

**KLASIFIKASI KEBAKARAN HUTAN BERBASIS CITRA MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK***

**SKRIPSI**

Oleh :  
**NURUL IZZAH**  
**NIM. 210605110005**



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

**KLASIFIKASI KEBAKARAN HUTAN BERBASIS CITRA  
MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

**SKRIPSI**

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

Oleh :  
**NURUL IZZAH**  
**NIM. 210605110005**

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

**HALAMAN PERSETUJUAN**

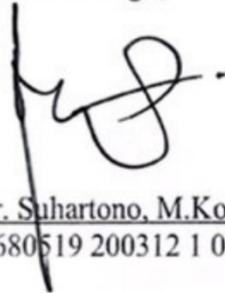
**KLASIFIKASI KEBAKARAN HUTAN BERBASIS CITRA  
MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

**SKRIPSI**

Oleh :  
**NURUL IZZAH**  
**NIM. 210605110005**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 17 Maret 2025

Pembimbing I,



Prof. Dr. Suhartono, M.Kom  
NIP. 19680519 200312 1 001

Pembimbing II,



Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

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



Dr. Ir. Fachrul Kurniawan, M.MT., IPU  
NIP. 19771020 200912 1 001

**HALAMAN PENGESAHAN**

**KLASIFIKASI KEBAKARAN HUTAN BERBASIS CITRA  
MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

**SKRIPSI**

Oleh :  
**NURUL IZZAH**  
**NIM. 210605110005**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 15 Mei 2025

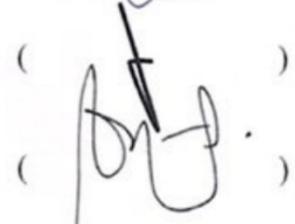
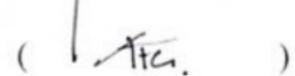
**Susunan Dewan Penguji**

Ketua Penguji : Dr. M. Amin Hariyadi, M.T  
NIP. 19670118 200501 1 001

Anggota Penguji I : Prof. Dr. Muhammad Faisal, M.T  
NIP. 19740510 200501 1 007

Anggota Penguji II : Prof. Dr. Suhartono, M.Kom  
NIP. 19680519 200312 1 001

Anggota Penguji III : Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

 )  
(  )  
(  )

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



  
**Dr. Ir. Fachrul Kurniawan, M.MT., IPU**  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Nurul Izzah  
NIM : 210605110005  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Klasifikasi Kebakaran Hutan Berbasis Citra  
Menggunakan *Convolutional Neural Network*

Menyatakan dengan sebenarnya bahwa Skripsi 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 skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 15 Mei 2025  
Yang membuat pernyataan,



Nurul Izzah  
NIM.210605110005

## **MOTTO**

*“You can't fall if you don't climb. But there's no joy in living your whole life on the ground.”*

*“Live a life you will remember.”*

*“No need to rush, you're doing fine. Just stay as you are.”*

*~ My my - Seventeen*

## HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT atas limpahan rahmat dan petunjuk-Nya, sehingga skripsi ini dapat diselesaikan dengan baik.

Skripsi ini penulis persembahkan untuk:

Orang tua dan ketiga adik penulis yang menjadi sumber kekuatan dan doa.

Alm. nenek tercinta yang selalu memberi dukungan semasa hidup.

Sahabat yang kebersamai dalam suka dan duka.

Temam seperjuangan selama perkuliahan dan penyusunan skripsi.

Para guru, dosen, dan pembimbing yang telah membagikan ilmu dan arah.

Semua pihak yang turut hadir memberi semangat, baik secara langsung maupun tidak.

Dan untuk diri sendiri, yang terus berjuang, bertumbuh, dan tidak menyerah meski dalam diam.

Semoga karya ini dapat menjadi langkah awal yang bermanfaat dalam perjalanan ilmu dan kehidupan.

## KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarakatuh.*

*Alhamdulillah rabbil 'alamin*, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan inayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Klasifikasi Kebakaran Hutan Berbasis Citra Menggunakan *Convolutional Neural Network*”. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh ilmu dan cahaya Islam.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam proses penyusunannya, penulis banyak menerima dukungan, doa, semangat, dan bantuan dari berbagai pihak yang sangat berarti. Oleh karena itu, dengan penuh rasa hormat dan tulus, penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Prof. Dr. Suhartono, M.Kom., selaku Dosen Pembimbing I dan Bapak Fatchurrohman, M.Kom., selaku Dosen Pembimbing II yang telah meluangkan waktu dan dengan sabar telah memberikan arahan, masukan, dan motivasi kepada penulis selama proses penyusunan skripsi ini sehingga penulis dapat menyelesaikan penelitian pada skripsi ini dengan baik.
5. Dr. M. Amin Hariyadi, M.T., selaku Ketua Penguji dan Prof. Dr. Muhammad Faisal, M.T., selaku Dosen Penguji I yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan skripsi dengan baik.
6. Segenap dosen serta jajaran staf Program Studi Teknik Informatika yang telah memberikan ilmu, bimbingan, dan inspirasi selama masa studi di Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Kedua orang tua tercinta, Abi Burhan dan Umi Azizah, yang selalu menjadi tempat pulang dan sumber kekuatan. Terima kasih atas doa yang selalu mengiringi setiap langkah, kepercayaan dan pengorbanan yang tak terhitung, serta dukungan yang menjadi sumber kekuatan di setiap langkah. Untuk adik-adik tersayang, Nanda, Umam, dan Raisya, atas kehadirannya yang menjadi penyemangat di tengah proses ini.
8. Almarhumah Omah, nenek tercinta, yang semasa hidupnya selalu menjadi tempat mengadu dan menceritakan keluh kesah. Meski kini telah tiada, kasih sayang dan keteladanannya tetap hidup dalam ingatan, menjadi bagian dari kekuatan yang menemani setiap langkah dalam menyelesaikan pendidikan ini.

9. Keluarga besar, terkhusus Mami Lia dan Papi Fauzi, yang selalu memberikan semangat, dukungan, dan doa dari jauh maupun dekat.
10. Rifdah, sahabat jauh yang tetap terasa dekat, yang selalu menyediakan ruang aman untuk bercerita dan berbagi keresahan dalam proses yang tidak mudah ini.
11. Atun, sepupu yang menjadi teman tumbuh dan berbagi sepanjang perjalanan, yang senantiasa memberi semangat dalam bentuk-bentuk sederhana namun berarti.
12. Teman-teman seperjuangan selama masa kuliah, Arin, Nopa, Muizzul, Ima, Nia, dan Andien, yang telah membersamai penulis sejak awal perjalanan ini. Terima kasih telah menguatkan saat jatuh dan tertawa bersama di tengah segala tekanan.
13. Adila, teman seperjuangan dalam menyelesaikan skripsi ini, yang tak pernah lelah membantu, mendengar, dan menguatkan. Terima kasih telah berbagi semangat dan membersamai di masa paling berat.
14. Teman-teman kos Didoakan I, Petir, Nenden, dan Lala, yang kebersamaannya membuat hari-hari menjadi lebih ringan dan bermakna.
15. Raniah, teman seperantauan yang membawa rasa hangat dalam lingkungan baru, menjadi pengingat bahwa perantauan tak harus selalu terasa asing.
16. Charles, teman pendakian yang telah menemani menaklukkan berbagai jalur dan puncak, serta berbagi lelah dan tawa di setiap langkah perjalanan.

17. Teman-teman selama masa kuliah: Rafi, Irul, Pian, Najah, Enggar, Heny, Dila, Jati, Kimal, Abdillah, dan Suci, yang telah memberi warna dalam kehidupan perkuliahan dengan canda, diskusi, dan semangat kebersamaan.
18. Komunitas GDSC dan Senyum Anak Nusantara Chapter Malang, yang telah memberi ruang untuk bertumbuh, belajar, dan berkontribusi bersama.
19. Seluruh teman seangkatan Teknik Informatika 2021, “Aster 21”, yang telah menjadi teman seperjalanan dalam proses menempuh ilmu dan melewati berbagai fase bersama.
20. Wonwoo, sosok idola yang diam-diam menjadi penguat. Kata-katanya yang sederhana namun dalam, terus mengingatkan untuk tidak menyerah dan tetap bertahan meski hari terasa berat.
21. BTS dan Seventeen, yang melalui lagu-lagunya telah memberikan motivasi, kenyamanan, dan semangat untuk terus melangkah maju.
22. Semua pihak yang terlibat, secara langsung maupun tidak, dalam proses penyusunan skripsi ini.
23. Diri sendiri, yang terus melangkah meski berkali-kali ragu, yang tetap mencoba meski tak selalu yakin, yang bertahan melewati malam-malam panjang dan pagi yang berat. Terima kasih telah tidak menyerah.

Malang, 24 Mei 2025

Penulis

## DAFTAR ISI

<b>HALAMAN PENGAJUAN</b> .....	<b>ii</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>v</b>
<b>MOTTO</b> .....	<b>vi</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vii</b>
<b>KATA PENGANTAR</b> .....	<b>viii</b>
<b>DAFTAR ISI</b> .....	<b>xii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>DAFTAR TABEL</b> .....	<b>xv</b>
<b>ABSTRAK</b> .....	<b>xvi</b>
<b>ABSTRACT</b> .....	<b>xvii</b>
<b>مستخلص البحث</b> .....	<b>xviii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	6
1.3 Batasan Masalah .....	6
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	6
<b>BAB II STUDI PUSTAKA</b> .....	<b>7</b>
2.1 Penelitian Terkait .....	7
2.2 Kebakaran Hutan .....	12
2.3 <i>Convolutional Neural Network</i> .....	13
<b>BAB III DESAIN DAN IMPLEMENTASI</b> .....	<b>17</b>
3.1 Desain Penelitian .....	17
3.2 Pengumpulan Data .....	18
3.3 Desain Sistem .....	20
3.3.1 Input Citra .....	21
3.3.2 <i>Preprocessing</i> Citra .....	21
3.3.3 Normalisasi .....	22
3.3.4 Augmentasi Citra .....	22
3.4 Implementasi Metode .....	25
3.4.1 <i>Input Layer</i> .....	27
3.4.2 <i>Convolution Layer</i> .....	27
3.4.3 <i>Max Pooling Layer</i> .....	30
3.4.4 <i>Fully Connected Layers</i> .....	31
3.5 Proses <i>Training</i> .....	36
3.5.1 <i>Feedforward</i> .....	38
3.5.2 <i>Backpropagation</i> .....	41
3.6 Skenario Pengujian .....	44
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>50</b>
4.1 Hasil Uji Coba .....	50
4.1.1 Skenario Uji Pembagian Data .....	50

4.1.2 Skenario Uji Perubahan Arsitektur .....	55
4.1.3 Evaluasi Menggunakan <i>K-Fold Cross Validation</i> .....	60
4.2 Pembahasan.....	61
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>73</b>
5.1 Kesimpulan .....	73
5.2 Saran.....	74
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 2. 1 Bentuk Umum CNN.....	15
Gambar 3.1 Desain Penelitian.....	17
Gambar 3.2 Desain Sistem.....	20
Gambar 3.3 <i>Preprocessing</i> Citra.....	22
Gambar 3.4 <i>Rotate</i> .....	23
Gambar 3.5 <i>Shear</i> .....	24
Gambar 3.6 <i>Zoom</i> .....	24
Gambar 3.7 <i>Horizontal Flip</i> .....	25
Gambar 3.8 Arsitektur CNN .....	26
Gambar 3.9 Proses Konvolusi.....	28
Gambar 3.10 Perhitungan Proses Konvolusi .....	28
Gambar 3.11 Grafik fungsi aktivasi ReLU .....	30
Gambar 3.12 Penerapan Fungsi Aktivasi ReLU .....	30
Gambar 3.13 Penerapan <i>Max Pooling</i> .....	31
Gambar 3.14 Penerapan Proses <i>Flattening</i> .....	31
Gambar 3.15 Penerapan <i>Input Layer</i> ke <i>Hidden Layer</i> .....	33
Gambar 3.16 Grafik Fungsi Aktivasi <i>Sigmoid</i> .....	34
Gambar 3.17 Penerapan <i>Hidden Layer</i> ke <i>Output Layer</i> .....	35
Gambar 3.18 Alur Proses <i>Training</i> .....	37
Gambar 3.19 Alur Proses <i>Testing</i> .....	47
Gambar 4.1 Grafik Proses <i>Training</i> Skenario 1-A-Y .....	50
Gambar 4.2 <i>Confussion Matrix</i> Skenario 1-A-Y .....	51
Gambar 4.3 Grafik Proses <i>Training</i> Skenario 2-A-Y .....	52
Gambar 4.4 <i>Confussion Matrix</i> Skenario 2-A-Y .....	52
Gambar 4.5 Grafik Proses <i>Training</i> Skenario 3-A-Y .....	53
Gambar 4.6 <i>Confussion Matrix</i> Skenario 3-A-Y .....	54
Gambar 4.7 Grafik Proses <i>Training</i> Skenario 2-A-Z.....	56
Gambar 4.8 <i>Confussion Matrix</i> Skenario 2-A-Z.....	56
Gambar 4.9 Grafik Proses <i>Training</i> Skenario 2-B-Y.....	57
Gambar 4.10 <i>Confussion Matrix</i> Skenario 2-B-Y.....	58
Gambar 4.11 Grafik Proses <i>Training</i> Skenario 2-B-Z.....	59
Gambar 4.12 <i>Confussion Matrix</i> Skenario 2-B-Z.....	59
Gambar 4.13 Perbandingan Metrik Evaluasi Skenario Pembagian Data.....	62
Gambar 4.14 Perbandingan Metrik Evaluasi Skenario Perubahan Arsitektur.....	64
Gambar 4.15 Grafik Hasil Akurasi per <i>Fold</i> .....	66
Gambar 4.16 Grafik Hasil <i>Loss</i> per <i>Fold</i> .....	68

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait .....	10
Tabel 3.1 Contoh Citra <i>Fire</i> dan <i>Non-Fire</i> .....	18
Tabel 3.2 <i>Hyperparameter</i> .....	38
Tabel 3.3 Pembagian Dataset untuk Skenario Uji 1 .....	44
Tabel 3.4 Jumlah Lapisan Konvolusi untuk Skenario Uji 2 .....	45
Tabel 3.5 Metode Klasifikasi untuk Skenario Uji 2.....	45
Tabel 3.6 Kombinasi Skenario Pengujian.....	46
Tabel 3.7 Nilai <i>Hyperparameter</i> .....	46
Tabel 4.1 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 1-A-Y.....	51
Tabel 4.2 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 2-A-Y.....	53
Tabel 4.3 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 3-A-Y.....	54
Tabel 4.4 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 2-A-Z.....	57
Tabel 4.5 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 2-B-Y.....	58
Tabel 4.6 Nilai Presisi, <i>Recall</i> , <i>F1-Score</i> pada Skenario 2-B-Z.....	60
Tabel 4.7 Hasil Evaluasi <i>K-fold</i> pada Skenario 2-A-Y.....	60
Tabel 4.8 Hasil Evaluasi <i>K-fold</i> pada Skenario 2-B-Z.....	61
Tabel 4.9 Hasil Uji Setiap Skenario.....	62

## ABSTRAK

Izzah, Nurul. 2025. **Klasifikasi Kebakaran Hutan Berbasis Citra Menggunakan *Convolutional Neural Network***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, M.Kom (II) Fatchurrohman, M.Kom.

**Kata Kunci:** *Convolutional Neural Network*, Klasifikasi Citra, Kebakaran Hutan.

Kebakaran hutan merupakan salah satu bencana lingkungan yang berdampak besar terhadap ekosistem dan kehidupan manusia. Penelitian ini bertujuan untuk mengklasifikasikan citra kebakaran hutan menggunakan metode *Convolutional Neural Network* (CNN) serta menganalisis pengaruh variasi pembagian data dan perubahan arsitektur terhadap performa model. Dataset yang digunakan berasal dari Kaggle dengan total 1.900 citra yang terdiri atas dua kelas, yaitu *fire* dan *non-fire*. Data tersebut diproses melalui tahapan normalisasi dan augmentasi untuk meningkatkan keragaman data pelatihan. Model CNN yang digunakan merupakan model kustom dengan dua lapisan konvolusi dan klasifikasi biner. Penelitian ini dilakukan melalui dua jenis skenario: variasi rasio pembagian data pelatihan dan pengujian, serta variasi jumlah lapisan dan metode klasifikasi. Berdasarkan hasil pengujian, skenario terbaik untuk pembagian data diperoleh pada rasio 80:20 (skenario 2-A-Y) dengan akurasi sebesar 96%, sedangkan pada pengujian perubahan arsitektur, hasil terbaik dicapai oleh model dengan tiga lapisan konvolusi dan metode klasifikasi *argmax* (skenario 2-B-Z) dengan akurasi sebesar 97%. Selain itu, pengujian menggunakan metode *K-Fold Cross Validation* menunjukkan bahwa skenario 2-A-Y memiliki rata-rata akurasi sebesar 96,05% dan skenario 2-B-Z sebesar 95,89%, yang keduanya menunjukkan performa model yang stabil serta kemampuan generalisasi yang baik terhadap variasi distribusi data. Hasil ini membuktikan bahwa variasi pembagian data serta modifikasi arsitektur CNN dapat memberikan dampak signifikan terhadap akurasi klasifikasi citra kebakaran hutan.

## ABSTRACT

Izzah, Nurul. 2025. **Forest Fire Classification Based on Image Using Convolutional Neural Network**. Thesis. Departement of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University. Supervisor: (I) Prof. Dr. Suhartono, M.Kom (II) Fatchurrohman, M.Kom.

**Kata Kunci:** *Convolutional Neural Network, Image Classification, Forest Fire.*

Forest fires are one of the most significant environmental disasters, causing severe impacts on ecosystems and human life. This study aims to classify forest fire images using the Convolutional Neural Network (CNN) method and to analyze the effects of data partitioning variations and architectural modifications on model performance. The dataset used in this study was obtained from Kaggle, consisting of a total of 1,900 images divided into two classes: fire and non-fire. The images were processed through normalization and augmentation stages to enhance the diversity of the training data. The CNN model implemented in this study is a custom model consisting of two convolutional layers and a binary classification output. The research was conducted through two types of scenarios: variations in the training and testing data ratios, and variations in the number of layers and classification methods. Based on the results, the best scenario for data partitioning was achieved with a ratio of 80:20 (scenario 2-A-Y), reaching an accuracy of 96%, while the best result in the architectural modification test was obtained by the model with three convolutional layers and the argmax classification method (scenario 2-B-Z), achieving an accuracy of 97%. In addition, testing using the K-Fold Cross Validation method showed that scenario 2-A-Y had an average accuracy of 96.05%, and scenario 2-B-Z achieved 95.89%, both demonstrating stable model performance and good generalization capabilities across different data distributions. These results confirm that both data partitioning and CNN architecture modifications can significantly impact the accuracy of forest fire image classification.

## مستخلص البحث

عزة، نورل. 2025. تصنيف حرائق الغابات بالاعتماد على الصور باستخدام الشبكة العصبية التلافيفية. الأطروحة. برنامج  
دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف:  
(الأول) أ. د. سوهارتونو، ماجستري. (الثاني) فتح الرحمن، ماجستري.

الكلمات الرئيسية: الشبكة العصبية التلافيفية، تصنيف الصور، حرائق الغابات.

حرائق الغابات هي إحدى الكوارث البيئية التي لها تأثير كبير على النظم البيئية و حياة البشر. تهدف هذه الدراسة إلى تصنيف صور حرائق الغابات باستخدام طريقة الشبكة العصبية التلافيفية (CNN) و تحليل تأثير تباين تقسيم البيانات و تغيير البنية على أداء النموذج. تم الحصول على مجموعة البيانات المستخدمة من Kaggle و التي تضم 1900 صورة مقسمة إلى فئتين: حرائق و غير حرائق. تمت معالجة البيانات من خلال مراحل التطبيع و التوسيع لزيادة تنوع بيانات التدريب. النموذج CNN المستخدم هو نموذج مخصص ذو طبقتين من التحويل و تصنيف ثنائي. أجريت هذه الدراسة من خلال نوعين من السيناريوهات: تباين نسبة تقسيم بيانات التدريب و الاختبار، و تباين عدد الطبقات و طرق التصنيف. بناءً على نتائج الاختبار، تم الحصول على أفضل سيناريو لتقسيم البيانات بنسبة 80:20 (سيناريو 2-A-Y) بدقة 96%، بينما في اختبار تغيير البنية، تم تحقيق أفضل نتيجة من خلال النموذج ذي ثلاث طبقات التحويل و طريقة التصنيف *argmax* (سيناريو 2-B-Z) بدقة 97%. بالإضافة إلى ذلك، أظهرت الاختبارات باستخدام طريقة *K-Fold Cross Validation* أن السيناريو 2-A-Y حقق متوسط دقة بنسبة 96.05% و السيناريو 2-B-Z بنسبة 95.89%، و كلاهما أظهر أداءً مستقرًا للنموذج و قدرة جيدة على التعميم تجاه تباين توزيع البيانات تثبت هذه النتائج أن تنوع تقسيم البيانات و تعديل بنية CNN يمكن أن يكون له تأثير كبير على دقة تصنيف صور حرائق الغابات.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kebakaran hutan merupakan salah satu isu lingkungan yang sangat serius di berbagai belahan dunia, termasuk Indonesia. Kebakaran ini tidak hanya menyebabkan kerusakan besar pada keanekaragaman hayati, tetapi juga berkontribusi pada polusi udara yang membahayakan kesehatan manusia. Dampak ekonomi yang signifikan juga dirasakan oleh masyarakat, terutama di daerah-daerah yang bergantung pada sektor pariwisata (Suhardono *et al.*, 2024). Kebakaran hutan gambut di Indonesia, khususnya di Kalimantan Tengah, menyebabkan peningkatan konsentrasi gas dan partikel berbahaya di udara, seperti karbon monoksida (CO) dan partikel halus PM<sub>2.5</sub>, yang sangat berisiko bagi kesehatan masyarakat. Pemantauan yang dilakukan menunjukkan bahwa selama musim kemarau, konsentrasi CO dan PM<sub>2.5</sub> jauh lebih tinggi dibandingkan musim hujan, dengan tingkat CO mencapai hingga 22.127,5 µg/m<sup>3</sup> dan PM<sub>2.5</sub> mencapai 360,5 µg/m<sup>3</sup> di beberapa lokasi. Hal ini menggarisbawahi pentingnya program penanganan kebakaran yang lebih serius untuk mencegah dampak kesehatan yang parah di masa mendatang (Kusin *et al.*, 2022).

