

**IMPLEMENTASI METODE *TRANSFER LEARNING* VGG-16 UNTUK  
KLASIFIKASI CITRA X-RAY TULANG BELAKANG**

**(Skripsi)**

**Oleh**

**HARUM APRELINA RAHMAD**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2024**

## **ABSTRACT**

### **IMPLEMENTATION OF VGG-16 TRANSFER LEARNING METHOD FOR THE VERTEBRAE X-RAY IMAGE CLASSIFICATION**

**By**

**HARUM APRELINA RAHMAD**

The problem of vertebral disorders is increasingly becoming an important concern in the medical field, given its significant impact on daily life. This can be handled by utilizing technology in the diagnosis and management process, through image classification that helps identify vertebral disorders. However, there are obstacles in the automation of image classification such as; medical data sets that are difficult to access, professional expertise in labelling images, and time. Thus, machine learning is needed to improve the efficiency and accuracy of image classification. This study applies a machine learning model inspired by the advantages of CNN (Convolutional Neural Network) in image classification. Applying the Transfer Learning method using the VGG-16 model to the vertebrae x-ray image classification, can increase accuracy with a data set that is not too large. The dataset used in this study was obtained from the Kaggle site, which contains the vertebrae x-ray image data covering 338 images. This study involved six classification experiments, namely experiments A, B, and C using CNN, and experiments D, E, and F using Transfer Learning VGG-16. The architecture of the VGG-16 Transfer Learning model includes six main layers, namely the VGG-16 base layer, convolution, max pooling, dropout, flatten, and fully connected. The results of the F experiment showed the best performance with an accuracy of 98.52%, a precision of 99%, a sensitivity of 97%, and an F1 score of 98%. This superior performance factor is supported by changes in data composition, the addition of dropout layers, and adjustments to the learning rate. The application of the VGG-16 Transfer Learning model has been proven to provide significant performance in the classification of the vertebrae x-ray images, so it is expected to be an effective alternative for the classification of the vertebrae x-ray images.

**Keywords** : The Vertebrae, Machine Learning, Transfer Learning, CNN, VGG-16

## ABSTRAK

### IMPLEMENTASI METODE *TRANSFER LEARNING* VGG-16 UNTUK KLASIFIKASI CITRA X-RAY TULANG BELAKANG

Oleh

HARUM APRELINA RAHMAD

Masalah kelainan tulang belakang semakin menjadi perhatian penting di bidang medis, mengingat dampaknya yang signifikan terhadap kehidupan sehari-hari. Hal tersebut dapat ditangani dengan memanfaatkan teknologi dalam proses diagnosis maupun pengelolaannya, melalui klasifikasi citra yang membantu mengidentifikasi kelainan tulang belakang. Akan tetapi, terdapat kendala dalam otomatisasi klasifikasi citra seperti; kumpulan data medis yang sulit diakses, keahlian profesional dalam memberikan label pada gambar, dan waktu. Sehingga, diperlukan penerapan *machine learning* untuk meningkatkan efisiensi dan ketepatan pada klasifikasi Citra. Penelitian ini menerapkan model *machine learning* yang terinspirasi oleh keunggulan CNN (*Convolutional Neural Network*) dalam klasifikasi citra. Dengan menerapkan metode *Transfer Learning* menggunakan model VGG-16 pada klasifikasi citra *x-ray* tulang belakang mampu meningkatkan akurasi dengan Kumpulan data yang tidak terlalu besar. Kumpulan data yang digunakan pada penelitian ini diperoleh dari situs *Kaggle* berisi data citra *x-ray* tulang belakang mencakup 338 citra. Penelitian ini melibatkan enam percobaan klasifikasi, yaitu percobaan A, B, dan C menggunakan CNN, serta percobaan D, E, dan F menggunakan *Transfer Learning* VGG-16. Arsitektur model *Transfer Learning* VGG-16 mencakup enam lapisan utama, yaitu lapisan dasar VGG-16, konvolusi, *max pooling*, *dropout*, *flatten*, dan *fully connected*. Hasil dari percobaan F menunjukkan kinerja terbaik dengan akurasi sebesar 98,52%, presisi 99%, sensitivitas 97%, dan skor F1 sebesar 98%. Faktor kinerja unggul ini didukung oleh perubahan komposisi data, penambahan lapisan *dropout*, dan penyesuaian *learning rate*. Penerapan model *Transfer Learning* VGG-16 ini terbukti memberikan performa yang signifikan dalam klasifikasi citra *x-ray* tulang belakang, sehingga diharapkan dapat menjadi alternatif yang efektif untuk klasifikasi citra *x-ray* tulang belakang.

**Kata Kunci** : Tulang Belakang, *Machine Learning*, *Transfer Learning*, CNN, VGG-16

**IMPLEMENTASI METODE *TRANSFER LEARNING* VGG-16 UNTUK  
KLASIFIKASI CITRA X-RAY TULANG BELAKANG**

**Oleh**

**HARUM APRELINA RAHMAD  
2017031092**

**Skripsi**

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA MATEMATIKA**

Pada

Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2024**

Judul Skripsi : **IMPLEMENTASI METODE *TRANSFER LEARNING* VGG-16 UNTUK KLASIFIKASI CITRA X-RAY TULANG BELAKANG**

Nama Mahasiswa : **Harum Aprefina Rahmad**

Nomor Pokok Mahasiswa : **2017031092**

Jurusan : **Matematika**


Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

**MENYETUJUI**

**1. Komisi Pembimbing**



**Dr. Dian Kurniasari, S.Si., M.Sc.**  
NIP. 19690305 199603 2 001



**Ir. Warsono, M.S., Ph.D.**  
NIP. 19630216 198703 1 003

**2. Ketua Jurusan Matematika**



**Dr. Aang Nuryaman, S.Si., M.Si.**  
NIP. 19740316 200501 1 001

**MENGESAHKAN**

1. Tim Penguji

Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.**



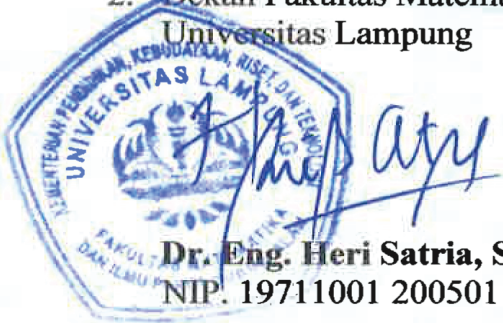
Sekretaris : **Ir. Warsono, M.S., Ph.D.**



Penguji  
Bukan Pembimbing : **Prof. Dra. Wamiliana, M.A., Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi : **4 November 2024**

## PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama : **Harum Aprelina Rahmad**  
Nomor Pokok Mahasiswa : **2017031092**  
Jurusan : **Matematika**  
Judul Skripsi : **Implementasi Metode *Transfer Learning* VGG-16  
untuk Klasifikasi Citra *X-Ray* Tulang Belakang**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan apabila di kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 4 November 2024

Penulis



**Harum Aprelina Rahmad**  
**NPM. 2017031092**

## RIWAYAT HIDUP

Penulis bernama lengkap Harum Aprelina Rahmad lahir di Sumpalsari, Metro pada tanggal 8 April 2002. Penulis merupakan anak pertama dari pasangan Bapak Basuki Rahmad dan Ibu Diana Mardiyanti.

Penulis memulai perjalanan pendidikannya dari taman kanak-kanak (TK) di TK Aisyiyah Sumpalsari Bantul pada tahun 2006-2008. Kemudian melanjutkan pendidikan Sekolah Dasar (SD) di SD IT Al-Muhsin Metro pada tahun 2008-2014. Lalu penulis melanjutkan pendidikan Sekolah Menengah Pertama (SMP) di SMP Negeri 2 Metro pada tahun 2014-2017. Pada tahun 2017-2020, penulis melanjutkan pendidikan Sekolah Menengah Atas (SMA) di SMA Negeri 1 Metro. Pada tahun 2020 penulis melanjutkan pendidikan Strata Satu (S1) di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Lampung pada tahun 2020 melalui jalur SBMPTN.

Selama perkuliahan, penulis pernah menjadi pengurus HIMATIKA Unila dan menjabat sebagai anggota Bidang Keilmuan pada tahun 2022. Penulis melaksanakan Kerja Praktek (KP) di Dinas Komunikasi dan Informatika Kota Metro pada bulan Januari-Februari 2023. Lalu penulis juga melaksanakan Kuliah Kerja Nyata (KKN) pada bulan Juni-Agustus 2023 di Desa Sidoluhur, Kecamatan Bangun Rejo, Lampung Tengah. Pada bulan Agustus sampai dengan Desember 2023 penulis mengikuti kegiatan Merdeka Belajar – Kampus Merdeka (MBKM) yaitu Magang dan Studi Independen Bersertifikat (MSIB) Angkatan 5 dan lolos pada kegiatan Studi Independen di Rakamin Academy untuk *Data Science Intensive Bootcamp*.



## **KATA INSPIRASI**

*“Cukuplah Allah (menjadi penolong) bagi kami dan Dia sebaik-baiknya pelindung.”*  
(Q.S Al-Imran: 173)

*”Maka sesungguhnya bersama kesulitan ada kemudahan, sesungguhnya beserta kesulitan ada kemudahan.”*  
(Q.S Ash-Sharh: 5 - 6)

*“Bermimpi dan berdoa karena Tuhan akan memeluk mimpi-mimpimu itu.”*  
(Andrea Hirata)

*“You are on your own, kid”*  
(Taylor Swift)

## **PERSEMBAHAN**

Alhamdulillahirobbil'amin

Puji syukur kepada Allah SWT karena atas nikmat dan karunia-Nya, Shalawat serta salam selalu tercurah kepada baginda Nabi Muhammad SAW yang telah memberikan kabar gembira kepada umat manusia.

Dengan tulus hati, ku persembahkan karya sederhana penuh perjuangan dan kesabaran ini untuk:

**Ayahanda, Ibunda dan Adikku**

Terima kasih telah memberikan dukungan, serta selalu mendoakanku tanpa henti untuk menyelesaikan setiap langkah perjalananku dan atas bantuan yang terus menerus kalian berikan.

**Dosen Pembimbing dan Pembahas**

Terima kasih kepada dosen pembimbing dan pembahas yang telah membantu, memberikan motivasi, arahan serta ilmu yang berharga kepada penulis. Semoga Allah SWT senantiasa melimpahkan rahmat keberkahan, serta membalas semua kebaikan dosen pembimbing dan pembahas dengan paha yang berlimpah.

**Sahabat-sahabatku**

Terima kasih atas semua keceriaan dan semangat yang telah diberikan.

**Almamater Tercinta Universitas Lampung**

## SANWACANA

Puji syukur kehadirat Allah SWT atas limpahan Rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Model *Transfer Learning* VGG-16 Untuk Klasifikasi Citra *X-Ray* Tulang Belakang”. Dalam proses penyusunan skripsi ini, banyak pihak yang telah memberikan bimbingan, motivasi, bantuan, dan saran sehingga skripsi ini dapat terselesaikan. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku dosen Pembimbing I yang selalu berkenan memberikan arahan, bimbingan, saran, serta dukungan kepada penulis dalam penyusunan skripsi ini.
2. Bapak Ir. Warsono, M.S., Ph.D. selaku dosen Pembimbing II yang telah memberikan arahan, bimbingan, dan dukungan kepada penulis.
3. Ibu Prof. Dra. Wamiliana, M.A., Ph.D. selaku dosen Penguji yang telah memberikan kritik dan saran, serta evaluasi kepada penulis sehingga skripsi ini dapat lebih baik lagi.
4. Bapak Prof. Dr. La Zakaria, S.Si., M.Sc. selaku dosen pembimbing akademik yang telah memberikan bimbingan, motivasi, dan nasehat selama penulis menjalankan perkuliahan.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
7. Seluruh dosen, staff, dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.
8. Orang tua serta seluruh keluarga besar yang selalu memberikan motivasi dan

dukungan serta selalu mendoakan untuk kesuksesan penulis.

9. Sahabat-sahabat penulis Aulia Diah, Defina, Intan Putri, Nunung, dan Deta yang telah memberikan semangat, motivasi dan dukungan kepada penulis.
10. Mida Widya dan Tamam Abdiella yang selalu setia menemani penulis dalam menyelesaikan penyusunan skripsi ini.
11. Teman-teman satu bimbingan yang telah memberikan semangat, saran, dan bantuan kepada penulis.
12. Teman-teman mahasiswa Jurusan Matematika angkatan 2020.
13. Semua pihak yang terlibat dalam menyelesaikan skripsi ini tidak dapat disebutkan satu persatu.
14. Almamater tercinta Universitas Lampung.

Semoga skripsi ini dapat memberikan banyak manfaat untuk kita semua. Penulis juga menyadari bahwa skripsi ini masih jauh dari kata sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menyempurnakan skripsi ini.

