.3. Grafik Loss Training NN Model E2

Pada Gambar 8.3 menunjukkan nilai *loss* pada proses *training* model E2 yang terus mengalami penurunan hingga mencapai titik konvergen, sehingga model dianggap sudah optimal. Didapatkan nilai *accuracy* sebesar 86.53%, *precision* sebesar 86.68%, *recall* sebesar 86.53%, dan F1-Score sebesar 86.54%. Model

optimal yang telah didapatkan, kemudian digunakan untuk memprediksi area tutupan lahan dari data *input* citra Vaihingen yang berjumlah 9 citra beresolusi tinggi. Hasil pengujian model E2 dalam melakukan klasifikasi area tutupan lahan lebih detail disajikan dalam ditampilkan dalam Tabel 8.5.

Tabel 8.5. Beberapa Hasil Klasifikasi Model NN E2 Terhadap Citra Beresolusi Tinggi

Kode Area	Citra Beresolusi Tinggi	Ground Truth	Hasil Klasifikasi Tutupan Lahan
Area 1			
Area 7			
Area 26			

Hasil klasifikasi model NN E2 terhadap 9 citra beresolusi tinggi memiliki nilai metrik yang berbeda. Dalam Tabel 8.5, terlihat model E2 dapat melakukan pada jenis tutupan lahan permukaan kedap air dengan lebih baik dibandingkan model E2. Lebih detail, untuk menghitung nilai metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU dibutuhkan nilai TP, TN, FP, dan FN dari hasil klasifikasi tutupan lahan yang secara detail disajikan dalam Tabel 8.6.

Tabel 8.6. Nilai TP, TN, FP, FN Setiap Area Hasil Klasifikasi NN E2

Data	Kelas	TP	TN	FP	FN
Area 1	Bangunan	1260267	2708753	368393	592498
	Permukaan kedap air	977336	2899058	277690	775827
	<i>Low vegetation</i>	355816	3804347	504029	265719
	Pepohonan	464759	4094342	209446	161364
	<i>Background</i>	0	4800084	129825	2
	Mobil	32479	4503717	349871	43844
Area 3	Bangunan	868137	3595082	804700	764123
	Permukaan kedap air	797765	3772114	200560	1261603
	<i>Low vegetation</i>	726327	3789107	1085778	430830
	Pepohonan	790188	4545511	359137	337206
	<i>Background</i>	0	6000542	31500	0
	Mobil	24161	5632390	343789	31702
Area 7	Bangunan	780928	3226915	293052	524164
	Permukaan kedap air	905188	3092694	239971	587206
	<i>Low vegetation</i>	218861	3762660	425169	418369
	Pepohonan	1155851	3134267	408089	126852
	<i>Background</i>	0	4760024	65035	0
	Mobil	43141	4427645	289774	64499
Area 11	Bangunan	489735	3475918	477123	414662
	Permukaan kedap air	760678	3117639	131067	848054
	<i>Low vegetation</i>	400072	3351678	697608	408080
	Pepohonan	1235204	3003698	384711	233825
	<i>Background</i>	0	4829688	27750	0
	Mobil	30214	4567034	223276	36914
Area 15	Bangunan	660182	3559282	287783	414988
	Permukaan kedap air	312741	3899144	167979	542371
	<i>Low vegetation</i>	488039	3039492	572846	821858
	Pepohonan	1345103	2537764	741282	298086
	<i>Background</i>	6	4865148	49904	7177
	Mobil	16538	4610719	279832	15146
Area 23	Bangunan	482202	3773073	186681	403082
	Permukaan kedap air	250683	3929429	113957	550969
	<i>Low vegetation</i>	706183	2763078	736232	639545
	Pepohonan	1464513	2416190	639677	324658
	<i>Background</i>	25	4814057	23225	7731
	Mobil	8397	4596328	233263	7050
Area 26	Bangunan	949573	3552964	228657	608891

Data	Kelas	TP	TN	FP	FN
Area 26	Permukaan kedap air	979779	3333921	355501	670884
	<i>Low vegetation</i>	322690	4360164	477965	179266
	Pepohonan	893161	4070466	183994	192464
	<i>Background</i>	386111	4711532	181419	61023
	Mobil	35649	4898256	345586	60594
Area 32	Bangunan	420689	2644825	1143161	850225
	Permukaan kedap air	1065598	2125135	317077	1551090
	<i>Low vegetation</i>	205389	4067093	629211	157207
	Pepohonan	384396	4345272	117354	211878
	<i>Background</i>	593	4849066	154232	55009
	Mobil	49541	4330415	571659	107285
Area 37	Bangunan	604257	2621793	437648	318322
	Permukaan kedap air	516446	2893765	167139	404670
	<i>Low vegetation</i>	1077416	1998359	429114	477131
	Pepohonan	315848	3206014	210872	249286
	<i>Background</i>	0	3971370	10650	0
	Mobil	4167	3754913	208463	14477

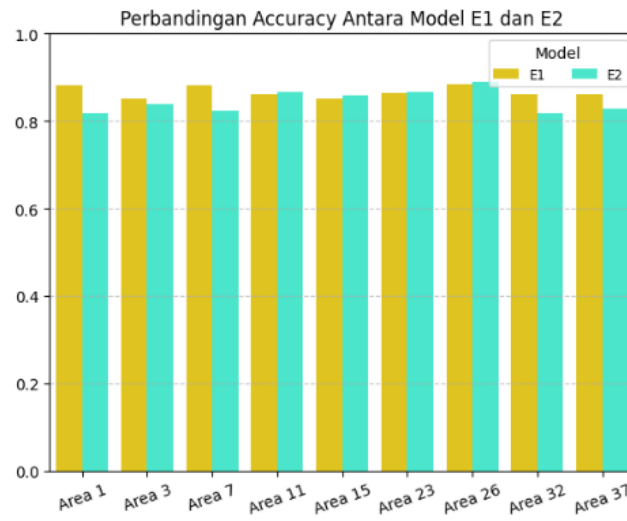
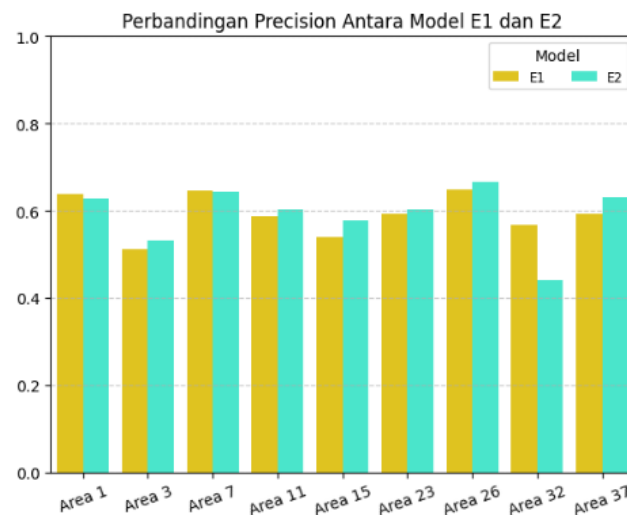
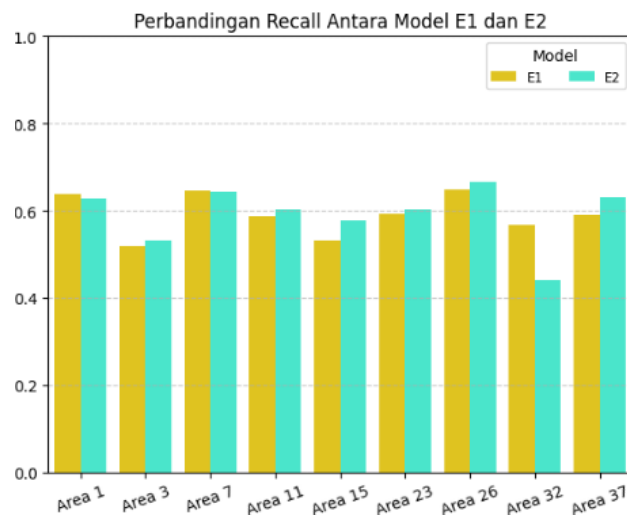
Setelah didapatkan nilai TP, TN, FP, dan FN dari setiap kelas pada masing-masing area citra beresolusi tinggi. Langkah selanjutnya adalah menghitung metrik evaluasi yang digunakan untuk mengukur performa model NN E2 untuk mengklasifikasikan 9 citra beresolusi tinggi. Lebih lengkap mengenai performa hasil pengujian model NN E2 disajikan dalam Tabel 8.7.

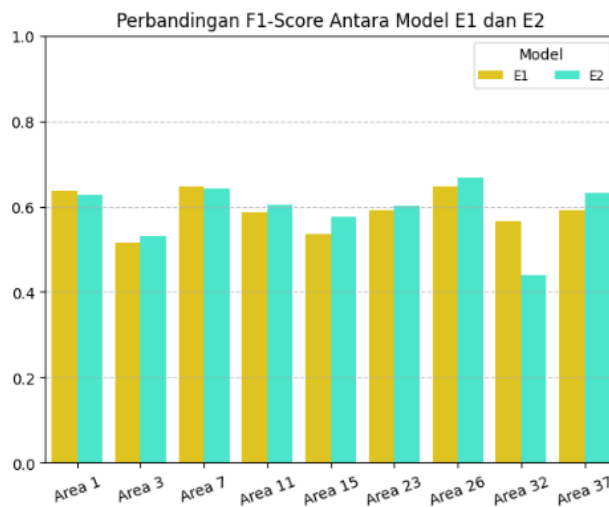
Tabel 8.7. Performa Hasil Pengujian Model NN E2

Kode Area Citra Beresolusi Tinggi	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
Area 1	81.76	62.68	62.71	62.69	45.68
Area 3	83.88	53.04	53.14	53.10	36.16
Area 7	82.41	64.23	64.30	64.26	47.33
Area 11	86.72	60.38	60.38	60.38	43.25
Area 15	85.87	57.65	57.65	57.65	40.48
Area 23	86.70	60.16	60.16	60.16	43.02
Area 26	88.90	66.71	66.71	66.71	50.04
Area 32	81.77	43.91	43.91	43.91	28.12
Area 37	82.89	63.05	63.05	63.05	46.07

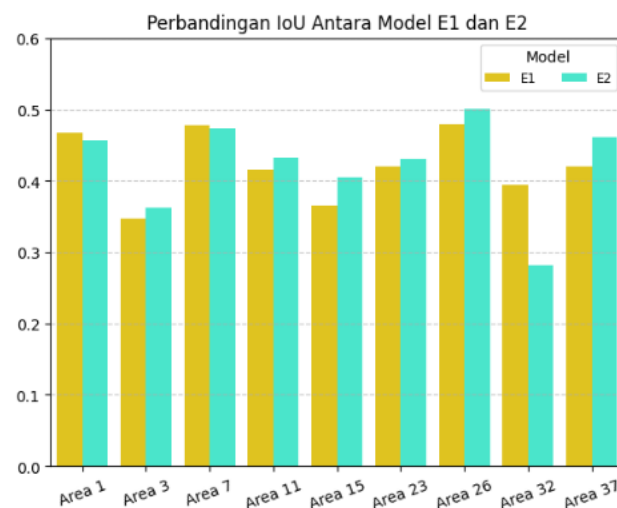
8.5 Komparasi Performa NN Model E1 dan E2

Selanjutnya dilakukan komparasi antara NN model E1 dan E2 untuk memudahkan perbandingan performa antar model. Secara berurutan perbandingan model ditampilkan dalam Gambar 8.4 hingga Gambar 8.8.

Gambar 8.4. Perbandingan *Accuracy* Antara Model NN E1 dan E2Gambar 8.5. Perbandingan *Precision* Antara Model NN E1 dan E2Gambar 8.6. Perbandingan *Recall* Antara Model NN E1 dan E2



Gambar 8.7. Perbandingan F1-score Antara Model NN E1 dan E2



Gambar 8.8. Perbandingan IoU Antara Model NN E1 dan E2

Perbandingan antara model NN E1 yang menggunakan 5 fitur GLCM dan model NN E2 yang menggunakan 6 fitur GLCM menunjukkan hasil yang berbeda-beda. Meskipun pada NN satu hingga tiga *hidden layer* menunjukkan bahwa model dengan GLCM 6 fitur selalu lebih unggul dibandingkan dengan model dengan GLCM 5 fitur, namun pada NN empat dan lima *hidden layer* mulai menunjukkan perbedaan, yang mana pada beberapa Area model dengan GLCM 5 fitur dapat mengungguli performa model yang menggunakan GLCM 6 fitur.

Dari segi metrik *accuracy*, meskipun pada beberapa Area model E1 memiliki akurasi yang lebih rendah dibanding model E2, namun secara keseluruhan model E1 memiliki keunggulan dibandingkan dengan model E2. Dari perbandingan yang dilakukan, model E1 lebih unggul dalam 5 Area citra beresolusi tinggi, yakni pada Area 1, Area 3, Area 7, Area 32, dan Area 37.