Menurut laporan dari *Global Forest Watch* pada tahun 2021, data baru dari *University of Maryland* (UMD) menunjukkan bahwa kebakaran, baik alami maupun yang disebabkan oleh manusia, merupakan penyebab signifikan hilangnya tutupan pohon secara langsung, yang dapat membingungkan tren jangka panjang

deforestasi permanen karena emisi karbon, pengeringan hutan, dan peningkatan frekuensi kebakaran (Goldman & Weisse, 2022). Akibatnya tidak hanya berdampak pada kerusakan ekosistem hutan dan hilangnya keanekaragaman hayati, tetapi juga meningkatkan emisi karbon di atmosfer, yang dapat memperburuk perubahan iklim dan mempercepat siklus kebakaran di masa depan.

*Machine learning* merupakan salah satu bagian dari kecerdasan buatan yang fokus pada pengembangan sistem yang dapat berkembang dan belajar secara mandiri tanpa perlu dikontrol secara eksplisit oleh manusia (Faisal, 2022). Salah satu teknik yang banyak digunakan dalam *machine learning* khususnya dalam bidang *computer vision* adalah *Convolutional Neural Network*. CNN telah memberikan kontribusi besar dalam berbagai bidang seperti pengenalan objek, klasifikasi gambar, segmentasi gambar, pemrosesan bahasa alami, dan pengenalan suara. Kemajuan signifikan dalam arsitektur CNN, penggunaan fungsi aktivasi alternatif, regulasi, serta optimasi parameter telah mendorong peningkatan kinerja yang luar biasa dalam tugas-tugas ini. Keberadaan data dalam jumlah besar serta perangkat keras yang lebih mudah diakses juga membuka peluang baru untuk penelitian lebih lanjut dalam pengembangan CNN (Bhatt *et al.*, 2021).

Dalam konteks klasifikasi kebakaran hutan berbasis citra, CNN memiliki keunggulan yang kuat. Arsitekturnya dirancang untuk memanfaatkan informasi spasial dan saluran dalam data input, memungkinkan model ini untuk secara efektif mengidentifikasi pola visual, termasuk dalam konteks klasifikasi kebakaran hutan (Chen *et al.*, 2024). Evolusi CNN, terutama dalam eksploitasi spasial, pemrosesan multi-jalur, dan peningkatan kedalaman, secara signifikan telah berperan penting

dalam meningkatkan kemampuan untuk memproses gambar yang kompleks, mengakomodasi variasi pencahayaan, sudut, dan latar belakang (Purwono *et al.*, 2022). Oleh karena itu, dengan menggunakan implementasi model berbasis CNN, penelitian ini dapat memanfaatkan kekuatan arsitektur *deep learning* untuk mendeteksi dan mengklasifikasikan kebakaran hutan dengan tingkat akurasi yang tinggi.

Beberapa penelitian telah menunjukkan bahwa performa CNN bisa sangat baik dalam mendeteksi dan juga mengklasifikasi objek kebakaran hutan. Penelitian oleh Khan & Khan, (2022) menunjukkan bahwa metode CNN mampu memberikan akurasi yang sangat tinggi dalam mendeteksi kebakaran hutan. Dengan menggunakan model FFireNet dan dataset yang terdiri dari 1.900 citra beresolusi 250x250 piksel, penelitian ini berhasil mencapai akurasi sebesar 98.42% pada proses pengujian. Hasil ini menegaskan bahwa CNN efektif dalam menganalisis fitur citra untuk mendeteksi kebakaran. Namun, penelitian ini menggunakan arsitektur yang sudah kompleks dan memerlukan sumber daya komputasi yang signifikan. Selain itu, analisis pengaruh jumlah lapisan CNN terhadap performa model tidak dibahas secara rinci.

Penelitian lain oleh Dwiasnati *et al.*, (2024) juga membuktikan kemampuan CNN dalam mendeteksi area kebakaran hutan. Dengan menggunakan dataset dari wilayah Nusa Tenggara Timur, penelitian ini mencapai akurasi sebesar 89%. Namun, penelitian tersebut kurang menjelaskan beberapa aspek penting, seperti akuisisi data, pengaturan *hyperparameter*, dan struktur arsitektur model CNN yang

digunakan. Keterbatasan ini menyulitkan peneliti lain untuk mereplikasi atau memperluas penelitian.

Berdasarkan latar belakang yang telah diuraikan, penelitian ini dilakukan dengan mengimplementasikan model *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi kebakaran hutan menggunakan data berbasis citra. Model ini dipilih karena terbukti efektif dalam mengklasifikasikan citra dengan akurasi yang tinggi. Data yang digunakan berasal dari dataset citra kebakaran hutan yang tersedia di platform Kaggle, yang diharapkan dapat membantu dalam pengklasifikasian dan pemetaan area kebakaran hutan secara akurat, sehingga dapat mendukung upaya mitigasi kebakaran hutan dengan lebih efektif.

Penelitian ini penting, mengingat kebakaran hutan merupakan bencana alam yang terjadi setiap tahun dengan dampak yang merugikan. Allah SWT berfirman dalam Al-Qur'an:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

*“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: 41)*

Ayat ini menekankan bahwa kerusakan yang terjadi di bumi, baik di daratan maupun di lautan, disebabkan oleh ulah manusia sendiri. Menurut tafsir Kementerian Agama Jilid 7, kosakata "*fasād*" (kerusakan) dalam ayat ini keluarnya sesuatu dari keseimbangan baik sedikit maupun banyak atau bermakna rusak. Istilah ini digunakan untuk menunjuk kerusakan, baik jasmani, jiwa, maupun hal-hal lain. Dalam konteks lingkungan, *fasād* ini dapat diartikan sebagai kerusakan ekosistem,

seperti hancurnya flora dan fauna, penggundulan hutan, dan pencemaran. Allah SWT memberikan bencana ini sebagai peringatan agar manusia merasakan sebagian dari akibat perbuatannya dan kembali ke jalan yang benar (Kementerian Agama RI, 2011).

Penelitian ini juga diharapkan berkontribusi pada usaha manusia dalam menjaga alam dan lingkungan, sebagaimana amanat yang diberikan Allah SWT dalam Al-Qur'an Surah Al-A'raf ayat 56, dimana manusia diingatkan untuk tidak membuat kerusakan di muka bumi:

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

*"Dan janganlah kamu membuat kerusakan di muka bumi setelah (Allah) memperbaikinya dan berdoalah kepada-Nya dengan rasa takut (tidak akan diterima) dan penuh harap (akan dikabulkan). Sesungguhnya rahmat Allah sangat dekat kepada orang-orang yang berbuat kebaikan." (QS. Al-A'raf: 56)*

Dalam ayat ini Allah SWT melarang manusia agar tidak membuat kerusakan di muka bumi. Larangan membuat kerusakan tersebut mencakup semua bidang, seperti merusak pergaulan, jasmani dan rohani orang lain, kehidupan dan sumber-sumber penghidupan (pertanian, perdagangan, dan lain-lain), merusak lingkungan, dan lain sebagainya. Allah SWT telah menciptakan bumi dengan seluruh kelengkapan alam seperti gunung, sungai, hutan, dan lautan sebagai sarana bagi manusia untuk dikelola dengan bijak demi kesejahteraan bersama. Oleh karena itu, manusia wajib menjaga dan tidak merusaknya (Kementerian Agama RI, 2011).

Dengan memanfaatkan model CNN, penelitian ini diharapkan dapat membantu manusia agar lebih efektif dalam mitigasi bencana khususnya kebakaran hutan dan mencegah kerusakan lebih lanjut.

## 1.2 Identifikasi Masalah

Bagaimana menerapkan metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan kebakaran hutan menggunakan data berbasis citra serta menganalisis pengaruh variasi pembagian data dan perubahan arsitektur terhadap performa model?

## 1.3 Batasan Masalah

1. Dataset yang digunakan adalah dataset publik didapat dari Kaggle dengan judul "*Forest Fire Dataset*" yang berisi citra kebakaran hutan berformat jpg.
2. Penelitian ini hanya berfokus pada klasifikasi citra kebakaran hutan dengan dua kelas, yaitu "*fire*" (kebakaran) dan "*non-fire*" (tidak kebakaran).
3. Pengujian dilakukan dalam dua skenario utama, yaitu skenario pengaruh pembagian data dan skenario pengaruh perubahan arsitektur.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menerapkan metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan kebakaran hutan menggunakan data berbasis citra serta menganalisis pengaruh variasi pembagian data dan perubahan arsitektur terhadap performa model.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat menjadi dasar untuk pengembangan sistem deteksi kebakaran yang lebih cepat dan efisien, sehingga dapat membantu dalam penanggulangan dan pencegahan kebakaran hutan.

## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian tentang klasifikasi kebakaran hutan telah berkembang pesat dengan berbagai metode *deep learning* yang digunakan untuk meningkatkan akurasi dan efisiensi dalam mendeteksi kebakaran. Beberapa studi terdahulu telah memanfaatkan *Convolutional Neural Network* (CNN) dengan berbagai arsitektur dan teknik untuk melakukan klasifikasi citra kebakaran hutan maupun bencana alam lainnya.

Madhuri *et al.*, (2024) melakukan penelitian yang berfokus pada identifikasi kebakaran hutan dari citra udara menggunakan pendekatan gabungan dari tiga arsitektur CNN, yaitu InceptionV3, VGG-16, dan ResNet50. Dalam penelitian ini, data gambar kebakaran hutan diproses dan diklasifikasikan menggunakan kombinasi model tersebut. Dengan akurasi mencapai 95.8%, pendekatan ini menunjukkan bahwa gabungan model gabungan mampu meningkatkan hasil klasifikasi dibandingkan dengan model individu. Penelitian ini memberikan bukti bahwa pendekatan hibrida dari beberapa model arsitektur CNN mampu meningkatkan akurasi secara signifikan dalam tugas klasifikasi kebakaran hutan.

Selanjutnya, Bahhar *et al.*, (2023) mengajukan metode deteksi kebakaran hutan dan asap menggunakan model *You Only Look Once* (YOLO) yang dikombinasikan dengan beberapa model CNN seperti XceptionNet, MobileNetV2, dan ResNet50. Dataset yang digunakan adalah dataset FLAME, yang terdiri dari

citra kebakaran dan non-kebakaran. Melalui kombinasi deteksi multi-tahap, CNN digunakan untuk melakukan klasifikasi sementara YOLO dipakai untuk mendeteksi objek kebakaran atau asap. Dengan pencapaian F1-score sebesar 0.95, akurasi 0.99, dan sensitivitas sebesar 0.98, pendekatan ini terbukti sangat efektif dalam mendeteksi kebakaran hutan secara cepat dan efisien. Hal ini menjadi penting dalam pengawasan kebakaran hutan untuk memastikan deteksi dini dan respon yang cepat terhadap bencana kebakaran.

Khan & Khan, (2022) mengembangkan model *Fire Forest Network* (FFireNet) yang didasarkan pada arsitektur CNN untuk klasifikasi kebakaran hutan. Model ini dilatih menggunakan dataset kebakaran hutan dan mencapai akurasi sebesar 98.42%. Hasil ini memperlihatkan bahwa arsitektur yang dikembangkan secara khusus untuk klasifikasi kebakaran hutan mampu menghasilkan akurasi tinggi dan dapat diimplementasikan pada sistem pengawasan kebakaran di wilayah perkotaan maupun pedesaan.

Di sisi lain, Saputra & Adhinata, (2023) melakukan penelitian yang berfokus pada deteksi kebakaran hutan dan lahan di Indonesia. Mereka menggunakan model *transfer learning* dengan arsitektur DenseNet201 untuk mendeteksi kebakaran berdasarkan citra satelit. Mereka menggunakan dataset yang terdiri dari 1900 gambar yang dibagi menjadi dua kelas, yaitu *fire* dan *non-fire*. Hasil penelitian ini menunjukkan akurasi yang sangat tinggi, yakni sebesar 99%, yang menunjukkan bahwa DenseNet201 merupakan model yang sangat efektif untuk mendeteksi kebakaran hutan di wilayah tropis seperti Indonesia.

Dalam penelitian lainnya, Pratiwi & Nudin, (2023) mengembangkan sistem deteksi dini kebakaran berbasis *website* dengan menggunakan model ResNet50. Dataset yang digunakan dalam penelitian ini terdiri dari 3600 citra yang terbagi menjadi tiga kelas: *fire*, *smoke*, dan *non-fire*. Penelitian ini berhasil mencapai akurasi pelatihan sebesar 98%, namun akurasi menurun menjadi 78% pada tahap pengujian. Penurunan akurasi ini mengindikasikan adanya tantangan dalam implementasi model pada lingkungan nyata, seperti variasi gambar di luar data pelatihan.

Abror, (2019) juga berkontribusi dalam penelitian klasifikasi kebakaran dengan mengembangkan model CNN kustom yang dirancang khusus untuk klasifikasi gambar kebakaran. Model ini diuji menggunakan dataset citra kebakaran dan menghasilkan akurasi pelatihan sebesar 98.8% serta akurasi pengujian sebesar 90%. Hasil ini menunjukkan potensi penggunaan model CNN kustom untuk menangani masalah klasifikasi dalam kebakaran hutan, meskipun tantangan terkait generalisasi model tetap perlu diperhatikan.

Terakhir, Khan *et al.*, (2022) mengembangkan sistem pemadam kebakaran hutan berbasis UAV dengan integrasi AI untuk pengawasan hutan berkelanjutan dan deteksi kebakaran. Data baru dikumpulkan sebanyak 1900 citra berwarna dengan masing-masing 950 untuk kelas *fire* dan *non-fire*. Model *transfer learning* VGG-19 digunakan sebagai usulan model pada penelitian ini, dengan akurasi yang didapat sebesar 95% dalam mengklasifikasi citra kebakaran hutan yang nantinya akan digunakan dalam sistem deteksi.

Berbagai model *deep learning* telah diusulkan oleh beberapa peneliti untuk meningkatkan akurasi dalam klasifikasi kebakaran hutan dan bencana alam lainnya. Masing-masing penelitian menggunakan model dan dataset yang berbeda, serta menghasilkan tingkat akurasi yang bervariasi. Rangkuman dari penelitian-penelitian tersebut beserta perbedaannya dengan penelitian ini disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait

No	Peneliti	Judul Penelitian	Metode	Hasil	Perbedaan Penelitian
1	Madhuri et al. (2024)	<i>Accurate Classification of Forest Fires in Aerial Images Using Ensemble Model</i>	InceptionV3, VGG-16, ResNet50	Akurasi 95.8% didapat setelah menggabungkan ketiga model	Berbeda dengan penelitian ini yang menggunakan CNN kustom dan dua kelas ( <i>fire</i> dan <i>non-fire</i> ), penelitian oleh Madhuri et al. menggunakan pendekatan dari tiga model <i>pretrained</i> dan dataset tiga kelas ( <i>fire, smoke, non-fire</i> )
2	Bahhar et al. (2023)	<i>Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN</i>	XceptionNet, MobileNetV2, ResNet50	Akurasi 99% didapatkan untuk masing-masing model	Penelitian oleh Bahhar et al. menggunakan kombinasi YOLO dan tiga model CNN dengan dataset FLAME, berbeda dengan penelitian ini yang menggunakan CNN kustom tanpa YOLO dan dataset dua kelas dari Kaggle
3	Khan & Khan (2022)	<i>FFireNet: Deep Learning Based Forest Fire Classification and Detection in Smart Cities</i>	FfireNet	Didapat akurasi sebesar 98.42% pada model yang diajukan	Penelitian oleh Khan mengusulkan arsitektur khusus FFireNet, sedangkan penelitian ini menggunakan arsitektur CNN sederhana dengan variasi jumlah lapisan dan metode klasifikasi untuk menguji pengaruhnya terhadap performa

No	Peneliti	Judul Penelitian	Metode	Hasil	Perbedaan Penelitian
4	Saputra & Adhinata (2023)	Model Deteksi Kebakaran Hutan dan Lahan Menggunakan <i>Transfer Learning</i> DenseNet201	DenseNet201	Didapat akurasi terbaik sebesar 99%	Berbeda dengan penelitian ini yang menggunakan CNN kustom dan uji pembagian data serta arsitektur, penelitian ini menggunakan pendekatan <i>transfer learning</i> dengan model DenseNet201 tanpa pengujian variasi arsitektur
5	Pratiwi & Nudin (2023)	Pengembangan Sistem Deteksi Dini Kebakaran Menggunakan Pendekatan <i>Transfer Learning</i> dengan Model ResNet50 Berbasis <i>Website</i>	ResNet50	Didapat akurasi tertinggi proses pelatihan sebesar 98% dan 78% pada pengujian	Penelitian oleh Pratiwi dan Nudin mengembangkan sistem berbasis <i>web</i> menggunakan <i>transfer learning</i> (ResNet50) dengan dataset tiga kelas, berbeda dengan penelitian ini yang fokus pada pengaruh pembagian data dan arsitektur CNN custom menggunakan dua kelas
6	Abror (2019)	Klasifikasi Citra Kebakaran dan Non-Kebakaran Menggunakan <i>Convolutional Neural Network</i>	CNN <i>Custom</i>	Didapat akurasi validasi sebesar 98.8% dan akurasi pengujian sebesar 90%	Meskipun sama-sama menggunakan CNN custom, penelitian ini menggunakan dataset dari hasil <i>crawling</i> sebanyak 950 citra dan tidak menguji pengaruh arsitektur atau pembagian data seperti yang dilakukan dalam penelitian ini
7	Khan et al. (2022)	<i>DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection</i>	VGG-19	Didapati akurasi sebesar 95%	Penelitian ini menggunakan <i>transfer learning</i> dengan VGG-19 dan sistem pemadam kebakaran berbasis UAV, sedangkan penelitian ini menggunakan CNN custom dan fokus pada evaluasi performa berdasarkan variasi arsitektur dan rasio data.

Berdasarkan Tabel 2.1, dapat dilihat bahwa metode *deep learning*, khususnya yang menggunakan *Convolutional Neural Network* (CNN), memberikan hasil yang sangat baik dalam mendeteksi dan mengklasifikasikan kebakaran hutan serta bencana alam lainnya. Keunggulan CNN terletak pada kemampuannya untuk mengenali pola yang kompleks dalam data citra, sehingga meningkatkan akurasi dalam identifikasi kebakaran hutan dibandingkan dengan metode tradisional. Pada penelitian ini, peneliti akan mengadopsi arsitektur kustom yang dirancang oleh Abror, (2019) dikarenakan model yang dirancang terbukti dapat menghasilkan akurasi dengan nilai tinggi meskipun dengan arsitektur yang lebih sederhana. Pemilihan arsitektur ini juga bertujuan untuk mengurangi masalah komputasi pada saat model dijalankan di perangkat yang akan digunakan.

## **2.2 Kebakaran Hutan**

Kebakaran hutan merupakan fenomena yang sering terjadi di berbagai belahan dunia, termasuk Indonesia. Kebakaran hutan merupakan bencana yang terjadi akibat interaksi antara faktor manusia dan kondisi lingkungan yang mendukung, seperti cuaca ekstrem dan rendahnya curah hujan. Aktivitas manusia, seperti pembukaan lahan pertanian dengan cara membakar dan penggunaan api yang tidak bertanggung jawab, menjadi penyebab utama kebakaran. Selain itu, kondisi alam seperti lahan yang kering dan suhu yang panas memperburuk situasi, meningkatkan durasi kebakaran dan memicu penyebaran api yang lebih cepat. Keterlambatan dalam deteksi dini kebakaran hutan juga merupakan kendala serius dalam upaya penanggulangan, di mana sistem pemantauan tradisional memiliki keterbatasan waktu dan biaya yang signifikan (Anhar *et al.*, 2022).

Dampak dari kebakaran hutan sangat beragam dan dapat memengaruhi baik manusia maupun lingkungan. Dari segi kesehatan, kebakaran dapat menyebabkan masalah pernapasan akibat kabut asap, yang berdampak langsung pada kesehatan masyarakat. Selain itu, dampak sosial juga terlihat melalui penurunan interaksi antar masyarakat serta pengaruh terhadap ekonomi rumah tangga yang berkurang. Dari perspektif lingkungan, kebakaran menyebabkan hilangnya vegetasi, penurunan kualitas udara, dan berkurangnya sumber daya air, yang semua itu berkontribusi pada kerusakan ekosistem secara keseluruhan (Hadiwijoyo, 2023).