Bandar Lampung, 4 November 2024

Penulis

Harum Aprelina Rahmad

NPM. 2017031092

## DAFTAR ISI

	Halaman
DAFTAR TABEL.....	xv
DAFTAR GAMBAR .....	xvi
I. PENDAHULUAN .....	1
1.1 Latar Belakang dan Masalah .....	1
1.2 Tujuan Penelitian.....	4
1.3 Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terkait .....	5
2.2 <i>Machine Learning</i> .....	9
2.3 <i>Deep Learning</i> .....	10
2.4 <i>Preprocessing Data</i> .....	10
2.5 <i>Convolutional Neural Network</i> .....	11
2.5.1 Lapisan Konvolusi .....	13
2.5.2 Lapisan <i>Pooling</i> .....	15
2.5.3 Lapisan <i>Fully-Connected</i> .....	16
2.6 <i>Visual Geometry Group (VGG-16)</i> .....	18
2.7 <i>Transfer Learning</i> .....	18
2.8 Pengujian Kinerja Model.....	19
III. METODOLOGI PENELITIAN.....	22
3.1 Tempat dan Waktu Penelitian .....	22
3.2 Data Penelitian .....	23
3.3 Metode Penelitian.....	24
3.3.1 <i>Data Preparation</i> .....	24

3.3.2 Modeling.....	29
IV. HASIL DAN PEMBAHASAN .....	59
4.1 Hasil.....	59
4.1.1 Hasil Pembentukan Model.....	59
4.1.2 Hasil Pelatihan Model.....	71
4.1.3 Evaluasi Model .....	84
4.2 Pembahasan .....	102
V. KESIMPULAN .....	105
5.1 Kesimpulan.....	105
DAFTAR PUSTAKA .....	106
LAMPIRAN.....	110

## DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terkait .....	5
2. <i>Confusion Matrix Multiclass</i> .....	19
3. Jumlah Kumpulan Data .....	25
4. Contoh Kumpulan Data Citra <i>X-Ray Tulang Belakang</i> .....	25
5. Pembagian Komposisi Kumpulan Data Setiap Percobaan .....	28
6. <i>Hyperparameter Percobaan F - Transfer Learning VGG16</i> .....	30
7. <i>Hyperparameter Percobaan A - CNN From Scratch</i> .....	61
8. <i>Hyperparameter Percobaan B - CNN From Scratch</i> .....	63
9. <i>Hyperparameter Percobaan C - CNN From Scratch</i> .....	65
10. <i>Hyperparameter Percobaan D – Transfer Learning VGG16</i> .....	67
11. <i>Hyperparameter Percobaan E - Transfer Learning VGG16</i> .....	69
12. <i>Hyperparameter Percobaan F – Transfer Learning VGG16</i> .....	71
13. <i>Training Model Percobaan A - CNN From Scratch</i> .....	72
14. <i>Training Model Percobaan B - CNN From Scratch</i> .....	73
15. <i>Training Model Percobaan C - CNN From Scratch</i> .....	75
16. <i>Training Model Percobaan D – Transfer Learning VGG16</i> .....	76
17. <i>Training Model Percobaan E - Transfer Learning VGG16</i> .....	79
18. <i>Training Model Percobaan F - Transfer Learning VGG16</i> .....	81
19. Hasil <i>Confusion Matrix</i> Percobaan A - <i>CNN From Scratch</i> .....	85
20. Hasil <i>Confusion Matrix</i> Percobaan B - <i>CNN From Scratch</i> .....	87
21. Hasil <i>Confusion Matrix</i> Percobaan C - <i>CNN From Scratch</i> .....	90
22. Hasil <i>Confusion Matrix</i> Percobaan D - <i>Transfer Learning VGG16</i> .....	93
23. Hasil <i>Confusion Matrix</i> Percobaan E - <i>Transfer Learning VGG16</i> .....	96
24. Hasil <i>Confusion Matrix</i> Percobaan F - <i>Transfer Learning VGG16</i> .....	99
25. Ringkasan Hasil Penelitian .....	102

## DAFTAR GAMBAR

Gambar	Halaman
1. Perbedaan Arsitektur MLP dan CNN .....	12
2. Ilustrasi <i>Padding</i> .....	13
3. Ilustrasi Operasi <i>Max Pooling</i> .....	15
4. Arsitektur VGG-16 .....	18
5. Ilustrasi <i>Resize Image</i> .....	27
6. Ilustrasi Augmentasi Data .....	29
7. <i>Flowchart</i> Arsitektur Percobaan F Model <i>Transfer Learning</i> VGG-16 .....	30
8. <i>Model summary</i> Arsitektur Model <i>Transfer Learning</i> VGG-16 (Terbaik).....	31
9. Output Tahapan <i>Transfer Learning</i> VGG-16 .....	33
10. Nilai Pixel RGB Kumpulan Data Label Skoliosis .....	34
11. <i>Kernel 3x3</i> .....	34
12. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Red</i> .....	35
13. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Red</i> .....	36
14. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Red</i> .....	37
15. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Red</i> .....	38
16. Hasil Tahapan Perhitungan Lapisan Konvolusi <i>Channel Red</i> .....	38
17. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Green</i> .....	39
18. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Green</i> .....	40
19. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Green</i> .....	41
20. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Green</i> .....	42
21. Hasil Tahapan Perhitungan Lapisan Konvolusi <i>Channel Green</i> .....	42
22. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Blue</i> .....	43
23. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Blue</i> .....	44
24. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Blue</i> .....	45
25. Ilustrasi Perhitungan pada Lapisan Konvolusi <i>Channel Blue</i> .....	46
26. Hasil Tahapan Perhitungan Lapisan Konvolusi <i>Channel Blue</i> .....	46
27. Hasil Perhitungan Lapisan Konvolusi <i>Channel Red, Green, Dan Blue</i> .....	47
28. Total Perhitungan Konvolusi .....	47
29. Hasil Konvolusi #1 .....	48
30. Ilustrasi Perhitungan Lapisan <i>Max Pooling</i> .....	49
31. Ilustrasi Proses Lapisan <i>Flatten</i> .....	50



32. Ilustrasi Prose <i>Dense + Softmax</i> .....	51
33. <i>Flowchart</i> Metode Penelitian.....	58
34. <i>Flowchart</i> Proses <i>Training</i> Model Percobaan A - <i>CNN From Scratch</i> .....	60
35. <i>Model Summary</i> Percobaan A - <i>CNN From Scratch</i> .....	60
36. <i>Flowchart</i> Proses <i>Training</i> Model Percobaan B - <i>CNN From Scratch</i> .....	62
37. <i>Model Summary</i> Percobaan B - <i>CNN From Scratch</i> .....	62
38. <i>Flowchart</i> Proses <i>Training</i> Model Percobaan C - <i>CNN From Scratch</i> .....	64
39. <i>Model Summary</i> Percobaan C - <i>CNN From Scratch</i> .....	64
40. Proses <i>Training</i> Model Percobaan D - <i>Transfer Learning</i> VGG16.....	66
41. <i>Model Summary</i> Percobaan D - <i>Transfer Learning</i> VGG16.....	66
42. <i>Flowchart</i> Proses <i>Training</i> Model Percobaan E - <i>Transfer Learning</i> VGG16.....	68
43. <i>Model Summary</i> Percobaan E - <i>Transfer Learning</i> VGG16.....	68
44. <i>Flowchart</i> Proses <i>Training</i> Model Percobaan F - <i>Transfer Learning</i> VGG16.....	70
45. <i>Model Summary</i> Percobaan F - <i>Transfer Learning</i> VGG16.....	70
46. Hasil Proses <i>Training</i> Percobaan A - <i>CNN From Scratch</i> .....	72
47. Hasil Proses <i>Training</i> Percobaan B - <i>CNN From Scratch</i> .....	74
48. Hasil Proses <i>Training</i> Percobaan C - <i>CNN From Scratch</i> .....	75
49. Hasil Proses <i>Training</i> Percobaan D - <i>Transfer Learning</i> VGG16.....	78
50. Hasil Proses <i>Training</i> Percobaan E - <i>Transfer Learning</i> VGG16.....	80
51. Hasil Proses <i>Training</i> Percobaan F - <i>Transfer Learning</i> VGG16.....	83
52. <i>Confusion Matrix</i> Percobaan A - <i>CNN From Scratch</i> .....	84
53. Hasil Pengujian Model Percobaan A - <i>CNN From Scratch</i> .....	86
54. <i>Confusion Matrix</i> Percobaan B - <i>CNN From Scratch</i> .....	87
55. Hasil Pengujian Model Percobaan B - <i>CNN From Scratch</i> .....	89
56. <i>Confusion Matrix</i> Percobaan C - <i>CNN From Scratch</i> .....	90
57. Hasil Pengujian Model Percobaan C - <i>CNN From Scratch</i> .....	92
58. <i>Confusion Matrix</i> Percobaan D - <i>Transfer Learning</i> VGG16.....	93
59. Hasil Pengujian Model Percobaan D - <i>Transfer Learning</i> VGG16.....	95
60. <i>Confusion Matrix</i> Percobaan E - <i>Transfer Learning</i> VGG16.....	96
61. Hasil Pengujian Model Percobaan E - <i>Transfer Learning</i> VGG16.....	98
62. <i>Confusion Matrix</i> Percobaan F - <i>Transfer Learning</i> VGG16.....	99
63. Hasil Pengujian Model Percobaan F - <i>Transfer Learning</i> VGG16.....	101

## I. PENDAHULUAN

### 1.1 Latar Belakang dan Masalah

*X-ray* merupakan bentuk pencitraan medis yang menggunakan sinar-X berupa radiasi elektromagnetik berenergi tinggi. Teknologi ini mampu memberikan gambaran struktur internal tubuh manusia berupa kondisi organ dalam tubuh dan kondisi tulang. Dengan demikian, *x-ray* mampu menjadi alat diagnostik dalam mengidentifikasi kelainan pada organ dan fraktur tulang (Ramadhan dkk., 2022).

Salah satu tulang yang dapat dilakukan pemeriksaan secara *x-ray* adalah tulang belakang. Pemeriksaan lebih dini mampu menghindari munculnya berbagai kelainan tulang belakang, yang disebabkan oleh pergeseran ruas tulang, postur tubuh, atau cedera tulang. Kelainan tulang belakang yang tidak terdeteksi dengan cepat dapat menyebabkan masalah serius, termasuk rasa nyeri dan perubahan postur tubuh (Solehatin dkk., 2022). Rasa nyeri yang muncul pun tidak hanya menjadi sumber ketidaknyamanan, melainkan berdampak pada keseimbangan tubuh yang terganggu.

Semakin banyaknya masyarakat yang memiliki masalah tulang belakang menjadi perhatian penting dalam dunia medis. Belum ada data tentang jumlah masyarakat yang menderita gangguan tulang di Indonesia, namun masyarakat dihadapkan dengan kasus kelainan tulang belakang yang semakin dirasakan dan dampak kelainan tulang belakang yang berpengaruh dalam kehidupan sehari-hari. Hal ini menunjukkan perlunya pencegahan dan penanganan yang lebih efektif.

Penanganan masalah tulang belakang tidak hanya melibatkan aspek klinis, melainkan juga menyangkut peran teknologi dalam proses diagnosis maupun pengelolaan. Klasifikasi citra menjadi elemen utama dalam dunia medis yang memungkinkan identifikasi lebih akurat terhadap kelainan tulang belakang. Teknologi ini membuka jalan untuk peningkatan efisiensi serta akurasi dalam pengenalan dan penanganan gangguan kesehatan tulang belakang, yang pada gilirannya dapat meningkatkan hasil perawatan dan kualitas hidup pasien secara keseluruhan (Sanjaya dkk., 2023).

Klasifikasi citra dalam dunia medis memainkan peran penting. Pada saat ini, perkembangan teknologi di bidang medis sudah berkembang dengan pesat. Klasifikasi citra adalah teknik pengelompokan piksel pada suatu citra ke dalam kelas tertentu sehingga setiap kelas dapat menggambarkan suatu entitas dengan karakter yang unik (Ambarwari dkk., 2023). Klasifikasi citra *x-ray* akan menambah efisiensi di bidang medis. Akan tetapi, terdapat kendala dalam otomatisasi klasifikasi citra seperti; kumpulan data medis yang sulit diakses, keahlian profesional dalam memberikan label pada gambar, dan waktu. Oleh karena itu, diperlukan penerapan *machine learning* (ML) dalam menangani masalah klasifikasi citra medis secara efektif (Yadav dan Jadhav, 2019). Sehingga, berdasarkan hasil klasifikasi citra menerapkan ML, dokter dapat menentukan diagnosis yang tepat dan merencanakan pengobatan yang sesuai.

*Convolutional Neural Network* (CNN) merupakan salah satu implementasi metode *machine learning* yang memiliki keunggulan dalam melakukan klasifikasi citra (Indrawan dkk., 2022). Model CNN merupakan hasil pengembangan dari metode *Multilayer Perceptron* (MLP) yang awalnya diberi nama *NeoCognitron*, kemudian dikembangkan menjadi arsitektur LeNet. Model CNN tersusun dari jaringan saraf tiruan yang dirancang untuk mengenali pola dalam citra. Jaringan tersebut terdiri dari jaringan yang saling terhubung dan dapat mengenali fitur tertentu. Namun, model CNN memiliki kelemahan, yaitu memerlukan kumpulan data yang banyak dan waktu yang lama. Hal ini karena, model CNN yang dilatih merupakan hasil

pengenalan pola bergantung pada kumpulan data yang diberikan (Ersyad dkk., 2020).

Masalah dalam melakukan klasifikasi citra adalah jumlah kumpulan data yang terbatas. *Transfer Learning* merupakan model yang memiliki kemampuan dalam menghasilkan informasi yang lebih banyak untuk klasifikasi karena memiliki lapisan yang lebih mendalam dibandingkan CNN sederhana (Rozaqi dkk., 2021). *Transfer Learning* memanfaatkan pengetahuan dimana mempelajari model yang sudah dilatih pada kumpulan data sebelumnya digunakan kembali pada kumpulan data yang baru, dan hal ini dapat membantu mengurangi kebutuhan data (Wijaya dkk., 2021). *Transfer Learning* juga dapat membantu mengurangi waktu dengan memulai dari model yang sudah dilatih. Terdapat beberapa jenis model *Transfer Learning* diantaranya adalah *Visual Geometry Group (VGG-16)*, *GoogLeNet*, *ResNet-50*, *MobileNet*, dan lainnya.

Penelitian yang pernah dilakukan terkait menggunakan metode *Transfer Learning* dengan membandingkan 14 *pre-trained model* untuk klasifikasi citra *x-ray* tulang belakang dengan pembagian data menjadi tiga kelas, yaitu normal, skoliosis dan spondylolisthesis. Hasil penelitian menunjukkan bahwa nilai akurasi tertinggi diperoleh *pre-trained model* *DenseNet-201* adalah 96,73% (Fraiwan dkk., 2022).

Penelitian lainnya, menggunakan metode *Transfer Learning* dengan *pre-trained model* *VGG-16*, *Interception-13* dan *ResNet-50* untuk mengklasifikasikan penyakit daun tanaman tomat. Klasifikasi dilakukan dengan membagi kelas menjadi 9 penyakit daun tomat dan 1 daun sehat. Hasil penelitian menunjukkan *pre-trained model* *VGG-16* paling stabil dan menghasilkan akurasi 97,78% (Rozi dkk., 2023).

Penelitian terkait lainnya, perbandingan kemampuan *Transfer Learning* *ResNet-50* dan *VGG-16* untuk klasifikasi citra *x-ray* paru-paru. Penelitian ini menggunakan kumpulan data berlabel *Covid-19*, *lung opacity*, *viral pneumonia*, dan normal. Hasil penelitian menunjukkan nilai akurasi antara *VGG-16* dan *ResNet-50* tidak terdapat

perbedaan yang signifikan. Dilihat dari kecepatan, hasil menunjukkan bahwa model *Transfer Learning* VGG-16 merupakan model yang hasil lebih optimal dibandingkan dengan ResNet-50 (Berliani dkk., 2023).