Lalu berikutnya dari segi metrik *precision*, meskipun pada beberapa Area model E2 memiliki akurasi lebih rendah dibanding model E1, namun secara keseluruhan model E2 unggul dibandingkan dengan model E1. Dari perbandingan yang dilakukan, model E2 unggul dalam 6 Area dibandingkan dengan model E1, yakni pada Area 3, Area 11, Area 15, Area 23, Area 26, dan Area 37.

Selanjutnya dari metrik *recall*, model E2 menunjukkan keunggulan dalam 6 Area dibandingkan model E1 yang hanya mengungguli model E2 dalam 3 Area. Performa metrik *recall* model E2 mengungguli model E1 pada Area 3, Area 11, Area 15, Area 23, Area 26, dan Area 37.

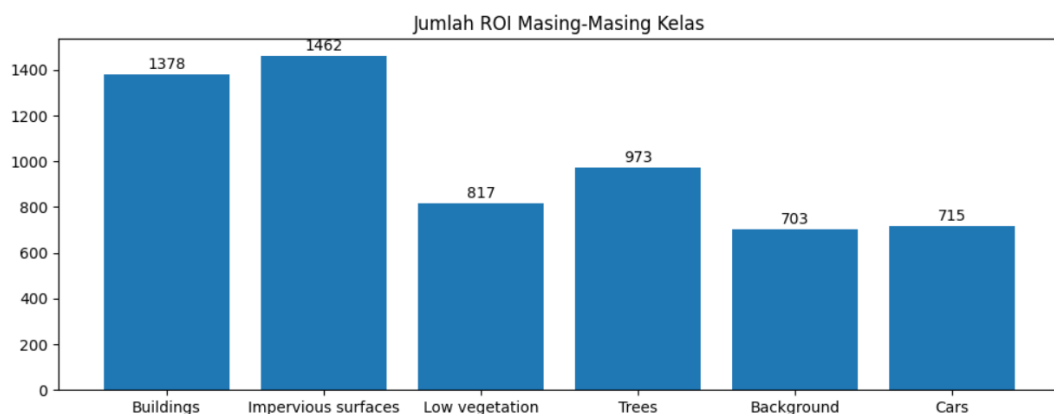
Berikutnya pada metrik *F1-Score* model E2 yang menggunakan GLCM 6 fitur menunjukkan keunggulan pada 6 Area dibandingkan model E1 dengan GLCM 5 fitur yang unggul hanya pada 3 Area. Model E2 mengungguli model E1 pada Area 3, Area 11, Area 15, Area 23, Area 26, dan Area 37.

Terakhir adalah perbandingan pada metrik IoU. Didapatkan model E2 dengan GLCM 6 fitur mengungguli secara mayoritas dibandingkan model E1 dengan GLCM 5 fitur. Model E2 mampu mengungguli model E1 pada Area 3, Area 11, Area 15, Area 23, Area 26, dan Area 37.

BAB IX

PEMBAHASAN

Dalam melakukan analisis performa model untuk melakukan klasifikasi tutupan lahan, perlu terlebih dahulu untuk menganalisa distribusi dataset yang digunakan dalam penelitian. Dataset yang dipelajari oleh model NN dalam penelitian ini merupakan data ROI dari masing-masing kelas pada citra beresolusi tinggi yang berjumlah 6048 citra ROI. Lebih detail mengenai jumlah distribusi data ROI dari masing-masing kelas ditampilkan pada Gambar 9.1.



Gambar 9.1. Distribusi Data ROI Masing-Masing Kelas

Berdasarkan Gambar 9.1, dapat dilihat bahwa dataset yang digunakan pada penelitian ini memiliki distribusi yang tidak seimbang yang terbukti dari beberapa kelas yang jumlahnya sekitar setengah dari kelas dengan jumlah ROI terbanyak. Kelas yang mendominasi adalah kelas *buildings* dengan jumlah ROI sebanyak 1378 citra, dan kelas *impervious surfaces* dengan jumlah ROI sebanyak 1462 citra. Banyaknya jumlah ROI pada kelas *buildings* disebabkan oleh keragaman arsitektur, bentuk, warna, dan ukuran objek bangunan. Oleh karena itu, jumlah ROI yang tinggi dibutuhkan untuk mencakup keragaman tersebut, sehingga model NN dapat

mengenali berbagai jenis bangunan. Keragaman pada kelas *buildings* juga terjadi pada kelas *impervious surfaces* yang mencakup jalan raya serta lahan parkir, yang mana beberapa area pada jalan raya dan lahan parkir tertutup oleh bayangan dari objek lain sehingga sukar untuk dikenali oleh model. Hal ini menjadi dasar diperlukan jumlah ROI dalam jumlah yang lebih banyak yang mencakup area kelas *impervious surfaces* baik yang tertutup oleh bayangan maupun yang tidak agar model NN dapat mengenali kelas *impervious surfaces* dengan lebih baik.

Selanjutnya pada kelas *low vegetation* dan *trees*, meskipun kedua kelas ini memiliki cakupan area yang luas pada setiap citra beresolusi tinggi Vaihingen, namun variasi bentuk, warna, serta tekstur dari dua kelas ini terbatas. Berbeda dengan kelas *building* yang memiliki keragaman bentuk, warna, tekstur, dan ukuran objek bangunan, kelas *low vegetation* dan *trees* cenderung seragam meskipun terletak pada area yang berbeda. Sehingga pada penelitian ini ROI pada kelas *low vegetation* diambil sebanyak 817 citra, dan untuk kelas *trees* diambil sebanyak 973 citra.

Terakhir pada kelas *background* dan *cars* yang memiliki jumlah ROI paling sedikit terjadi karena keterbatasan jumlah area *background* pada citra beresolusi tinggi Vaihingen. Area kelas *background* hanya terdapat pada beberapa Area citra Vaihingen, secara khusus Area 26 memiliki kelas *background* paling luas dibandingkan dengan Area lainnya. Keterbatasan inilah yang mengakibatkan jumlah ROI dari kelas *background* sedikit. Lalu untuk kelas *cars*, sedikitnya jumlah ROI diakibatkan oleh sedikitnya jumlah objek mobil pada setiap Area citra beresolusi tinggi Vaihingen. Sehingga setelah dilakukan pemilihan ROI, didapatkan

jumlah ROI untuk kelas *background* adalah sejumlah 703 citra, dan untuk jumlah ROI pada kelas *cars* didapatkan sejumlah 715 citra.

Berdasarkan analisis distribusi data ROI pada setiap kelas citra beresolusi tinggi Vaihingen, dapat disimpulkan bahwa jumlah distribusi data ROI yang digunakan setiap kelas yang digunakan pada penelitian ini tidak seimbang atau *imbalance*. Data yang *imbalance* ini dapat menyebabkan hasil klasifikasi yang dilakukan oleh model NN cenderung bias pada kelas dengan jumlah data yang lebih banyak dibandingkan kelas lainnya yang memiliki jumlah data jauh lebih sedikit.

Setelah didapatkan hasil analisis mengenai distribusi data yang digunakan dalam proses *training* dan *testing* oleh model NN, selanjutnya dilakukan analisis hasil dari skenario pengujian yang telah dijalankan, didapatkan hasil yang berbeda dari setiap model. Proses *training* berjalan dengan mengeluarkan *output* berupa nilai *loss* yang terus menurun seiring berjalannya proses *training*. Model yang telah optimal kemudian digunakan untuk menguji dan melakukan klasifikasi tutupan lahan pada 9 Area citra beresolusi tinggi.

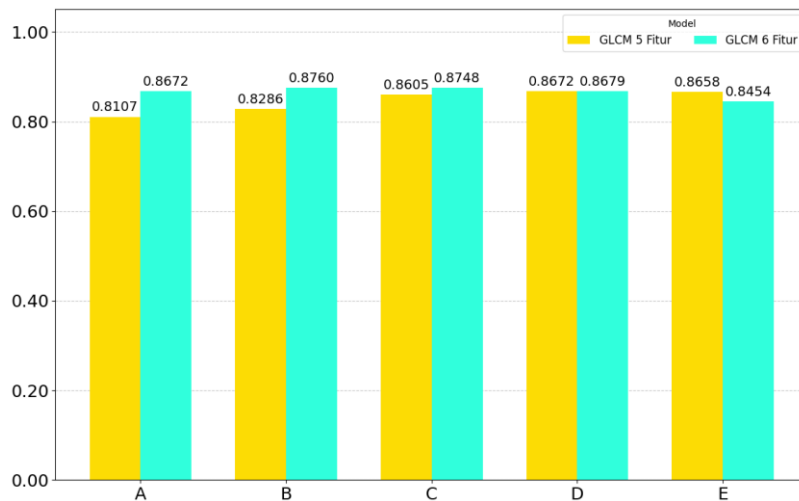
Hasil klasifikasi model terhadap tutupan lahan pada 9 Area citra beresolusi tinggi menunjukkan hasil metrik yang beragam. Hasil pengujian berupa metrik *accuracy*, *precision*, *recall*, *F1-Score*, dan *IoU* dari setiap model pada masing-masing Area citra beresolusi tinggi yang telah didapatkan kemudian dirata-rata dan dibandingkan hasil performanya dengan model lain. Lebih detail rangkuman hasil rata-rata performa model pada setiap metrik disajikan dalam Tabel 9.1.

Tabel 9.1. Perbandingan Rata-Rata Performa Antara Semua Model

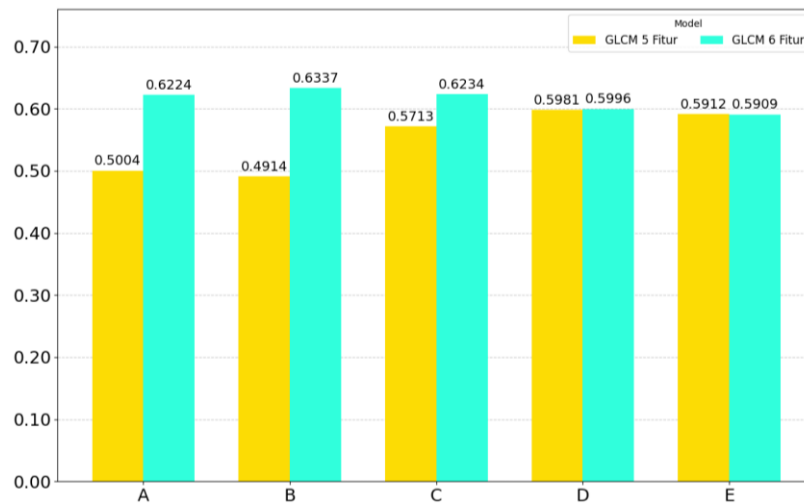
Model	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>IoU (%)</i>
A1	81.07	50.04	49.96	49.98	33.41
A2	86.72	62.24	61.94	62.09	45.15
B1	82.86	49.14	48.85	48.99	33.43

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
B2	87.60	63.37	63.40	63.39	48.22
C1	86.05	57.13	57.15	57.14	40.11
C2	87.48	62.34	62.33	62.34	45.39
D1	86.72	59.81	59.34	59.54	42.57
D2	86.79	59.96	59.93	59.95	42.97
E1	86.58	59.12	59.09	59.10	42.10
E2	84.54	59.09	59.11	59.10	42.24

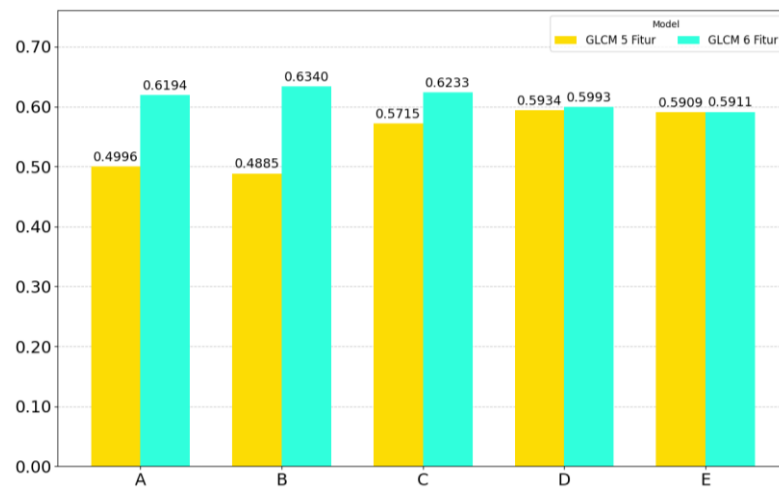
Hasil performa yang telah dirangkum dalam Tabel 9.1 kemudian dilakukan visualisasi guna mempermudah proses perbandingan antar model. Sehingga terlihat dampak dari perbedaan penggunaan jumlah *hidden layer* dan fitur GLCM pada performa model. Perbandingan pertama adalah dengan membandingkan dampak penggunaan jumlah fitur GLCM terhadap performa model dalam melakukan klasifikasi tutupan lahan yang secara lebih detail ditampilkan dalam Gambar 9.2 hingga Gambar 9.6.