Data dari Badan Nasional Penanggulangan Bencana (BNPB) menunjukkan bahwa kebakaran hutan dan lahan (karhutla) mendominasi bencana di Indonesia pada bulan Juli tahun 2019 dan 2023. Tercatat pada tahun 2019 terdapat 115 kejadian kebakaran hutan, sedangkan pada tahun 2023 meningkat tajam menjadi 269 kejadian (BNBP, 2024). Kebakaran ini tidak hanya mengakibatkan kerusakan ekologis yang signifikan, tetapi juga memicu krisis kesehatan akibat kabut asap yang berdampak pada ribuan orang dengan penyakit pernapasan. Dalam konteks ini, deteksi dini kebakaran hutan menjadi salah satu kendala utama, di mana sistem pemantauan tradisional, seperti satelit dan patroli darat, sering kali mengalami keterbatasan baik dalam hal waktu maupun biaya.

### ***2.3 Convolutional Neural Network***

Jaringan saraf tiruan merupakan model komputasi yang terinspirasi dari cara kerja jaringan saraf biologis dalam memproses informasi. Jaringan ini memiliki beberapa karakteristik utama, yaitu struktur koneksi antar neuron dalam jaringan (yang membentuk arsitektur model), metode menetapkan bobot dalam setiap

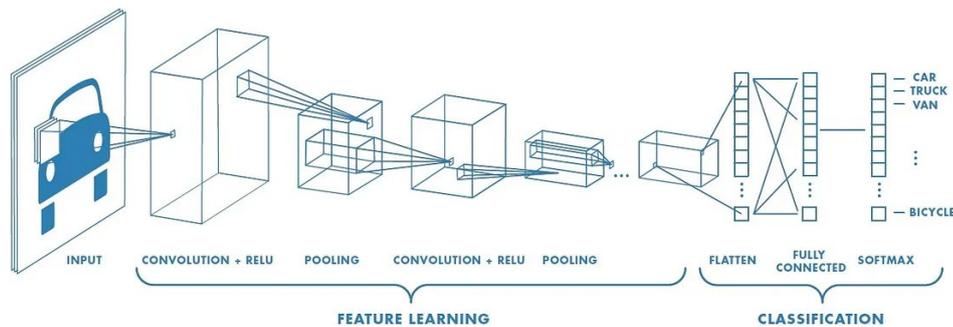
sambungan melalui proses pembelajaran, serta penggunaan fungsi aktivasi untuk menentukan keluaran dari neuron. Sistem jaringan saraf tiruan banyak digunakan di berbagai bidang seperti kedokteran, bisnis, keuangan, serta elektronika, termasuk dalam pemrosesan sinyal dan sistem kontrol (Suhartono, 2018).

Salah satu arsitektur *deep learning* berupa jaringan saraf tiruan yang dirancang untuk menganalisis data visual adalah *Convolutional Neural Network* (CNN). CNN bekerja dengan cara mengekstraksi fitur secara otomatis dari data gambar melalui serangkaian lapisan yang terdiri dari lapisan konvolusi, *pooling*, dan *fully-connected*. Proses ini memungkinkan jaringan untuk mendeteksi pola penting pada gambar, seperti garis, tepi, tekstur, dan bentuk objek, tanpa memerlukan teknik ekstraksi fitur manual (Sakib *et al.*, 2018).

Ciri khas utama dari CNN adalah penggunaan filter atau kernel yang diterapkan pada input gambar untuk mendeteksi fitur lokal. Filter-filter ini bekerja dengan cara mengonvolusi gambar, yaitu menggeser filter melintasi seluruh area gambar dan melakukan operasi perkalian serta penjumlahan antara nilai piksel dan bobot filter. Hasil dari proses ini adalah *feature map* yang menunjukkan lokasi serta keberadaan pola atau fitur penting seperti tepi, garis atau tekstur dalam gambar. Selama pelatihan, filter belajar secara otomatis dengan menyesuaikan bobotnya menggunakan algoritma *backpropagation*. Algoritma ini menghitung kesalahan klasifikasi berdasarkan perbedaan antara prediksi model dan nilai aktual, kemudian memperbarui bobot untuk meminimalkan kesalahan tersebut. Dengan semakin bertambahnya kedalaman jaringan, filter mampu mengenali pola yang lebih

kompleks, sehingga CNN dapat mengenali objek meskipun terjadi variasi orientasi, skala, atau pencahayaan pada gambar (Taye, 2023).

Arsitektur CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi, lapisan *pooling*, dan lapisan *fully connected*. Lapisan konvolusi dan lapisan *pooling* merupakan lapisan yang berada pada tahap ekstraksi fitur dimana gambar input diekstraksi menjadi fitur berupa angka-angka yang merepresentasikan gambar tersebut (Saleem *et al.*, 2022). Bentuk umum dari arsitektur CNN dapat dilihat pada Gambar 2.1 (Matlab, 2021).



Gambar 2.1 Bentuk Umum CNN

### 1. Convolution Layer

*Convolution layer* atau lapisan konvolusi merupakan lapisan pada arsitektur CNN yang berguna untuk mengekstrak fitur dari gambar input. Setiap lapisan konvolusi memiliki sejumlah filter atau kernel yang digeser melintasi gambar input untuk menghasilkan *feature map*. Setiap filter bertugas mendeteksi fitur tertentu seperti tepi, sudut, dan pola-pola lainnya.

### 2. Pooling Layer

*Pooling layer* berfungsi untuk melakukan *down-sampling* atau proses pengurangan dimensi dari *feature map* yang telah dihasilkan pada lapisan konvolusi. *Pooling* menghasilkan *feature map* dengan resolusi lebih rendah namun

tetap mempertahankan informasi penting dari gambar input. Terdapat beberapa jenis *pooling*, diantaranya *max pooling*, *average pooling*, dan *L2-norm pooling*.

### 3. *Fully Connected Layer*

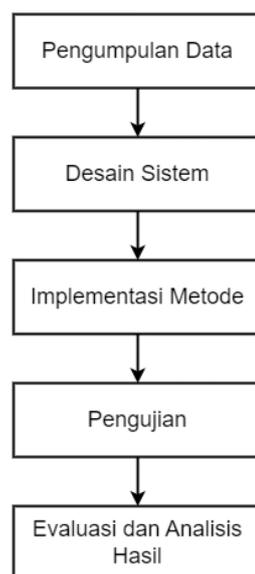
Lapisan *fully connected* melakukan klasifikasi berdasarkan fitur yang telah diekstraksi oleh lapisan sebelumnya. Sejumlah neuron diambil sebagai input dan diklasifikasikan ke dalam kelas yang relevan.

## BAB III

### DESAIN DAN IMPLEMENTASI

#### 3.1 Desain Penelitian

Desain penelitian merupakan serangkaian langkah yang dilakukan oleh peneliti, dimulai dari proses pengumpulan data hingga tahap implementasi model. Desain penelitian dibuat untuk menggambarkan urutan proses yang terjadi di dalam sistem. Proses-proses yang ada dalam desain tersebut kemudian diimplementasikan untuk memungkinkan model mempelajari data yang diberikan. Untuk memahami kinerja terbaik dari model dan faktor-faktor yang mempengaruhinya, diperlukan beberapa skenario uji coba. Setiap skenario uji coba akan menghasilkan kinerja model yang berbeda, sehingga dapat diidentifikasi skenario yang memberikan performa terbaik. Langkah terakhir dalam desain penelitian ini adalah mengevaluasi dan menganalisis hasil uji coba yang telah dilakukan. Desain penelitian ini ditampilkan pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

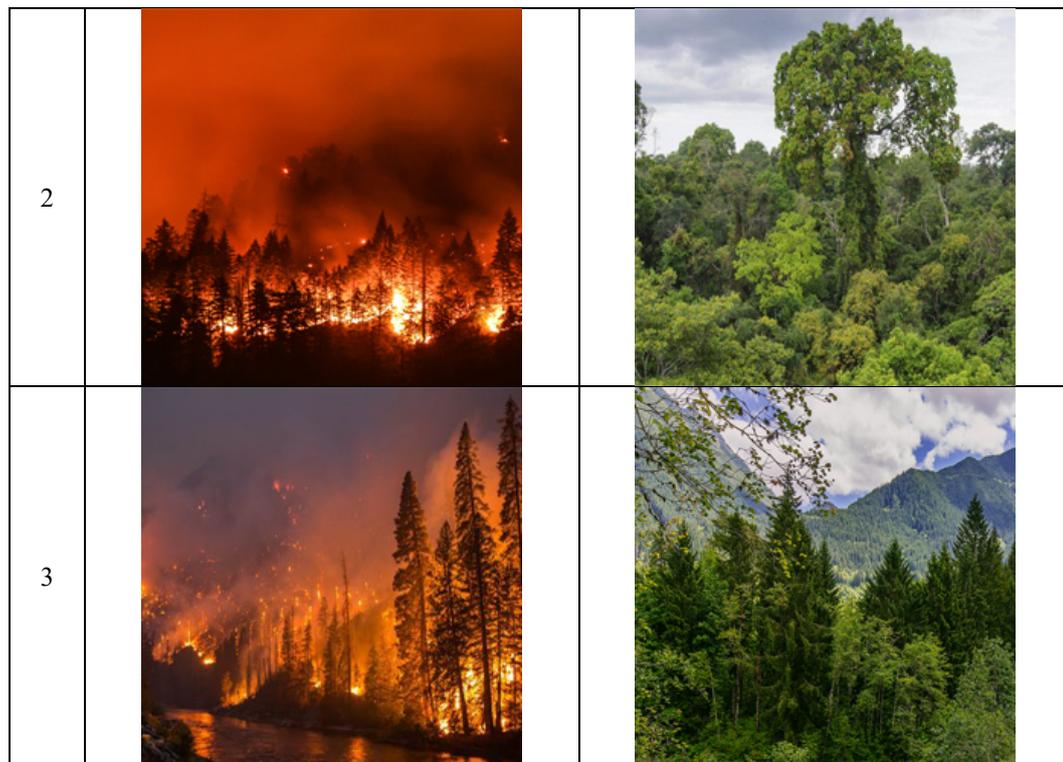
### 3.2 Pengumpulan Data

Dalam penelitian ini, untuk melakukan klasifikasi kebakaran hutan berdasarkan citra, diperlukan dataset yang berisi gambar area hutan yang mengalami kebakaran dan area yang tidak mengalami kebakaran. Dataset yang digunakan adalah "*Forest Fire Dataset*", yang diperoleh dari situs Kaggle.com. Dataset ini dirancang untuk menyelesaikan masalah deteksi kebakaran hutan dalam bentuk klasifikasi biner, yaitu *fire* (kebakaran) dan *non-fire* (tidak kebakaran). Dataset ini terdiri dari total 1900 citra berformat .jpg, dengan masing-masing kelas memiliki 950 citra. Semua gambar dalam dataset ini memiliki resolusi  $250 \times 250$  piksel dengan 3 *channel* (RGB).

Dataset citra kebakaran hutan ini awalnya dikumpulkan dan dikembangkan oleh Khan *et al.*, (2022) dalam penelitian mereka yang berjudul "*DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection*" yang kemudian diunggah ke Kaggle agar dapat digunakan oleh peneliti lain untuk pengembangan lebih lanjut. Contoh sampel citra dari dataset ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh Citra *Fire* dan *Non-Fire*

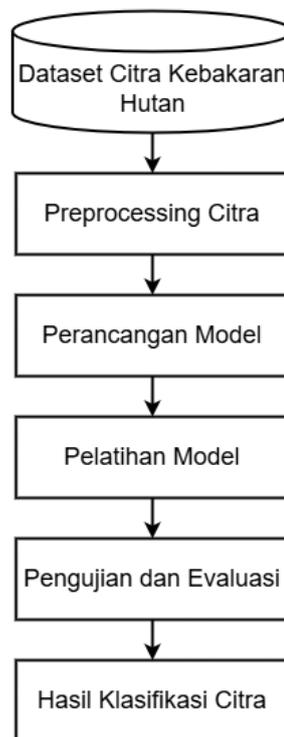
No	Fire	Non-Fire
1		



Tabel 3.1 diatas menampilkan beberapa sampel citra yang mewakili dua kategori utama dalam dataset, yaitu *fire* dan *non-fire*. Citra *fire* menunjukkan area hutan yang terbakar, di mana api yang menyala jelas terlihat, serta adanya asap dan kerusakan pada vegetasi sekitar. Kondisi ini menggambarkan situasi kebakaran hutan yang nyata dan membantu sistem dalam mengenali karakteristik visual dari area yang terkena kebakaran. Di sisi lain, citra *non-fire* menampilkan hutan yang tidak terbakar, yang umumnya hanya memperlihatkan pohon-pohon dan vegetasi tanpa adanya tanda-tanda kebakaran. Perbedaan visual yang signifikan antara kedua kelas ini memungkinkan model *Convolutional Neural Network* (CNN) untuk belajar dan melakukan klasifikasi secara akurat berdasarkan fitur-fitur yang ada pada citra.

### 3.3 Desain Sistem

Desain sistem dalam penelitian ini menerangkan alur kerja sistem yang digunakan untuk melakukan klasifikasi kebakaran hutan, mulai dari input citra hingga menghasilkan *output* berupa kategori *fire* atau *non-fire*. Rincian dari keseluruhan desain sistem ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Desain Sistem

Berdasarkan Gambar 3.2, sistem klasifikasi kebakaran hutan berbasis citra menggunakan CNN dimulai dengan pengumpulan dataset citra yang terdiri dari gambar kebakaran dan non-kebakaran hutan. Dataset tersebut kemudian diproses melalui tahapan *preprocessing*, termasuk pembagian dataset menjadi data *training* dan *testing*, normalisasi citra untuk mengubah nilai piksel menjadi rentang 0 hingga 1, serta augmentasi citra untuk meningkatkan keragaman data pelatihan. Selanjutnya, model CNN kustom dirancang dengan lapisan konvolusi dan lapisan

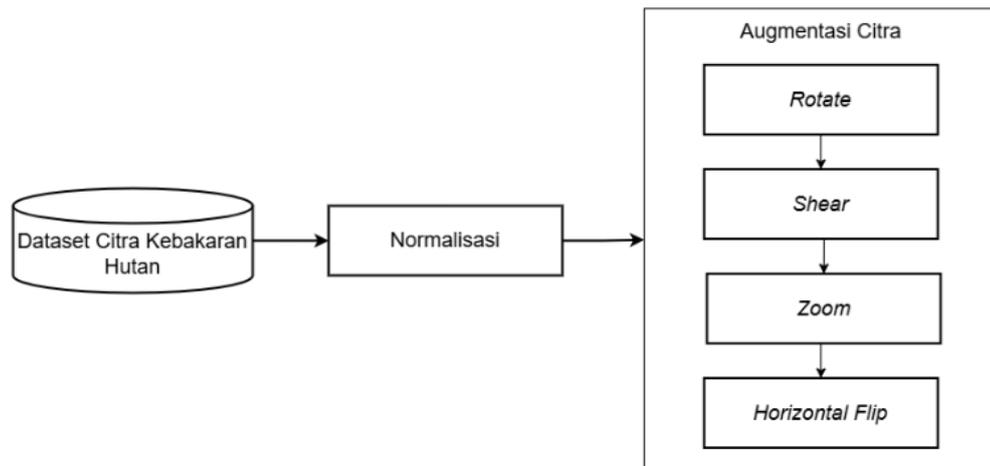
*pooling* untuk ekstraksi fitur, dilanjutkan dengan lapisan *fully connected* untuk klasifikasi. Model kemudian dilatih menggunakan proses *feedforward* dan *backpropagation*, dimana bobot-bobot model diperbarui untuk meminimalkan *loss*. Setelah pelatihan selesai, model diuji dan dievaluasi menggunakan data pengujian untuk mengukur kinerja dengan metrik seperti akurasi, *precision*, *recall*, dan *F1-score*. Hasil klasifikasi citra berupa label *fire* atau *non-fire* kemudian ditampilkan, yang menunjukkan kemampuan model dalam mendeteksi kebakaran hutan.

### **3.3.1 Input Citra**

Citra kebakaran hutan yang telah diperoleh digunakan sebagai input dalam sistem yang akan dibangun. Dataset citra kebakaran hutan ini terdiri dari dua kelas, yaitu *fire* dan *non-fire*, dengan format .jpg dengan ukuran citra  $250 \times 250$ . Dataset tersebut berisi total 1900 citra, di mana setiap kelas memiliki 950 citra. Citra-citra ini nantinya akan digunakan untuk melatih serta menguji model dalam klasifikasi area hutan yang mengalami kebakaran dan yang tidak mengalami kebakaran.

### **3.3.2 Preprocessing Citra**

*Preprocessing* merupakan langkah penting dalam mempersiapkan citra sebelum dimasukkan ke dalam model. Proses ini memastikan bahwa citra yang digunakan dalam pelatihan model telah diolah dengan cara yang sesuai, untuk meningkatkan efektivitas dan efisiensi model dalam melakukan klasifikasi. Pada penelitian ini, *preprocessing* melibatkan dua tahap utama, yang dijelaskan dalam Gambar 3.3.

Gambar 3.3 *Preprocessing* Citra

### 3.3.3 Normalisasi

Normalisasi bertujuan untuk mengurangi beban komputasi dalam proses pelatihan model. Proses ini dilakukan dengan membagi setiap nilai piksel pada citra dengan 255, sehingga setiap nilai piksel berada dalam rentang  $[0, 1]$ . Proses normalisasi citra ini dapat dinyatakan dengan rumus yang terlihat pada Persamaan 3.1.

$$I_{norm} = \frac{I_{pixel}}{255} \quad (3.1)$$

Keterangan:

$I_{norm}$  = nilai *pixel* yang sudah dinormalisasi dengan rentang nilai antara 0 dan 1

$I_{pixel}$  = nilai asli *pixel* pada citra

### 3.3.4 Augmentasi Citra

Augmentasi citra adalah teknik yang digunakan untuk meningkatkan variasi data pelatihan dengan memodifikasi citra yang ada tanpa menambah jumlah data secara langsung. Tujuan dari augmentasi ini adalah untuk membantu model belajar lebih baik dengan meningkatkan daya generalisasi terhadap berbagai kondisi citra

(Ghahremani et al., 2024). Pada penelitian ini, augmentasi yang digunakan meliputi *rotate*, *shear*, *zoom*, dan *horizontal flip*. Berikut penjelasan dari masing-masing teknik pada augmentasi citra:

### 1. *Rotate*

*Rotate* bertujuan untuk memutar citra dalam rentang sudut tertentu. Dalam penelitian ini, rotasi dilakukan untuk menciptakan variasi sudut pandang dari objek di dalam citra, dimana citra diputar dalam rentang 0 hingga 90 derajat untuk memberikan variasi sudut pandang. Berikut contoh hasil dari proses *rotate* dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Rotate*

### 2. *Shear*

*Shear* adalah proses penggeseran satu bagian dari citra sehingga menciptakan efek miring pada gambar. Pada penelitian ini, proses *shear* dilakukan dengan memberikan distorsi pada citra dalam rentang 20 kemiringan. Berikut contoh hasil dari proses *shear* dapat dilihat pada Gambar 3.5.

Gambar 3.5 *Shear*

### 3. *Zoom*

*Zoom* dilakukan dengan memperbesar atau memperkecil objek dalam citra, sehingga model dapat mengenali objek meskipun ukurannya bervariasi. Dalam proses augmentasi ini, citra diperbesar atau diperkecil dalam rentang 20 persen, yang membantu meningkatkan kemampuan model dalam mendeteksi kebakaran hutan dari skala yang berbeda-beda. Berikut contoh hasil dari proses *zoom* dapat dilihat pada Gambar 3.6.