Berdasarkan penelitian terdahulu terdapat metode *Transfer Learning* yang memiliki keunggulan meningkatkan akurasi dengan kumpulan data yang tidak terlalu besar. Oleh karena itu, penulis tertarik untuk melakukan penelitian mengenai klasifikasi data citra *x-ray* tulang belakang menggunakan metode *Transfer Learning* VGG-16.

## 1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Menerapkan sistem klasifikasi citra *x-ray* tulang belakang dengan menggunakan *Transfer Learning* dengan *pretrained* model VGG-16.
2. Mengetahui kinerja model dari metode *Transfer Learning* VGG-16 untuk klasifikasi citra *x-ray* tulang belakang

## 1.3 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Memberikan pengetahuan mengenai implementasi metode *Transfer Learning* VGG-16 untuk klasifikasi citra *x-ray* tulang belakang menjadi 3 kelas yaitu normal, *scoliosis*, dan *spondylolisthesis*.
2. Membantu perkembangan teknologi di bidang radiologi sebagai sistem pendukung dalam mengambil keputusan.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Penelitian ini tidak terlepas dari hasil penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Hasil penelitian yang dijadikan perbandingan berkaitan dengan topik penelitian yaitu tentang klasifikasi citra menggunakan metode *Transfer Learning* VGG-16 dan hasil kinerja model melihat nilai *accuracy* (*acc*), *precision* (*prec*), *recall* (*rec*), dan *F1-score* (*F1*). Berdasarkan penelitian sebelumnya, berikut ini contoh penerapan metode *Transfer Learning* digunakan sebagai kajian yang jelaskan secara singkat melalui Tabel 1.

Tabel 1. Penelitian Terkait

No	Penelitian	Data	Metode	Hasil			
				<i>Acc</i>	<i>Prec</i>	<i>Rec</i>	<i>F1</i>
1.	<i>Using deep Transfer Learning to detect scoliosis and spondylolisthesis from x-ray images</i> (Fraiwandkk., 2022).	<i>X-ray images</i> Jumlah : 338 <i>Scoliosis</i> : 188 <i>Spondylolisthesis</i> : 79 Normal : 71  Sumber : Data Mendeley	Metode Klasifikasi : <i>Transfer Learning</i> (DenseNet-201)	97,0%	97,9%	95,0%	96,3%

2	<p>Identifikasi Kinerja Arsitektur <i>Transfer Learning</i> VGG16, ResNet-50, dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat (Rozi dkk., 2023).</p>	<p>Gambar Daun Kentang          Jumlah : 10000 data          latihan 1000 data uji, dengan masing masing kelas 1000 data latihan dan 100 data uji</p> <ul style="list-style-type: none"> <li>- <i>Tomato Mosaic Virus</i></li> <li>- <i>Target Spot</i></li> <li>- <i>Bacterial Spot</i></li> <li>- <i>Tomato Yellow Leaf Curl Virus</i></li> <li>- <i>Late Blight</i></li> <li>- <i>Leaf Mold</i></li> <li>- <i>Early Blight</i></li> <li>- <i>Septoria Leaf Spot</i></li> <li>- <i>Spider mites Two spotted spider mite</i></li> <li>- <i>Healthy Tomato Leaf</i></li> </ul> <p>Sumber : Kaggle dan GitHub</p>	<p>Metode Klasifikasi : <i>Transfer Learning</i> VGG-16, ResNet-50, dan Inception-V3</p>	<p>VGG-16 : 97,78%</p> <p>ResNet50 : 94,67%</p> <p>Inception-V3 : 95,4%</p>			
3	<p>Perbandingan Kemampuan Klasifikasi Citra X-ray Paru-paru</p>	<p><i>X-ray images</i>          Jumlah : 21165          Covid-19 : 3616  <i>Lung opacity</i> : 6012</p>	<p>Metode Klasifikasi: ResNet50 dan VGG-16</p>	<p>ResNet-50 : 95,1%</p> <p>VGG-16 : 91,4%</p>	<p>ResNet-50 : 92,8%</p> <p>VGG-16 : 92,5%</p>	<p>ResNet-50 : 85,0%</p> <p>VGG-16 : 90,3%</p>	<p>ResNet-50 : 88,8%</p> <p>VGG-16 : 95,6%</p>

	mengguna kan <i>Transfer Learning</i> ResNet-50 dan VGG- 16 (Berli ani dkk., 2023).	<i>Viral pneumonia</i> : 1345 Normal : 10192 Sumber : Github dan Kaggle dari University of Montreal, dan Qatar University berkolaborasi dengan University of Dhaka Bangladesh dengan lisensi CC BY-SA 4.0					
--	--	---	--	--	--	--	--

Berdasarkan Tabel 1 dapat diuraikan sebagai berikut :

a. Penelitian Pertama (Fraiwan dkk., 2022)

Penelitian ini pernah dilakukan terkait menggunakan metode *Transfer Learning* untuk klasifikasi citra *x-ray* tulang belakang. Kumpulan data sebanyak 338 citra dengan pembagian data menjadi tiga kelas, yaitu 71 citra berlabel normal, 188 citra berlabel *scoliosis* dan 79 citra berlabel *spondylolisthesis*. Data yang digunakan bersumber dari Mendeley yang dikumpulkan dari King Abdullah University Hospital, Jordan University of Science and Technology, Irbid, Jordan.

Penelitian ini menggunakan 14 *pre-trained model* yang digunakan yaitu SqueezeNet, GoogLeNet, Inception-V3, DenseNet-201, MobileNets, ResNet-101, ReNet-50, ResNet-18, The Xception, The Inception-ResNet-v2, ShuffleNet, NASnetMobile, DarkNet-53, dan EfficientNet-b0. Hasil penelitian menunjukkan bahwa *pre-trained model* DenseNet-201 memiliki nilai akurasi 97,0%, *Precision* 97,9%, *Recall* 95,0%, dan *F1-Score* 96,3%.

b. Penelitian Kedua (Rozi dkk., 2023)



Penelitian selanjutnya membahas mengenai metode *Transfer Learning* dengan *pre-trained model* VGG-16, Inception-13 dan ResNet-50 untuk mengklasifikasikan penyakit daun tanaman tomat. Data yang digunakan adalah 10000 citra gambar data latih dan 1000 citra gambar validasi. Dalam penelitian ini terdapat 10 kelas berisi 9 jenis penyakit citra daun tomat dan 1 citra daun tomat yang sehat. Penyakit daun tomat diantaranya yaitu *Tomato Mosaic Virus*, *Target Spot*, *Bacterial Spot*, *Tomato Yellow Leaf Curl Virus*, *Late Blight*, *Leaf Mold*, *Early Blight*, *Septoria Leaf Spot*, dan *Spider mites Two-spotted spider mite*.

Penelitian dilakukan terhadap penggunaan CNN pada *Transfer Learning* VGG-16, ResNet-50, dan Inception V-3. Masing-masing arsitektur diperkuat dengan penambahan 2 lapisan *Dense* (4096 node) dan 2 lapisan *dropout* (probabilitas 0,5). Penggunaan *fine-tuning* dalam pelatihan model dinilai lebih efektif dan efisien dibandingkan dengan pelatihan model secara manual. Hasil penelitian menunjukkan kinerja model VGG-16 akurasi yang tinggi, yaitu 97,7%.

c. Penelitian Ketiga (Berliani dkk., 2023)

Penelitian lainnya, perbandingan kemampuan *Transfer Learning* ResNet-50 dan VGG-16 untuk klasifikasi citra *x-ray* paru-paru. Penelitian ini menggunakan kumpulan data sebanyak 3616 citra berlabel Covid-19, 6012 citra berlabel *lung opacity*, 1345 citra berlabel *viral pneumonia* dan 10.192 citra berlabel normal. Sumber data yang digunakan untuk penelitian ini mengambil dua kumpulan data publik dari Kaggle dan Github dengan lisensi CC BY-SA 4.0 yang bersumber dari University of Montreal, dan Qatar University berkolaborasi dengan University of Dhaka Bangladesh

Pada percobaan dilakukan *fine-tuning* parameter nilai epoch untuk meningkatkan akurasi. Hasil penelitian ini, menunjukkan perbedaan akurasi dari nilai epoch 10, epoch 100, dan epoch 400 sangat berbeda jauh, model VGG-16 menghasilkan nilai akurasi 91,45% sedangkan ResNet-50 menghasilkan nilai 95,13%. Penelitian menunjukkan nilai akurasi antara VGG-16 dan ResNet-50 tidak terdapat perbedaan

yang signifikan. Dilihat dari kecepatan, model *Transfer Learning* VGG-16 merupakan model yang hasilnya lebih optimal dibandingkan dengan ResNet-50.

## 2.2 Machine Learning

*Machine learning* (ML) adalah bagian dari bidang kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan komputer untuk belajar dari data sehingga berpotensi memecahkan masalah dari berbagai bidang (Roihan dkk., 2020). *Machine learning* menerapkan teknik statistik dan algoritma dalam mengotomatisasi model melalui pelatihan model, komputer dapat menggunakan pola untuk membuat keputusan. *Machine learning* bekerja dengan cara menggunakan data untuk melatih komputer menemukan pembelajaran tersembunyi tanpa pemrograman eksplisit saat mencarinya (Kamil dkk., 2021). Data yang digunakan dapat berupa data teks, gambar, suara dan lain-lain. *Machine learning* dibagi menjadi tiga kategori, berdasarkan data untuk memecahkan masalah diantaranya *supervised learning*, *unsupervised learning*, dan *reinforcement learning*

### 1. *Supervised Learning*

*Supervised learning* merupakan algoritma yang proses pembelajarannya dibawah pengawasan. Pembelajaran ini dapat mengatasi masalah yang rumit dengan melatih sistem saraf buatan dengan data yang sudah berlabel. *Supervised learning* banyak digunakan dalam masalah klasifikasi dan regresi (Sidik dan Ansawarman, 2022).

### 2. *Unsupervised Learning*

*Unsupervised learning* merupakan algoritma yang proses pembelajarannya tanpa pengawasan. Pembelajaran tak terarah dimana data tidak diberi label untuk data pelatihannya. *Unsupervised learning* banyak digunakan dalam penyelesaian masalah *clustering* dan *dimensionality reduction* (Sidik dan Ansawarman, 2022).

### 3. *Reinforcement learning*

*Reinforcement learning* merupakan algoritma yang bisa membuat *agent software* mesin bekerja secara otomatis untuk menentukan perilaku ideal sehingga dapat memaksimalkan kinerja algoritma. Penggunaan algoritma ini diaplikasikan pada navigasi robot, *game AI*, *real-time decision*, dan *real-time navigation* (Sidik dan Ansawarman, 2022).

### **2.3 Deep Learning**

*Deep learning* (DL) merupakan salah satu bidang ML yang melakukan ekstraksi fitur, identifikasi pola, dan klasifikasi dengan menggunakan banyak lapisan dalam pengolahan informasi nonlinear (Efrian dan Latifa, 2022). Jaringan saraf adalah pembangun dasar model pada DL. Berbeda dengan ML yang belajar menggunakan model statistik, DL belajar menggunakan algoritma jaringan syaraf tiruan dalam menyelesaikan masalah. *Deep learning* adalah sebuah metodologi pemecahan masalah dalam sistem pembelajaran komputer yang memanfaatkan konsep hierarki. Komputer dapat mempelajari konsep yang rumit dengan menggabungkan konsep yang lebih sederhana sesuai dengan konsep hierarki (Hermawati dan Zai, 2021)

Secara signifikan, DL terletak pada fakta bahwa jumlah lapisan tersembunyi lebih penting daripada luasnya jaringan. Banyak lapisan tersembunyi sering kali dibutuhkan dalam permasalahan praktis agar input dapat menghasilkan output. Input ke dalam sebuah representasi diwakili oleh setiap lapisan tersembunyi. Dengan kata lain, kualitas input diubah menjadi bentuk baru yang dipelajari oleh lapisan tersembunyi satu sama lain (Harani dkk., 2019).

### **2.4 Preprocessing Data**

*Preprocessing* data merupakan suatu proses untuk menghilangkan *noise* atau bagian yang tidak diperlukan pada gambar (Febrianti dkk., 2020). *Preprocessing* data adalah tahap penting dalam persiapan data sebelum diproses melakukan pelatihan data pada jaringan saraf untuk menyeragamkan ukuran citra. Hasil

*preprocessing* kumpulan data gambar digunakan model dalam mendeteksi objek dapat bekerja dengan optimal. *Preprocessing* yang digunakan sebagai berikut :

1. *Resize*

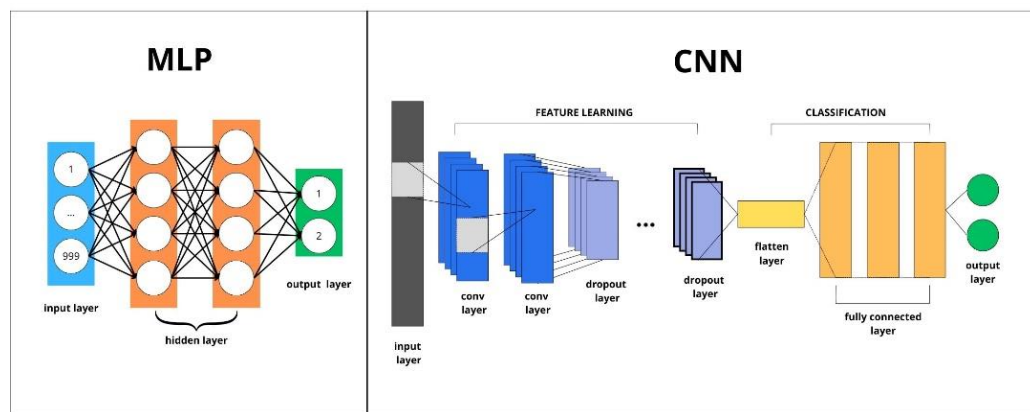
*Resize* adalah proses mengubah ukuran citra menjadi lebih kecil. Proses *resize* tidak memerlukan metode khusus, dengan cara membandingkan ukuran citra dari segmentasi dengan target ukuran citra. *Resize* diperlukan untuk mempercepat dan tidak menghabiskan penyimpanan dalam komputasi.

2. *Rescale*

*Rescale* adalah proses mengubah skala nilai piksel dalam citra. Hal ini, dimana setiap nilai piksel dijadikan dalam rentang antara 0 hingga 1 dengan membagi setiap nilai piksel dengan 255. *Rescale* merupakan langkah umum dalam *preprocessing* citra karena intensitas piksel biasanya berada pada rentang 0 hingga 255.