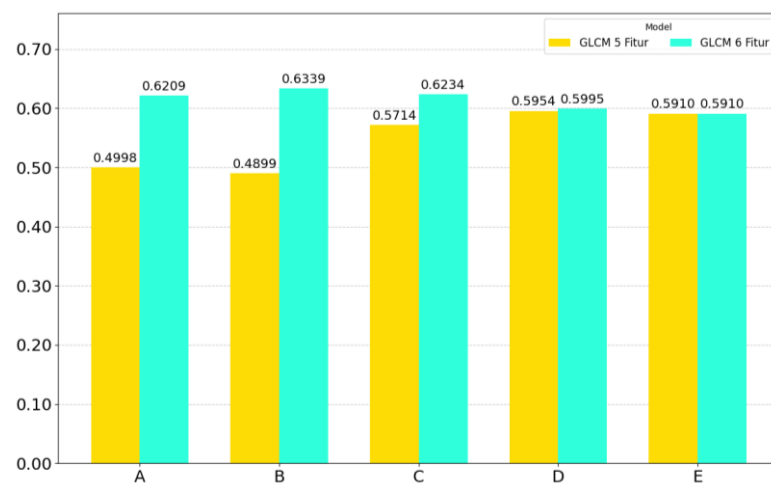
Gambar 9.2. Perbandingan Metrik *Accuracy* Antar Model Berdasarkan Jumlah Fitur GLCM



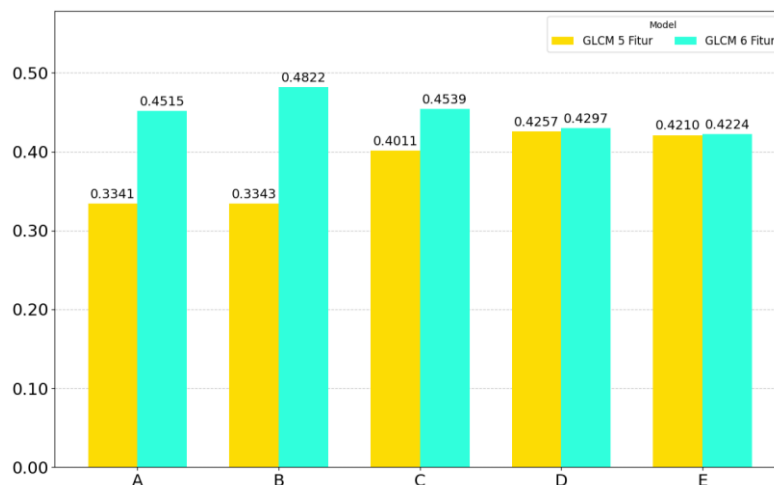
Gambar 9.3. Perbandingan Metrik *Precision* Antar Model Berdasarkan Jumlah Fitur GLCM



Gambar 9.4. Perbandingan Metrik *Recall* Antar Model Berdasarkan Jumlah Fitur GLCM



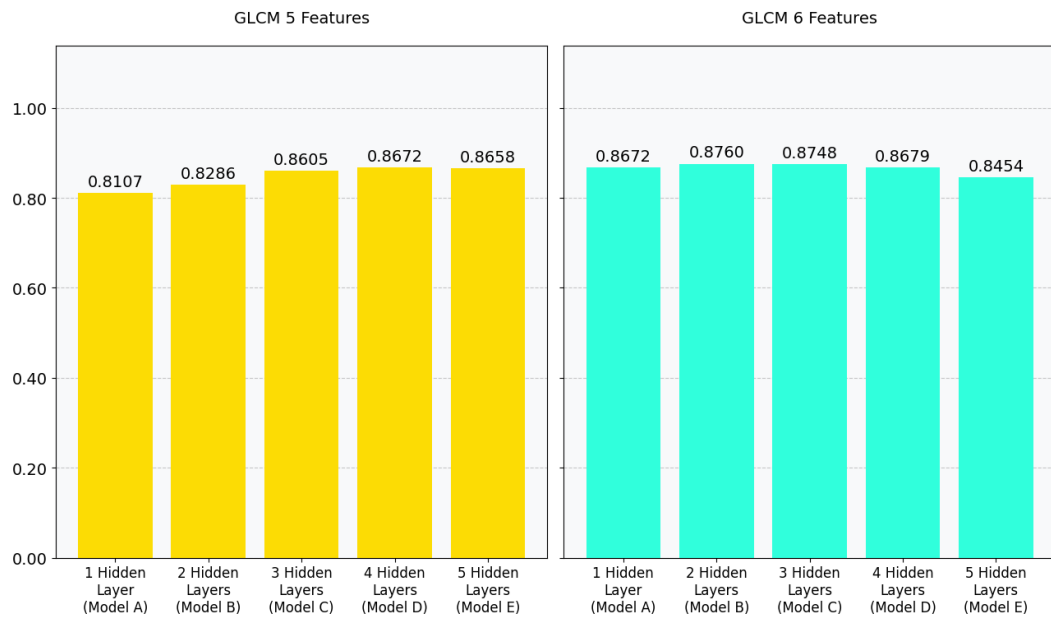
Gambar 9.5. Perbandingan Metrik *F1-Score* Antar Model Berdasarkan Jumlah Fitur GLCM



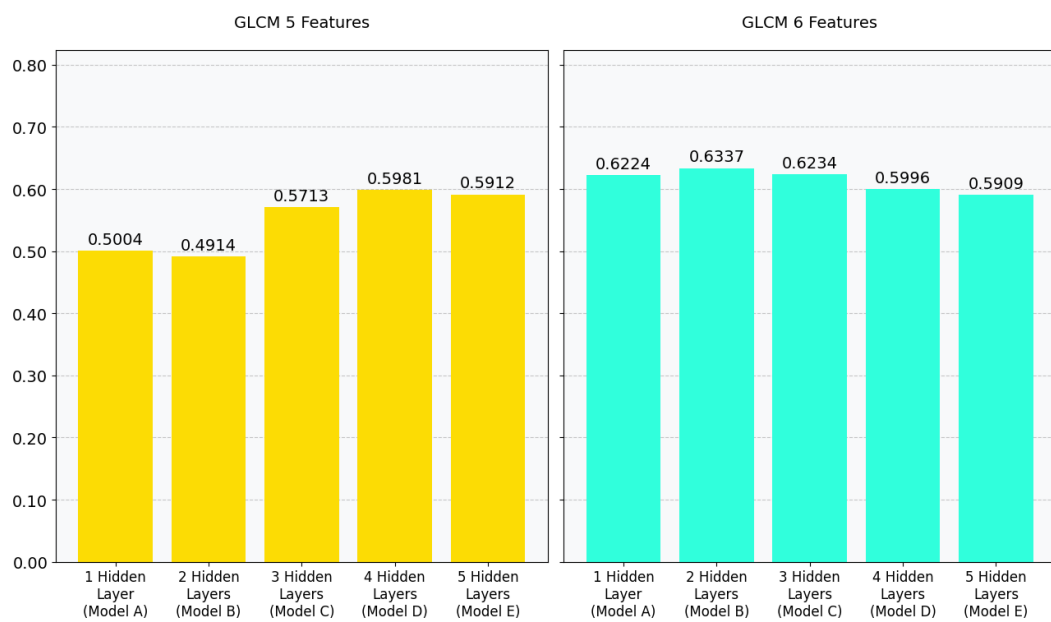
Gambar 9.6. Perbandingan Metrik IoU Antar Model Berdasarkan Jumlah Fitur GLCM

Dari perbandingan mengenai dampak penggunaan jumlah fitur GLCM terhadap performa model yang ditampilkan dalam Gambar 9.2 hingga Gambar 9.3, meskipun terdapat penurunan performa khususnya pada metrik *accuracy* dan *precision* pada model E, secara keseluruhan dari visualisasi perbandingan pada Gambar 9.2 hingga Gambar 9.6 didapatkan bahwa secara garis besar dengan menambah jumlah fitur GLCM dapat meningkatkan performa model baik dari segi *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU.

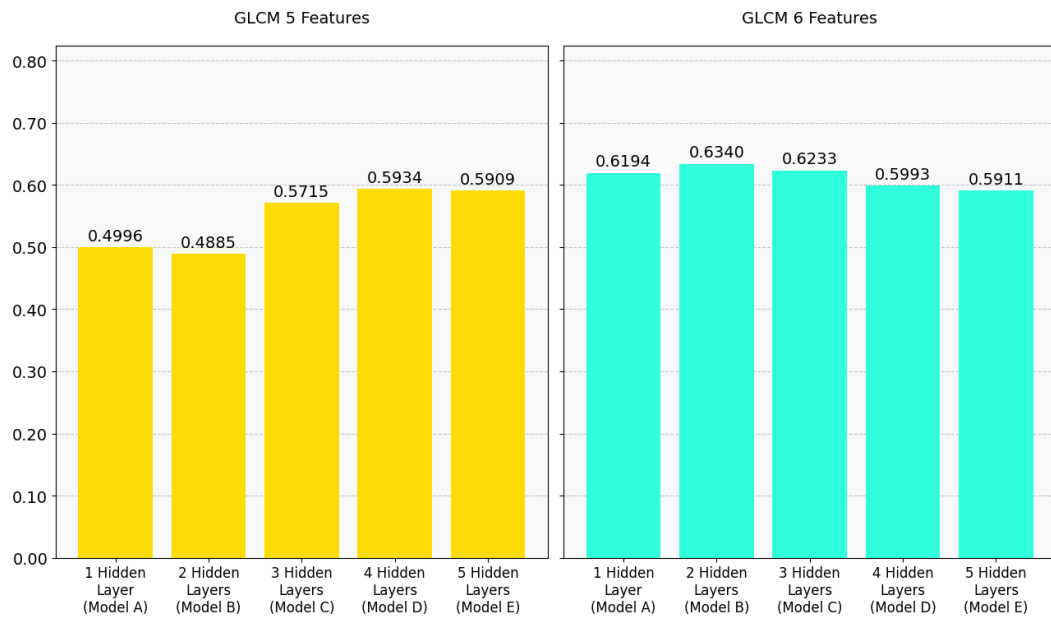
Dari analisa perbandingan mengenai penambahan jumlah fitur GLCM yang awalnya berjumlah 5 fitur menjadi 6 fitur menunjukkan peningkatan performa pada semua jenis metrik, baik pada *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU. Perbandingan berikutnya adalah dengan membandingkan dampak penggunaan jumlah *hidden layer* terhadap performa model dalam melakukan klasifikasi tutupan lahan yang secara lebih detail ditampilkan dalam Gambar 9.7 hingga Gambar 9.11.



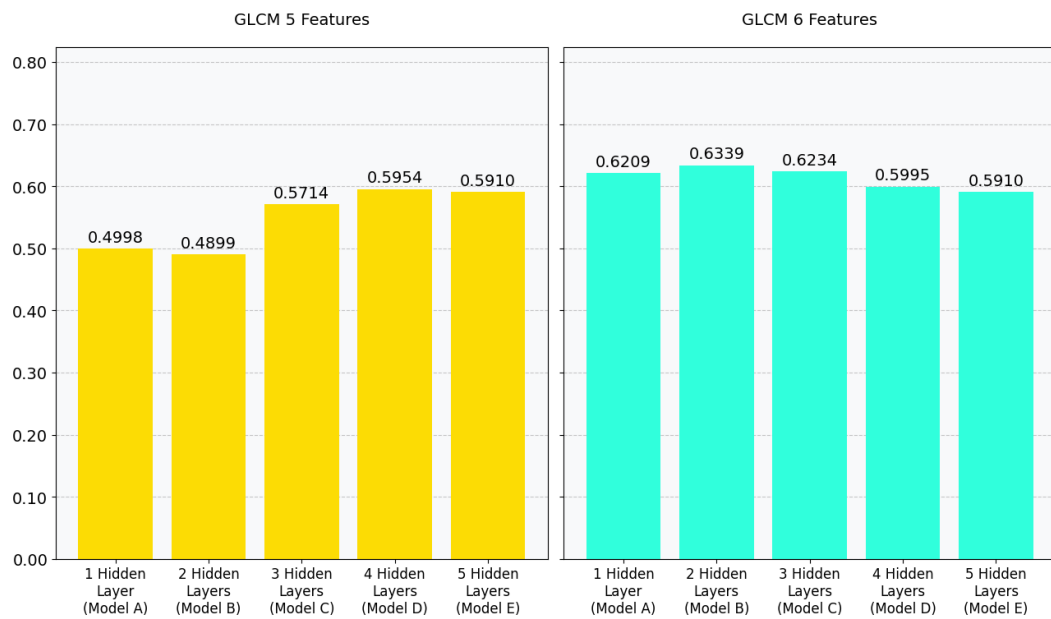
Gambar 9.7. Perbandingan Metrik *Accuracy* Antar Model Berdasarkan Jumlah *Hidden Layer*