Gambar 3.6 *Zoom*

#### 4. *Horizontal Flip*

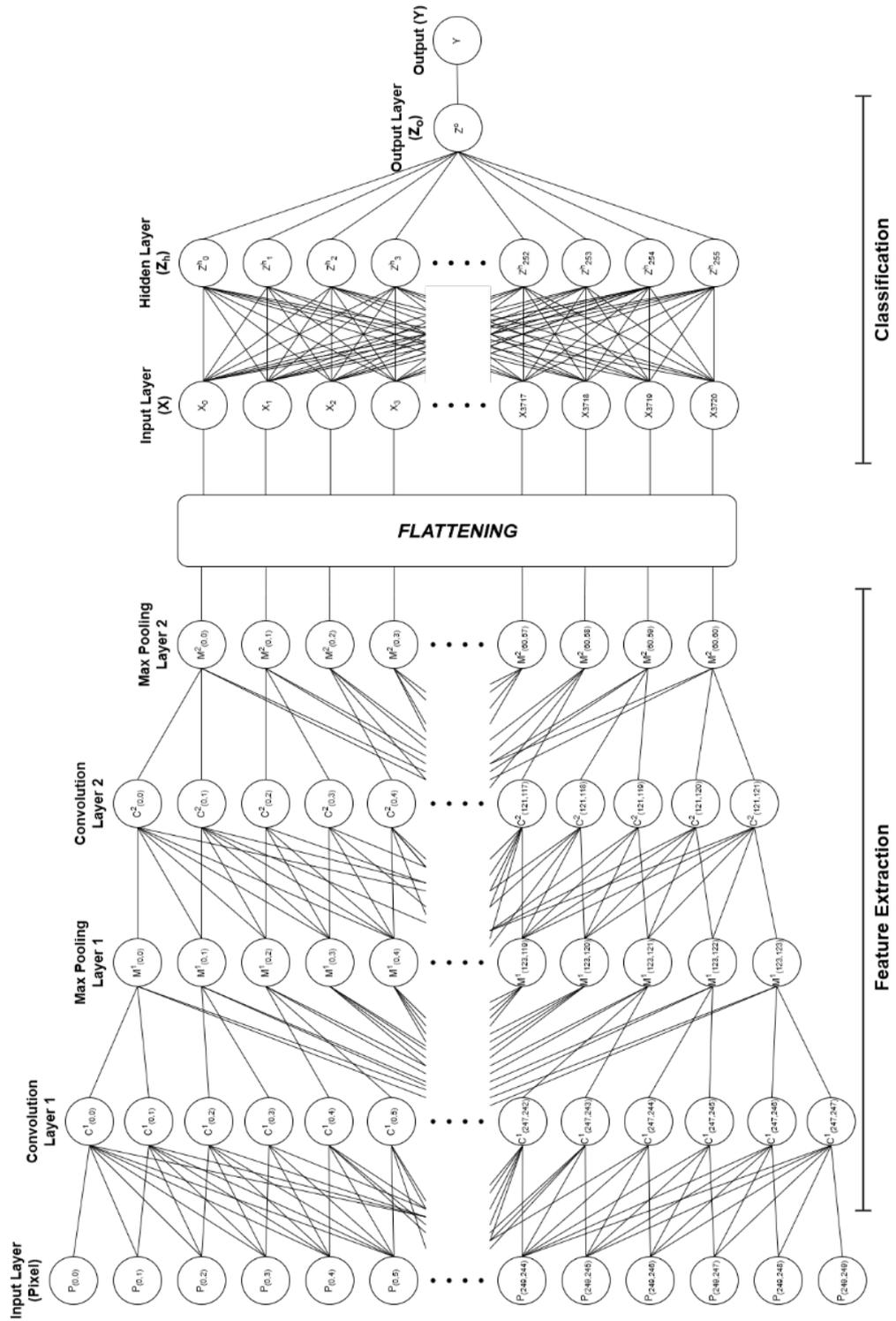
*Horizontal flip* adalah proses membalik citra secara horizontal (dari kiri ke kanan). Teknik ini menambah variasi orientasi dari objek di dalam citra, sehingga model mampu mengenali pola objek yang sama meskipun orientasinya terbalik. Berikut contoh hasil dari proses *horizontal flip* dapat dilihat pada Gambar 3.7.



Gambar 3.7 *Horizontal Flip*

### 3.4 Implementasi Metode

Metode yang digunakan untuk mengklasifikasikan citra kebakaran hutan dalam penelitian ini adalah *Convolutional Neural Network* (CNN), yang merujuk pada arsitektur yang digunakan oleh Abror, (2019). Bentuk arsitektur yang digunakan dalam penelitian ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Arsitektur CNN

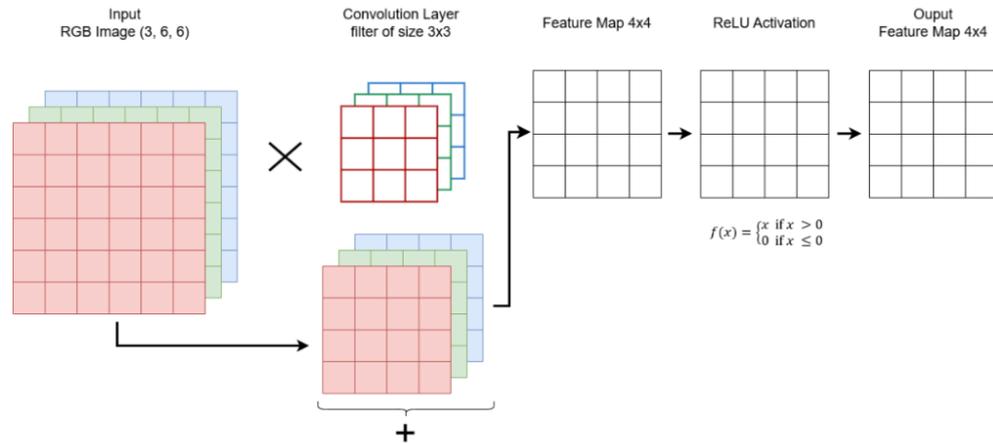
Implementasi metode ini dibagi menjadi dua tahapan utama, yaitu tahap ekstraksi fitur dan tahap klasifikasi. Tahap ekstraksi fitur berfokus pada pengenalan karakteristik visual penting dari citra kebakaran hutan, dimana proses tersebut terjadi pada *input layer*, *2 convolutional layer*, dan *2 max-pooling layer*. Sementara tahap klasifikasi bertanggung jawab untuk menentukan apakah gambar tersebut termasuk kategori "*fire*" atau "*non-fire*" yang terjadi pada *fully connected layers*.

### **3.4.1 Input Layer**

Lapisan awal berperan sebagai *input layer* yang menerima data piksel dari citra dengan dimensi  $250 \times 250$  piksel dalam format RGB, sehingga ukuran input menjadi (250, 250, 3). Data ini diperoleh setelah melalui tahap *preprocessing* citra kebakaran hutan, di mana setiap nilai piksel dinormalisasi ke dalam rentang 0 hingga 1 untuk mengurangi skala data dan membantu dalam proses pelatihan model.

### **3.4.2 Convolution Layer**

Pada lapisan konvolusi, sebuah kernel akan diterapkan pada input gambar. Setiap nilai dalam kernel dikalikan dengan nilai piksel yang bersesuaian pada citra input, dan hasilnya dijumlahkan untuk membentuk satu nilai pada *output*. Kernel tersebut bergerak secara sistematis di sepanjang citra, baik secara horizontal maupun vertikal, untuk mengekstrak fitur dari seluruh bagian gambar. Proses ini menghasilkan *feature map*, yang merupakan peta fitur dari citra input berdasarkan kernel yang digunakan. Proses konvolusi dapat dilihat pada Gambar 3.9.



Gambar 3.9 Proses Konvolusi

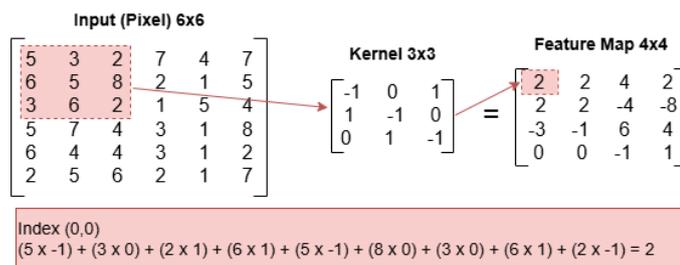
Operasi dari proses konvolusi tersebut dapat dihitung menggunakan Persamaan 3.2 (Goodfellow et al., 2016).

$$F(i, j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I(i + m, j + n)K(m, n) \tag{3.2}$$

Keterangan:

- $F(i, j)$  = output konvolusi dengan indeks ke-  $(i, j)$
- $k$  = ukuran kernel
- $I(i + m, j + n)$  = nilai piksel dari citra input pada posisi  $(i, j)$
- $K(m, n)$  = nilai kernel pada posisi  $(m, n)$
- $m, n$  = koordinat pada kernel
- $i, j$  = koordinat pada *feature map*

Berikut contoh perhitungan manual proses konvolusi berdasarkan Gambar 3.9, dimana terdapat input gambar berukuran 6×6 dan kernel berukuran 3×3, disajikan pada Gambar 3.10.



Gambar 3.10 Perhitungan Proses Konvolusi

Ukuran *output* dari *feature map* bergantung pada ukuran input, kernel, *stride*, dan juga *padding* yang digunakan. Pada penelitian ini, kernel yang digunakan adalah kernel berukuran 3x3 dengan *stride* 1 dan *padding* 0. Ukuran *output* pada *feature map* dapat diketahui menggunakan Persamaan 3.3 (Suyanto, 2018).

$$Feature\ Map = \frac{(I - K + 2P)}{S} + 1 \quad (3.3)$$

Keterangan:

$I$  = dimensi input gambar  
 $K$  = ukuran kernel  
 $P$  = padding  
 $S$  = stride

Setelah melewati operasi konvolusi, fungsi aktivasi *Rectified Linear Unit* (ReLU) akan digunakan untuk mengubah semua nilai negatif dalam *feature map* menjadi nol. Fungsi aktivasi ReLU dapat dilihat dalam Persamaan 3.4.

$$f(x) = \max(0, x) \quad (3.4)$$

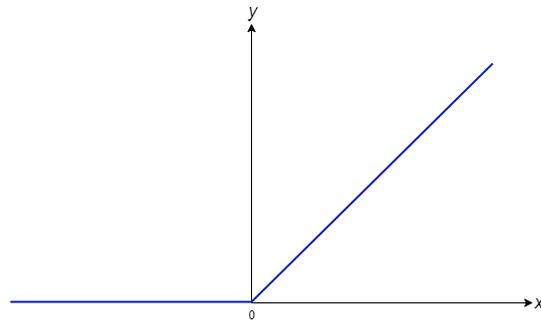
Keterangan:

$f(x)$  = fungsi aktivasi ReLU  
 $x$  = input

Jika nilai input lebih besar dari nol, maka *output* akan sama dengan nilai input itu sendiri. Sedangkan jika input kurang dari nol, maka nilai *ouput* akan berubah menjadi nol. Secara matematis, fungsi ReLU juga dapat dipahami dengan Persamaan 3.5.

$$f(x) = \begin{cases} x & \text{jika } x > 0 \\ 0 & \text{jika } x \leq 0 \end{cases} \quad (3.5)$$

Grafik fungsi aktivasi ReLU dapat dilihat pada Gambar 3.11.



Gambar 3.11 Grafik fungsi aktivasi ReLU

Contoh penerapan fungsi aktivasi ReLU terhadap *feature map* dapat dilihat pada Gambar 3.12.

$$\begin{array}{c}
 \text{Feature Map 4x4} \\
 \left[ \begin{array}{cccc}
 2 & 2 & 4 & 2 \\
 2 & 2 & -4 & -8 \\
 -3 & -1 & 6 & 4 \\
 0 & 0 & -1 & 1
 \end{array} \right] \xrightarrow{\text{ReLU Activation}} \left[ \begin{array}{cccc}
 2 & 2 & 4 & 2 \\
 2 & 2 & 0 & 0 \\
 0 & 0 & 6 & 4 \\
 0 & 0 & 0 & 1
 \end{array} \right]
 \end{array}$$

Gambar 3.12 Penerapan Fungsi Aktivasi ReLU

### 3.4.3 Max Pooling Layer

Setelah lapisan konvolusi, *pooling layer* diterapkan untuk mengurangi dimensi *output* dan mengurangi kompleksitas komputasi. Jenis *pooling* yang digunakan pada penelitian ini adalah *max pooling* dengan ukuran kernel  $2 \times 2$  dan stride 2. *Max pooling* bekerja dengan cara mengambil nilai maksimum dari area tertentu pada *feature map*. Perhitungan *max pooling* dapat dilihat pada Persamaan 3.6.

$$m(x, y) = \max \{x_{i+m, j+n} \mid 0 \leq m < k, 0 \leq n < k\} \quad (3.6)$$

Keterangan:

$m(x, y)$  = *output* pada posisi  $(x, y)$  setelah operasi *max pooling*

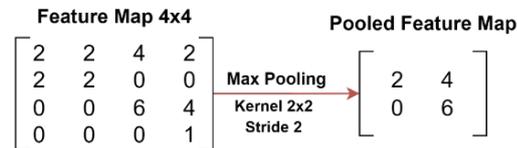
$x_{i+m, j+n}$  = nilai piksel pada posisi  $(i + m, j + n)$  dalam jendela *pooling*

$i, j$  = koordinat pada *feature map* input

$m, n$  = variabel untuk menghitung posisi relatif di dalam jendela *pooling*

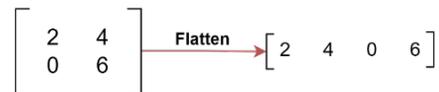
$k$  = ukuran *max pooling*

Contoh penerapan *max pooling* terhadap *feature map* dapat dilihat pada Gambar 3.13.



Gambar 3.13 Penerapan *Max Pooling*

Sebelum input hasil ekstraksi masuk ke lapisan *fully connected*, input tersebut diubah menjadi bentuk vektor satu dimensi melalui proses *flattening*. Berikut hasil dari proses *flattening* dapat dilihat pada Gambar 3.14.



Gambar 3.14 Penerapan Proses *Flattening*

#### 3.4.4 *Fully Connected Layers*

Lapisan *fully connected* terdiri dari sejumlah neuron dimana setiap neuron tersebut terhubung ke semua neuron pada lapisan sebelumnya. Setiap neuron menghitung *output*-nya dengan menerima input dari lapisan sebelumnya dan bobot yang telah ditentukan. Tujuan utama dari lapisan ini adalah untuk menggabungkan dan menginterpretasikan fitur yang telah diekstraksi melalui lapisan konvolusi dan *pooling* menjadi *output* yang lebih terstruktur, yaitu kelas prediksi. Berikut lapisan-lapisan yang terdapat dalam lapisan *fully connected* beserta jumlah neuron yang terdapat dalam masing-masing lapisan:

1. *Input Layer*

*Input layer* pada lapisan *fully connected* merupakan lapisan pertama berupa data input yang diubah ke dalam bentuk vektor satu dimensi. Neuron pada lapisan ini menyesuaikan pada *feature map* yang sudah melalui proses *flattening*.

## 2. *Hidden Layer*

*Hidden layer* merupakan lapisan diantara *input layer* dan *output layer*, dimana pada lapisan tersebut terdapat proses aktivasi menggunakan fungsi aktivasi ReLU. Neuron yang digunakan pada lapisan ini berjumlah 256 neuron.

## 3. *Output Layer*

*Output layer* merupakan lapisan terakhir dalam kumpulan lapisan *fully connected*. *Output layer* memberikan prediksi akhir atau hasil dari model, dimana pada penelitian ini digunakan fungsi aktivasi *sigmoid*. Hanya terdapat satu neuron pada lapisan ini, karena fungsi klasifikasi yang digunakan untuk memprediksi kelas adalah fungsi *threshold*, yang hanya membutuhkan satu nilai *output*.

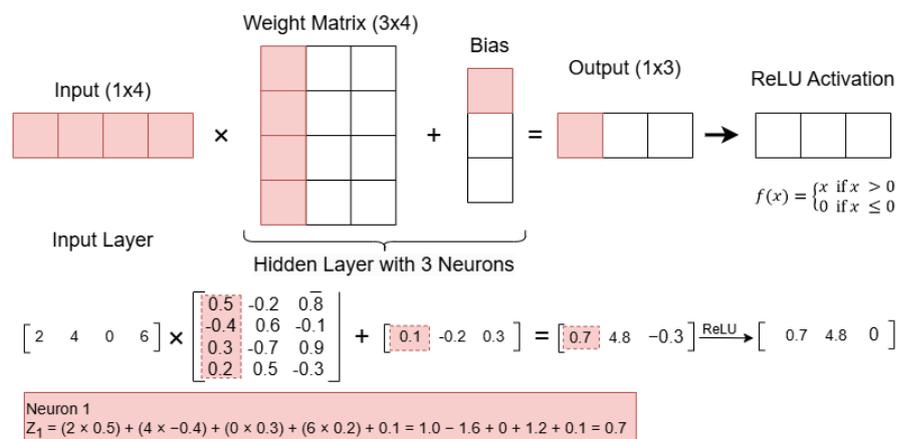
Langkah yang terjadi pada lapisan *fully connected* dimulai dengan meneruskan hasil akhir *feature map* yang telah di-*flatten* dari *input layer* ke *hidden layer*. Setiap nilai dalam vektor tersebut dikalikan dengan bobot dan ditambahkan dengan bias tertentu yang terhubung dengan setiap neuron di *hidden layer*. Operasi yang digunakan dapat dilihat pada Persamaan 3.7.

$$z_j = \sum_{i=1}^n x_i \cdot w_{ij} + b_j \quad (3.7)$$

Keterangan:

- $z_j$  = *output* neuron ke-j pada lapisan *fully connected*
- $w_{ij}$  = bobot neuron ke-i pada *Input Layer* dan neuron ke-j pada *hidden layer*
- $x_i$  = nilai input
- $b_j$  = bias neuron ke-j
- $n$  = jumlah neuron pada lapisan sebelumnya

Setelah proses tersebut, nilai hasil perkalian bobot dan penambahan bias ini akan dihitung menggunakan fungsi aktivasi ReLU seperti pada Persamaan 3.4, dimana setiap *output* yang bernilai negatif akan diubah menjadi 0. Berikut contoh penerapan seluruh proses pada lapisan *fully connected* yang terjadi antara *input layer* dengan *hidden layer* ditampilkan pada Gambar 3.15.



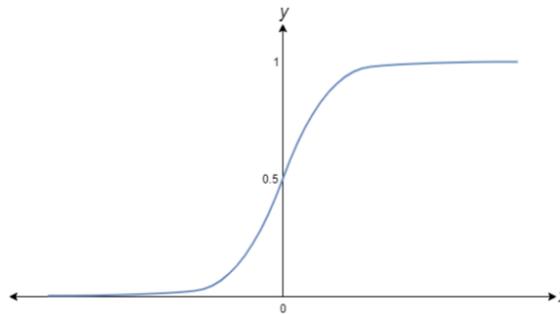
Gambar 3.15 Penerapan *Input Layer* ke *Hidden Layer*

Proses serupa dilakukan pada *output layer*, dimana *output* yang sudah diaktivasi pada lapisan sebelumnya digunakan sebagai input perhitungan *output layer*. Setiap neuron pada *output layer* menerima input dari *hidden layer* yang sudah diproses. Nilai tersebut kembali dikalikan dengan bobot dan ditambah bias, kemudian melalui fungsi aktivasi *sigmoid*. Fungsi aktivasi *sigmoid* mengubah nilai input menjadi nilai antara 0 dan 1 sebagai representasi probabilitas untuk kelas “*fire*” dan “*non-fire*”. Fungsi ini sering digunakan dalam *Neural Network* untuk menghasilkan *output* dari klasifikasi biner yang digunakan sebagai prediksi kelas. Rumus fungsi aktivasi *sigmoid* dapat dilihat pada Persamaan 3.8 dan grafik fungsi aktivasi *sigmoid* pada Gambar 3.16.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.8)$$

Keterangan:

- $\sigma(x)$  = *output* dari fungsi *sigmoid* untuk input  $x$
- $e$  = bilangan euler, dengan nilai sekitar 2.71828
- $x$  = nilai dari input hasil perhitungan bobot dan bias pada *layer*



Gambar 3.16 Grafik Fungsi Aktivasi *Sigmoid*

Dapat dilihat pada Gambar 3.16, bahwa grafik fungsi aktivasi *sigmoid* memiliki bentuk “S”. Ketika input  $x$  bernilai positif, maka *output* dari fungsi aktivasi *sigmoid* akan mendekati 1. Apabila input  $x$  bernilai negatif, maka *output* dari fungsi aktivasi akan mendekati 0.

Hasil dari nilai fungsi *sigmoid* kemudian melalui proses penentuan kelas prediksi menggunakan fungsi *threshold*, dimana hasil *output* dari fungsi *sigmoid* dibandingkan dengan nilai ambang batas atau *threshold* yang bernilai 0.5. Jika *output* lebih besar dari 0.5, maka prediksi diklasifikasikan sebagai kelas "*non-fire*" [1], sedangkan jika kurang dari atau sama dengan 0.5, maka diklasifikasikan sebagai kelas "*fire*" [0]. Proses tersebut merupakan langkah akhir yang terjadi pada tahapan implementasi metode, dimana setiap piksel akan melalui tahapan tersebut

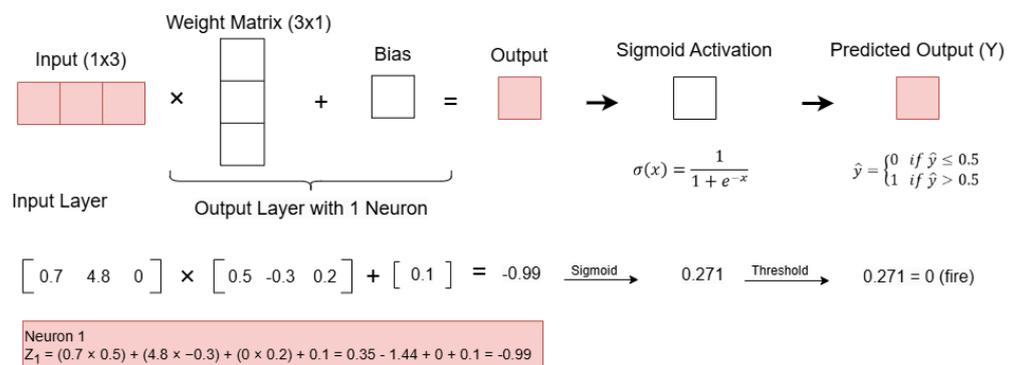
untuk mendapatkan hasil prediksi. Berikut persamaan yang digunakan untuk penentuan kelas prediksi menggunakan fungsi *threshold*:

$$\hat{y} = \begin{cases} 0 & \text{jika } \hat{y} \leq 0.5 \\ 1 & \text{jika } \hat{y} > 0.5 \end{cases} \quad (3.9)$$

Keterangan:

$\sigma(x)$  = output dari fungsi *sigmoid* untuk input  $x$

Berikut adalah contoh penerapan seluruh proses yang terjadi pada *hidden layer* ke *output layer*, dimulai dari perhitungan antara bobot neuron dengan input, hingga penentuan klasifikasi menggunakan fungsi *threshold*. Proses tersebut ditampilkan pada Gambar 3.17.