## **2.5 Convolutional Neural Network**

*Convolutional Neural Network* (CNN) merupakan salah satu metode dalam DL yang telah menunjukkan kemampuannya dalam menangkap representasi bermakna, termasuk dalam implementasi klasifikasi dan pemodelan citra (Irfansyah dkk., 2021). Asal mula CNN diambil karena pada jaringan tersebut terdapat suatu operasi matematika yang disebut konvolusi. Model CNN tercipta karena terinspirasi oleh sistem saraf biologis untuk menggabungkan input dari neuron yang berbeda, didalamnya tersusun lapisan-lapisan pemrosesan yang beroperasi secara paralel untuk melakukan tindakan konvolusi (Batubara dan Awangga, 2020). Perbedaan arsitektur *Multi Layer Perceptron* (MLP) sederhana dengan CNN diilustrasikan pada Gambar 1 sebagai berikut :



Gambar 1. Perbedaan Arsitektur MLP dan CNN

Model CNN mampu mengatasi permasalahan MLP dalam konteks pelatihan model untuk validasi (Muhammad dan Wibowo, 2021). Model CNN dapat menyelesaikan masalah ini dengan menggunakan sebuah filter pendeteksi yang fungsinya mendeteksi gambar di posisi manapun. model CNN adalah arsitektur yang dapat dilatih dan terdiri dari beberapa lapisan. Setiap lapisan CNN menerima input dari lapisan sebelumnya dan menghasilkan output yang dikirim ke lapisan selanjutnya. Input dan output dari masing-masing lapisan adalah representasi dari input yang telah diproses oleh lapisan sebelumnya dikenal sebagai *feature map* (Wahyuni dan Sulaeman, 2022).

Data gambar diproses secara bertahap menggunakan arsitektur CNN. Konvolusi adalah tahap awal, ketika karakteristik fundamental dalam gambar dideteksi menggunakan *kernel*. Selanjutnya tahap aktivasi, ketika hasil konvolusi dinormalisasi menggunakan fungsi *Rectified Linear Unit* (ReLU). Untuk meminimalkan jumlah parameter yang perlu dipelajari. Tahap terakhir, yang dikenal sebagai *pooling*, tahap ini bertujuan untuk memperkecil ukuran *feature map*. Proses ini diulang hingga *feature map* yang dihasilkan sudah sesuai dengan kebutuhan model sampai pada titik di mana jaringan saraf dapat bekerja dengan baik tanpa kehilangan informasi penting untuk melakukan tugas klasifikasi (Yanto dkk., 2021). Masing-masing tahap terdiri dari tiga lapisan yaitu lapisan konvolusi, lapisan *pooling*, dan lapisan *fully-connected*.

## 2.5.1 Lapisan Konvolusi

Lapisan konvolusi merupakan pondasi utama untuk membangun CNN. Lapisan ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Lapisan konvolusi pada dasarnya digunakan untuk mendeteksi keberadaan fitur dalam citra (tepi, lekukan, tekstur, dll). Lapisan konvolusi memiliki peran dalam melakukan pengenalan gambar, banyak jenis fitur yang harus dipertimbangkan dalam kombinasi antar fitur, sehingga menghasilkan satu lapisan konvolusi yang sering kali menyertakan sejumlah *feature map* yang berbeda (Putro, dkk., 2020). Tujuan melakukan konvolusi pada data citra yaitu untuk mengekstraksi fitur dari citra input (Cahya dkk., 2021).

### 2.5.1.1 *Padding*

*Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah piksel (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi output dari *feature map*. Penggunaan *padding* dapat untuk mengatur dimensi output agar tetap sama seperti dimensi input atau setidaknya tidak berkurang drastis sehingga dapat dilakukan ekstraksi *feature* yang lebih mendalam (Meliuwati dan Kurniati, 2022). Ilustrasi *Padding* disajikan pada Gambar 2 sebagai berikut :

0	0	0	0	0	0	0	0
0	60	113	56	139	85	0	0
0	73	121	54	84	128	0	0
0	131	99	70	129	127	0	0
0	80	57	115	69	134	0	0
0	104	126	123	95	130	0	0
0	0	0	0	0	0	0	0

Kernel		
0	-1	0
-1	5	-1
0	-1	0

114				

Gambar 2. Ilustrasi *Padding* (Yenusi dkk, 2023)

### 2.5.1.2 *Stride*

*Stride* adalah parameter yang menentukan seberapa banyak filter bergeser saat dikonvolusi dengan input. Nilai *stride* yang lebih kecil menghasilkan *feature map* yang lebih kecil, tetapi juga menangkap lebih banyak detail dari input. Nilai *stride* yang lebih besar menghasilkan *feature map* yang lebih besar, tetapi menangkap lebih sedikit detail dari input (Meliuwati dan Kurniati, 2022).

### 2.5.1.3 *Rectified Linear Unit (ReLU)*

Operasi pada lapisan ini sama dengan operasi konvolusi yaitu melakukan operasi kombinasi linier filter terhadap daerah lokal yang disebut operasi ReLU. Filter ini merupakan representasi bidang reseptif dari neuron yang terhubung ke dalam daerah lokal pada input gambar (Darmanto, 2019). Tujuan dari ReLU adalah untuk mengurangi linearitas akibat dari proses konvolusi sehingga CNN lebih mudah untuk mencapai hasil yang optimum. Lapisan ini menggantikan angka negatif dari lapisan konvolusi dengan 0 (nol), yang membantu untuk pelatihan yang lebih cepat dan efektif sehingga didapatkan hasil konvolusi. Persamaan (2.1) ReLU adalah

$$f(x) = \max(0, x) \begin{cases} x, & \text{jika } x \geq 0 \\ 0, & \text{jika } x < 0 \end{cases} \quad (2.1)$$

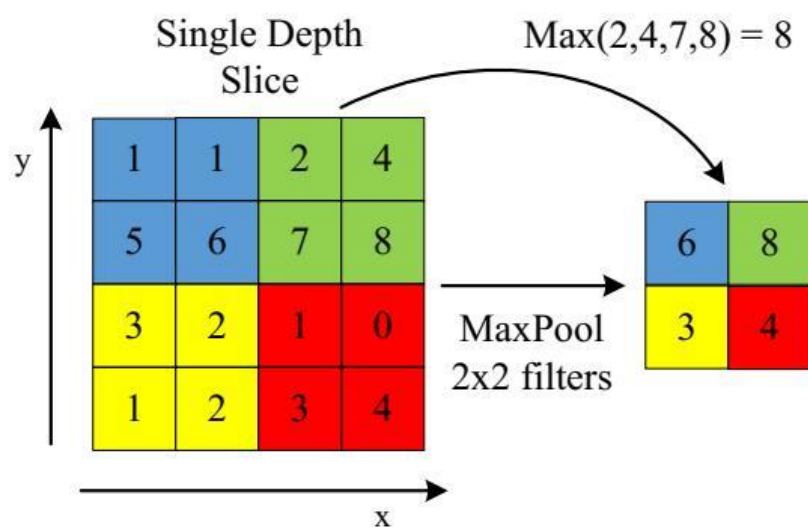
Keterangan :

$f(x)$  = nilai dari aktivasi ReLU

$x$  = nilai matriks dari citra

### 2.5.2 Lapisan *Pooling*

Lapisan *pooling* menerima output dari lapisan konvolusi, pada lapisan ini ukuran data citra akan direduksi lapisan konvolusi. Lapisan *pooling* memiliki cara kerja yang sama seperti lapisan konvolusi, yang membedakan adalah *filter* yang digunakan lapisan *pooling* bukan bertujuan untuk mendeteksi sesuatu melainkan antara mencari nilai terbesar atau rata-rata yang biasa disebut *Max Pooling* dan *Average Pooling*. Lapisan *pooling* digunakan untuk mengecilkan ukuran *feature map* dengan cara merangkum informasi penting. Proses ini bertujuan untuk mengurangi dimensi data sambil tetap mempertahankan fitur-fitur utama, sehingga mengurangi beban komputasi dan mencegah *overfitting* pada model. Prinsipnya lapisan *pooling* terdiri dari *filter* dengan ukuran tertentu dan *stride* kemudian bergeser ke seluruh area *feature map*. Ilustrasi operasi *Max Pooling* dapat dilihat pada Gambar 3 sebagai berikut :



Gambar 3. Ilustrasi Operasi *Max Pooling* (Miftahuddin dan Adani, 2022)

Pada arsitektur CNN metode *pooling* yang sering digunakan adalah *Max Pooling*. *Max Pooling* umumnya menggunakan *kernel size* 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari inputnya. (Shukla dan Fricklas, 2018).



Output dari proses *pooling* adalah matriks dengan dimensi yang lebih kecil dibanding dengan matriks awal. Proses konvolusi dan *pooling* dilakukan beberapa kali sehingga didapatkan *feature map* dengan ukuran yang dikehendaki. *Feature map* tersebut akan menjadi input bagi jaringan saraf *fully-connected*.

### 2.5.3 Lapisan *Fully-Connected*

Lapisan *fully-connected* mengambil input dari hasil output lapisan *pooling* yang berupa *feature map*. *Feature map* tersebut masih berbentuk *multi-dimensional array* maka lapisan ini akan melakukan *flatten* dan menghasilkan vektor sebanyak  $n$ -dimensi bisa digunakan sebagai input dari lapisan *fully-connected*. Lapisan *Fully-connected* yang dimaksud disini adalah MLP yang sudah pernah dipelajari (Nurani dkk., 2022).

Lapisan *fully-connected* adalah lapisan terakhir di mana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan saraf tiruan biasa (Darmanto, 2019). Setiap aktivasi dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di lapisan. Lapisan *Fully-connected* biasanya digunakan pada metode MLP untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara lapisan *fully-connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada input, sementara lapisan *fully-connected* memiliki neuron yang secara keseluruhan terhubung. Berikut perhitungan yang terdapat pada di dalam yang didefinisikan pada Persamaan (2.2) sebagai berikut :

$$z_j = \sum_{i=1}^c W_{i,j} x_i + b_j \quad (2.2)$$

Keterangan :

$z_j$  = nilai keluaran dari jaringan

$x_i$  = nilai masukan hasil ekstraksi fitur

$W_{i,j}$  = bobot dari jaringan berukuran  $i \times j$

$i$  = jumlah fitur masukan

$j$  = jumlah target kelas

$b_j$  = bias dari jaringan

### 2.5.3.1 Softmax

Lapisan *fully-connected* terdiri dari lapisan input, lapisan tersembunyi dan lapisan output. Pada lapisan *fully-connected* semua neuron di lapisan terhubung secara menyeluruh. Pada lapisan tersembunyi terdapat fungsi aktivasi, yang umum digunakan yaitu ReLU, dan begitu pula di lapisan output, juga terdapat fungsi aktivasi, dalam kasus klasifikasi yang umum digunakan yaitu *softmax*. Aktivasi *softmax* digunakan untuk menghitung kemungkinan dari setiap kelas atas semua kelas yang memungkinkan dan membantu untuk menentukan kelas target untuk input citra yang diberikan (Paraijun dkk., 2022). *Softmax* memiliki fungsi untuk mengubah logits menjadi 3 output, sehingga *softmax* cocok untuk menjadi fungsi aktivasi untuk kumpulan dengan 3 atau lebih kategori. Persamaan (2.3) *softmax* sebagai berikut :

$$S_j = \frac{e^{z_j}}{\sum_{d=1}^c e^{z_d}} \quad (2.3)$$

Keterangan :

$z_j$  = nilai keluaran dari jaringan

$S_j$  = probabilitas dari  $z_j$  untuk setiap C kelas yang berbeda

$\sum_c = \sum_{d=1}^c e^{z_d}$

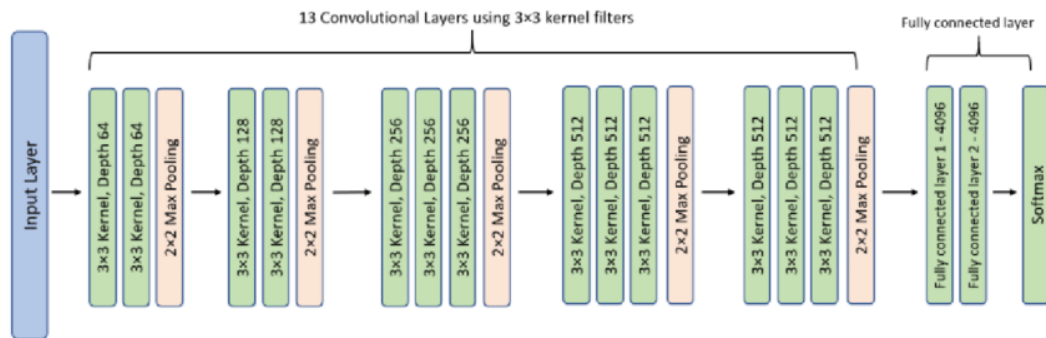
$e$  = eksponen

$i$  = jumlah fitur masukan

$j$  = jumlah target kelas

## 2.6 Visual Geometry Group (VGG-16)

*Visual Geometry Group* merupakan sebuah *pre-trained* model CNN yang menggunakan lapisan konvolusi dengan spesifikasi filter konvolusi yang kecil ( $3 \times 3$ ) (Pangestu dan Bunyamin, 2018). Arsitektur VGG-16 yang berupa input, lapisan konvolusi, lapisan konvolusi, *max-pooling*, lapisan konvolusi, lapisan konvolusi, lapisan konvolusi, *max-pooling* kemudian masuk kedalam dense. Model VGG-16 menggunakan ukuran  $3 \times 3$  sebagai ukuran *kernel* dan tiga lapisan terakhir adalah lapisan *fully-connected* dengan ukuran 4096 (Hasan dkk., 2021). Struktur Arsitektur VGG-16 dapat dilihat pada Gambar 4



Gambar 4. Arsitektur VGG-16 (Rismiyati dan Luthfiarta, 2021)

Model VGG-16 dapat diakses melalui ImageNet ini memiliki kemampuan untuk mengekstrak fitur-fitur, seperti spasial, garis tepi, rotasi, pencahayaan, dan bentuk. VGG-16 dapat digunakan dalam proses *Transfer Learning* dengan cara mengekstrak fitur pada citra memasukkan dan mengirimkannya sebagai masukan bagi algoritma lain.