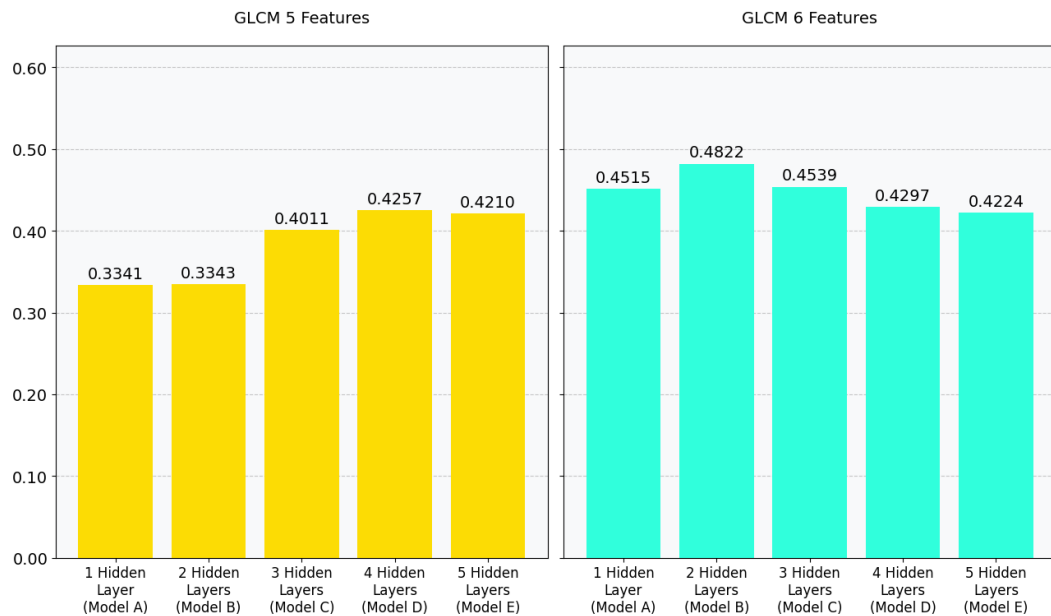
Gambar 9.8. Perbandingan Metrik *Precision* Antar Model Berdasarkan Jumlah *Hidden Layer*



Gambar 9.9. Perbandingan Metrik *Recall* Antar Model Berdasarkan Jumlah *Hidden Layer*



Gambar 9.10. Perbandingan Metrik *F1-Score* Antar Model Berdasarkan Jumlah *Hidden Layer*



Gambar 9.11. Perbandingan Metrik IoU Antar Model Berdasarkan Jumlah *Hidden Layer*

Berbeda dengan perbandingan jumlah fitur GLCM yang secara keseluruhan performa akan meningkat dengan ditambahkannya jumlah fitur GLCM untuk dipelajari model, hasil dari perbandingan dampak penggunaan jumlah *hidden layer* terhadap performa model memperlihatkan hasil performa yang tidak selalu meningkat seiring bertambahnya *hidden layer*.

Perbandingan penggunaan jumlah *hidden layer* pada model A hingga model E yang menggunakan GLCM 5 fitur menunjukkan pola yang sama, yakni terjadi penurunan performa model dari 1 *hidden layer* ke 2 *hidden layer* dan kemudian performa model mengalami kenaikan secara berkala seiring ditambahkannya jumlah *hidden layer* hingga 4 *hidden layer* lalu mengalami penurunan performa pada penambahan jumlah *hidden layer* ke-5.

Hal yang sebaliknya terjadi pada perbandingan penggunaan jumlah *hidden layer* pada model A hingga model E yang menggunakan 6 fitur GLCM, yakni

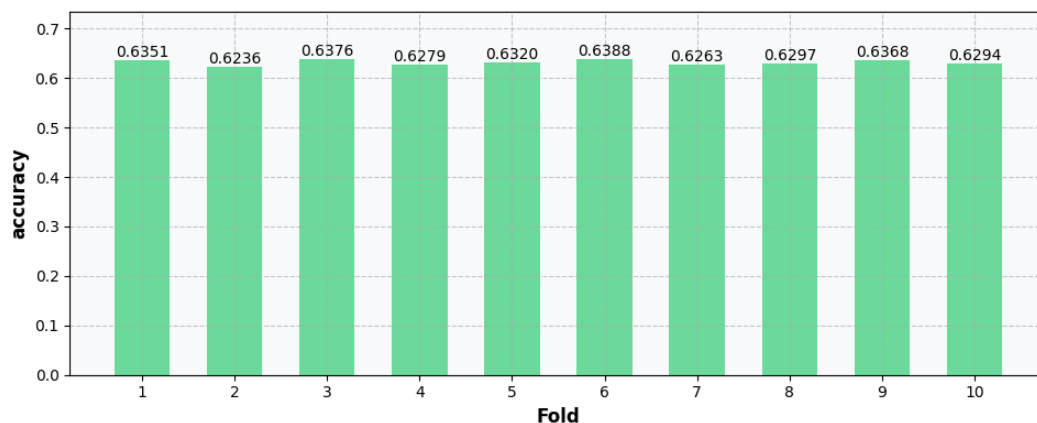
terjadi peningkatan performa model dari 1 *hidden layer* ke 2 *hidden layer* dan kemudian terjadi penurunan performa model secara berkala seiring ditambahkannya jumlah dari 1 *hidden layer* hingga 5 *hidden layers*. Dari seluruh perbandingan kinerja model yang dilakukan, didapatkan bahwa model B2 atau model dengan 2 *hidden layer* yang memanfaatkan 6 fitur GLCM mendapatkan hasil performa terbaik dibandingkan model lainnya.

Dari analisa perbandingan mengenai penambahan jumlah *hidden layer* pada NN, diketahui bahwa dengan menambahkan jumlah *hidden layer* tidak selalu berhasil meningkatkan kinerja model dan dapat menyebabkan model menjadi *overfitting*. Isaac dan Zareinia (2022) dalam penelitiannya menyatakan bahwa *overfitting* adalah kondisi dimana model terlalu banyak mempelajari data *training* sehingga menghasilkan kinerja yang bagus dari proses *training*, namun menunjukkan kinerja yang buruk di saat proses *testing*. Dalam penelitian yang sama Isaac dan Zareinia (2022) juga menjelaskan tentang salah satu penyebab dari *overfitting* adalah karena terlalu banyak menambahkan *layer* pada arsitektur NN. Oleh karena itu, dapat disimpulkan bahwa hasil eksperimen dari skenario pengujian yang telah ditetapkan diawal penelitian ini, kombinasi arsitektur dan jumlah fitur GLCM yang menghasilkan performa terbaik didapatkan oleh model B2 yang memiliki 2 *hidden layer* dan menggunakan 6 fitur GLCM.

Selanjutnya pada penelitian ini digunakan *k-fold cross validation* untuk mengetahui kemampuan dan seberapa stabil performa model dalam melakukan klasifikasi tutupan lahan pada semua distribusi data. Dataset dibagi menjadi beberapa bagian yang disebut sebagai *fold* atau biasa diwakili oleh variabel *k*.

Proses *training* akan berjalan dengan memanfaatkan setiap bagian *fold* dataset, lalu kemudian dilakukan pengujian dan dilanjutkan pada *fold* berikutnya. Hasil metrik yang didapatkan dari pengujian yang memanfaatkan masing-masing *fold* dataset kemudian dirata-rata dan dibandingkan. Nilai k yang digunakan dalam penelitian ini adalah 10. Hal ini didasarkan pada penelitian oleh Wong (2015) yang menyatakan bahwa memanfaatkan *k-fold cross validation* dalam melakukan estimasi kemampuan model untuk melakukan prediksi atau klasifikasi dengan nilai k sebesar 10 direkomendasikan untuk dapat mengetahui seberapa stabil kemampuan model dalam melakukan klasifikasi.

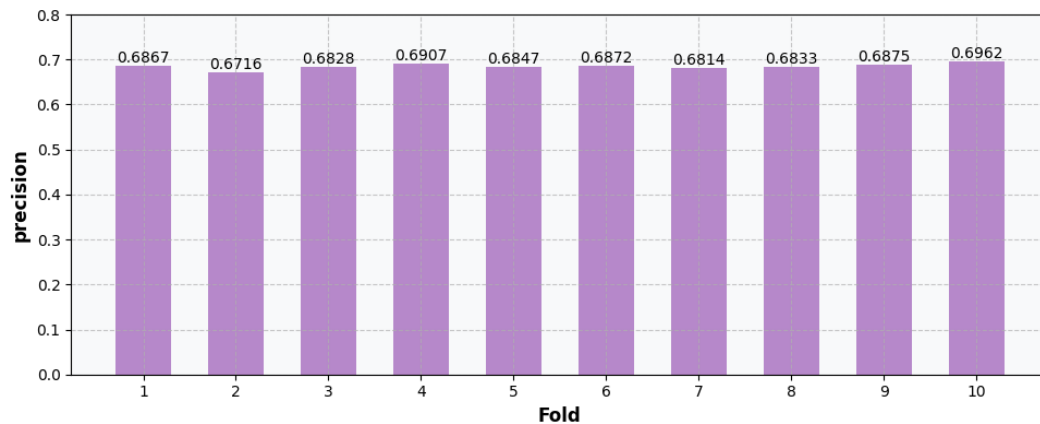
Untuk mempermudah proses perbandingan hasil *k-fold cross validation* pada model B2, selanjutnya dilakukan visualisasi untuk hasil dari masing-masing metrik. Untuk metrik yang pertama adalah metrik *accuracy* yang ditampilkan pada Gambar 9.12.



Gambar 9.12. Hasil *k-fold cross validation* model B2 pada metrik *accuracy* dengan nilai $k=10$

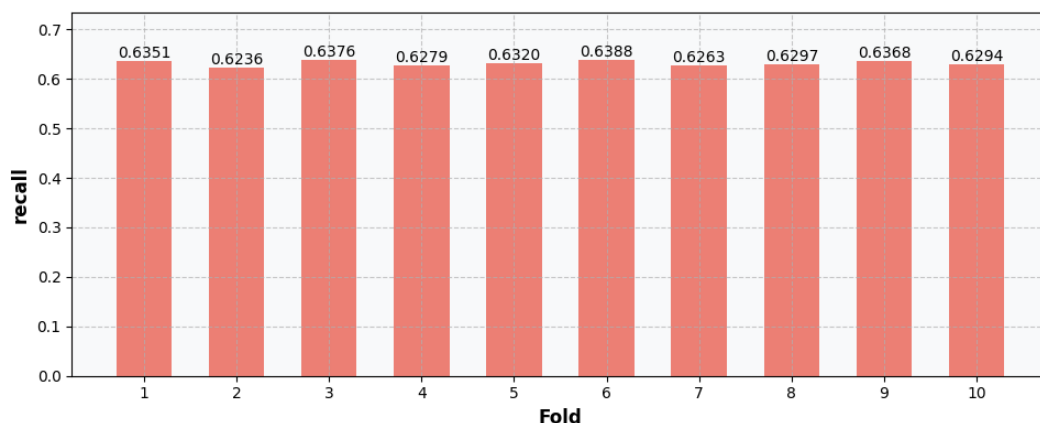
Hasil *k-fold cross validation* metrik *accuracy* dari model B2 dengan nilai $k=10$ pada Gambar 9.12 memiliki rata-rata nilai *accuracy* sebesar 63.17%. Nilai *accuracy* terendah didapatkan pada *fold* 2 dan 7 yang sama-sama memiliki nilai

accuracy sebesar 62.63%. Kemudian untuk nilai *accuracy* tertinggi didapatkan pada *fold* 6 dengan nilai *accuracy* sebesar 63.88%. Kemudian untuk nilai standar deviasi yang didapatkan adalah 0.0052. Lalu selanjutnya adalah perbandingan nilai metrik *precision* pada masing-masing *fold* yang ditampilkan pada Gambar 9.13.



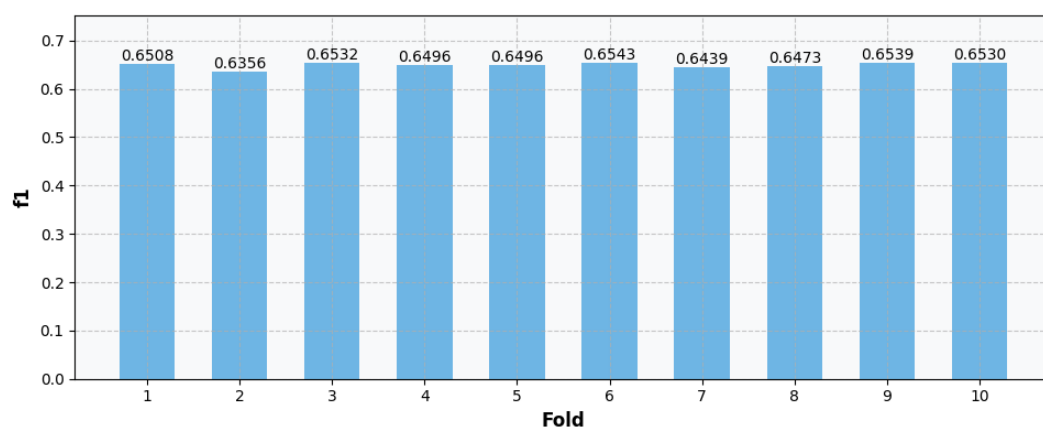
Gambar 9.13. Hasil *k-fold cross validation* model B2 pada metrik *precision* dengan nilai $k=10$

Hasil *k-fold cross validation* metrik *precision* dari model B2 dengan nilai $k=10$ pada Gambar 9.13 memiliki rata-rata nilai *precision* sebesar 68.52%. Nilai *precision* terendah didapatkan pada *fold* 2 dengan nilai *precision* sebesar 67.16%. Kemudian untuk nilai *precision* tertinggi didapatkan pada *fold* 10 dengan nilai *precision* sebesar 69.62%. Kemudian untuk nilai standar deviasi yang didapatkan adalah 0.0064. Lalu selanjutnya adalah perbandingan nilai metrik *recall* pada masing-masing *fold* yang ditampilkan pada Gambar 9.14.