Gambar 3.17 Penerapan *Hidden Layer* ke *Output Layer*

Setelah model CNN melakukan penentuan klasifikasi citra kebakaran hutan, perhitungan error dilakukan menggunakan fungsi *loss binary cross-entropy*. Fungsi tersebut digunakan karena pada penelitian ini hanya terdapat dua kelas, yaitu *fire* dan *non-fire*. Perhitungan *loss* menggunakan fungsi *binary cross-entropy* dilakukan menggunakan Persamaan 3.10.

$$BCE Loss = -\frac{1}{n} \sum_{i=1}^n [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (3.10)$$

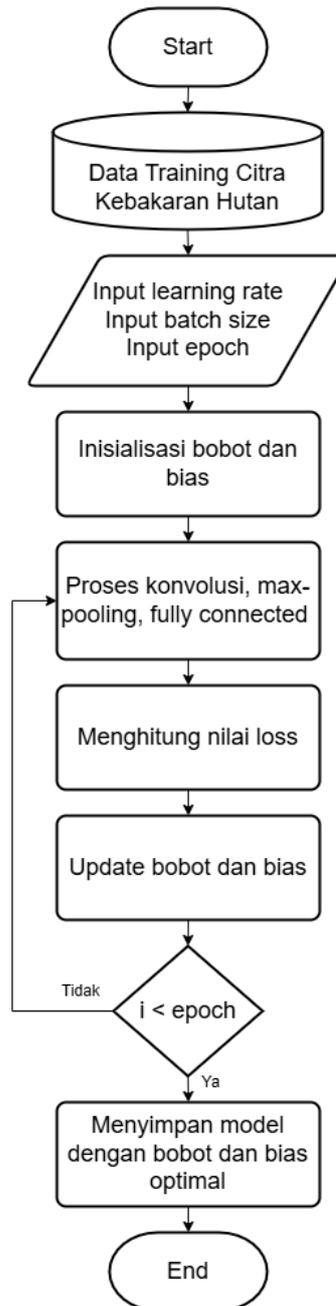
Keterangan:

- $n$  = jumlah *pixel*
- $y_i$  = label asli dari sampel ke- $i$
- $\hat{y}_i$  = probabilitas prediksi model untuk sampel ke- $i$

Jika prediksi lebih dekat dengan nilai target, *loss* yang dihasilkan akan rendah, mengindikasikan model telah memprediksi dengan lebih akurat. Sebaliknya, jika hasil prediksi jauh dari target, nilai *loss* akan lebih tinggi.

### 3.5 Proses *Training*

Proses *training* pada model *Convolutional Neural Network* (CNN) adalah langkah-langkah yang dilakukan untuk melatih model mengenali pola dari data. Dalam konteks klasifikasi kebakaran hutan berbasis citra, model CNN dilatih menggunakan dataset gambar untuk menghasilkan *output* berupa prediksi yang benar berdasarkan input yang diberikan. Tujuan dari *training* adalah agar model belajar membuat prediksi yang akurat dengan mengoptimalkan bobot dan bias di setiap lapisan melalui proses pelatihan. Proses *training* dapat dilihat pada diagram alur Gambar 3.18.

Gambar 3.18 Alur Proses *Training*

Proses *training* dimulai dengan memanggil data latih yang akan digunakan dan juga menentukan nilai *hyperparameter* seperti *learning rate*, *batch size*, dan jumlah *epoch*. Pengertian dari setiap *hyperparameter* ditampilkan pada Tabel 3.2.

Tabel 3.2 *Hyperparameter*

<b><i>Hyperparameter</i></b>	<b>Pengertian</b>
<i>Learning Rate</i>	Parameter yang menentukan seberapa besar langkah yang diambil saat meng- <i>update</i> bobot
<i>Batch Size</i>	Ukuran <i>batch</i> menentukan jumlah sampel data yang digunakan dalam pelatihan sebelum melakukan <i>update</i> bobot
<i>Epoch</i>	Jumlah iterasi di mana seluruh dataset dilalui oleh model untuk dipelajari

Proses *training* dilanjutkan dengan inisialisasi bobot dan bias, kemudian data *training* berupa gambar melalui *feedforward* melewati lapisan konvolusi, *pooling*, dan *fully connected* untuk menghasilkan prediksi. Setelah prediksi dibuat, *loss* dihitung dengan membandingkan hasil prediksi dengan label sebenarnya menggunakan fungsi *loss binary cross-entropy*. Model kemudian melakukan *backpropagation* untuk menghitung gradien dan memperbarui. Proses ini diulang dalam beberapa *epoch* sampai model mencapai kinerja yang diinginkan.

### 3.5.1 *Feedforward*

Proses *feedforward* adalah tahap dalam *neural network* dimana input diproses maju melalui jaringan dari lapisan input hingga lapisan *output* tanpa ada aliran balik. Pada setiap lapisan, input dari lapisan sebelumnya dikalikan dengan bobot, ditambahkan bias, kemudian hasilnya diterapkan fungsi aktivasi. *Output* akhir dari proses ini adalah prediksi yang dihasilkan oleh model. Berikut langkah-langkah yang dilakukan pada proses *feedforward*:

1. Ekstraksi fitur

Input berupa gambar melalui serangkaian proses pada lapisan konvolusi dan lapisan *pooling* untuk ekstraksi fitur dan mendapatkan *output* berupa *feature map*

yang merepresentasikan fitur seperti tepi dan tekstur. Hasil berupa *feature map* dapat dihitung menggunakan Persamaan 3.11.

$$Feature\ Map = MaxPooling(ReLU(Convolution(X,K))) \quad (3.11)$$

Keterangan:

*MaxPooling()* = proses *max pooling*  
*ReLU()* = proses fungsi aktivasi ReLU  
*Convolution()* = proses konvolusi  
*X* = input gambar  
*K* = kernel

## 2. Perhitungan input pada *hidden layer*

Setiap neuron pada *hidden layer* menerima nilai dari *input layer* berupa *feature map* yang sudah diubah menjadi vektor satu dimensi melalui proses *flattening*. Nilai tersebut kemudian dikalikan dengan bobot dan ditambahkan dengan bias. Perhitungan tersebut dapat dilihat pada Persamaan 3.12.

$$Z_h = X \cdot W_1 + b_1 \quad (3.12)$$

Keterangan:

$Z_h$  = *output* hasil input dikali bobot ditambah bias  
*X* = input data berupa vektor satu dimensi  
 $W_1$  = bobot antara *input layer* dengan *hidden layer*  
 $b_1$  = bias pada *hidden layer*

## 3. Fungsi aktivasi pada *hidden layer*

*Output* dari *hidden layer* kemudian diaktivasi menggunakan fungsi ReLU dengan Persamaan 3.13.

$$\begin{aligned} \sigma(Z_h) &= \max(0, Z_h) \\ a_h &= \sigma(Z_h) \end{aligned} \quad (3.13)$$

Keterangan:

$\sigma(Z_h)$  = *output* setelah aktivasi

#### 4. Perhitungan input pada *output layer*

Pada *output layer*, input lapisan sebelumnya yang sudah diaktivasi juga dikalikan dengan bobot dan ditambah dengan bias dengan Persamaan 3.14.

$$Z_o = a_h \cdot W_2 + b_2 \quad (3.14)$$

Keterangan:

$Z_o$  = *output* hasil input dikali bobot ditambah bias  
 $a_h$  = input data hasil dari aktivasi *output hidden layer* sebelumnya  
 $W_2$  = bobot antara *hidden layer* dengan *output layer*  
 $b_2$  = bias pada *output layer*

#### 5. Fungsi aktivasi pada *output layer*

Setelah mendapatkan nilai *output layer* yang sudah dikalikan dengan bobot dan ditambah dengan bias, proses berikutnya adalah mengaktivasi nilai tersebut menggunakan fungsi aktivasi *sigmoid* menggunakan Persamaan 3.15.

$$a_o = \frac{1}{1 + e^{-Z_o}} \quad (3.15)$$

$$\hat{y} = a_o$$

Keterangan:

$a_o$  = *output* dari fungsi *sigmoid* (hasil prediksi input)

#### 6. *Output* prediksi

Hasil akhir dengan aktivasi *sigmoid* menampilkan nilai probabilitas antara 0 dan 1. Penentuan kelas *output* dilakukan menggunakan fungsi *threshold*, dimana apabila nilai *output* lebih besar dari 0.5, maka prediksi akan dianggap sebagai kelas 1 yaitu “*non-fire*”. Sedangkan apabila nilai *output* lebih kecil dari 0.5, maka prediksi dianggap sebagai kelas 0 yaitu “*fire*”. Cara menentukan kelas pada *output* dapat dilihat pada Persamaan 3.16.

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{y} \leq 0.5 \\ 1 & \text{if } \hat{y} > 0.5 \end{cases} \quad (3.16)$$

## 7. Perhitungan *Loss*

Setelah prediksi didapatkan, nilai eror dihitung menggunakan *binary cross-entropy*. Eror yang didapat akan menunjukkan jumlah kesalahan model dalam melakukan prediksi. Perhitungan *binary cross-entropy* dapat dilihat pada Persamaan 3.17.

$$BCE\ Loss = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})) \quad (3.17)$$

Keterangan:

- $n$  = jumlah *pixel*
- $y$  = label asli dari sampel ke- $i$
- $\hat{y}$  = probabilitas prediksi model untuk sampel ke- $i$

### 3.5.2 *Backpropagation*

Proses *backpropagation* dilakukan untuk memperbarui bobot dan bias pada jaringan dengan menghitung gradien dari fungsi *loss* terhadap setiap parameter bobot dan bias. Hal tersebut berguna untuk mengurangi nilai *loss* atau eror pada model. Berikut langkah-langkah yang dilakukan pada proses *backpropagation*:

#### 1. Perhitungan eror pada *output layer*

Eror pada *output layer* didapat dengan menghitung selisih antara *output* prediksi  $\hat{y}$  dengan target aktual  $y$ . Rumus eror tersebut yaitu:

$$\delta_2 = \hat{y} - y \quad (3.18)$$

Keterangan:

- $\delta_2$  = eror pada *output layer*
- $\hat{y}$  = *output* prediksi
- $y$  = nilai target sebenarnya

## 2. Perhitungan gradien untuk *output layer*

Gradien dari eror terhadap bobot dan bias pada *output layer* dapat dihitung dengan Persamaan 3.19 dan 3.20.

$$\frac{\partial L}{\partial W_2} = \delta_2 \cdot a_h^T \quad (3.19)$$

$$\frac{\partial L}{\partial b_2} = \delta_2 \quad (3.20)$$

Keterangan:

$\frac{\partial L}{\partial W_2}$  = gradien eror terhadap bobot pada *output layer*  
 $\frac{\partial L}{\partial b_2}$  = gradien eror terhadap bias pada *output layer*  
 $a_h^T$  = nilai transpose output dari *hidden layer*

## 3. Perhitungan eror pada *hidden layer*

Eror pada *hidden layer* dihitung dengan menjumlahkan eror dari *output layer* yang diteruskan kembali ke *hidden layer*, dikalikan dengan turunan fungsi aktivasi ReLU pada *hidden layer*.

$$\delta_1 = (W_2^T \cdot \delta_2) \cdot \sigma'(Z_h) \quad (3.21)$$

Keterangan:

$\delta_1$  = eror pada *hidden layer*  
 $W_2^T$  = nilai transpose bobot dari *hidden layer* ke *output layer*  
 $\sigma'(Z_h)$  = turunan dari fungsi aktivasi pada *hidden layer*

## 4. Menghitung gradien bobot dan bias untuk *hidden layer*

Gradien dari eror terhadap bobot dan bias pada *output layer* dapat dihitung dengan Persamaan 3.22 dan 3.23.

$$\frac{\partial L}{\partial W_1} = \delta_1 \cdot X^T \quad (3.22)$$

$$\frac{\partial L}{\partial b_1} = \delta_1 \quad (3.23)$$

Keterangan:

$$\begin{aligned} \frac{\partial L}{\partial W_1} &= \text{gradien eror terhadap bobot pada } \textit{hidden layer} \\ \frac{\partial L}{\partial W_1} &= \text{gradien eror terhadap bias pada } \textit{hidden layer} \\ X^T &= \text{nilai transpose } \textit{output} \text{ dari } \textit{input layer} \end{aligned}$$

### 5. Update bobot dan bias

Setelah nilai gradien didapatkan, bobot dan bias diperbarui menggunakan *learning rate*. Berikut rumus pada Persamaan 3.24, 3.25, 3.26, dan 3.27 yang digunakan untuk memperbarui nilai bobot dan bias.

$$W_2 = W_2 - \eta \cdot \frac{\partial L}{\partial W_2} \quad (3.24)$$

$$W_1 = W_1 - \eta \cdot \frac{\partial L}{\partial W_1} \quad (3.25)$$

$$b_1 = b_1 - \eta \cdot \frac{\partial L}{\partial b_1} \quad (3.26)$$

$$b_2 = b_2 - \eta \cdot \frac{\partial L}{\partial b_2} \quad (3.27)$$

Keterangan:

$$\eta = \textit{learning rate}$$

Proses *training* akan terus berjalan sampai batas *epoch* yang ditentukan sebelumnya. Ketika iterasi proses *training* berhenti, itu menandakan bahwa bobot dan bias optimal telah diperoleh dan akan disimpan untuk digunakan dalam proses *testing*. Setelah penyimpanan, model yang telah dilatih perlu diuji untuk memastikan performanya dapat diandalkan pada data yang belum pernah dilihat sebelumnya, serta untuk mengevaluasi kemampuan model dalam menggeneralisasi informasi.

### 3.6 Skenario Pengujian

Dalam penelitian ini, pengujian diterapkan dengan menyusun dua jenis skenario untuk mengevaluasi performa model berdasarkan variasi pembagian data dan perubahan arsitektur CNN. Masing-masing skenario akan melalui proses *training* dan *testing* untuk menilai pengaruh parameter yang diuji terhadap performa model. Pengujian ini bertujuan untuk mengetahui pengaruh pembagian data dan perubahan arsitektur model dalam menentukan konfigurasi yang optimal sehingga model dapat mencapai hasil klasifikasi kebakaran hutan terbaik.

Skenario pertama berfokus pada pengaruh pembagian data menjadi data *training* dan *testing*. Terdapat tiga variasi pembagian data yang diuji, dengan proporsi 70:30, 80:20, dan 90:10. Pembagian dataset ini dilakukan untuk menguji bagaimana variasi jumlah data pada masing-masing kategori mempengaruhi performa model dalam melakukan klasifikasi kebakaran hutan. Dalam pengujian ini, bentuk arsitektur yang digunakan tetap dengan 2 lapisan konvolusi dan metode klasifikasi *threshold*. Hal ini dilakukan untuk memastikan bahwa perbedaan performa model sepenuhnya disebabkan oleh variasi proporsi data yang digunakan pada *training* dan *testing*. Pembagian dataset ini disajikan dalam Tabel 3.3.

Tabel 3.3 Pembagian Dataset untuk Skenario Uji 1

Skenario	Data train	Data test
1	70%	30%
2	80%	20%
3	90%	10%

Skenario kedua berfokus pada pengaruh perubahan arsitektur CNN yang mencakup jumlah lapisan konvolusi dan jenis metode klasifikasi yang digunakan. Pada skenario ini, proporsi data ditetapkan sebesar 80:20 untuk memastikan bahwa

perbedaan performa model disebabkan oleh perubahan arsitektur dan bukan karena perbedaan jumlah data pada *training* dan *testing*.

Dalam arsitektur yang telah diusulkan, model menggunakan dua lapisan konvolusi. Untuk mengevaluasi apakah penambahan lapisan konvolusi dapat meningkatkan performa model, dilakukan pengujian dengan menambahkan jumlah lapisan konvolusi menjadi tiga. Pembagian jumlah lapisan konvolusi ini disajikan dalam Tabel 3.4.

Tabel 3.4 Jumlah Lapisan Konvolusi untuk Skenario Uji 2

Skenario	Jumlah lapisan konvolusi
A	2
B	3

Metode yang digunakan untuk menentukan kelas prediksi pada tahap klasifikasi juga dapat memengaruhi hasil akhir model. Pada bagian implementasi metode di BAB III, peneliti mengusulkan metode *threshold* sebagai metode yang digunakan untuk menentukan kelas prediksi. Namun penggunaan metode lain berupa *argmax* juga diajukan sebagai bahan skenario pengujian untuk mengetahui apakah ada perbedaan performa antara kedua metode klasifikasi tersebut dalam menentukan kelas prediksi. Skenario penggunaan metode klasifikasi ini disajikan dalam Tabel 3.5.

Tabel 3.5 Metode Klasifikasi untuk Skenario Uji 2

Skenario	Metode klasifikasi
Y	<i>Threshold</i>
Z	<i>Argmax</i>

Berdasarkan dua jenis skenario tersebut, maka didapatkan kombinasi skenario pengujian yang akan diterapkan pada model. Dimana dua lapisan konvolusi dan metode klasifikasi *threshold* akan digunakan pada skenario uji

pembagian dataset, sedangkan proporsi data 80:20 akan ditetapkan pada skenario uji perubahan arsitektur CNN. Kombinasi yang digunakan pada penelitian ini disajikan dalam Tabel 3.6.

Tabel 3.6 Kombinasi Skenario Pengujian

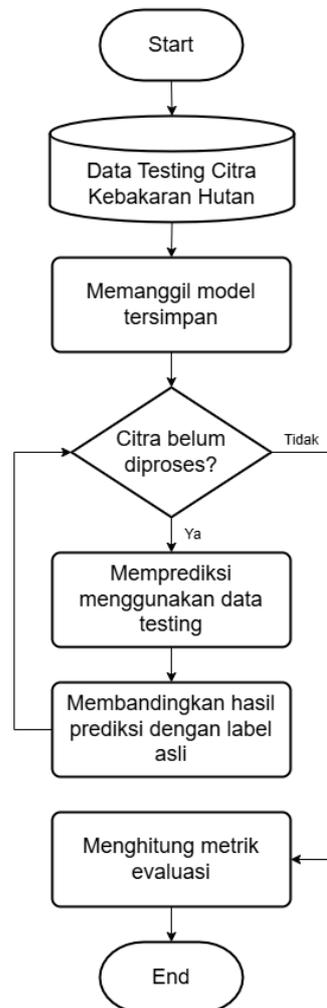
Jenis Skenario	Skenario	Variasi Parameter
Pembagian Data	1-A-Y	70:30
Pembagian Data	2-A-Y	80:20
Pembagian Data	3-A-Y	90:10
Perubahan Arsitektur	2-A-Y	2 Lapisan Konvolusi + <i>Threshold</i>
Perubahan Arsitektur	2-A-Z	2 Lapisan Konvolusi + <i>Argmax</i>
Perubahan Arsitektur	2-B-Y	3 Lapisan Konvolusi + <i>Threshold</i>
Perubahan Arsitektur	2-B-Z	3 Lapisan Konvolusi + <i>Argmax</i>

Nilai dari setiap *hyperparameter* juga ditentukan untuk diimpelentasikan dalam proses *training* dengan nilai yang sama pada setiap skenario pengujian. Berdasarkan arsitektur yang merujuk pada penelilitan Abror, (2019), Berikut nilai dari setiap *hyperparameter* yang akan digunakan pada penelitian ini ditampilkan pada Tabel 3.7.

Tabel 3.7 Nilai Hyperparameter

<i>Hyperparameter</i>	Nilai
<i>Learning Rate</i>	0.001
<i>Batch Size</i>	32
<i>Epoch</i>	25

Setelah model dilatih menggunakan data *training* untuk mempelajari citra, selanjutnya proses *testing* dilakukan untuk mengevaluasi performa akhir dan kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Sehingga berdasarkan skenario uji yang diusulkan, dapat dilihat model mana yang menghasilkan performa terbaik untuk digunakan pada implementasi nyata. Proses *testing* dapat dilihat pada diagram alur Gambar 3.19.



Gambar 3.19 Alur Proses *Testing*

Proses *testing* dimulai dengan memanggil data *testing* citra kebakaran hutan dan model tersimpan dengan bobot dan bias optimal yang diperoleh dari proses *training* sebelumnya. Selanjutnya nilai piksel pada setiap citra dalam data *testing* diprediksi untuk menghasilkan nilai prediksi antara kebakaran dan tidak kebakaran. Kemudian hasil prediksi tersebut dibandingkan dengan label asli untuk menghitung metrik evaluasi.