## 2.7 Transfer Learning

*Transfer Learning* adalah metode di mana pengetahuan yang telah dipelajari dari satu tugas digunakan untuk meningkatkan kinerja pada tugas terkait (Albani dan

Andhi, 2023). *Transfer Learning* dapat digunakan untuk menyempurnakan beberapa lapisan akhir sehingga lebih disesuaikan dengan kumpulan data yang digunakan, atau dapat digunakan untuk menghapus lapisan akhir yang sepenuhnya terhubung dari model untuk membuat ekstraktor fitur tetap (Zuhan dan Kristian, 2023).

*Transfer Learning* melatih model dengan memanfaatkan ImageNet sebagai pengetahuan umum untuk model. Dalam *Transfer Learning* terdapat beberapa istilah seperti *pre-trained* dan *fine-tuning*. *Transfer Learning* nantinya akan melatih model dengan menggunakan data besar, inilah yang disebut *pre-trained*. Selain *pre-trained*, *fine-tuning* juga merupakan sebuah cara yang biasanya dipakai dalam melatih model. Model yang sudah dilakukan *pre-trained* akan dilatih kembali menggunakan data baru untuk kebutuhan lain (Wahid dkk., 2023).

## 2.8 Pengujian Kinerja Model

Pengujian kinerja model digunakan untuk mengevaluasi kinerja model klasifikasi dan pengenalan objek pada ML, *Confusion Matrix* memberikan output berupa nilai kinerja model klasifikasi berdasarkan objek berupa akurasi, *precision* dan *recall* (Tangkelayuk dan Mailoa, 2022). *Confusion Matrix* disajikan dalam Tabel 2 sebagai berikut :

Tabel 2. *Confusion Matrix Multiclass* (Markoulidakis dkk., 2021)

		Predict Class			
		$C_1$	$C_2$	...	$C_N$
Actual Class	$C_1$	$C_{1,1}$	FP	...	$C_{1,N}$
	$C_2$	FN	TP	...	FN
	...	...	...	...	...
	$C_N$	$C_{N,1}$	FP	...	$C_{N,N}$

Terdapat empat istilah untuk merepresentasikan hasil proses klasifikasi tersebut, yaitu :

- TP (*True Positive*) : data positif yang terbukti kebenarannya.
- FP (*False Positive*) : data negatif yang terbukti kebenarannya.
- TN (*True Negative*) : data positif yang tidak terbukti kebenarannya.
- FN (*False Negative*) : data negatif yang tidak terbukti kebenarannya.

Penjelasan dan persamaan yang digunakan dalam menentukan kinerja model sebagai berikut:

### 1. *Recall*

*Recall* merupakan pembagian dari jumlah total data positif yang diprediksi bernilai benar dibagi dengan jumlah total data positif (Markoulidakis dkk., 2021). *Recall* dihitung dengan Persamaan (2.4) sebagai berikut:

$$Recall(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (2.4)$$

### 2. *Precision*

*Precision* dapat didefinisikan sebagai rasio dari jumlah total data positif yang diprediksi bernilai benar dengan jumlah total data positif yang diprediksi (Markoulidakis dkk., 2021). *Precision* dihitung dengan Persamaan (2.5) sebagai berikut:

$$Precision(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (2.5)$$

### 3. *F1-Score*

*F1-score* adalah *Harmonic Mean* antara *precision* dan *recall*. Nilai *F1-score* berada antara 0 dan 1 dimana 1 adalah nilai terbaik. *F1-Score* yang baik mengindikasikan model klasifikasi memiliki *precision* dan *recall* yang baik (Markoulidakis dkk., 2021). *F1-Score* dihitung dengan Persamaan (2.6) sebagai berikut:

$$F1\ Score(C_i) = \frac{2 \times Precision(C_i) \times Recall(C_i)}{Precision(C_i) + Recall(C_i)} \quad (2.6)$$

#### 4. Akurasi

Akurasi dapat menunjukkan seberapa akurat model dapat mengklasifikasikan dengan benar berdasarkan pembagian jumlah prediksi yang benar (positif dan negatif) dengan keseluruhan (Markoulidakis dkk., 2021). Akurasi dihitung dengan Persamaan (2.7) sebagai berikut:

$$Accuracy = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N TP(C_{i,j})} \quad (2.7)$$

### **III. METODOLOGI PENELITIAN**

#### **3.1 Tempat dan Waktu Penelitian**

Tempat dan Waktu Penelitian ini dapat diuraikan sebagai berikut :

##### **1. Tempat Penelitian**

Penelitian ini dilaksanakan secara studi literatur bertempat di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

##### **2. Waktu Penelitian**

Penelitian ini dilakukan pada semester ganjil tahun akademik 2023/2024. Tahapan perencanaan dilakukan untuk mengestimasi proses penyelesaian skripsi di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Setelah berkonsultasi dan mengetahui sistem yang akan dibuat tahapan perencanaan dan melakukan estimasi waktu yang digunakan. Rincian tahapan perencanaan dibagi menjadi 3 tahap sebagai berikut :

- Tahap awal dilakukan pengajuan tema, kemudian studi literatur sesuai dengan tema yang sudah diajukan. Kemudian, dilakukan pencarian dan pengumpulan data sebagai objek penelitian. Pada tahap ini juga dilakukan penyusunan proposal penelitian.
- Tahap kedua, tahapan ini merupakan tahap paling penting yaitu tahap pengerjaan hasil.
- Tahapan terakhir, penyusunan hasil penelitian dan menyampaikan hasil dan kesimpulan yang didapat dari hasil penelitian pada seminar hasil penelitian. Penelitian ini diakhiri dengan uji komprehensif.

### 3.2 Data Penelitian

Data dan alat yang digunakan pada penelitian ini adalah sebagai berikut :

#### a. Data

Data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari situs Kaggle

<https://www.kaggle.com/datasets/yasserhessein/the-vertebrae-xray-images>.

Data citra *x-ray* tulang belakang dikumpulkan secara lokal di Rumah Sakit Universitas King Abdullah, Universitas Sains dan Teknologi Yordania, Irbid, Yordania. Gambar-gambar tersebut diambil sebagai bagian dari prosedur diagnostik standar di King Abdullah University Hospital (KAUH) untuk pasien yang menderita penyakit tulang belakang. Gambar-gambar tersebut dipesan oleh para spesialis di KAUH dan dilakukan oleh ahli radiologi yang tepat di rumah sakit yang sama.

Data yang digunakan dalam penelitian ini berisi kumpulan data mencakup 338 citra (240 perempuan, 98 laki-laki). Jumlah citra dengan label normal adalah 71 (40 perempuan, 31 laki-laki). Jumlah citra yang diberi label spondylolisthesis adalah 79 (49 perempuan, 30 laki-laki). Jumlah citra yang diberi label skoliosis adalah 188 (151 perempuan, 37 laki-laki). Terdapat beberapa ukuran gambar pada kumpulan data, penelitian ini menggunakan ukuran sebenarnya sebelum dimanipulasi.

#### b. Alat

Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut :

##### 1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut :

- Processor : AMD Athlon with Radeon Graphics 2.30 GHz
- Installed RAM : 4,00 GB

##### 2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut :

- Sistem Operasi Windows 10 Pro 64-bit



- Jupyter Notebook Python 3.10.12

Adapun *packages* yang digunakan dalam penelitian ini adalah sebagai berikut :

- *pandas* versi 1.5.3
- *pathlib* versi 1.0.1
- *numpy* versi 1.25.2
- *matplotlib* versi 3.7.1
- *keras* versi 2.15.0
- *scikit-learn* versi 1.2.2
- *tensorflow* versi 2.15.0

### **3.3 Metode Penelitian**

#### **3.3.1 Data Preparation**

Pada tahap ini mencakup proses mengumpulkan data dan mempelajari data untuk mengenal data yang akan dipakai. Fase ini, dilakukan identifikasi masalah yang berkaitan dengan kualitas data. Dalam penelitian ini, data yang akan digunakan adalah kumpulan data yang digunakan pada penelitian ini adalah kumpulan data sekunder dari internet, yang diperoleh dari *website Kaggle*.

##### **3.3.1.1 Pengumpulan Data**

Penelitian ini mengumpulkan data untuk memperoleh data atau informasi yang dibutuhkan dalam melaksanakan penelitian. Adapun metode pengumpulan data yang digunakan pada penelitian ini adalah kumpulan data sekunder dari internet, yang diperoleh dari *website Kaggle*, melalui

<https://www.kaggle.com/s/yasserhessein/the-vertebrae-xray-images>.

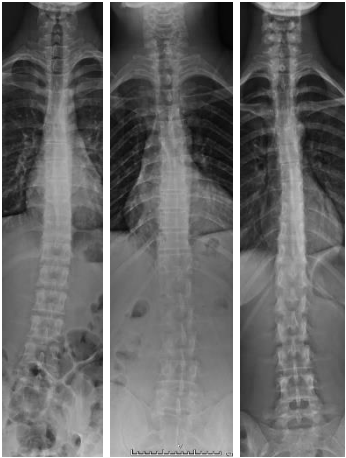
Data ini berupa citra *x-ray* tulang belakang yang diambil dari Rumah Sakit Universitas King Abdullah, Universitas Sains dan Teknologi Yordania, Irbid, Yordania. Total dari kumpulan data ini berjumlah 338 yang dibagi kedalam 3 kelas yaitu normal, spondylolisthesis, dan skoliosis.

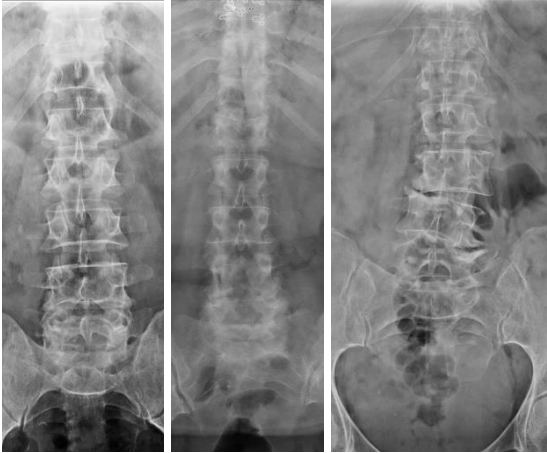
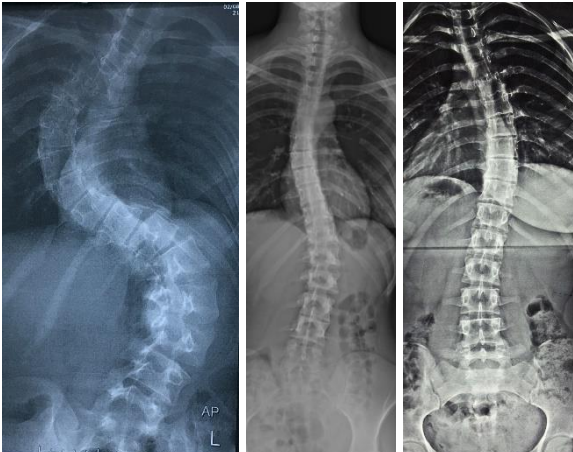
Tabel 3. Jumlah kumpulan data

No	Kelas Citra <i>X-ray</i> Tulang Belakang	Jumlah Citra
1	Normal	71
2	Spondylolisthesis	79
3	Skoliosis	188

Berdasarkan Tabel 3 di atas dapat dilihat bahwa kelas citra *x-ray* tulang belakang untuk normal memiliki data sebanyak 71 citra yang memiliki format .jpg, kelas citra *x-ray* tulang belakang untuk spondylolisthesis memiliki data sebanyak 79 citra dengan format .jpg, dan kelas citra *x-ray* tulang belakang untuk skoliosis yang memiliki citra sebanyak 188 citra dengan format .jpg. Setiap citra merepresentasikan kelas citra, contoh citra yang diambil dari kumpulan data tersebut dapat dilihat pada Tabel 4, sebagai berikut :

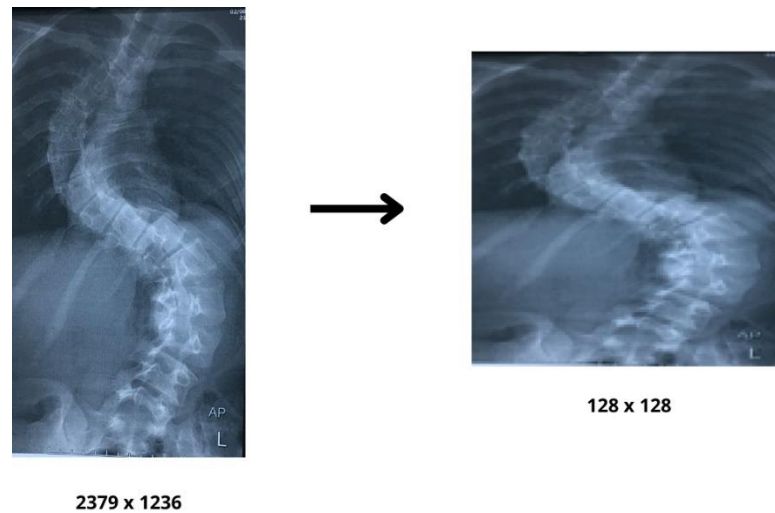
Tabel 4. Contoh kumpulan data citra *x-ray* tulang belakang

No	Citra	Kelas
1		Normal
2		Spondylolisthesis

		
3		Skoliosis

### 3.3.1.2 Preprocessing Data

Pada tahap ini, dilakukan proses perubahan ukuran citra *resize* yaitu mengubah ukuran citra yang bervariasi menjadi seragam dengan ukuran 128x128. Proses ini dilakukan karena pada data citra dari kumpulan data *x-ray* tulang belakang memiliki orientasi, sudut pandang, dan ukuran yang berbeda-beda. Tahapan *preprocessing* pada kumpulan data yang ada dengan tujuan untuk menyamakan ukuran citra agar proses pelatihan model menjadi lebih cepat dan efisien. Adapun ilustrasi pada proses *resizing Image* seperti Gambar 5 berikut :



Gambar 5. Ilustrasi *Resize Image*

### 3.3.1.3 *Splitting Data*

Pada tahap ini, pengelompokan data berdasarkan jumlah keseluruhan kumpulan data yaitu 338 data dikelompokkan menjadi 3 kelompok dengan komposisi data *training*, *validation*, dan *testing*. Pengelompokan ini harus dilakukan tanpa tumpang tindih antar kelompok untuk memastikan keakuratan proses pelatihan model. Data *training* yang terlalu kecil membuat model tidak dapat belajar dengan baik. Oleh karena itu, ukuran data *training* jauh lebih besar dari data *validation* dan data *testing*.