Gambar 9.14. Hasil *k-fold cross validation* model B2 pada metrik *recall* dengan nilai $k=10$

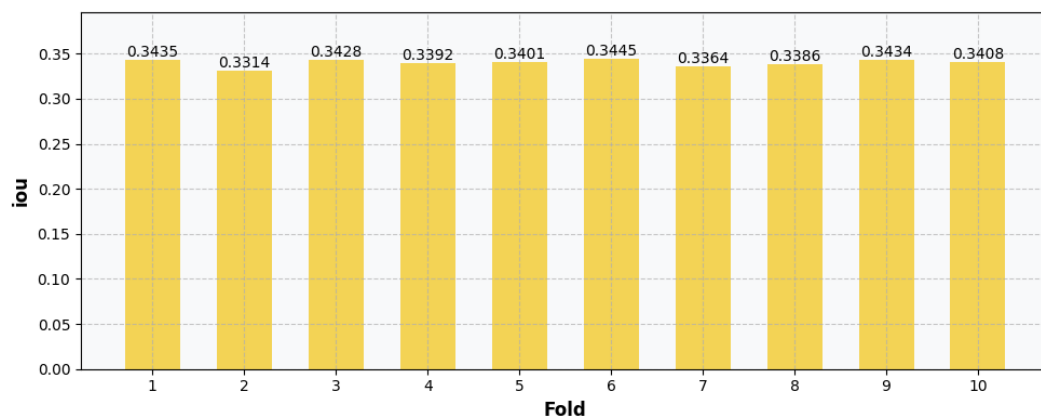
Hasil *k-fold cross validation* metrik *recall* dari model B2 dengan nilai $k=10$ pada Gambar 9.14 memiliki rata-rata nilai *recall* sebesar 63.17%. Nilai *recall* terendah didapatkan pada *fold* 2 dengan nilai *recall* sebesar 62.36%. Kemudian untuk nilai *recall* tertinggi didapatkan pada *fold* 6 dengan nilai *recall* sebesar 63.88%. Kemudian untuk nilai standar deviasi yang didapatkan adalah 0.0052. Lalu selanjutnya adalah perbandingan nilai metrik *F1-Score* pada masing-masing *fold* yang ditampilkan pada Gambar 9.15.



Gambar 9.15. Hasil *k-fold cross validation* model B2 pada metrik *F1-Score* dengan nilai $k=10$

Hasil *k-fold cross validation* metrik *F1-Score* dari model B2 dengan nilai $k=10$ pada Gambar 9.15 memiliki rata-rata nilai *F1-Score* sebesar 64.91%. Nilai

F1-Score terendah didapatkan pada *fold* 2 dengan nilai *recall* sebesar 63.56%. Kemudian untuk nilai *F1-Score* tertinggi didapatkan pada *fold* 6 dengan nilai *F1-Score* sebesar 65.43%. Kemudian untuk nilai standar deviasi yang didapatkan adalah 0.0058. Terakhir adalah perbandingan nilai metrik IoU pada masing-masing *fold* yang ditampilkan pada Gambar 9.16.



Gambar 9.16. Hasil *k-fold cross validation* model B2 pada metrik IoU dengan nilai $k=10$

Hasil *k-fold cross validation* metrik IoU dari model B2 dengan nilai $k=10$ pada Gambar 9.16 memiliki rata-rata nilai IoU sebesar 34.01%. Nilai IoU terendah didapatkan pada *fold* 2 dengan nilai *recall* sebesar 33.14%. Kemudian untuk nilai IoU tertinggi didapatkan pada *fold* 6 dengan nilai IoU sebesar 34.45%. Kemudian untuk nilai standar deviasi yang didapatkan adalah 0.0040. Berdasarkan hasil evaluasi yang dilakukan pada model B2 menggunakan *k-fold cross validation* didapatkan bahwa model B2 menunjukkan performa yang stabil pada seluruh distribusi data.

Tujuan dari penelitian ini adalah untuk mengembangkan model *machine learning* yang dapat melakukan klasifikasi tutupan lahan. Meskipun model yang dibangun melalui berbagai skenario pengujian telah mampu mengklasifikasikan

berbagai jenis tutupan lahan, masih terdapat banyak keterbatasan dan kekurangan seperti tingkat performa model yang diukur menggunakan berbagai metrik belum maksimal. Selain itu skenario pengujian pada penelitian ini juga masih terbatas, sehingga masih membutuhkan pengembangan dan penelitian lebih lanjut agar model yang dibangun dapat memberikan hasil yang lebih optimal. Upaya untuk dapat mendapatkan hasil terbaik merupakan hal yang sesuai dengan firman Allah yang berbunyi:

وَأَنْ لَّيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ﴿٣٩﴾

“Bahwa manusia hanya memperoleh apa yang telah diusahakannya” (QS. An-Najm [62]: 39).

Syaikh Abdurrahman bin Nashir as-Sa’di dalam tafsir as-Sa’di karyanya menyatakan bahwa makna dari QS. An-Najm [62]:39 menunjukkan bahwa seseorang tidaklah mendapatkan selain yang ia kerjakan sendiri. Karena di dalam ayat ini Allah Swt. menjanjikan balasan bagi orang yang berusaha dengan maksimal, untuk itu setiap pribadi dari kita harus berusaha keras untuk mendapatkan hasil sebaik mungkin (Nabila *et al.*, 2021).

Hasil terbaik yang didapatkan dari penelitian ini bertujuan untuk mempermudah proses klasifikasi tutupan lahan dan membantu ikhtiar manusia untuk menjaga dan merawat lingkungan. Selain itu juga dapat membantu instansi terkait untuk melakukan pengawasan terhadap perkembangan area tutupan lahan dengan cepat dan efisien. Perbuatan saling membantu atau tolong-menolong ini berkaitan dengan firman Allah berikut ini:

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ ﴿٢٠﴾

“Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.” (QS. Al-Maidah [5]: 2).

Mengutip dari Tafsir Almaraghi, Surat Al-Maidah ayat 2 ini dapat dipahami sebagai perintah untuk saling membantu dalam melakukan kebaikan dan ketakwaan, yang mana ini merupakan salah satu ajaran sosial dalam Al-Qur'an. Allah mewajibkan manusia untuk saling menolong dalam melakukan segala sesuatu yang bermanfaat bagi umat manusia, baik secara individu maupun kelompok, dalam urusan agama maupun dunia. Karena dengan bekerja sama dalam ketakwaan, manusia diharapkan dapat mencegah kerusakan dan bahaya yang mengancam keselamatan mereka (Puspitasari, 2022).

Lebih dalam mengenai tolong-menolong antar sesama manusia, diriwayatkan daripada Abu Hurairah ra. Daripada Rasulullah SAW. Baginda bersabda yang artinya:

“Sesiapa yang meringankan daripada seorang mukmin satu kesusahan daripada kesusahan-kesusahan dunia, Allah akan meringankan daripadanya satu kesusahan daripada kesusahan-kesusahan Hari Kiamat. Sesiapa yang memudahkan kepada orang yang kesempitan, Allah akan memudahkan kepadanya di dunia dan akhirat. Sesiapa yang menutup (kebaikan) seorang Muslim, Allah akan menutup keaibannya di dunia dan akhirat. Allah senantiasa menolong seseorang hamba selama mana hamba itu menolong saudaranya. Sesiapa yang melalui satu jalan untuk mencari ilmu, Allah akan memudahkan baginya satu jalan ke surga. Sesuatu kaum (kelompok manusia) tidaklah berkumpul di satu rumah (masjid) daripada rumah-rumah Allah, sambil membaca Al-Quran, saling memperbetulkan bacaan dan memahami kandungannya, kecuali as-Sakinah akan turun kepada mereka dan mereka diselubungi oleh rahmat. Mereka juga dilingkungi oleh para malaikat serta disebut dan dipuji oleh Allah di kalangan (para malaikat) yang ada di sisi-Nya. Sesiapa yang diperlambat oleh amalannya (karena amalannya sedikit), dia tidak akan dipercepatkan oleh keturunan (keturunannya tidak akan mengangkat derajatnya)” (HR. Muslim).

Kalimat *“Allah senantiasa menolong hamba-Nya selama hamba-Nya itu suka menolong saudaranya”* pada hadis diatas bermakna bahwa jika seseorang berkeinginan membantu saudaranya, maka sudah sepatutnya untuk dikerjakan dalam bentuk kata ataupun pembelaan atas kebenaran yang didasari dengan iman kepada Allah. Apabila seorang Muslim memberi bantuan kepada saudara Muslim lainnya dengan ikhlas, maka balasan terbaik akan diberikan oleh Allah untuk hambanya, yaitu dilepaskan seorang Muslim tersebut dari kesulitan pada hari Kiamat kelak. Karena itulah sangat dianjurkan bagi seorang Muslim untuk membantu sesama saudara Muslim (Alfian, 2021).

Perbuatan saling tolong menolong dalam hal menjaga dan merawat lingkungan ini diharapkan dapat membantu kita terhindar dari firman Allah yang berjanji akan membuat manusia merasakan akibat dari perbuatan kerusakan yang dilakukan dengan tangannya. Hal ini terdapat dalam firman Allah berikut:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ ﴿٤١﴾

“Telah tampak kerusakan di darat dan di laut disebabkan perbuatan tangan manusia. (Melalui hal itu) Allah membuat mereka merasakan sebagian dari (akibat) perbuatan mereka agar mereka kembali (ke jalan yang benar)” (QS. Ar-Rum [30]: 41).

Mengutip dari Tafsir Al-Azhar, QS. Ar-Rum (30) ayat 41 ini menjelaskan bahwa kerusakan yang terjadi di muka bumi adalah akibat dari ulah manusia. Dalam tafsir ini juga menganggap jika perkembangan teknologi saat ini adalah baik untuk perkembangan pembangunan. Jika manusia tidak mengikuti perkembangan teknologi dan pengetahuan yang ada di zamannya, maka manusia tersebut akan semakin tertinggal dan sengsara. Karena sejatinya tujuan Allah mengatur manusia

tinggal di bumi ini untuk menjadi khalifah. Khalifah disini merupakan pelayan dan wakil Allah untuk menjaga apa yang ada di bumi, untuk itu sudah seharusnya manusia terutama umat Muslim bisa memperbaiki, menjaga, dan merawat apa yang ada di bumi Allah, terutama lingkungan hidup (Rizkiyah dan Erwanto, 2023). Secara lebih terperinci Hatami (2025) menjelaskan bahwa terdapat 4 nilai yang dapat diterapkan oleh manusia sebagai khalifah di bumi dengan merujuk pada QS. Al-Qashash ayat 77:

وَابْتَغِ فِيمَا آتَاكَ اللَّهُ الدَّارَ الْآخِرَةَ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ وَلَا تَبْغِ الْفُسَادَ فِي الْأَرْضِ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ ﴿٧٧﴾

“Dan, carilah pada apa yang telah dianugerahkan Allah kepadamu (pahala) negeri akhirat, tetapi janganlah kamu lupakan bagianmu di dunia. Berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik kepadamu dan janganlah kamu berbuat kerusakan di bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.”