Evaluasi model dilakukan untuk menilai seberapa baik model dalam melakukan klasifikasi. Metode evaluasi yang digunakan pada penelitian ini adalah

*confussion matrix*, yang sudah sangat umum digunakan dalam penelitian klasifikasi. *Confusion matrix* merupakan tabel yang menampilkan hasil prediksi model terhadap data uji dalam empat kategori utama, yang memberikan gambaran menyeluruh mengenai performa model (Suryadewiansyah *et al.*, 2020). *Confusion matrix* dalam klasifikasi biner berisi empat nilai sebagai berikut:

1. *True Positive* (TP), yaitu dimana model memprediksi kelas positif dan prediksi tersebut benar.
2. *True Negative* (TN), yaitu dimana model memprediksi kelas negatif dan prediksi tersebut benar.
3. *False Positive* (FP), yaitu dimana model memprediksi kelas positif, namun sebenarnya kelas tersebut negatif.
4. *False Negative* (FN), yaitu dimana model memprediksi kelas negatif, namun sebenarnya kelas tersebut positif.

Kemudian berdasarkan hasil dari *confussion matrix* yang didapat, selanjutnya evaluasi lain yaitu akurasi, presisi, *recall*, dan *F1-score* dapat dihitung menggunakan persamaan berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.28)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3.29)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.30)$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (3.31)$$

Untuk memastikan keandalan model dalam menghadapi variasi data dan menghindari ketergantungan pada satu subset data tertentu, proses validasi juga dilakukan dengan penerapan metode *K-Fold Cross Validation*. *K-fold* dilakukan terhadap keseluruhan dataset dengan tujuan untuk mengevaluasi *robustness* model. Validasi menggunakan *K-fold* difokuskan pada dua skenario, yaitu skenario 2-A-Y dan skenario 2-B-Z yang dipilih karena mewakili konfigurasi arsitektur model yang kontras. Skenario 2-A-Y menggunakan arsitektur dengan struktur dasar, sedangkan skenario 2-B-Z mengadopsi perubahan signifikan pada arsitektur model. Pemilihan dua skenario ini memungkinkan evaluasi model dilakukan secara lebih menyeluruh terhadap variasi desain dan distribusi data.

Dalam penelitian ini, data dibagi menjadi 5 bagian (*fold*), di mana masing-masing *fold* secara bergantian digunakan sebagai data uji, sementara empat *fold* lainnya digunakan sebagai data pelatihan. Proses ini diulang sebanyak lima kali hingga seluruh data pernah menjadi data uji. Hasil evaluasi dari setiap *fold* kemudian dirata-ratakan untuk memperoleh metrik performa yang lebih stabil dan menyeluruh. Evaluasi pada setiap *fold* mencakup metrik seperti akurasi dan *loss*, dan hasilnya disajikan dalam bentuk grafik untuk menggambarkan tren performa selama proses pelatihan. Dengan pendekatan ini, validasi model menjadi lebih komprehensif karena mempertimbangkan variasi distribusi data yang mungkin terjadi di berbagai subset.

## BAB IV

### HASIL DAN PEMBAHASAN

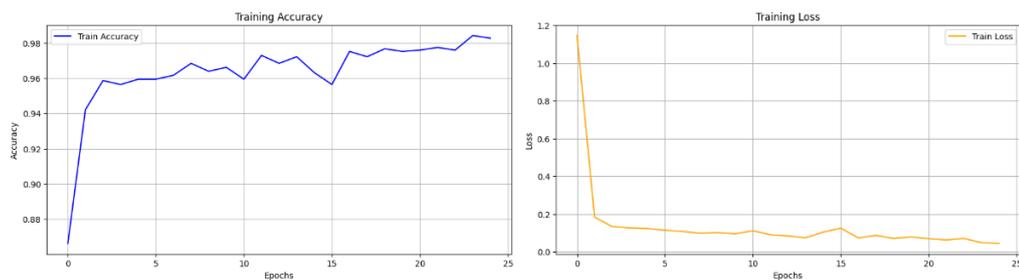
#### 4.1 Hasil Uji Coba

##### 4.1.1 Skenario Uji Pembagian Data

Skenario uji pembagian data bertujuan untuk mengevaluasi pengaruh perbedaan proporsi pembagian data *training* dan *testing* terhadap performa model dalam klasifikasi citra kebakaran hutan. Terdapat tiga variasi pembagian data yang diuji, yaitu Skenario 1-A-Y (70:30), 2-A-Y (80:20), dan 3-A-Y (90:10).

##### 4.1.1.1 Skenario 1-A-Y

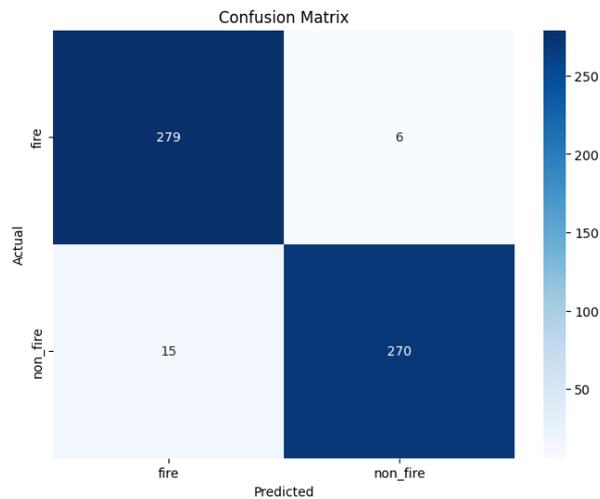
Pada skenario ini, data dibagi dengan proporsi 70% (1330 data) untuk *training* dan 30% (570 data) untuk *testing*. Model dilatih menggunakan arsitektur dengan 2 lapisan konvolusi dan metode klasifikasi *threshold*. Grafik akurasi dan *loss* dari proses *training* pada skenario 1-A-Y ditampilkan pada Gambar 4.1.



Gambar 4.1 Grafik Proses *Training* Skenario 1-A-Y

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 97.44% pada proses *training* dengan nilai *loss* sebesar 0.0707. Model hasil proses

*training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.2 menampilkan hasil *confussion matrix* pada data testing skenario 1-A-Y.



Gambar 4.2 *Confussion Matrix* Skenario 1-A-Y

Gambar 4.2 memperlihatkan bahwa model berhasil memprediksi dengan benar 279 data citra *fire* dan 270 data citra *non-fire*. Namun, masih terdapat 6 data citra *fire* yang salah diprediksi sebagai *non-fire* dan 15 sampel *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung nilai akurasi, presisi, *recall*, dan *F1-score*.

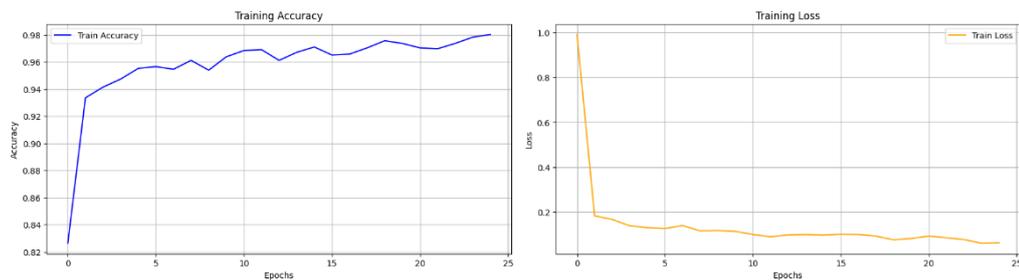
Berdasarkan hasil *confussion matrix*, didapati nilai akurasi pada proses *testing* mencapai 96% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.1:

Tabel 4.1 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 1-A-Y

Kelas	Presisi	<i>Recall</i>	<i>F1-Score</i>
Fire	95%	98%	96%
Non-Fire	98%	95%	96%

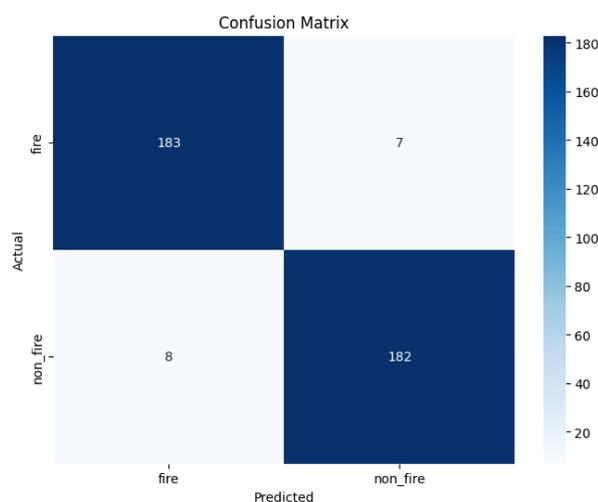
#### 4.1.1.2 Skenario 2-A-Y

Pada skenario ini, data dibagi dengan proporsi 80% (1520 data) untuk *training* dan 20% (380 data) untuk *testing*. Model dilatih menggunakan arsitektur dengan 2 lapisan konvolusi dan metode klasifikasi *threshold*. Grafik akurasi dan *loss* dari proses *training* pada skenario 2-A-Y ditampilkan pada Gambar 4.3.



Gambar 4.3 Grafik Proses *Training* Skenario 2-A-Y

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 96.36% pada proses *training* dengan nilai *loss* sebesar 0.1255. Model hasil proses *training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.4 menampilkan hasil *confusion matrix* pada data *testing* skenario 2-A-Y.



Gambar 4.4 *Confusion Matrix* Skenario 2-A-Y

Gambar 4.4 memperlihatkan bahwa model berhasil memprediksi dengan benar 183 data citra *fire* dan 182 data citra *non-fire*. Namun, masih terdapat 7 data citra *fire* yang salah diprediksi sebagai *non-fire* dan 8 sampel *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

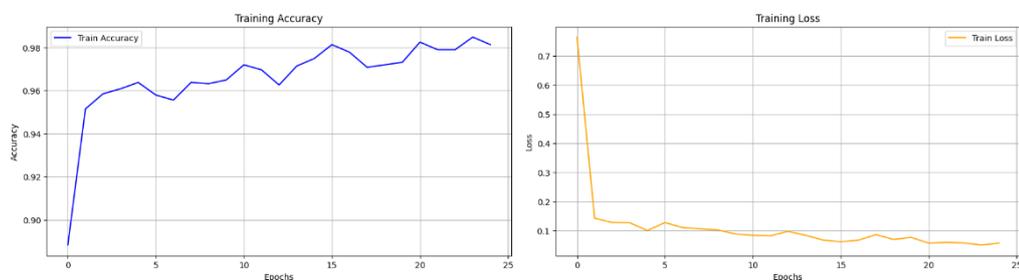
Berdasarkan hasil *confusion matrix*, didapati nilai akurasi pada proses *testing* mencapai 96% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.2:

Tabel 4.2 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 2-A-Y

Kelas	Presisi	<i>Recall</i>	<i>F1-Score</i>
Fire	96%	96%	96%
Non-Fire	96%	96%	96%

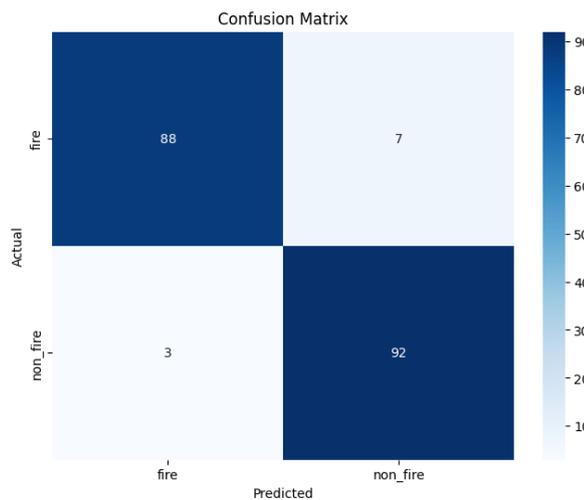
#### 4.1.1.3 Skenario 3-A-Y

Pada skenario ini, data dibagi dengan proporsi 90% (1710 data) untuk *training* dan 10% (190 data) untuk *testing*. Model dilatih menggunakan arsitektur dengan 2 lapisan konvolusi dan metode klasifikasi *threshold*. Grafik akurasi dan *loss* dari proses *training* pada skenario 3-A-Y ditampilkan pada Gambar 4.5.



Gambar 4.5 Grafik Proses *Training* Skenario 3-A-Y

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 92.83% pada proses *training* dengan nilai *loss* sebesar 0.1913. Model hasil proses *training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.6 menampilkan hasil *confusion matrix* pada data testing skenario 3-A-Y.



Gambar 4.6 *Confussion Matrix* Skenario 3-A-Y

Gambar 4.6 memperlihatkan bahwa model berhasil memprediksi dengan benar 88 data citra *fire* dan 92 data citra *non-fire*. Namun, masih terdapat 7 data citra *fire* yang salah diprediksi sebagai *non-fire* dan 3 sampel *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

Berdasarkan hasil *confussion matrix*, didapati nilai akurasi pada proses *testing* mencapai 95% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.3:

Tabel 4.3 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 3-A-Y

Kelas	Presisi	<i>Recall</i>	<i>F1-Score</i>
Fire	97%	93%	95%
Non-Fire	93%	97%	95%

### 4.1.2 Skenario Uji Perubahan Arsitektur

Skenario uji perubahan arsitektur bertujuan untuk mengevaluasi pengaruh perubahan arsitektur terhadap performa model. Perubahan yang diuji meliputi jumlah lapisan konvolusi dan metode klasifikasi yang digunakan, yaitu *threshold* dan *argmax*. Metode *threshold* menentukan kelas berdasarkan nilai probabilitas dengan ambang batas, sedangkan metode *argmax* memilih kelas dengan nilai probabilitas tertinggi. Selain itu, model diuji dengan dua dan tiga lapisan konvolusi untuk melihat pengaruh kedalaman arsitektur terhadap akurasi dan *loss*.

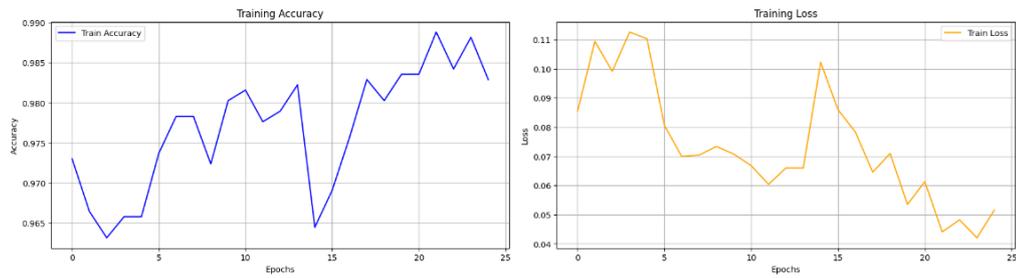
Terdapat empat kombinasi skenario pengujian yang akan digunakan pada skenario uji perubahan arsitektur, yaitu 2-A-Y (2 lapisan, *threshold*), 2-A-Z (2 lapisan, *argmax*), 2-B-Y (3 lapisan, *threshold*), dan 2-B-Z (3 lapisan, *argmax*).

#### 4.1.2.1 Skenario 2-A-Y

Skenario ini telah dijelaskan sebelumnya pada bagian 4.1.1.2 , dimana model menggunakan 2 lapisan konvolusi dengan metode klasifikasi *threshold* dan data dibagi dengan proporsi 80% *training* dan 20% *testing*.

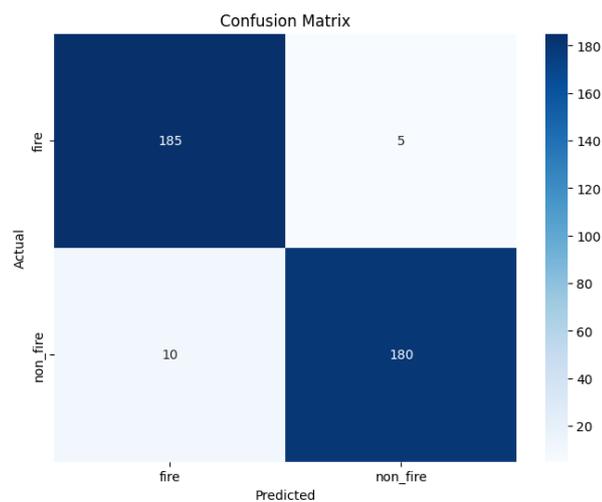
#### 4.1.2.2 Skenario 2-A-Z

Pada skenario ini, model menggunakan 2 lapisan konvolusi dengan metode klasifikasi *argmax*. Data yang digunakan dibagi dengan proporsi 80% (1520 data) untuk *training* dan 20% (380 data) untuk *testing*. Grafik akurasi dan *loss* dari proses *training* pada skenario 2-A-Z ditampilkan pada Gambar 4.7.



Gambar 4.7 Grafik Proses *Training* Skenario 2-A-Z

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 96.61% pada proses *training* dengan nilai *loss* sebesar 0.0821. Model hasil proses *training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.8 menampilkan hasil *confusion matrix* pada data *testing* skenario 2-A-Z.



Gambar 4.8 *Confusion Matrix* Skenario 2-A-Z

Gambar 4.8 memperlihatkan bahwa model berhasil memprediksi dengan benar 185 data citra *fire* dan 180 data citra *non-fire*. Namun, masih terdapat 5 data citra *fire* yang salah diprediksi sebagai *non-fire* dan 10 sampel *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

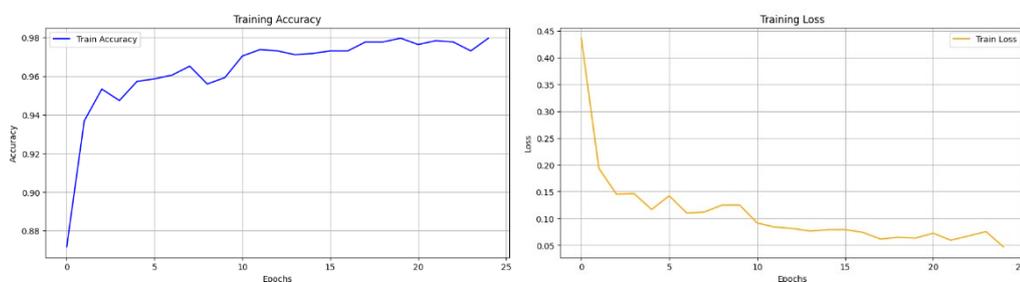
Berdasarkan hasil *confusion matrix*, didapati nilai akurasi pada proses *testing* mencapai 96% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.4:

Tabel 4.4 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 2-A-Z

Kelas	Presisi	<i>Recall</i>	<i>F1-Score</i>
Fire	95%	97%	96%
Non-Fire	97%	95%	96%

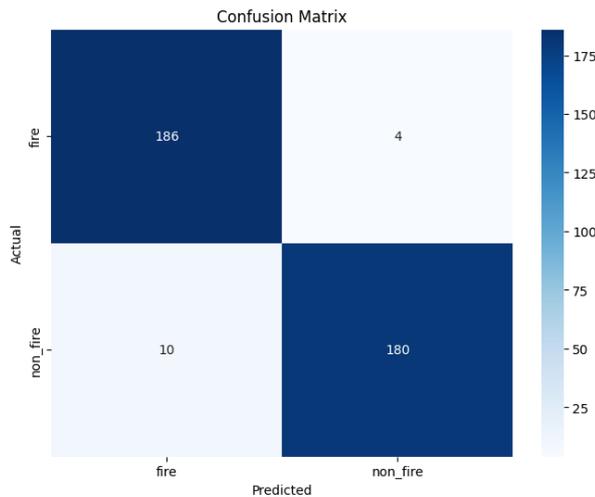
#### 4.1.2.3 Skenario 2-B-Y

Pada skenario ini, model menggunakan 3 lapisan konvolusi dengan metode klasifikasi *threshold*. Data yang digunakan dibagi dengan proporsi 80% (1520 data) untuk *training* dan 20% (380 data) untuk *testing*. Grafik akurasi dan *loss* dari proses *training* pada skenario 2-B-Y ditampilkan pada Gambar 4.9.



Gambar 4.9 Grafik Proses *Training* Skenario 2-B-Y

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 97.20% pada proses *training* dengan nilai *loss* sebesar 0.1098. Model hasil proses *training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.10 menampilkan hasil *confusion matrix* pada data *testing* skenario 2-B-Y.