Adapun fungsi dari data *training* digunakan untuk melatih model, sementara data *validation* untuk mengevaluasi proses pelatihan model yang akan menghasilkan *loss function*, untuk pertimbangan data tersebut *overfitting* atau *underfitting*. Data *testing* digunakan untuk menguji model sebagai simulasi penggunaan model pada dunia nyata dan tidak boleh digunakan dalam proses pelatihan sebelumnya.

Pada penelitian ini, kumpulan data citra *x-ray* tulang belakang dibagi menjadi data *training*, *validation*, dan *testing* dengan komposisi di setiap percobaan tersaji pada Tabel 5, sebagai berikut :

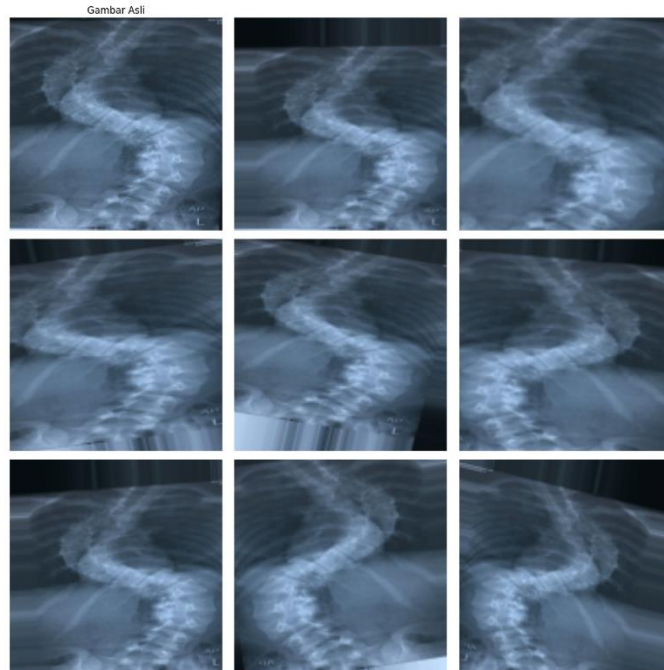
Tabel 5. Pembagian komposisi kumpulan data di setiap percobaan

No	Nama	<i>Training</i>	<i>Validation</i>	<i>Test</i>
1	Percobaan A	60%	20%	20%
2	Percobaan B	80%	10%	10%
3	Percobaan C	90%	5%	5%
4	Percobaan D	90%	5%	5%
5	Percobaan E	80%	10%	10%
6	Percobaan F	60%	20%	20%

Pada Tabel 5 menunjukkan pembagian komposisi kumpulan data untuk setiap percobaan yang dilakukan dalam penelitian ini. Setiap percobaan menggunakan persentase yang berbeda untuk data *training*, *validation*, dan *testing*. Penentuan komposisi data ini juga yang bertujuan mengevaluasi pengaruh proporsi data terhadap pelatihan dan kinerja model yang telah dibuat, semakin banyak dan bervariasi data yang dilatih maka kemungkinan model mendapatkan akurasi terbaik akan semakin meningkat.

### 3.3.1.4 Data Augmentasi

Jumlah data yang digunakan selama pelatihan sangat mempengaruhi kinerja model pembelajaran mesin. Semakin banyak data pelatihan yang tersedia, semakin tinggi akurasi model yang dapat dicapai. Sebaliknya, jumlah data yang terbatas dapat menyebabkan masalah seperti *overfitting*. Salah satu cara untuk mengatasi masalah ini adalah dengan augmentasi data, yaitu manipulasi data melalui teknik seperti *rotasi*, *zoom*, *horizontal flip*, *width shift*, *height shift*, dan lainnya. Adapun ilustrasi Augmentasi data pada Gambar 6 sebagai berikut :



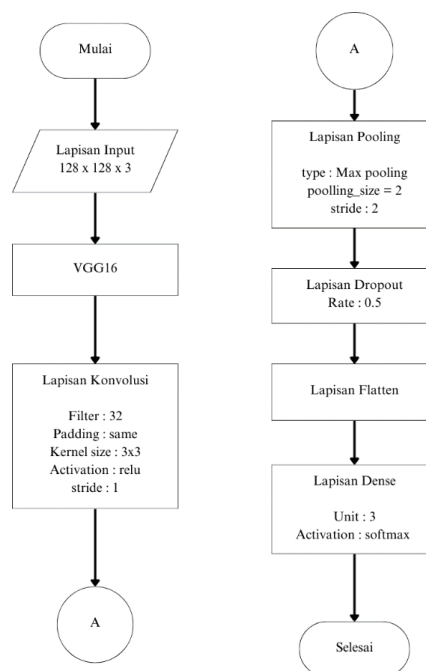
Gambar 6. Ilustrasi Augmentasi data

Penelitian ini menggunakan teknik augmentasi meliputi *Rescale* =  $1/225$ , *Rotation range* = 15, *Zoom range* = 0.2, *Width shift range* = 0.1, *Height shift range* = 0.1, *Shear range* = 0.2, *horizontal flip* = *True*, dan mengisi piksel kosong setelah transformasi dengan menerapkan *Fill Mode* = *nearest*.

### 3.3.2 Modeling

Tahap ini mencakup pemilihan teknik *deep learning* yang akan digunakan, menentukan tools yang sesuai, memilih algoritma *deep learning*, serta menetapkan *hyperparameter* yang optimal untuk mencapai akurasi terbaik. Metode yang diterapkan ialah metode klasifikasi pada *deep learning* yang dilakukan percobaan sebanyak 6 kali, yaitu percobaan A - CNN *From Scratch*, percobaan B - CNN *From Scratch*, percobaan C - CNN *From Scratch*, percobaan D - model *Transfer Learning* VGG16, percobaan E - model *Transfer Learning* VGG16, percobaan F - model *Transfer Learning* VGG16.

Secara umum, arsitektur model CNN terdiri dari beberapa lapisan seperti lapisan konvolusi, lapisan *pooling*, lapisan *dropout*, lapisan *flatten*, dan lapisan *fully connected*. Sedangkan, pada model *Transfer Learning* VGG16 terdiri dari lapisan basis menggunakan VGG16 yang sudah dilatih, kemudian ditambahkan lapisan tambahan di atasnya seperti Conv2D, MaxPooling2D, Dropout, Flatten, dan Dense.. Bagian ini akan memahami cara kerja lapisan tersebut maka diambil sampel percobaan menggunakan model *Transfer Learning* VGG16. Adapun *flowchart* arsitektur dan *hyperparameter* percobaan menggunakan model *Transfer Learning* VGG16 sebagai berikut :



Gambar 7. *Flowchart* Arsitektur Percobaan Model *Transfer Learning* VGG-16

Dari Gambar 7 dapat dilihat *flowchart* percobaan menggunakan model *Transfer Learning* VGG16 untuk memaksimalkan akurasi, maka dilakukan *trial and error* dalam menemukan *hyperparameter* yang optimal. Percobaan ini menggunakan *hyperparameter* yang disajikan pada Tabel 6 sebagai berikut:

Tabel 6. *Hyperparameter* Percobaan Model *Transfer Learning* VGG16

No	<i>Hyperparameter</i>	
1	<i>Split Data</i>	60% <i>Train</i> , 20% <i>Valid</i> , 20% <i>Test</i>
2	<i>Dropout rate</i>	0,5
3	<i>Learning rate</i>	0,001

Pada Tabel 6 menunjukkan *hyperparameter* yang digunakan dalam percobaan F, dengan komposisi data dikelompokkan menjadi 3 yaitu 60% *Training*, 20% *Validation*, 20% *Testing*. Arsitektur terdiri dari 6 lapisan yaitu lapisan basis model VGG16, lapisan konvolusi, lapisan *max pooling*, lapisan *dropout*, lapisan *flatten*, dan lapisan *fully connected*. Pada lapisan *dropout*, nilai *dropout rate* sebesar 0.5 digunakan untuk mengurangi *overfitting* dengan secara acak menonaktifkan setengah dari neuron selama pelatihan. Penggunaan *learning rate* 0,001 dalam penelitian berarti bahwa setiap kali algoritma optimasi melakukan pembaruan pada bobot model selama pelatihan, dimana semakin kecil suatu *learning rate* maka model akan melakukan pembaruan yang lebih kecil dan terukur pada setiap iterasi sehingga model mempelajari data *training* semakin detail.

Arsitektur model pada percobaan menggunakan model *Transfer Learning* VGG16 terdiri dari 6 lapisan. *Feature map* yang berhasil di-extract dari input berukuran 128x128 dengan *channel* 3 (*Red, Green, Blue*). Ringkasan model arsitektur percobaan ini dapat dilihat pada Gambar 8 sebagai berikut :