Nilai pertama didapatkan dari tafsir Ibnu Katsir, pada potongan ayat *“Dan, carilah pada apa yang telah dianugerahkan Allah kepadamu (pahala) negeri akhirat...”* bermakna agar manusia dengan terkhusus umat Muslim dapat memanfaatkan semua fasilitas baik sumber daya alam, ilmu, teknologi, harta, dan semua hal lainnya yang diberikan oleh Allah di dunia ini untuk beribadah dan mendekatkan diri kepada Allah dengan tujuan akhirat. Nilai kedua masih dalam tafsir Ibnu Katsir, potongan berikutnya dari ayat yang sama *“... tetapi janganlah kamu lupakan bagianmu di dunia...”* menjelaskan mengenai betapa pentingnya seorang muslim memahami bahwa kehidupan dunia ini merupakan titik awal untuk dapat menggapai kehidupan akhirat, sehingga sangat penting untuk menjaga keseimbangan kehidupan dunia dan akhirat. Nilai ketiga juga didapatkan dari tafsir

Ibnu Katsir, pada potongan berikutnya “... *Berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik kepadamu ...*” mengajarkan bahwa manusia yang secara terkhusus adalah umat Muslim dapat berbuat baik kepada sesamanya dengan tidak terbatas pada harta dan materi saja, namun juga seperti sikap, tindakan, serta perkataan. Artinya seorang Muslim, selain harus cerdas secara keilmuan, juga harus mengedepankan karakter dan akhlak yang mulia. Lalu pada potongan ayat terakhir “... *dan janganlah kamu berbuat kerusakan di bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.*”, merujuk pada tafsir Ibnu Katsir, potongan ayat ini merupakan pengingat agar manusia yang secara terkhusus adalah umat muslim agar tidak berbuat kerusakan di bumi Allah. Kerusakan ini dapat berupa kerusakan lingkungan ataupun zalim dan merugikan orang lain. Untuk itu Hatami (2025) menekankan dalam penelitiannya yang membahas mengenai tafsir QS. Al-Qashash ayat 77 dengan salah satunya merujuk pada tafsir Ibnu Katsir, bahwasannya manusia hidup didunia harus selalu mendorong dirinya untuk menggunakan akhirat sebagai fokus utama, dan dunia adalah sebagai perantaranya. Sehingga dalam kehidupan sehari-hari seorang Muslim dapat selalu berusaha untuk terus berbuat kebaikan dan memberikan manfaat pada sekitarnya, baik dari segi ilmu pengetahuan, melestarikan dan menjaga lingkungan, serta dari tindakan kebaikan dan membantu sesama manusia.

BAB X

PENUTUP

10.1 Kesimpulan

Merujuk kembali pada pembahasan kerangka konsep dalam penelitian ini yang mengusulkan bahwa penggunaan GLCM 6 fitur yang kemudian dipelajari oleh model NN, baik yang memanfaatkan satu hingga lima *hidden layer* belum diimplementasikan pada penelitian-penelitian sebelumnya. Dari pengujian yang dilakukan pada berbagai skenario model, didapatkan beberapa kesimpulan:

- a. Penambahan jumlah fitur GLCM yang awalnya berjumlah 5 fitur menjadi 6 fitur dapat meningkatkan kinerja NN dalam melakukan klasifikasi tutupan lahan. Hal ini dapat dilihat dari peningkatan performa model NN pada semua jenis metrik, baik pada *accuracy*, *precision*, *recall*, *F1-Score*, dan IoU yang divisualisasikan dalam Gambar 9.1 hingga Gambar 9.5.
- b. Menambahkan jumlah *hidden layer* tidak selalu menaikkan performa kinerja NN dalam melakukan klasifikasi tutupan lahan. Setelah dilakukan penambahan *layer* pada model NN satu *hidden layer* menjadi dua *hidden layer*, model NN mengalami peningkatan performa pada semua metrik. Namun setelah ditambahkan lagi *layer* pada model NN tiga *hidden layer* hingga lima *hidden layer*, performa model NN dalam melakukan klasifikasi tutupan lahan mengalami penurunan pada semua metrik. Dapat disimpulkan bahwa penambahan *layer* pada model NN tidak selalu meningkatkan performa, dan justru berpotensi menimbulkan *overfitting* pada model jika terlalu banyak

menambahkan *layer* pada arsitektur NN. Hasil dari perbandingan tersebut divisualisasikan dalam Gambar 9.6 hingga Gambar 9.10.

- c. Model B2 dengan kombinasi dua *hidden layer* dan enam fitur GLCM memiliki performa terbaik dibandingkan model lainnya dalam mengklasifikasikan tutupan lahan. Model B2 mendapatkan performa pada metrik *accuracy* sebesar 87.60%, *precision* sebesar 63.37%, *recall* sebesar 63.40%, *F1-Score* sebesar 63.39%, dan IoU sebesar 48.22%.

10.2 Saran

Berdasarkan hasil performa model B2 dalam melakukan klasifikasi tutupan lahan, terdapat beberapa saran yang dapat menjadi pertimbangan pada penelitian berikutnya:

- a. Melakukan eksplorasi fitur yang digunakan untuk melakukan *training* terhadap model NN secara lebih luas. Seperti dengan menambahkan jenis fitur GLCM lainnya, atau menggunakan metode ekstraksi fitur selain GLCM.
- b. Melakukan pengujian dan eksplorasi terhadap dataset citra beresolusi tinggi untuk mengetahui kemampuan model NN dalam melakukan klasifikasi area tutupan lahan pada dataset selain dari ISPRS.
- c. Melakukan eksplorasi teknik selain dari teknik untuk menentukan jumlah *hidden layer* dan jumlah *node* didalamnya yang digunakan pada penelitian ini.
- d. Menambah jenis metode yang digunakan untuk melakukan klasifikasi area tutupan lahan sebagai pembanding dengan metode NN yang diangkat pada penelitian ini.

DAFTAR PUSTAKA

- Abbass, K., Qasim, M. Z., Song, H., Murshed, M., Mahmood, H., & Younis, I. (2022). A review of the global climate change impacts, adaptation, and sustainable mitigation measures. In *Environmental Science and Pollution Research* (Vol. 29, Issue 28, pp. 42539–42559). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1007/s11356-022-19718-6>
- Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., & He, Y. (2019). Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. *Computers and Electronics in Agriculture*, 167. <https://doi.org/10.1016/j.compag.2019.105091>
- Achmad Gusyairi. (2022). *Manusia Dalam Perspektif Al-Qur'an (Mengungkap Makna Konotatif Lafadz Al-Insan Secara Psikologis)*. Institut Perguruan Tinggi Ilmu Al-Qur'an Jakarta.
- Adugna, T., Xu, W., & Fan, J. (2022). Comparison of Random Forest and Support Vector Machine Classifiers for Regional Land Cover Mapping Using Coarse Resolution FY-3C Images. *Remote Sensing*, 14(3). <https://doi.org/10.3390/rs14030574>
- Alfian, F. (2021). *Hadis Ke-36 Al-Arba'in: Menunaikan Kebutuhan Sesama Muslim*. <http://markazsunnah.com/hadis-ke-36-al-arbain-menunaikan-kebutuhan-sesama-muslim/>
- Basheer, S., Wang, X., Farooque, A. A., Nawaz, R. A., Liu, K., Adekanmbi, T., & Liu, S. (2022). Comparison of Land Use Land Cover Classifiers Using Different Satellite Imagery and Machine Learning Techniques. *Remote Sensing*, 14(19). <https://doi.org/10.3390/rs14194978>
- Civco, D. L. (1993). Artificial neural networks for land-cover classification and mapping. *International Journal of Geographical Information Systems*, 7(2), 173–186. <https://doi.org/10.1080/02693799308901949>
- Eldosouky, A. M., El-Qassas, R. A. Y., Pour, A. B., Mohamed, H., & Sekandari, M. (2021). Integration of ASTER satellite imagery and 3D inversion of aeromagnetic data for deep mineral exploration. *Advances in Space Research*, 68(9), 3641–3662. <https://doi.org/10.1016/j.asr.2021.07.016>
- Equere, E. B., Ibem, E. O., & Alagbe, O. A. (2021). Impact of rapid population growth on public housing schemes and the influence on city resilience. *IOP Conference Series: Earth and Environmental Science*, 665(1). <https://doi.org/10.1088/1755-1315/665/1/012006>
- Eyster, H. N., & Beckage, B. (2024). Applying a deep learning pipeline to classify land cover from low-quality historical RGB imagery. *PeerJ Computer*

- Science*, 10. <https://doi.org/10.7717/peerj-cs.2003>
- Hatami, M. H. (2025). Pendidikan Qur ' ani : Kajian Tafsir QS . Al-Qashash Ayat 77 terhadap Nilai dan Prinsip. *Al-Fahmu: Jurnal Ilmu Al-Qur'an Dan Tafsir*, 4(1), 1–13. <https://doi.org/10.58363/alfahmu.v4i1.260>
- Isaac, C., & Zareinia, K. (2022). Effect of excessive neural network layers on overfitting. *World Journal of Advanced Research and Reviews*, 16(2), 1246–1257. <https://doi.org/10.30574/wjarr.2022.16.2.1247>
- Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining*, 15(4), 531–538. <https://doi.org/10.1002/sam.11583>
- Jozdani, S. E., Johnson, B. A., & Chen, D. (2019). Comparing deep neural networks, ensemble classifiers, and support vector machine algorithms for object-based urban land use/land cover classification. *Remote Sensing*, 11(14). <https://doi.org/10.3390/rs11141713>
- Koch, D., Despotovic, M., Thaler, S., & Zeppelzauer, M. (2021). Where do university graduates live? – A computer vision approach using satellite images. *Applied Intelligence*, 51(11), 8088–8105. <https://doi.org/10.1007/s10489-021-02268-8>
- Kumar, G., & Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. *International Conference on Advanced Computing and Communication Technologies, ACCT*, 5–12. <https://doi.org/10.1109/ACCT.2014.74>
- Lan, Z., & Liu, Y. (2018). Study on multi-scale window determination for GLCM texture description in high-resolution remote sensing image geo-analysis supported by GIS and domain knowledge. *ISPRS International Journal of Geo-Information*, 7(5). <https://doi.org/10.3390/ijgi7050175>
- Lawrence, D., Coe, M., Walker, W., Verchot, L., & Vandekar, K. (2022). The Unseen Effects of Deforestation: Biophysical Effects on Climate. *Frontiers in Forests and Global Change*, 5, 1–13. <https://doi.org/10.3389/ffgc.2022.756115>
- Li, H., Qiu, K., Chen, L., Mei, X., Hong, L., & Tao, C. (2021). SCAttNet: Semantic Segmentation Network with Spatial and Channel Attention Mechanism for High-Resolution Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters*, 18(5), 905–909. <https://doi.org/10.1109/LGRS.2020.2988294>
- Liu, J., Zhu, Y., Song, L., Su, X., Li, J., Zheng, J., Zhu, X., Ren, L., Wang, W., & Li, X. (2023). Optimizing window size and directional parameters of GLCM texture features for estimating rice AGB based on UAVs multispectral imagery. *Frontiers in Plant Science*, 14. <https://doi.org/10.3389/fpls.2023.1284235>
- Meng, Z., Zhao, F., & Liang, M. (2021). Ss-mlp: A novel spectral-spatial mlp