Gambar 4.10 *Confusion Matrix* Skenario 2-B-Y

Gambar 4.10 memperlihatkan bahwa model berhasil memprediksi dengan benar 186 data citra *fire* dan 180 data citra *non-fire*. Namun, masih terdapat 4 data citra *fire* yang salah diprediksi sebagai *non-fire* dan 10 sampel *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

Berdasarkan hasil *confusion matrix*, didapati nilai akurasi pada proses *testing* mencapai 96% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.5:

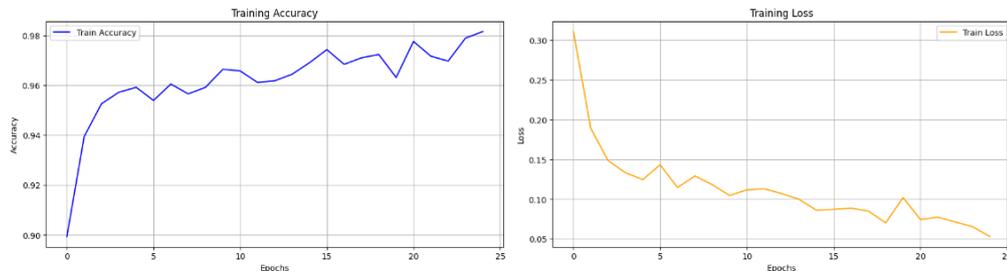
Tabel 4.5 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 2-B-Y

Kelas	Presisi	<i>Recall</i>	<i>F1-Score</i>
Fire	95%	98%	96%
Non-Fire	98%	95%	96%

#### 4.1.2.4 Skenario 2-B-Z

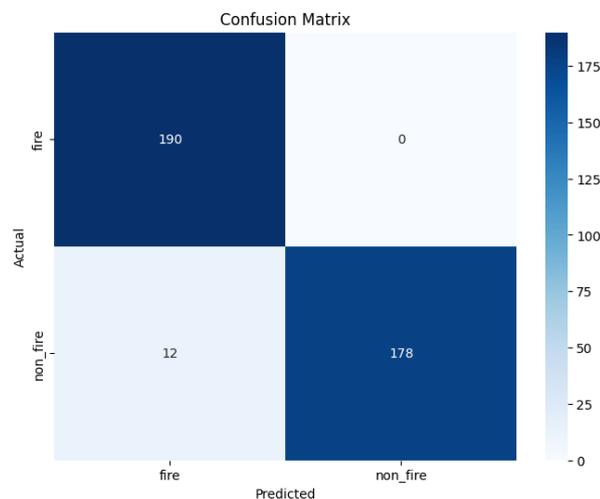
Pada skenario ini, model menggunakan 3 lapisan konvolusi dengan metode klasifikasi *argmax*. Data yang digunakan dibagi dengan proporsi 80% (1520 data)

untuk *training* dan 20% (380 data) untuk *testing*. Grafik akurasi dan *loss* dari proses *training* pada skenario 2-B-Z ditampilkan pada Gambar 4.11.



Gambar 4.11 Grafik Proses *Training* Skenario 2-B-Z

Grafik tersebut menunjukkan bahwa model mencapai akurasi sebesar 98.86% pada proses *training* dengan nilai *loss* sebesar 0.0503. Model hasil proses *training* kemudian disimpan dan digunakan pada proses *testing*. Gambar 4.12 menampilkan hasil *confusion matrix* pada data *testing* skenario 2-B-Z.



Gambar 4.12 *Confusion Matrix* Skenario 2-B-Z

Gambar 4.12 memperlihatkan bahwa model berhasil memprediksi dengan benar keseluruhan 190 data citra *fire* dan 178 data citra *non-fire*. Namun, masih terdapat 12 sampel data *non-fire* yang diprediksi sebagai *fire*. Nilai tersebut kemudian digunakan untuk menghitung akurasi, presisi, *recall*, dan *F1-score*.

Berdasarkan hasil *confusion matrix*, didapati nilai akurasi pada proses *testing* mencapai 97% setelah dihitung menggunakan Persamaan 3.28. Adapun nilai presisi, *recall*, dan *F1-score* yang didapatkan setelah dihitung menggunakan Persamaan 3.29, 3.30, dan 3.31 disajikan pada Tabel 4.6:

Tabel 4.6 Nilai Presisi, *Recall*, *F1-Score* pada Skenario 2-B-Z

<b>Kelas</b>	<b>Presisi</b>	<b>Recall</b>	<b>F1-Score</b>
Fire	94%	100%	97%
Non-Fire	100%	94%	97%

#### 4.1.3 Evaluasi Menggunakan *K-Fold Cross Validation*

Evaluasi model dengan metode *K-Fold Cross Validation* dilakukan pada dua skenario, yaitu skenario 2-A-Y dan skenario 2-B-Z. Pemilihan kedua skenario ini didasarkan pada karakteristik arsitektur yang kontras, sehingga diharapkan dapat memberikan gambaran yang lebih komprehensif terhadap kestabilan performa model.

Pada skenario 2-A-Y dengan arsitektur 2 lapisan konvolusi dan klasifikasi *threshold*, model diuji menggunakan seluruh dataset yang dibagi ke dalam lima *fold*. Hasil evaluasi menunjukkan tingkat akurasi yang cukup konsisten antar *fold*, dengan nilai rata-rata yang menggambarkan kinerja model secara umum. Rincian akurasi dan *loss* pada tiap *fold* dapat dilihat pada Tabel 4.7:

Tabel 4.7 Hasil Evaluasi *K-fold* pada Skenario 2-A-Y

<b>Fold</b>	<b>Akurasi</b>	<b>Loss</b>
1	96.58%	0.1529
2	95.53%	0.1611
3	96.05%	0.1104
4	96.32%	0.1305
5	95.79%	0.1567

Kemudian skenario 2-B-Z dengan arsitektur 3 lapisan konvolusi dan klasifikasi *argmax* juga divalidasi menggunakan pendekatan yang sama. Hasilnya menunjukkan adanya sedikit perbedaan akurasi per *fold* yang cukup terlihat namun masih menunjukkan kestabilan hasil. Berikut hasil *K-fold* pada skenario 2-B-Z ditampilkan pada Tabel 4.8:

Tabel 4.8 Hasil Evaluasi *K-fold* pada Skenario 2-B-Z

<b><i>Fold</i></b>	<b>Akurasi</b>	<b><i>Loss</i></b>
1	94.47%	0.1705
2	96.05%	0.1265
3	97.37%	0.1064
4	95.26%	0.1013
5	96.32%	0.1089

Visualisasi grafik akurasi dan *loss* pada masing-masing *fold* akan disajikan pada bagian pembahasan untuk memberikan gambaran lebih lanjut mengenai kestabilan model di tiap skenario.

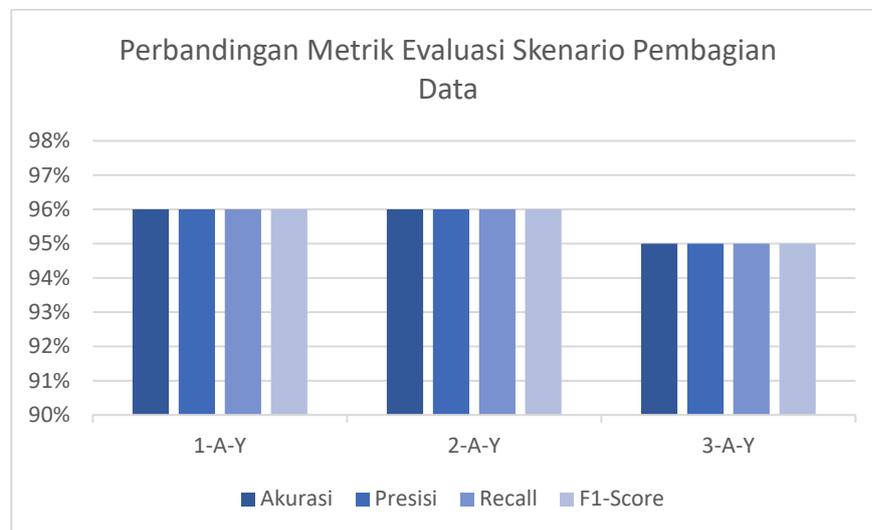
## 4.2 Pembahasan

Terdapat dua skenario utama yang diujikan dalam penelitian ini, yaitu skenario pembagian data yang terdiri dari 3 skenario (Skenario 1-A-Y, Skenario 2-A-Y, dan Skenario 3-A-Y) dan skenario perubahan arsitektur yang terdiri dari 4 skenario (Skenario 2-A-Y, Skenario 2-B-Y, Skenario 2-A-Z, dan Skenario 2-B-Z). Evaluasi dilakukan dengan mengukur metrik evaluasi yang terdiri dari akurasi, presisi, *recall*, dan *F1-score*, serta melakukan validasi terhadap model menggunakan *K-Fold Cross Validation*. Berdasarkan hasil pengujian yang telah ditampilkan pada bagian 4.1, berikut Tabel 4.9 menampilkan ringkasan hasil yang diperoleh dari seluruh skenario uji:

Tabel 4.9 Hasil Uji Setiap Skenario

Skenario	Akurasi	Presisi	Recall	F1-Score
<b>Pembagian Data</b>				
1-A-Y	96%	96%	96%	96%
2-A-Y	96%	96%	96%	96%
3-A-Y	95%	95%	95%	95%
<b>Perubahan Arsitektur</b>				
2-A-Y	96%	96%	96%	96%
2-A-Z	96%	96%	96%	96%
2-B-Y	96%	96%	96%	96%
2-B-Z	97%	97%	97%	97%

Skenario pertama adalah skenario pembagian data, dimana rasio data *training* dan data *testing* divariasikan menjadi tiga skenario, yaitu 70:30 (1-A-Y), 80:20 (2-A-Y), dan 90:10 (3-A-Y). Skenario uji ini dilakukan untuk mengamati bagaimana perubahan proporsi data dapat memengaruhi performa model dalam mengklasifikasikan kebakaran hutan. Gambar 4.13 menampilkan perbandingan metrik akurasi, presisi, *recall*, dan *F1-score* pada skenario pembagian data.



Gambar 4.13 Perbandingan Metrik Evaluasi Skenario Pembagian Data

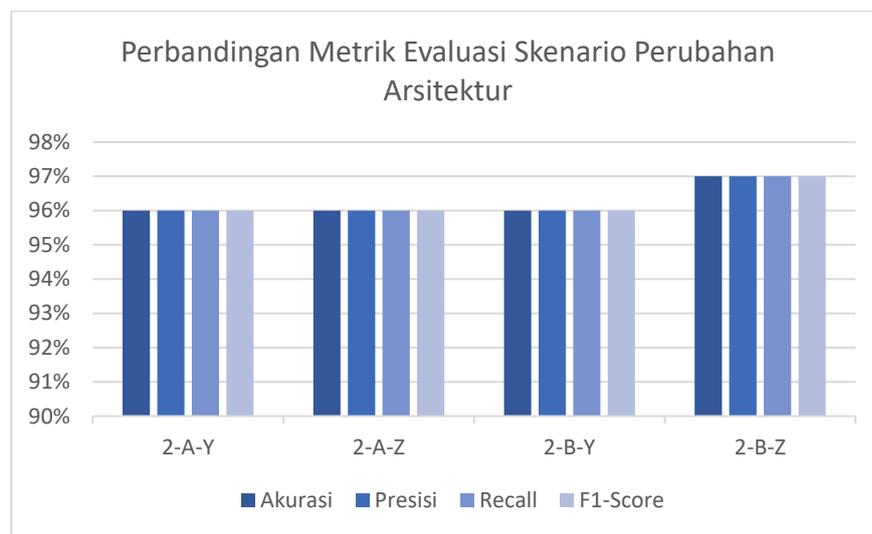
Pada perbandingan metrik evaluasi tersebut, hasil menunjukkan bahwa model dengan rasio 70:30 (1-A-Y) dan 80:20 (2-A-Y) memiliki akurasi yang sama, yaitu 96%, sedangkan model dengan rasio 90:10 (3-A-Y) sedikit lebih rendah

dengan 95%. Nilai presisi, *recall*, dan *F1-score* pada masing-masing skenario juga memiliki pola yang serupa, di mana skenario 1-A-Y dan 2-A-Y memberikan hasil yang lebih stabil dibandingkan dengan skenario 3-A-Y. Perbedaan akurasi ini dapat dijelaskan melalui jumlah data *training* yang tersedia. Pada rasio 70:30, model memiliki data *testing* yang lebih banyak, yang membantu dalam evaluasi generalisasi model. Pada rasio 80:20, keseimbangan antara data *training* dan data *testing* tetap terjaga sehingga model masih mampu belajar dengan baik. Namun, pada rasio 90:10, jumlah data *training* yang lebih banyak justru dapat meningkatkan risiko *overfitting*, di mana model menjadi terlalu menyesuaikan diri terhadap data *training* dan mengalami kesulitan dalam mengenali pola baru dari data *testing*, sehingga akurasinya sedikit menurun.

Meskipun secara umum jumlah data *training* yang lebih banyak dapat meningkatkan kemampuan model dalam memahami pola, dalam beberapa kasus, terutama dengan dataset yang terbatas, terlalu banyak data *training* dapat menyebabkan model terlalu spesifik terhadap data yang telah dilihat sebelumnya. Akibatnya, ketika dihadapkan pada data *testing* yang memiliki variasi lebih besar, model mengalami kesulitan dalam melakukan klasifikasi dengan akurasi yang optimal. Oleh karena itu, skenario dengan rasio 80:20 tampak menjadi pilihan yang lebih seimbang karena mempertahankan performa yang stabil di antara kedua kelas tanpa mengalami penurunan metrik evaluasi yang signifikan.

Skenario kedua yang dilakukan pada penelitian ini mengacu pada pengaruh perubahan arsitektur CNN terhadap performa model dalam mengklasifikasi kebakaran hutan. Pada skenario ini, empat variasi arsitektur diuji, yaitu 2-A-Y, 2-

A-Z, 2-B-Y, dan 2-B-Z dengan variasi jumlah lapisan konvolusi serta metode klasifikasi yang digunakan. Gambar 4.14 menampilkan perbandingan metrik akurasi, presisi, *recall*, dan *F1-score* pada skenario perubahan arsitektur.



Gambar 4.14 Perbandingan Metrik Evaluasi Skenario Perubahan Arsitektur

Berdasarkan hasil pengujian dari skenario perubahan arsitektur, dapat disimpulkan bahwa model 2-A-Y yang menggunakan dua lapisan konvolusi dan metode klasifikasi *threshold* berhasil mencapai akurasi sebesar 96%, dengan nilai presisi, *recall*, dan *F1-score* yang sama. Ketika metode klasifikasi diubah menjadi *argmax* dalam skenario 2-A-Z, nilai metrik performa tetap sama, yakni 96% untuk seluruh indikator. Hal ini menunjukkan bahwa pada arsitektur dengan dua lapisan konvolusi, perubahan metode klasifikasi dari *threshold* ke *argmax* belum memberikan dampak signifikan terhadap peningkatan performa.

Pada model 2-B-Y yang menggunakan tiga lapisan konvolusi namun tetap memakai metode klasifikasi *threshold*, hasil yang diperoleh juga identik, yaitu akurasi, presisi, *recall*, dan *F1-score* sebesar 96%. Artinya, penambahan jumlah lapisan konvolusi saja juga tidak menghasilkan peningkatan performa. Namun

peningkatan performa mulai terlihat saat dilakukan kombinasi antara penambahan lapisan konvolusi dan metode klasifikasi *argmax* dalam skenario 2-B-Z. Model ini berhasil mencapai akurasi sebesar 97%, dengan nilai presisi, *recall*, dan *F1-score* juga sebesar 97%. Peningkatan sebesar 1% ini menunjukkan bahwa pendekatan kombinatorial dari arsitektur yang lebih dalam dan metode klasifikasi *argmax* dapat memberikan keunggulan dalam mengenali pola-pola visual pada citra, terutama dalam konteks klasifikasi antara gambar kebakaran dan non-kebakaran. Metode klasifikasi *argmax* cenderung lebih efektif dibandingkan *threshold* karena langsung memilih kelas dengan probabilitas tertinggi tanpa perlu ditentukan ambang batas nilai tertentu. Pendekatan ini menjadi lebih optimal saat diterapkan pada model dengan arsitektur yang lebih kompleks, seperti pada skenario 2-B-Z.

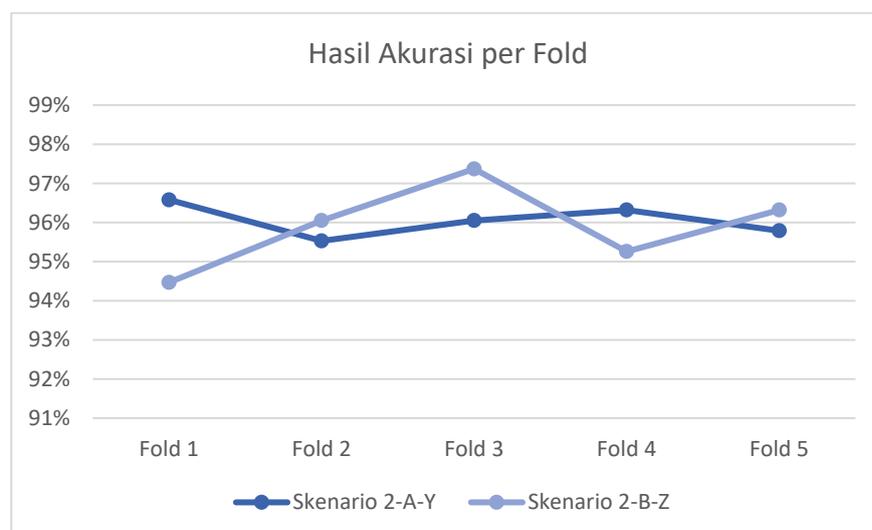
Kesamaan nilai presisi, *recall*, dan *F1-score* yang relatif tinggi dan stabil antar skenario menunjukkan bahwa model mampu mengenali pola antar kelas dengan baik, baik pada data latih maupun data uji. Hal ini juga dapat disebabkan oleh karakteristik dataset yang seimbang dan jelas membedakan antara dua kelas, sehingga meskipun proporsi data dibagi berbeda, model tetap dapat menghasilkan prediksi yang konsisten. Model CNN yang digunakan juga sudah cukup optimal dalam mengekstraksi fitur penting dari citra, sehingga performa antar skenario tidak menunjukkan perbedaan yang signifikan pada metrik evaluasi selain akurasi. Maka dapat dilihat bahwa performa model CNN dalam klasifikasi kebakaran hutan dapat ditingkatkan melalui penggabungan arsitektur yang lebih dalam dan pemilihan metode klasifikasi yang tepat. Pendekatan ini terbukti mampu meningkatkan

generalisasi model terhadap data baru dan memperkuat akurasi prediksi dalam skenario klasifikasi dua kelas berbasis citra.

Evaluasi lanjutan dilakukan dengan menguji validasi model terhadap variasi distribusi data menggunakan metode *K-Fold Cross Validation*. *K-fold* diterapkan pada dua skenario dengan konfigurasi arsitektur yang kontras, yaitu skenario 2-A-Y dan skenario 2-B-Z. Keduanya dipilih karena memiliki pendekatan berbeda dalam perubahan struktur CNN, sehingga dapat memberikan gambaran yang lebih menyeluruh terhadap robustness model dalam berbagai kondisi.

Sebelumnya, kedua skenario tersebut telah dievaluasi menggunakan pembagian data konvensional (tanpa *K-fold*), di mana skenario 2-A-Y menghasilkan akurasi sebesar 96.05% dengan nilai *loss* 0.1457, sementara skenario 2-B-Z mencapai akurasi 96.84% dan *loss* 0.1114. Nilai-nilai tersebut menjadi acuan awal untuk menilai konsistensi hasil dari validasi *K-fold*.

Berikut adalah grafik perbandingan akurasi pada masing-masing *fold* antara skenario 2-A-Y dan 2-B-Z:

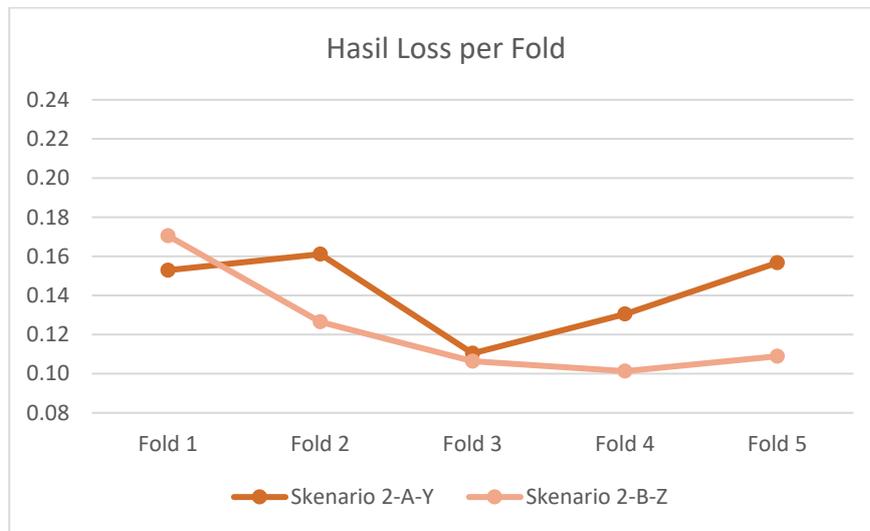


Gambar 4.15 Grafik Hasil Akurasi per *Fold*

Berdasarkan grafik yang ditampilkan, dapat diamati bahwa skenario 2-A-Y menunjukkan performa model yang relatif stabil pada seluruh *fold* dalam evaluasi *K-Fold Cross Validation*. Nilai akurasi yang diperoleh berada dalam rentang 95,53% hingga 96,58%, dengan rata-rata sebesar 96,05%. Nilai rata-rata tersebut identik dengan hasil akurasi awal sebelum penerapan K-fold, yang mengindikasikan bahwa variasi data pelatihan dan pengujian pada setiap *fold* tidak memberikan pengaruh signifikan terhadap performa model. Konsistensi ini mencerminkan bahwa model pada skenario 2-A-Y memiliki kemampuan generalisasi yang baik dan mampu mempertahankan akurasi yang tinggi meskipun diuji pada data yang berbeda-beda.