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
vgg16 (Functional)          (None, 4, 4, 512)          14714688
conv2d (Conv2D)             (None, 4, 4, 32)           147488
max_pooling2d (MaxPooling2D) (None, 2, 2, 32)           0
)
dropout (Dropout)           (None, 2, 2, 32)           0
flatten (Flatten)           (None, 128)                 0
dense (Dense)                (None, 3)                   387
-----
Total params: 14,862,563
Trainable params: 147,875
Non-trainable params: 14,714,688
-----

```

Gambar 8. *Model Summary* Arsitektur Model *Transfer Learning* VGG-16

Adapun tahapan-tahapan arsitektur pada Gambar 8 akan dijelaskan secara rinci sebagai berikut.



### 3.3.2.1 Lapisan Input

Lapisan pertama sebagai lapisan input menerima citra mentah dan diteruskan ke lapisan selanjutnya. Data citra yang telah di preprocessing menghasilkan *feature map* yang diekstraksi dari input memiliki ukuran 224x224 piksel dengan 3 *channel* (*Red, Green, Blue*).

### 3.3.2.2 Model *Transfer Learning* VGG16

Pada percobaan ini, VGG16 sebagai basis model yang terdiri dari 13 lapisan konvolusi dan 3 lapisan yang terhubung penuh, total 16 lapisan berat. Model ini sangat populer karena memiliki kelebihan yaitu VGG16 dapat menangkap fitur-fitur kompleks dari gambar secara efektif. dan dapat digunakan sebagai titik awal dan disesuaikan untuk tugas spesifik dengan data yang lebih sedikit dan waktu pelatihan yang lebih singkat.

Model VGG16 yang telah dilatih sebelumnya melalui *Transfer Learning* akan digunakan untuk kumpulan data *x-ray* tulang belakang dengan input berupa citra ukuran 128 dan input *channel* sebanyak 3. Output pada tahapan konvolusi pertama akan dijadikan input untuk tahapan konvolusi kedua dan seterusnya.

Ukuran citra input adalah 224 x 224 x 3 dan output dari tahapan ini adalah 4 x 4 x 512 dengan total parameter 14, 714, 688. Sebagaimana yang tertera pada Gambar 9 berikut ini :

block4_conv1 (Conv2D)	(None, 16, 16, 512)	1180160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

=====  
Total params: 14,714,688  
Trainable params: 14,714,688  
Non-trainable params: 0

---

Gambar 9. Output Tahapan *Transfer Learning* VGG-16

### 3.3.2.3 Lapisan Konvolusi

Setelah dilakukan *Transfer Learning* VGG16 maka masuk ke tahap selanjutnya yaitu konvolusi. Tahapan ini akan memanfaatkan input yang didapatkan sebelumnya yaitu output gambar dari VGG16 yaitu  $4 \times 4 \times 512$ . Konvolusi pada tahapan ini menggunakan filter 32, *kernel*  $3 \times 3$ , *padding same* (0), *activation* ReLU, *strides* 1.

Dalam memudahkan dalam penjelasan dan penulisan proses perhitungan pada lapisan konvolusi. Berikut proses konvolusi dengan memberikan nilai filter pada matriks :

Red					
0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

Green					
0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

Blue					
0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

Gambar 10. Nilai Pixel RGB Kumpulan Data Label Skoliosis

Gambar 10 menunjukkan proses konvolusi yaitu menggunakan matriks hasil *preprocessing* untuk mendapatkan *feature maps*. Proses ini menggunakan jenis *same padding*, dimana ditambahkan nilai 0 di sekeliling matriks atau disebut dengan *zero padding*. Hasil *feature maps* operasi konvolusi menghasilkan nilai piksel citra dengan 3 *channel* yaitu *red*, *green*, dan *blue* diambil masing-masing dan ditambahkan *padding* 0 (*same*) di setiap pikselnya.

Pada percobaan ini, proses konvolusi dengan memberikan nilai filter pada matriks digunakan *kernel* 3 x 3 dengan nilai seperti pada Gambar 11 sebagai berikut:

1	0	-1
1	0	-1
1	0	-1

Gambar 11. *Kernel* 3x3

Langkah selanjutnya dilakukan proses perhitungan di setiap *channel* dengan dikalikan dengan *kernel* ukuran 3x3 pada Gambar 12. Tahapan ini dilakukan berulang dengan pergeseran *kernel* sebanyak 1 *strides* di setiap *channel* sehingga didapatkan perhitungan di setiap *channel* dengan nilai sebagai berikut :

### Channel Red

#### Posisi 1 di Channel red

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned}
 & ((1*0)+(0*0)+((-1)*0)+ \\
 & (1*0)+(0*76)+((-1)*41)+ \\
 & (1*0)+(0*30)+((-1)*103)) = -144
 \end{aligned}$$

#### Posisi 2 di Channel red

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned}
 & ((1*0)+(0*0)+((-1)*0)+ \\
 & (1*76)+(0*41)+((-1)*78)+ \\
 & (1*30)+(0*103)+((-1)*152)) = -124
 \end{aligned}$$

#### Posisi 3 di Channel red

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned}
 & ((1*0)+(0*0)+((-1)*0)+ \\
 & (1*41)+(0*78)+((-1)*38)+ \\
 & (1*103)+(0*152)+((-1)*70)) = 36
 \end{aligned}$$

Gambar 12. Ilustrasi Perhitungan pada Lapisan Konvolusi Channel Red

Pada Gambar 12 menjelaskan ilustrasi perhitungan tahapan konvolusi. Sebagai contoh, pada posisi 1 *channel red* yang diberi tanda kotak merah akan dikalikan dengan nilai *kernel* 3x3 pada gambar 12, perkalian antara Matriks 3x3 yang dikalikan sesuai dengan baris dan kolomnya menghasilkan nilai (-144). Selanjutnya, pada posisi 2 *channel red*, kotak merah akan bergeser sebanyak 1 kolom sesuai dengan stridenya yaitu 1, sehingga menghasilkan nilai (-124). Tahapan ini akan terus berulang hingga posisi ke 16. Ilustrasi proses konvolusi secara detail untuk mendapatkan hasil proses konvolusi sebagai berikut :

Posisi 4 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*78)+(0*38)+((-1)*0)+ \\ & (1*152)+(0*70)+((-1)*0)) = 230 \end{aligned}$$

Posisi 5 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*76)+((-1)*41)+ \\ & (1*0)+(0*30)+((-1)*103)+ \\ & (1*0)+(0*7)+((-1)*120)) = -264 \end{aligned}$$

Posisi 6 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*76)+(0*41)+((-1)*78)+ \\ & (1*30)+(0*103)+((-1)*152)+ \\ & (1*7)+(0*120)+((-1)*176)) = -293 \end{aligned}$$

Posisi 7 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*41)+(0*78)+((-1)*38)+ \\ & (1*103)+(0*152)+((-1)*70)+ \\ & (1*120)+(0*176)+((-1)*95)) = 61 \end{aligned}$$

Posisi 8 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*78)+(0*38)+((-1)*0)+ \\ & (1*152)+(0*70)+((-1)*0)+ \\ & (1*176)+(0*95)+((-1)*0)) = 406 \end{aligned}$$

Gambar 13. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Red*

Posisi 9 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*30)+((-1)*103)+ \\ & (1*0)+(0*7)+((-1)*120)+ \\ & (1*0)+(0*35)+((-1)*167)) = -390 \end{aligned}$$

Posisi 10 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*30)+(0*103)+((-1)*152)+ \\ & (1*7)+(0*120)+((-1)*176)+ \\ & (1*35)+(0*167)+((-1)*186)) = -442 \end{aligned}$$

Posisi 11 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*103)+(0*152)+((-1)*70)+ \\ & (1*120)+(0*176)+((-1)*95)+ \\ & (1*167)+(0*186)+((-1)*167)) = 58 \end{aligned}$$

Posisi 12 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*152)+(0*70)+((-1)*0)+ \\ & (1*176)+(0*95)+((-1)*0)+ \\ & (1*186)+(0*167)+((-1)*0)) = 514 \end{aligned}$$

Posisi 13 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*7)+((-1)*120)+ \\ & (1*0)+(0*35)+((-1)*167)+ \\ & (1*0)+(0*0)+((-1)*0)) = -287 \end{aligned}$$

Gambar 14. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Red*

Posisi 14 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*7)+(0*120)+((-1)*176)+ \\ & (1*35)+(0*167)+((-1)*186)+ \\ & (1*0)+(0*0)+((-1)*0)) = -320 \end{aligned}$$

Posisi 15 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*120)+(0*176)+((-1)*95)+ \\ & (1*167)+(0*186)+((-1)*167)+ \\ & (1*0)+(0*0)+((-1)*0)) = 25 \end{aligned}$$

Posisi 16 di *Channel red*

0	0	0	0	0	0
0	76	41	78	38	0
0	30	103	152	70	0
0	7	120	176	95	0
0	35	167	186	167	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*176)+(0*95)+((-1)*0)+ \\ & (1*186)+(0*167)+((-1)*0)+ \\ & (1*0)+(0*0)+((-1)*0)) = 362 \end{aligned}$$

Gambar 15. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Red*

Setelah perhitungan di *channel red* maka didapatkan hasil seperti pada Gambar 16 sebagai berikut:

*Red*

-144	-124	36	230
-264	-293	61	406
-390	-442	58	514
-287	-320	25	362

Gambar 16. Hasil Tahapan Perhitungan Lapisan Konvolusi *Channel Red*

Selanjutnya dilakukan perhitungan di *channel green* yang dikalikan dengan *kernel* ukuran 3x3 pada Gambar 11 dengan tahapan perhitungan dan penjelasannya seperti pada *channel red* sebelumnya. Ilustrasi perhitungan *channel green* sebagai berikut:

### *Channel Green*

#### Posisi 1 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*0)+(0*67)+((-1)*38)+ \\ & (1*0)+(0*28)+((-1)*103)) = -141 \end{aligned}$$

#### Posisi 2 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*67)+(0*38)+((-1)*70)+ \\ & (1*28)+(0*103)+((-1)*152)) = -127 \end{aligned}$$

#### Posisi 3 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*38)+(0*70)+((-1)*33)+ \\ & (1*103)+(0*152)+((-1)*69)) = 39 \end{aligned}$$

Gambar 17. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Green*



Posisi 4 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*70)+(0*33)+((-1)*0)+ \\ & (1*152)+(0*69)+((-1)*0)) = 222 \end{aligned}$$

Posisi 5 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*67)+((-1)*38)+ \\ & (1*0)+(0*28)+((-1)*103)+ \\ & (1*0)+(0*7)+((-1)*123)) = -264 \end{aligned}$$

Posisi 6 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*67)+(0*38)+((-1)*70)+ \\ & (1*28)+(0*103)+((-1)*152)+ \\ & (1*7)+(0*123)+((-1)*182)) = -302 \end{aligned}$$

Posisi 7 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*38)+(0*70)+((-1)*33)+ \\ & (1*103)+(0*152)+((-1)*69)+ \\ & (1*123)+(0*182)+((-1)*96)) = 66 \end{aligned}$$

Posisi 8 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*70)+(0*33)+((-1)*0)+ \\ & (1*152)+(0*69)+((-1)*0)+ \\ & (1*182)+(0*96)+((-1)*0)) = 404 \end{aligned}$$

Gambar 18. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Green*

Posisi 9 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*28)+((-1)*103)+ \\ & (1*0)+(0*7)+((-1)*123)+ \\ & (1*0)+(0*32)+((-1)*157)) = -383 \end{aligned}$$

Posisi 10 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*28)+(0*103)+((-1)*152)+ \\ & (1*7)+(0*123)+((-1)*182)+ \\ & (1*32)+(0*157)+((-1)*178)) = -445 \end{aligned}$$

Posisi 11 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*103)+(0*152)+((-1)*69)+ \\ & (1*123)+(0*182)+((-1)*96)+ \\ & (1*157)+(0*178)+((-1)*159)) = 59 \end{aligned}$$

Posisi 12 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*152)+(0*69)+((-1)*0)+ \\ & (1*182)+(0*96)+((-1)*0)+ \\ & (1*178)+(0*159)+((-1)*0)) = 512 \end{aligned}$$

Posisi 13 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*7)+((-1)*123)+ \\ & (1*0)+(0*32)+((-1)*157)+ \\ & (1*0)+(0*0)+((-1)*0)) = -280 \end{aligned}$$

Gambar 19. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Green*

Posisi 14 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*7)+(0*123)+((-1)*182)+ \\ & (1*32)+(0*157)+((-1)*178)+ \\ & (1*0)+(0*0)+((-1)*0)) = -321 \end{aligned}$$

Posisi 15 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*123)+(0*187)+((-1)*96)+ \\ & (1*157)+(0*178)+((-1)*159)+ \\ & (1*0)+(0*0)+((-1)*0)) = 25 \end{aligned}$$

Posisi 16 di *Channel green*

0	0	0	0	0	0
0	67	38	70	33	0
0	28	103	152	69	0
0	7	123	182	96	0
0	32	157	178	159	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*182)+(0*96)+((-1)*0)+ \\ & (1*178)+(0*159)+((-1)*0)+ \\ & (1*0)+(0*0)+((-1)*0)) = 360 \end{aligned}$$

Gambar 20. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Green*

Setelah perhitungan di *channel green* maka didapatkan hasil sebagaimana pada Gambar sebagai berikut:

*Green*

-141	-127	39	222
-264	-302	66	404
-383	-445	59	512
-280	-321	25	360

Gambar 21. Hasil Tahapan Perhitungan Lapisan Konvolusi *Channel Green*

Selanjutnya dilakukan perhitungan di *channel blue* yang dikalikan dengan *kernel* ukuran 3x3 pada Gambar 11 dengan tahapan perhitungan sama sama penjelasannya seperti pada *channel red* sebelumnya, sehingga dilakukan perhitungan *channel blue* sebagai berikut :