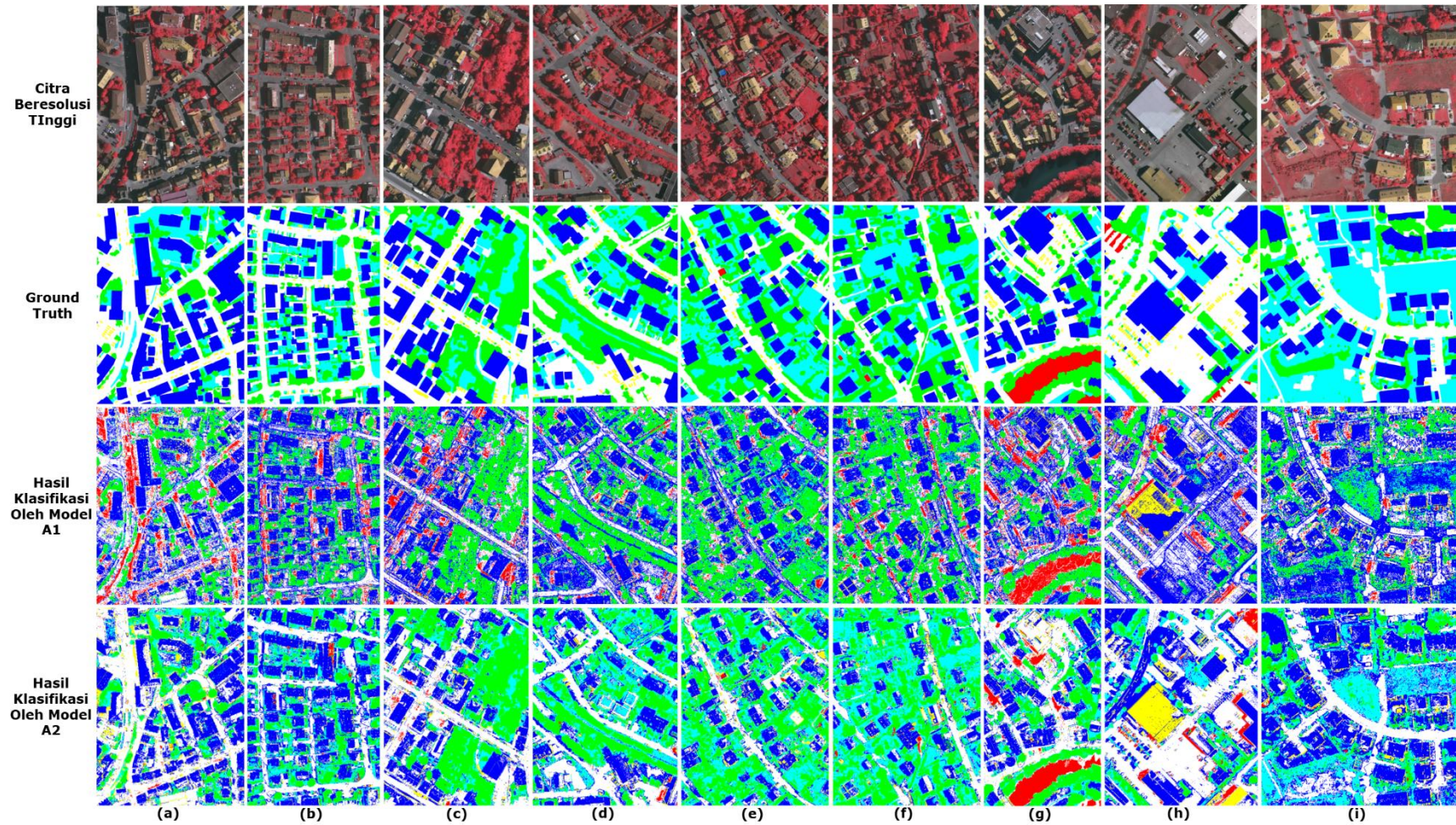
- architecture for hyperspectral image classification. *Remote Sensing*, 13(20). <https://doi.org/10.3390/rs13204060>
- Muhammad Mu'tiq Rosyadi. (2020). *Hak-Hak Alam Semesta Dalam Q.S. Al-A'raf [7]: 56-58 (Analisis Terhadap Tafsir Al-Misbah)* [Institut Agama Islam Nahdlatul Ulama (IAINU) Kebumen]. <http://www.iainukebumen.ac.id>
- Mujetahid, A., Nursaputra, M., & Soma, A. S. (2023). Monitoring Illegal Logging Using Google Earth Engine in Sulawesi Selatan Tropical Forest, Indonesia. *Forests*, 14(3). <https://doi.org/10.3390/f14030652>
- Mulder, C. (1996). Population and housing. *Population Research and Policy Review*, 15(5–6), 419–435. <https://doi.org/10.4054/demres.2006.15.13>
- Nabila, A., Sari Dewi, M., Damanik, S., & Sumatera Utara Medan, U. (2021). TAFSIR AYAT-AYAT TENTANG MOTIVASI KERJA. In *Cybernetics: Journal Educational Research and Social Studies* (Vol. 2, Issue 1). <http://pusdikra-publishing.com/index.php/jrss>
- Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1), 217–222. <https://doi.org/10.1080/01431160412331269698>
- Praseptiyana, W. I., Widodo, A. W., & Rahman, M. A. (2019). Pemanfaatan Ciri Gray Level Co-occurrence Matrix (GLCM) Untuk Deteksi Melasma Pada Citra Wajah. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(11), 10402–10409. <http://j-ptiik.ub.ac.id>
- Prawesti, A. D., Wardhani, P. I., & Surachman, D. (2024). The Impact of Population Growth on Land Covers in Tegal Regency, Indonesia in 2015 and 2023. *IOP Conference Series: Earth and Environmental Science*, 1357(1). <https://doi.org/10.1088/1755-1315/1357/1/012018>
- Puspitasari, M. (2022). KERJASAMA DALAM LEMBAGA PENDIDIKAN BERDASARKAN TAFSIR AL-QUR'AN SURAT AL-MAIDAH AYAT 2. *Jurnal Inovasi Penelitian Pendidikan Dan Pembelajaran*, 2, 209–221.
- Rachman, F., Huang, J., Xue, X., & Marfai, M. A. (2024). Insights from 30 Years of Land Use/Land Cover Transitions in Jakarta, Indonesia, via Intensity Analysis. *Land*, 13(4). <https://doi.org/10.3390/land13040545>
- Rajkumar, M., Nagarajan, S., & Dewitt, P. (2022). LAND USE LAND COVER MAPPING USING UAS IMAGERY: SCENE CLASSIFICATION and SEMANTIC SEGMENTATION. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 46(M-2–2022), 177–184. <https://doi.org/10.5194/isprs-Archives-XLVI-M-2-2022-177-2022>
- Ramadhan, C., Dina, R., & Nurjani, E. (2023). Spatial and temporal based deforestation proclivity analysis on flood events with applying watershed scale (case study: Lasolo watershed in Southeast Sulawesi, Central Sulawesi,

- and South Sulawesi, Indonesia). *International Journal of Disaster Risk Reduction*, 93. <https://doi.org/10.1016/j.ijdrr.2023.103745>
- Ritchie, H., Rod s-Guirao, L., Mathieu, E., Gerber Marcel, Ortiz-Ospina, E., Hasell, J., & Roser, M. (2023). *Population Growth*. Our World in Data. <https://ourworldindata.org/population-growth>
- Rizkiyah, Z. A., & Erwanto, D. (2023). MENGHADAPI FENOMENA KERUSAKAN DI MUKA BUMI (KAJIAN LAFADZ FASAD DALAM Q.S AR-RUM: 41). *MUSHAF JOURNAL : Jurnal Ilmu Al Quran Dan Hadis*, 3(2), 218–228.
- Santoso, I. B., Utama, S. N., & Supriyono. (2025). Meta-learning based softmax average of convolutional neural networks using multi-layer perceptron for brain tumour classification. *Array*, 26. <https://doi.org/10.1016/j.array.2025.100398>
- Seyam, M. M. H., Haque, M. R., & Rahman, M. M. (2023). Identifying the land use land cover (LULC) changes using remote sensing and GIS approach: A case study at Bhaluka in Mymensingh, Bangladesh. *Case Studies in Chemical and Environmental Engineering*, 7. <https://doi.org/10.1016/j.cscee.2022.100293>
- Si, Z., Yang, M., Yu, Y., & Ding, T. (2021). Photovoltaic power forecast based on satellite images considering effects of solar position. *Applied Energy*, 302. <https://doi.org/10.1016/j.apenergy.2021.117514>
- Su, Z., Li, W., Ma, Z., & Gao, R. (2022). An improved U-Net method for the semantic segmentation of remote sensing images. *Applied Intelligence*, 52(3), 3276–3288. <https://doi.org/10.1007/s10489-021-02542-9>
- Timilsina, S., Aryal, J., & Kirkpatrick, J. B. (2020). Mapping urban tree cover changes using object-based convolution neural network (OB-CNN). *Remote Sensing*, 12(18). <https://doi.org/10.3390/RS12183017>
- Wafiq, A. N., & Suryanto, S. (2021). The Impact of Population Density and Economic Growth on Environmental Quality: Study in Indonesia. *Jurnal Ekonomi & Studi Pembangunan*, 22(2), 301–312. <https://doi.org/10.18196/jesp.v22i2.10533>
- Wang, Y., & Gu, M. (2024). Classification methods for hyperspectral remote sensing images with weak texture features. *Journal of Radiation Research and Applied Sciences*, 17(3), 101019. <https://doi.org/10.1016/j.jrras.2024.101019>
- Wargadinata, E. L. (2021). KOMPLEKSITAS HUBUNGAN KEPENDUDUKAN DAN LINGKUNGAN. *Jurnal Ilmu Pemerintahan Widya Praja*, 47(1), 1–23. <https://doi.org/10.33701/jipwp.v47i1.1456>
- Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9), 2839–2846.

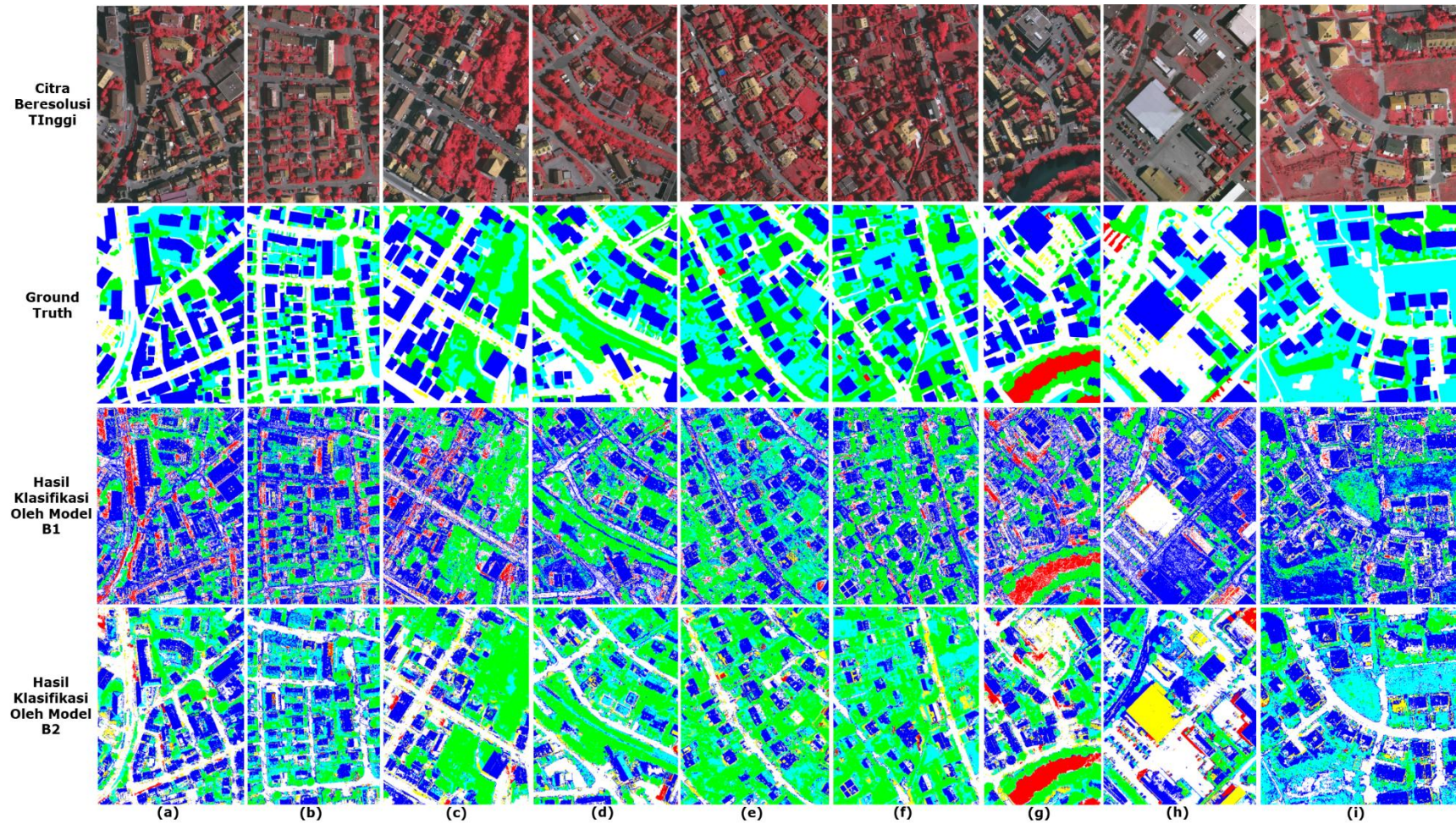
<https://doi.org/10.1016/j.patcog.2015.03.009>

- Xu, L., Thyagarajan, K. K., & Alshari, E. A. (2023). Classification of land use/land cover using artificial intelligence (ANN-RF). *Frontiers in Artificial Intelligence*, 5, 1–15. <https://doi.org/10.3389/frai.2022.964279>
- Yang, X., Fan, X., Peng, M., Guan, Q., & Tang, L. (2022). Semantic segmentation for remote sensing images based on an AD-HRNet model. *International Journal of Digital Earth*, 15(1), 2376–2399. <https://doi.org/10.1080/17538947.2022.2159080>
- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168(2). <https://doi.org/10.1088/1742-6596/1168/2/022022>
- Yu, J., Zeng, P., Yu, Y., Yu, H., Huang, L., & Zhou, D. (2022). A Combined Convolutional Neural Network for Urban Land-Use Classification with GIS Data. *Remote Sensing*, 14(5). <https://doi.org/10.3390/rs14051128>
- Yuan, X., Shi, J., & Gu, L. (2021). A review of deep learning methods for semantic segmentation of remote sensing imagery. In *Expert Systems with Applications* (Vol. 169). Elsevier Ltd. <https://doi.org/10.1016/j.eswa.2020.114417>
- Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., & Atkinson, P. M. (2019). Joint Deep Learning for land cover and land use classification. *Remote Sensing of Environment*, 221, 173–187. <https://doi.org/10.1016/j.rse.2018.11.014>
- Zhang, R., Duan, K., You, S., Wang, F., & Tan, S. (2020). A novel remote sensing detection method for buildings damaged by earthquake based on multiscale adaptive multiple feature fusion. *Geomatics, Natural Hazards and Risk*, 11(1), 1912–1938. <https://doi.org/10.1080/19475705.2020.1818637>