Sementara itu, skenario 2-B-Z menunjukkan fluktuasi nilai akurasi yang sedikit lebih besar, yaitu antara 94,47% hingga 97,37%, dengan rata-rata akurasi sebesar 95,89%. Meskipun nilai rata-rata ini sedikit lebih rendah dibandingkan akurasi awal sebesar 96,84%, perbedaan tersebut masih dalam batas yang wajar dan tidak signifikan. Hal ini menunjukkan bahwa model pada skenario 2-B-Z tetap mampu menjaga performa yang baik meskipun diuji pada data dengan pembagian yang bervariasi. Dengan demikian, kedua model pada skenario tersebut menunjukkan stabilitas yang cukup baik, dengan kemampuan generalisasi yang layak untuk diterapkan pada data nyata.

Evaluasi juga dilakukan dengan mengamati hasil nilai *loss* yang didapat selama proses validasi. Berikut grafik yang menampilkan perbandingan nilai *loss* antar *fold* untuk kedua skenario:



Gambar 4.16 Grafik Hasil *Loss* per *Fold*

Berdasarkan grafik tersebut, terlihat bahwa skenario 2-A-Y memiliki nilai *loss* yang relatif stabil, yaitu berada pada kisaran 0.1104 hingga 0.1611, dengan rata-rata sebesar 0.1423. Nilai rata-rata ini sangat mendekati nilai *loss* awal tanpa penerapan *K-fold*, yaitu sebesar 0,1457. Kedekatan nilai ini semakin memperkuat bahwa model pada skenario 2-A-Y menunjukkan performa yang konsisten meskipun diuji dengan variasi distribusi data pada setiap *fold* dan memiliki kemampuan untuk mempertahankan performa.

Adapun skenario 2-B-Z menunjukkan rentang nilai *loss* yang sedikit lebih lebar, yaitu antara 0.1013 hingga 0.1705, dengan rata-rata sebesar 0.1227. Meskipun terdapat satu *fold* yang menghasilkan nilai *loss* sedikit lebih tinggi, secara keseluruhan model tetap menunjukkan kestabilan dalam performa. Dibandingkan dengan nilai *loss* awal sebesar 0.1114, rata-rata hasil *K-fold* model pada skenario 2-B-Z tetap menunjukkan kemampuan generalisasi yang baik, serta mampu mempertahankan kualitas performa meskipun diuji pada data yang memiliki variasi distribusi.

Dari keseluruhan hasil yang didapat selama proses validasi menggunakan *K-Fold Cross Validation*, dapat disimpulkan bahwa kedua skenario menunjukkan robustness yang cukup kuat terhadap variasi data. Penerapan *K-Fold Cross Validation* berhasil menunjukkan bahwa model tidak *overfitting* pada subset tertentu, dan mampu mempertahankan performa saat diuji pada data yang berbeda. Perbandingan dengan hasil awal tanpa *K-fold* juga membantu memperkuat bahwa akurasi dan *loss* yang diperoleh sebelumnya bukanlah hasil dari pembagian data yang menguntungkan, melainkan benar-benar mencerminkan kualitas model.

Penelitian ini bertujuan untuk mengembangkan model klasifikasi kebakaran hutan berbasis citra menggunakan *Convolutional Neural Network* (CNN) sebagai upaya dalam mitigasi bencana kebakaran hutan. Dengan adanya sistem ini, diharapkan deteksi kebakaran dapat dilakukan lebih cepat dan akurat, sehingga memungkinkan tindakan pencegahan serta penanggulangan yang lebih efektif. Selain itu, penelitian ini juga bertujuan untuk memanfaatkan teknologi kecerdasan buatan sebagai sarana dalam menjaga kelestarian lingkungan, sejalan dengan prinsip Islam yang menekankan pentingnya menjaga keseimbangan alam dan mencegah kerusakan. Dalam Al-Qur'an, Allah SWT telah memperingatkan manusia agar tidak melakukan perbuatan yang merusak lingkungan. Hal ini disebutkan dalam Surah Al-Baqarah ayat 205:

وَإِذَا تَوَلَّى سَعَىٰ فِي الْأَرْضِ لِيُفْسِدَ فِيهَا وَيُهْلِكَ الْحَرْثَ وَالنَّسْلَ ۗ وَاللَّهُ لَا يُحِبُّ الْفُسَادَ

*"Dan apabila ia berpaling (dari kamu), ia berjalan di bumi untuk berbuat kerusakan padanya, dan merusak tanam-tanaman dan ternak. Dan Allah tidak menyukai kerusakan." (QS. Al-Baqarah:205)*

Menurut Tafsir Kementerian Agama RI Jilid 1 halaman 301, ayat ini menjelaskan tentang orang-orang yang tampak baik di hadapan manusia, tetapi sejatinya memiliki niat merusak. Setelah mereka berhasil menipu atau mendapatkan apa yang mereka inginkan, mereka justru melakukan kerusakan di bumi dengan merusak tanaman, membinasakan hewan ternak, dan bertindak sewenang-wenang ketika berkuasa. Perbuatan mereka membawa ketakutan dalam masyarakat, menghancurkan rumah tangga, serta mengancam kehidupan sosial dan moral. Allah SWT membenci perilaku semacam ini, karena mereka tidak hanya merugikan diri sendiri, tetapi juga makhluk hidup lain dan lingkungan di sekitar mereka (Kementerian Agama RI, 2011). Penelitian ini menjadi salah satu bentuk upaya untuk mengurangi dampak kebakaran hutan dengan teknologi yang mampu mendeteksi dan mengklasifikasikan kebakaran lebih awal. Dengan begitu, langkah-langkah pencegahan dapat segera diambil sebelum kebakaran meluas dan menyebabkan kehancuran yang lebih besar.

Allah SWT juga telah mengingatkan manusia akan pentingnya menjaga kehidupan dan mencegah kerusakan, sebagaimana disebutkan dalam Surah Al-Maidah ayat 32:

مِنْ أَجْلِ ذَلِكَ كَتَبْنَا عَلَىٰ بَنِي إِسْرَائِيلَ أَنَّهُ ۖ مَنْ قَتَلَ نَفْسًا بِغَيْرِ نَفْسٍ أَوْ فَسَادٍ فِي الْأَرْضِ فَكَأَنَّمَا قَتَلَ النَّاسَ جَمِيعًا ۖ وَمَنْ أَحْيَاهَا فَكَأَنَّمَا أَحْيَا النَّاسَ جَمِيعًا ۚ وَلَقَدْ جَاءَهُمْ رَسُولُنَا بِالْبَيِّنَاتِ ثُمَّ إِنَّ كَثِيرًا مِنْهُمْ بَعَدَ ذَلِكَ فِي الْأَرْضِ لُمُسْرِفُونَ

*“Oleh karena itu, Kami menetapkan (suatu hukum) bagi Bani Israil bahwa siapa yang membunuh seseorang bukan karena (orang yang dibunuh itu) telah membunuh orang lain atau karena telah berbuat kerusakan di bumi, maka seakan-akan dia telah membunuh semua manusia. Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia. Sungguh, rasul-rasul Kami benar-benar telah datang kepada mereka dengan (membawa) keterangan-keterangan yang jelas. Kemudian, sesungguhnya*

*banyak di antara mereka setelah itu melampaui batas di bumi.” (QS. Al-Maidah:32)*

Menurut Tafsir Wajiz Jilid 1, ayat ini menegaskan bahwa menjaga kehidupan manusia merupakan tindakan yang bernilai besar di sisi Allah SWT. Bagian ayat "*Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia.*" mengandung makna bahwa segala bentuk usaha yang bertujuan untuk menyelamatkan nyawa dan mencegah bencana adalah bagian dari tanggung jawab kemanusiaan. Dalam konteks penelitian ini, penerapan teknologi dalam mendeteksi kebakaran hutan lebih awal diharapkan dapat membantu pelaksanaan mitigasi bencana lebih cepat agar mengurangi risiko kebakaran yang berpotensi mengancam nyawa manusia serta keseimbangan ekosistem (Kementerian Agama RI, 2016).

Penelitian ini juga berkaitan dengan amanah yang diberikan kepada manusia dalam konteks tanggung jawab terhadap lingkungan. Tidak hanya dalam Al-Qur'an, Rasulullah SAW juga memberikan petunjuk dalam menjaga alam dan melarang perusakan lingkungan. Seperti sabda Rasulullah SAW dalam hadits riwayat Muslim yang menganjurkan umat muslim untuk senantiasa menjaga dan merawat alam yang artinya:

*“Tidaklah seorang muslim yang bercocok tanam, kecuali setiap tanaman yang dimakannya bernilai sedekah baginya, yang dicuri orang darinya menjadi sedekah baginya, yang dimakan binatang liar menjadi sedekah baginya, yang dimakan burung menjadi sedekah baginya, dan tidaklah seseorang mengambil darinya, melainkan menjadi sedekah baginya.” (HR. Muslim)*

Hadits ini menunjukkan bahwa setiap upaya manusia dalam menjaga dan melestarikan lingkungan memiliki nilai ibadah di sisi Allah SWT. Dengan

demikian, penelitian ini tidak hanya bertujuan untuk mengembangkan teknologi dalam mitigasi kebakaran hutan, tetapi juga merupakan bentuk tanggung jawab manusia dalam menjaga kelestarian alam sebagaimana yang diajarkan dalam Islam. Menanam sesuatu layaknya bersedekah bagi seorang muslim, baik hasil dari tanaman tersebut bermanfaat secara langsung baginya atau tidak. Dengan menanam sesuatu, secara tidak langsung kita juga sudah menjaga ekosistem lain yang mengambil manfaat dari apa yang kita tanam. Menanam sesuatu berarti menjaga kelangsungan flora dan fauna sekaligus (Chandra, 2016). Dengan kata lain, dengan menanam maka kesejahteraan semua makhluk yang ada di lingkungan tersebut akan terjaga dengan baik. Oleh karena itu, upaya untuk mencegah kebakaran hutan melalui teknologi kecerdasan buatan bukan hanya sebatas inovasi dalam bidang ilmu pengetahuan, tetapi juga selaras dengan ajaran Islam dalam menjaga kelestarian alam dan memberikan manfaat bagi seluruh makhluk hidup.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian ini telah menerapkan model *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi kebakaran hutan berbasis citra menggunakan dataset yang diperoleh dari Kaggle. Melalui eksperimen dengan dua skenario utama, yaitu variasi pembagian data dan perubahan arsitektur model, serta penggunaan metode *K-Fold Cross Validation*, diperoleh pemahaman yang lebih komprehensif mengenai performa dan stabilitas model terhadap dataset.

Berdasarkan hasil dari 7 skenario pengujian yang telah dilakukan untuk mengklasifikasikan kebakaran hutan menggunakan metode CNN, dapat disimpulkan bahwa variasi pada pembagian data dan arsitektur jaringan memiliki pengaruh terhadap performa model. Penelitian ini menggunakan dua pendekatan pengujian, yaitu pengaruh pembagian data (dengan rasio 70:30, 80:20, dan 90:10) dan perubahan arsitektur CNN (penambahan jumlah lapisan konvolusi dan metode klasifikasi).

Pada pengujian variasi pembagian data, skenario 2-A-Y (dengan pembagian data 90:10) memberikan hasil terbaik dengan akurasi sebesar 96%, serta nilai presisi, *recall*, dan *F1-score* sebesar 96% pada masing-masing kelas. Hasil ini menunjukkan bahwa semakin besar proporsi data latih yang digunakan, model dapat belajar pola lebih baik sehingga meningkatkan performa klasifikasi.

Sementara itu pada pengujian perubahan arsitektur, skenario 2-B-Z yang menggunakan tiga lapisan konvolusi dan metode klasifikasi *argmax* memberikan hasil terbaik dengan akurasi sebesar 97%. Pada skenario ini, nilai presisi mencapai 94% untuk kelas *fire* dan 100% untuk kelas *non-fire*, sedangkan nilai *recall* masing-masing sebesar 100% dan 94%, menghasilkan *F1-score* yang seimbang sebesar 97% untuk kedua kelas. Dibandingkan dengan skenario lainnya, model 2-B-Z menunjukkan bahwa kombinasi peningkatan kedalaman arsitektur dan penggunaan metode klasifikasi *argmax* dapat meningkatkan kemampuan model dalam mengekstraksi fitur dan membedakan antara citra kebakaran dan non-kebakaran.

Dengan demikian, tujuan dari penelitian ini berhasil dicapai, yaitu menerapkan CNN dalam klasifikasi kebakaran hutan menggunakan data citra serta menganalisis pengaruh pembagian data dan arsitektur terhadap performa model. Hasil penelitian menunjukkan bahwa baik jumlah data latih maupun arsitektur CNN yang digunakan memiliki kontribusi terhadap peningkatan akurasi klasifikasi.

## 5.2 Saran

Berdasarkan hasil yang diperoleh dari proses klasifikasi citra kebakaran hutan menggunakan model *Convolutional Neural Network*, peneliti menyadari bahwa masih terdapat ruang untuk pengembangan di masa mendatang. Oleh karena itu, beberapa saran yang dapat dipertimbangkan dalam penelitian selanjutnya antara lain:

1. Penelitian ini hanya menggunakan satu jenis dataset, yaitu “*Forest Fire Dataset*” dari Kaggle. Untuk memperluas cakupan dan menguji kemampuan generalisasi model, disarankan agar penelitian selanjutnya menggunakan

dataset citra lain yang lebih beragam atau mencakup kondisi lingkungan yang berbeda.

2. Arsitektur CNN yang digunakan dalam penelitian ini masih tergolong sederhana. Mengacu pada temuan bahwa penambahan lapisan dan perubahan strategi klasifikasi dapat meningkatkan performa model, maka pada penelitian berikutnya disarankan untuk mengeksplorasi arsitektur yang lebih kompleks atau melakukan modifikasi lanjutan terhadap konfigurasi jaringan.
3. Penelitian lanjutan dapat mengembangkan sistem klasifikasi ini menjadi sistem deteksi kebakaran hutan secara *real-time* yang terintegrasi dengan *drone* atau kamera pengawas untuk mendukung upaya mitigasi bencana.

## DAFTAR PUSTAKA

- Abror, Z. F. (2019). Klasifikasi Citra Kebakaran Dan Non Kebakaran Menggunakan Convolutional Neural Network. *Jurnal Ilmiah Teknologi Dan Rekayasa*, 24(2), 102–113. <https://doi.org/10.35760/tr.2019.v24i2.2389>
- Anhar, I. P., Rina Mardiana, & Rai Sita. (2022). Dampak Kebakaran Hutan dan Lahan Gambut terhadap Manusia dan Lingkungan Hidup (Studi Kasus: Desa Bunsur, Kecamatan Sungai Apit, Kabupaten Siak, Provinsi Riau). *Jurnal Sains Komunikasi Dan Pengembangan Masyarakat [JSKPM]*, 6(1), 75–85. <https://doi.org/10.29244/jskpm.v6i1.967>
- Bahhar, C., Ksibi, A., Ayadi, M., Jamjoom, M. M., Ullah, Z., Soufiene, B. O., & Sakli, H. (2023). Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics (Switzerland)*, 12(1), 1–15. <https://doi.org/10.3390/electronics12010228>
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics (Switzerland)*, 10(20), 1–28. <https://doi.org/10.3390/electronics10202470>
- BNBP. (2024). *Buletin Info Bencana Juni 2024*. 5(6).
- Chandra, A. F. (2016). HADIS-HADIS EKOLOGI DALAM KONTEKS PERINDUSTRIAN DI INDONESIA Agus Firdaus Chandra. *Jurnal Ilmiah Syari'ah*, 15(1).
- Chen, S.-W., Juang, J., Wang, C., Chang, H.-L., Hong, J.-S., & Hsiao, C. K. (2024). *A Convolutional Neural Network-based Ensemble Post-processing with Data Augmentation for Tropical Cyclone Precipitation Forecasts*. 1–21.
- Dwiasnati, S., Devianto, Y., Arif, S. M., & Avrizal, R. (2024). Pemodelan Wilayah Titik Api Kebakaran Hutan Menggunakan Deep Learning. *Jurnal Ilmiah FIFO*, 16(1), 1. <https://doi.org/10.22441/fifo.2024.v16i1.001>
- Faisal, M. (2022). *Data Sains Praktis Menggunakan Python* (1st ed.). UIN-Maliki Press.
- Ghahremani, T., Hoseyni, M., Ahmadi, M. J., Mehrabi, P., & Nikoofard, A. (2024). *AugmenTory: A Fast and Flexible Polygon Augmentation Library*. <http://arxiv.org/abs/2405.04442>
- Goldman, L., & Weisse, M. (2022). *Global Forest Watch's 2021 Data Update Explained*. Global Forest Watch. <https://www.globalforestwatch.org/blog/data-and-tools/2021-tree-cover-loss-data-explained/>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. In MIT Press. MIT Press. <https://books.google.co.id/books?id=omivDQAAQBAJ>

- Hadiwijoyo, E. (2023). Pola Kebakaran Hutan di Areal Konservasi Studi Kasus di Taman Nasional Bromo Tengger Semeru. *Journal of Tropical Silviculture*, 14(02), 114–118. <https://doi.org/10.29244/j-siltrop.14.02.114-118>
- Kementerian Agama RI. (2011a). Al-Qur'an dan Tafsirnya Jilid 1 (Juz 1-3). In *Departemen Agama RI* (Vol. 44, Issue 8). <https://doi.org/10.1088/1751-8113/44/8/085201>
- Kementerian Agama RI. (2011b). Al-Qur'an Dan Tafsirnya Jilid 3 (Juz 7-9). In *Widya Cahaya, Jakarta* (Vol. 44, Issue 8). <https://doi.org/10.1088/1751-8113/44/8/085201>
- Kementerian Agama RI. (2011c). Al-Qur'an Dan Tafsirnya Jilid 7 (Juz 19-21). In *Widya Cahaya, Jakarta*.
- Kementerian Agama RI. (2016). *Tafsir Wajiz Jilid 1* (Vol. 27). Lajnah Pentashihan Mushaf Al-Qur'an.
- Khan, A., Hassan, B., Khan, S., Ahmed, R., & Abuassba, A. (2022). DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mobile Information Systems*, 2022. <https://doi.org/10.1155/2022/5358359>
- Khan, S., & Khan, A. (2022). FFireNet: Deep Learning Based Forest Fire Classification and Detection in Smart Cities. *Symmetry*, 14(10). <https://doi.org/10.3390/sym14102155>
- Kusin, K., Sulistiyanto, Y., Usup, A., & Ardianor. (2022). Carbon Monoxide (CO) and Particulate Matter (PM2.5) Concentration at Central Kalimantan, Indonesia. *IOP Conference Series: Earth and Environmental Science*, 1111(1). <https://doi.org/10.1088/1755-1315/1111/1/012006>
- Madhuri, C. R., Jandhyala, S. S., Ravuri, D. M., & Babu, V. D. (2024). Accurate classification of forest fires in aerial images using ensemble model. *Bulletin of Electrical Engineering and Informatics*, 13(4), 2650–2658. <https://doi.org/10.11591/eei.v13i4.6527>
- Matlab. (2021). *What Is a Convolutional Neural Network? 3 things you need to know*. Mathworks.Com. <https://www.mathworks.com/discovery/convolutional-neural-network.html>
- Pratiwi, M. I., & Nudin, S. R. (2023). *Pengembangan Sistem Deteksi Dini Kebakaran Menggunakan Pendekatan Transfer Learning dengan Model ResNet50 Berbasis Website*. 1–12.
- Purwono, Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. U. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739–748. <https://doi.org/10.31763/ijrcs.v2i4.888>
- Sakib, S., Ahmed, N., Kabir, A. J., & Ahmed, H. (2018). An Overview of

Convolutional Neural Network: Its Architecture and Applications. *Preprints 2018, February*. <https://doi.org/10.20944/preprints201811.0546.v1>

Saleem, M. A., Senan, N., Wahid, F., Aamir, M., Samad, A., & Khan, M. (2022). Comparative Analysis of Recent Architecture of Convolutional Neural Network. *Mathematical Problems in Engineering*, 2022. <https://doi.org/10.1155/2022/7313612>

Saputra, R. A., & Adhinata, F. D. (2023). Model Deteksi Kebakaran Hutan dan Lahan Menggunakan Transfer Learning DenseNet201. *Journal of Intelligent System and Computation*, 5(2), 65–72. <https://doi.org/10.52985/insyst.v5i2.317>

Suhardono, S., Fitria, L., Suryawan, I. W. K., Septiariva, I. Y., Mulyana, R., Sari, M. M., Ulhasanah, N., & Prayogo, W. (2024). Human activities and forest fires in Indonesia: An analysis of the Bromo incident and implications for conservation tourism. *Trees, Forests and People*, 15(February), 100509. <https://doi.org/10.1016/j.tfp.2024.100509>

Suhartono. (2018). *Ilustrasi pemodelan pertumbuhan tanaman berbasis lingkungan : pendekatan jaringan saraf tiruan* (S. Zaman (ed.)). UIN-Maliki Press.

Suryadewiansyah, M. K., Endra, T., & Tju, E. (2020). Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 8(2), 81–88.

Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjut*. Informatika Bandung.

Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3). <https://doi.org/10.3390/computation11030052>