LAMPIRAN

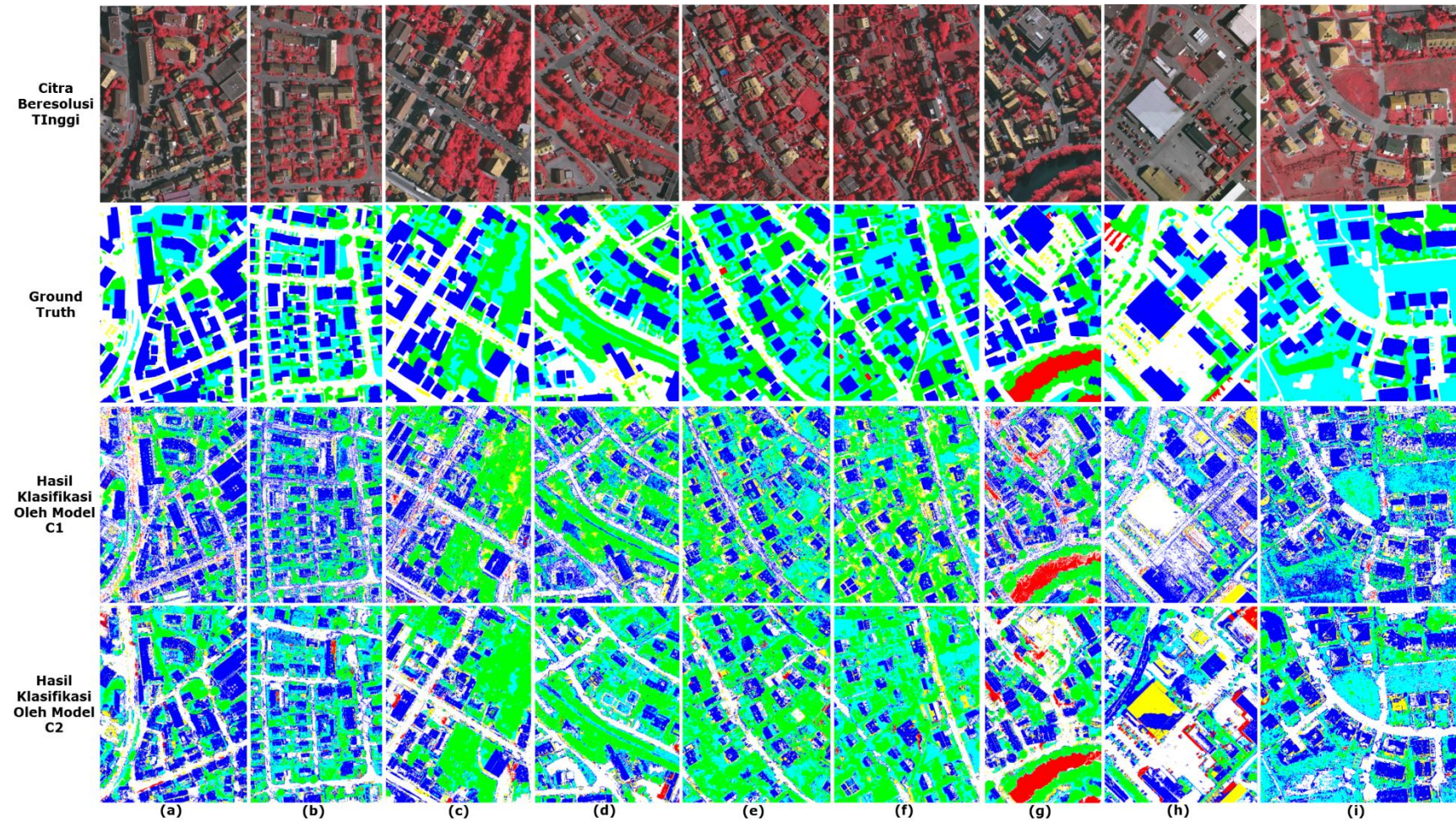


Hasil Klasifikasi Model NN A1 dan A2 Terhadap 9 Citra Beresolusi Tinggi
 (a) Area 1, (b) Area 3, (c) Area 7, (d) Area 11, (e) Area 15, (f) Area 23, (g) Area 26, (h) Area 32, (i) Area 37



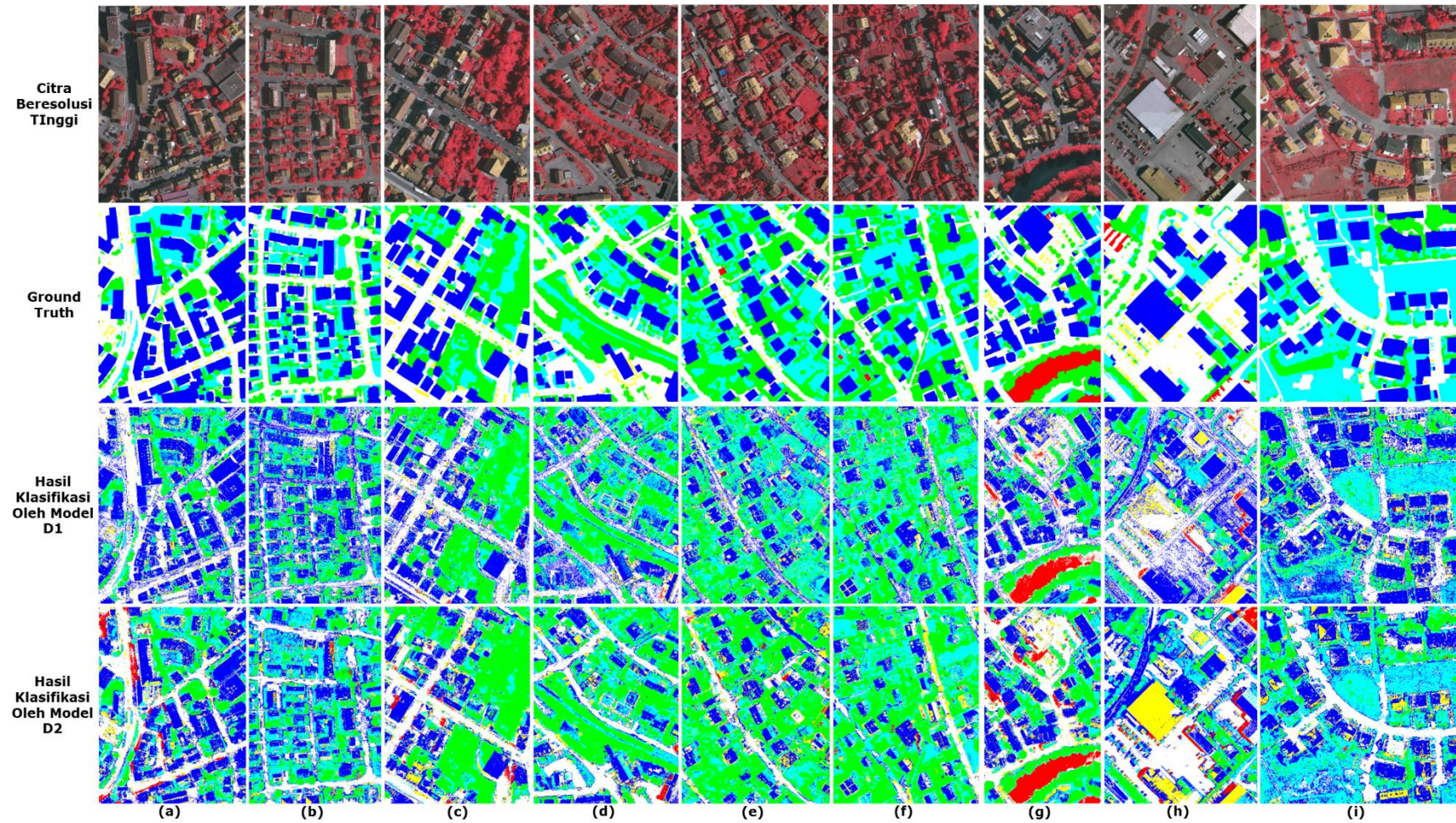
Hasil Klasifikasi Model NN B1 dan B2 Terhadap 9 Citra Beresolusi Tinggi

(a) Area 1, (b) Area 3, (c) Area 7, (d) Area 11, (e) Area 15, (f) Area 23, (g) Area 26, (h) Area 32, (i) Area 37



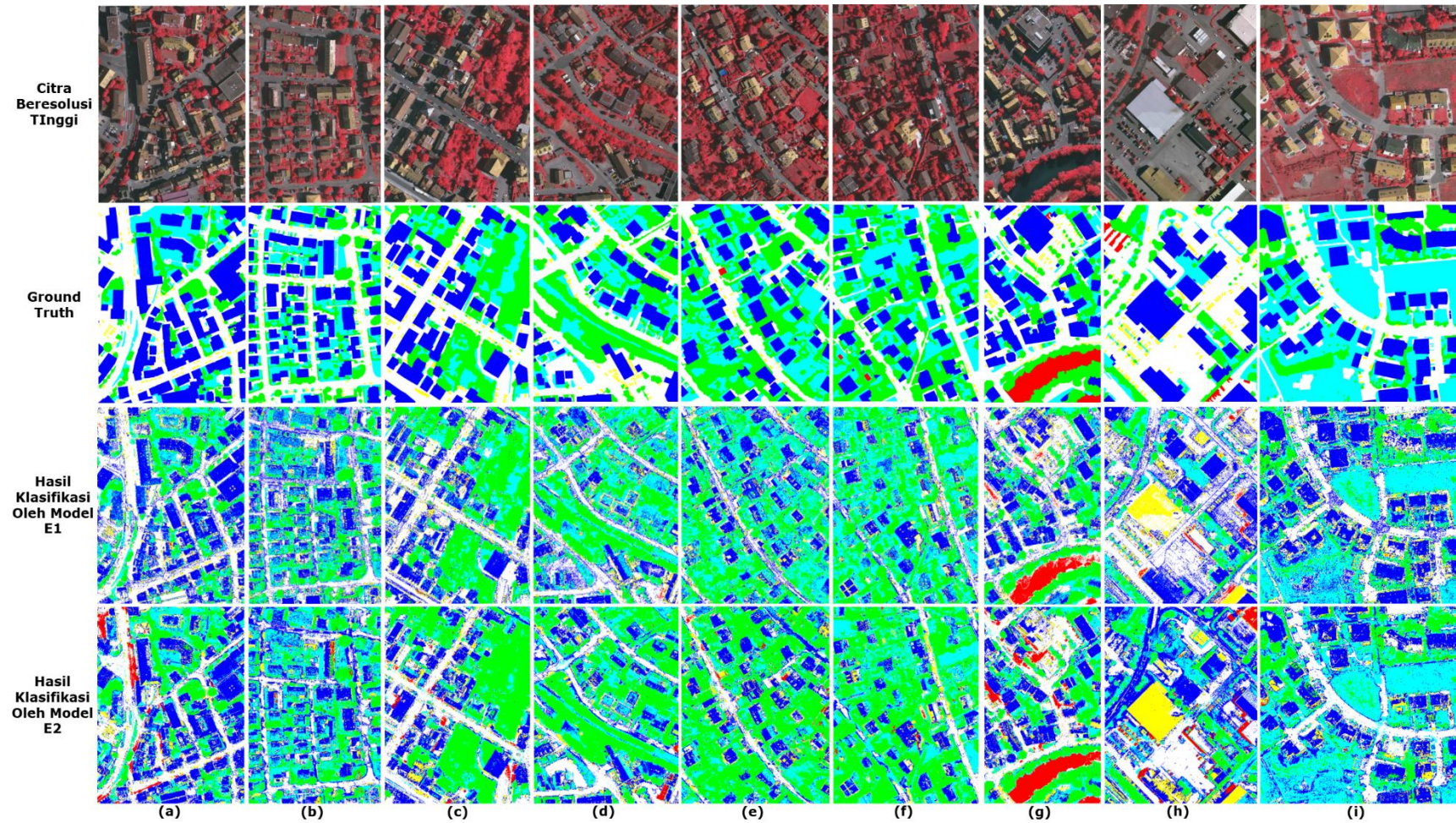
Hasil Klasifikasi Model NN C1 dan C2 Terhadap 9 Citra Beresolusi Tinggi

(a) Area 1, (b) Area 3, (c) Area 7, (d) Area 11, (e) Area 15, (f) Area 23, (g) Area 26, (h) Area 32, (i) Area 37



Hasil Klasifikasi Model NN D1 dan D2 Terhadap 9 Citra Beresolusi Tinggi

(a) Area 1, (b) Area 3, (c) Area 7, (d) Area 11, (e) Area 15, (f) Area 23, (g) Area 26, (h) Area 32, (i) Area 37



Hasil Klasifikasi Model NN E1 dan E2 Terhadap 9 Citra Beresolusi Tinggi

(a) Area 1, (b) Area 3, (c) Area 7, (d) Area 11, (e) Area 15, (f) Area 23, (g) Area 26, (h) Area 32, (i) Area 37