### *Channel Blue*

#### Posisi 1 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*0)+(0*57)+((-1)*32)+ \\ & (1*0)+(0*24)+((-1)*85)) = -117 \end{aligned}$$

#### Posisi 2 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*57)+(0*32)+((-1)*58)+ \\ & (1*24)+(0*85)+((-1)*126)) = -103 \end{aligned}$$

#### Posisi 3 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*32)+(0*58)+((-1)*28)+ \\ & (1*85)+(0*126)+((-1)*58)) = 31 \end{aligned}$$

Gambar 22. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Blue*

Posisi 4 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*0)+((-1)*0)+ \\ & (1*58)+(0*28)+((-1)*0)+ \\ & (1*126)+(0*58)+((-1)*0)) = 184 \end{aligned}$$

Posisi 5 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*57)+((-1)*32)+ \\ & (1*0)+(0*24)+((-1)*85)+ \\ & (1*0)+(0*7)+((-1)*101)) = -218 \end{aligned}$$

Posisi 6 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*57)+(0*32)+((-1)*58)+ \\ & (1*24)+(0*85)+((-1)*126)+ \\ & (1*7)+(0*101)+((-1)*151)) = -247 \end{aligned}$$

Posisi 7 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*32)+(0*58)+((-1)*28)+ \\ & (1*85)+(0*126)+((-1)*58)+ \\ & (1*101)+(0*151)+((-1)*81)) = 51 \end{aligned}$$

Posisi 8 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*58)+(0*28)+((-1)*0)+ \\ & (1*126)+(0*58)+((-1)*0)+ \\ & (1*151)+(0*81)+((-1)*0)) = 335 \end{aligned}$$

Gambar 23. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Blue*

Posisi 9 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*24)+((-1)*85)+ \\ & (1*0)+(0*7)+((-1)*101)+ \\ & (1*0)+(0*28)+((-1)*134)) = -320 \end{aligned}$$

Posisi 10 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*24)+(0*85)+((-1)*126)+ \\ & (1*7)+(0*101)+((-1)*151)+ \\ & (1*28)+(0*134)+((-1)*151)) = -369 \end{aligned}$$

Posisi 11 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*85)+(0*126)+((-1)*58)+ \\ & (1*101)+(0*151)+((-1)*81)+ \\ & (1*134)+(0*151)+((-1)*136)) = 45 \end{aligned}$$

Posisi 12 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*126)+(0*58)+((-1)*0)+ \\ & (1*151)+(0*81)+((-1)*0)+ \\ & (1*151)+(0*136)+((-1)*0)) = 428 \end{aligned}$$

Posisi 13 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*0)+(0*7)+((-1)*101)+ \\ & (1*0)+(0*28)+((-1)*134)+ \\ & (1*0)+(0*0)+((-1)*0)) = -235 \end{aligned}$$

Gambar 24. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Blue*

Posisi 14 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*7)+(0*101)+((-1)*151)+ \\ & (1*28)+(0*134)+((-1)*151)+ \\ & (1*0)+(0*0)+((-1)*0)) = -267 \end{aligned}$$

Posisi 15 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*101)+(0*151)+((-1)*81)+ \\ & (1*134)+(0*151)+((-1)*136)+ \\ & (1*0)+(0*0)+((-1)*0)) = 18 \end{aligned}$$

Posisi 16 di *Channel blue*

0	0	0	0	0	0
0	57	32	58	28	0
0	24	85	126	58	0
0	7	101	151	81	0
0	28	134	151	136	0
0	0	0	0	0	0

$$\begin{aligned} & ((1*151)+(0*81)+((-1)*0)+ \\ & (1*151)+(0*136)+((-1)*0)+ \\ & (1*0)+(0*0)+((-1)*0)) = 302 \end{aligned}$$

Gambar 25. Ilustrasi Perhitungan pada Lapisan Konvolusi *Channel Blue*

Setelah perhitungan di *channel blue* maka didapatkan hasil seperti Gambar 26 sebagai berikut:

*Blue*

-117	-103	31	184
-218	-247	51	335
-320	-369	45	428
-235	-267	18	302

Gambar 26. Hasil Tahapan Perhitungan Lapisan Konvolusi *Channel Blue*

Setelah melakukan tahapan konvolusi di setiap lapisan maka didapatkan nilai *red*, *green* dan *blue*. Hasil perhitungan tahapan konvolusi dapat dilihat pada Gambar 27 sebagai berikut :

Red	Green	Blue									
-144	-124	36	230	-141	-127	39	222	-117	-103	31	184
-264	-293	61	406	-264	-302	66	404	-218	-247	51	335
-390	-442	58	514	-383	-445	59	512	-320	-369	45	428
-287	-320	25	362	-280	-321	25	360	-235	-267	18	302

Gambar 27. Hasil Perhitungan Lapisan Konvolusi *Channel Red*, *Green*, dan *Blue*. Setiap nilai dari *channel* yang sudah melalui tahapan perhitungan pada Gambar 27 akan dijumlahkan sehingga menghasilkan output total. Hasil total penjumlahan dapat dilihat pada Gambar 28 sebagai berikut :

$$\text{Total} = \text{Red} + \text{Green} + \text{Blue}$$

-402	-354	106	636
-746	-842	178	1145
-1093	-1256	162	1454
-802	-908	68	1024

Gambar 28. Total Perhitungan Konvolusi

Pada Gambar 28, total perhitungan konvolusi dengan activation ReLU maka setiap nilai yang negative akan diubah menjadi angka 0, dan mempertahankan yang nilainya lebih besar dari 0. Contohnya pada elemen matriks baris 1 kolom 1 (1,1) memiliki nilai -402 setelah dilakukan operasi ReLU akan menjadi 0. Sehingga didapatkan hasil convolutional seperti pada Gambar 29 sebagai berikut :



Total = Red + Green + Blue

-402	-354	106	636
-746	-842	178	1145
-1093	-1256	162	1454
-802	-908	68	1024

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

Hasil ReLU

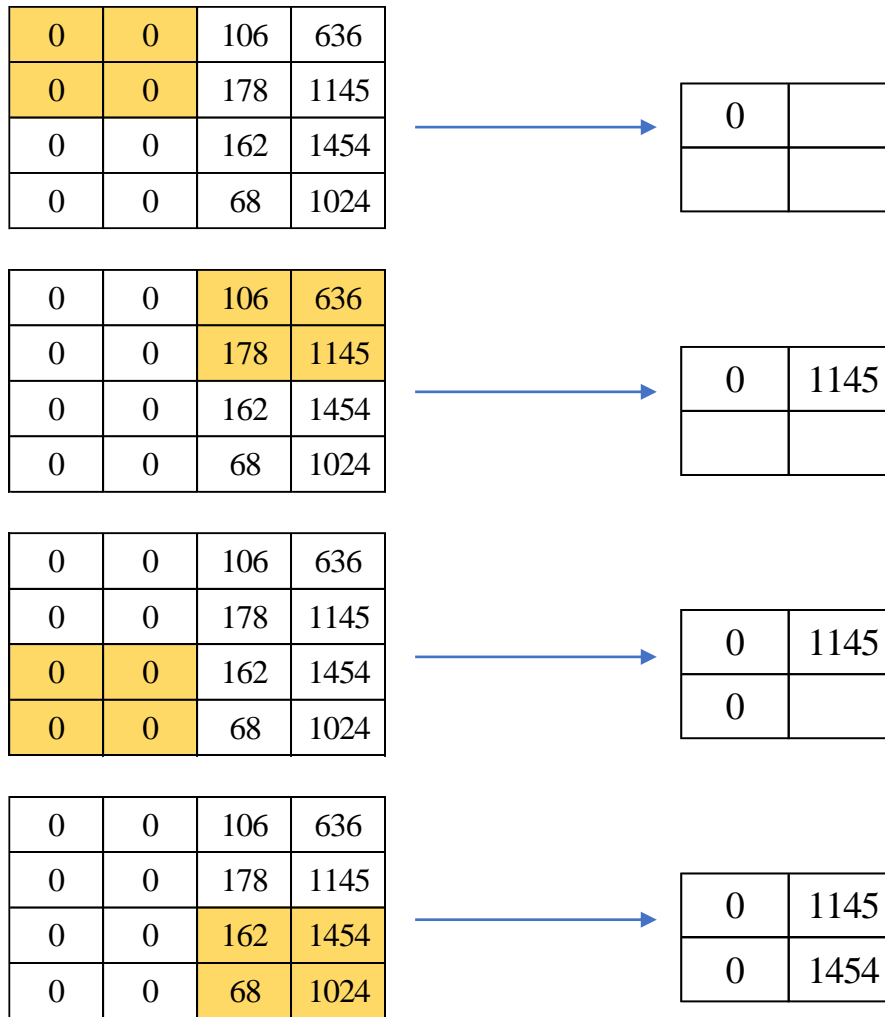
0	0	106	636
0	0	178	1145
0	0	162	1454
0	0	68	1024

Gambar 29. Hasil Konvolusi #1

Gambar 29 merupakan output dari salah satu *feature maps*. Proses ini dilakukan secara berulang sampai semua citra dengan menggunakan 32 jenis filter yang berbeda sehingga output dari tahapan konvolusi ini akan menghasilkan banyak *feature maps*.

### 3.3.2.4 Lapisan *Pooling*

Dari hasil pada Gambar 29 output tahapan konvolusi akan dijadikan sebagai input pada lapisan *pooling*, pada tahapan ini menggunakan *max pooling* dengan *kernel size* 2x2 dengan *stride* 2 dengan output dari tahapan konvolusi yaitu 4x4x32. Ilustrasi proses *max pooling* ditunjukkan seperti Gambar 30 sebagai berikut :



Gambar 30. Ilustrasi Perhitungan Lapisan *Max Pooling*

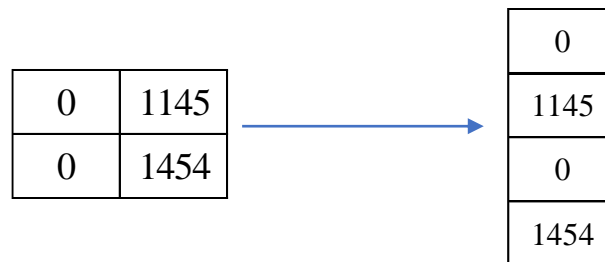
Gambar 30 adalah tahapan *max pooling* dimana tahapan ini dilakukan dengan ukuran *kernel size* 2x2, dari nilai *kernel* ini akan diambil nilai tertinggi sehingga didapatkan nilai pertama yaitu 0. Langkah selanjutnya dilakukan pergeseran *kernel* sebanyak 2 kolom sesuai *strides* 2. Sehingga didapatkan nilai *kernel* 2x2 pada Matriks selanjutnya dengan nilai [106, 636, 178, 1145] setelah dilakukan *max pooling* didapatkan nilai tertinggi yaitu 1145, dan begitu seterusnya untuk proses *max pooling*. Sehingga didapatkan output pada lapisan *max pooling* yaitu 2x2x32 dengan nilai [0, 1145, 0, 1454]. Output ini akan dijadikan inputan pada lapisan *dropout*.

### 3.3.2.5 Lapisan *Dropout*

Proses *dropout* adalah teknik yang digunakan untuk menghapus neuron-neuron pada lapisan tersembunyi dalam jaringan saraf. Neuron-neuron yang dihapus dipilih secara acak. *Dropout* berfungsi untuk mengurangi kompleksitas model yang telah dibangun dengan membuang *neural network* yang tidak terpakai. Ukuran model akan tetap  $2 \times 2 \times 32$ , tidak ada perubahan.

### 3.3.2.6 Lapisan *Flatten*

*Flatten* merupakan tahapan mengubah matriks  $2 \times 2$  menjadi vektor kolom. Dari output lapisan *dropout* sebesar  $2 \times 2 \times 32$  akan didapatkan bentuk vektor baru. Ilustrasi proses *flatten* dapat dilihat pada Gambar 31 sebagai berikut :

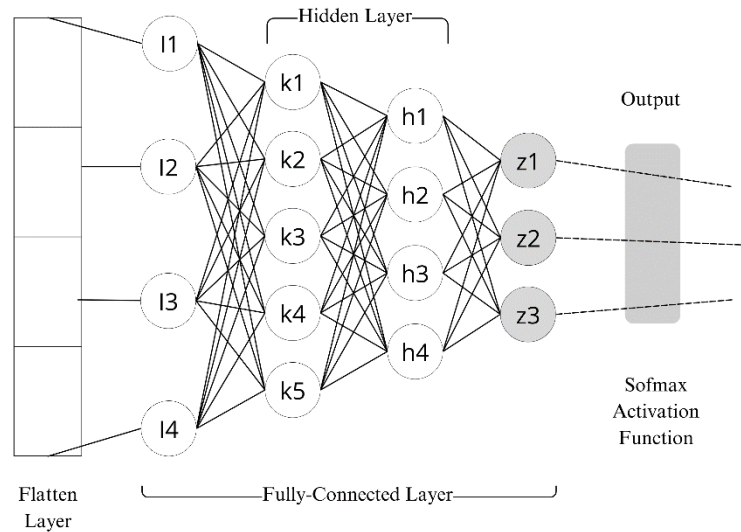


Gambar 31. Ilustrasi Proses Lapisan *Flatten*

Gambar 31 adalah proses *flatten* dimana Matriks  $2 \times 2$  diubah menjadi Matriks  $4 \times 1$  atau vektor kolom. Output pada tahapan ini adalah vektor kolom yang akan digunakan sebagai input di lapisan *fully connected (dense)*.

### 3.3.2.7 Lapisan *Fully Connected*

Output dari tahapan lapisan *flatten* yaitu vektor kolom, yang akan menjadi inputan pada tahapan ini. Ilustrasi lapisan *fully connected* seperti pada Gambar 32 sebagai berikut :



Gambar 32. Ilustrasi Prose *Dense + Softmax*

Tahapan akhir dari percobaan ini adalah *dense* dengan fungsi aktivasi *softmax*. Berikut perhitungan dari setiap lapisan tersembunyi :

$$k_j = \sum_{i=1}^c W_{i,j} l_i + b_j \quad (3.1)$$

$$h_j = \sum_{i=1}^c W_{i,j} k_i + b_j \quad (3.2)$$

$$z_j = \sum_{i=1}^c W_{i,j} h_i + b_j \quad (3.3)$$

Nilai output pada proses *flatten* pada Gambar 31 yang berbentuk vektor kolom kemudian digunakan sebagai vektor input dalam lapisan *fully connected* dengan perhitungan pada setiap lapisan tersembunyi sebagai berikut :

*Lapisan Tersembunyi Pertama*

- Vektor input :  $l_i = \begin{bmatrix} 0 \\ 1145 \\ 0 \\ 1454 \end{bmatrix}$
- Matriks bobot dengan ukuran  $5 \times 4$  :  $W_{5,4} = \begin{bmatrix} 0,02 & 0,04 & 0,06 & 0,03 \\ 0,01 & 0,03 & 0,05 & 0,02 \\ 0,03 & 0,07 & 0,02 & 0,05 \\ 0,04 & 0,01 & 0,03 & 0,07 \\ 0,05 & 0,02 & 0,04 & 0,06 \end{bmatrix}$
- Vektor bias :  $b_j = \begin{bmatrix} 0,01 \\ 0,02 \\ 0,03 \\ 0,04 \\ 0,05 \end{bmatrix}$

Menghitung output lapisan tersembunyi pertama

$$k_j = \sum_{i=1}^c W_{ij} l_i + b_j \quad (3.1)$$

$$k_j = \begin{bmatrix} 0,02 & 0,04 & 0,06 & 0,03 \\ 0,01 & 0,03 & 0,05 & 0,02 \\ 0,03 & 0,07 & 0,02 & 0,05 \\ 0,04 & 0,01 & 0,03 & 0,07 \\ 0,05 & 0,02 & 0,04 & 0,06 \end{bmatrix} \begin{bmatrix} 0 \\ 1145 \\ 0 \\ 1454 \end{bmatrix} + \begin{bmatrix} 0,01 \\ 0,02 \\ 0,03 \\ 0,04 \\ 0,05 \end{bmatrix}$$

$$k_1 = ((0,02 * 0) + (0,04 * 1145) + (0,06 * 0) + (0,03 * 1454)) + 0,01 = 89,43$$

$$k_2 = ((0,01 * 0) + (0,03 * 1145) + (0,05 * 0) + (0,02 * 1454)) + 0,02 = 63,45$$

$$k_3 = ((0,03 * 0) + (0,07 * 1145) + (0,02 * 0) + (0,05 * 1454)) + 0,03 = 152,88$$

$$k_4 = ((0,04 * 0) + (0,01 * 1145) + (0,03 * 0) + (0,07 * 1454)) + 0,04 = 113,27$$

$$k_5 = ((0,05 * 0) + (0,02 * 1145) + (0,04 * 0) + (0,06 * 1454)) + 0,05 = 110,19$$

Lapisan tersembunyi (*Hidden layer*) pertama di ilustrasikan dengan  $k_1, k_2, k_3, k_4, k_5$ .

Perhitungan lapisan tersembunyi pertama dengan menganggap beberapa nilai acak untuk matriks bobot dan vektor bias, setiap neuron dari  $l_1, l_2, l_3, l_4$  dikalikan dengan nilai bobot yang berbeda-beda. Kemudian, ditambah dengan bias menghasilkan nilai  $k_1 = 89,43 ; k_2 = 63,45 ; k_3 = 152,88 ; k_4 = 113,27 ; k_5 = 110,19$ .

Lapisan Tersembunyi kedua

- Vektor input :  $k_i = \begin{bmatrix} 89,43 \\ 63,45 \\ 152,88 \\ 113,27 \\ 110,19 \end{bmatrix}$

- Matriks bobot dengan ukuran 4x5 :

$$W_{4,5} = \begin{bmatrix} 0,03 & 0,07 & 0,02 & 0,04 & 0,01 \\ 0,05 & 0,06 & 0,03 & 0,02 & 0,08 \\ 0,01 & 0,03 & 0,05 & 0,07 & 0,04 \\ 0,02 & 0,05 & 0,07 & 0,06 & 0,03 \end{bmatrix}$$

- Vektor bias :  $b_j = \begin{bmatrix} 0,02 \\ 0,03 \\ 0,04 \\ 0,05 \end{bmatrix}$

Menghitung output lapisan tersembunyi kedua

$$h_j = \sum_{i=1}^c W_{i,j} k_i + b_j \quad (3.2)$$

$$h_j = \begin{bmatrix} 0,03 & 0,07 & 0,02 & 0,04 & 0,01 \\ 0,05 & 0,06 & 0,03 & 0,02 & 0,08 \\ 0,01 & 0,03 & 0,05 & 0,07 & 0,04 \\ 0,02 & 0,05 & 0,07 & 0,06 & 0,03 \end{bmatrix} \begin{bmatrix} 89,43 \\ 63,45 \\ 152,88 \\ 113,27 \\ 110,19 \end{bmatrix} + \begin{bmatrix} 0,02 \\ 0,03 \\ 0,04 \\ 0,05 \end{bmatrix}$$

$$h_1 = \left( (0,03 * 89,43) + (0,07 * 63,45) + (0,02 * 152,88) \right) + 0,02 = 15,83$$

$$+ (0,04 * 113,27) + (0,01 * 110,19)$$

$$h_2 = \left( (0,05 * 89,43) + (0,06 * 63,45) + (0,03 * 152,88) \right) + 0,03 = 23,97$$

$$+ (0,02 * 113,27) + (0,08 * 110,19)$$

$$h_3 = \left( (0,01 * 89,43) + (0,03 * 63,45) + (0,05 * 152,88) \right) + 0,04 = 22,82$$

$$+ (0,07 * 113,27) + (0,04 * 110,19)$$

$$h_4 = \left( (0,02 * 89,43) + (0,05 * 63,45) + (0,07 * 152,88) \right) + 0,05 = 25,81$$

$$+ (0,06 * 113,27) + (0,03 * 110,19)$$

Pada lapisan selanjutnya, perhitungan lapisan tersembunyi kedua setiap neuron dari  $k_1, k_2, k_3, k_4, k_5$  ini akan dikalikan lagi dengan nilai bobot yang berbeda-beda kemudian ditambah dengan bias untuk menghasilkan nilai  $h_1, h_2, h_3, h_4$  yang digunakan sebagai lapisan tersembunyi selanjutnya.

Setiap neuron dari  $h_1, h_2, h_3, h_4$  ini akan dikalikan lagi dengan bobot yang berbeda-beda kemudian ditambah dengan bias untuk menghasilkan nilai  $z_1, z_2, z_3$  sebagai lapisan output.

#### Lapisan Output

- Vektor input :  $h_i = \begin{bmatrix} 15,83 \\ 23,97 \\ 22,82 \\ 25,81 \end{bmatrix}$

- Matriks bobot dengan ukuran 4x5 :

$$W_{4,5} = \begin{bmatrix} 0,04 & 0,02 & 0,06 & 0,03 \\ 0,05 & 0,04 & 0,01 & 0,07 \\ 0,03 & 0,05 & 0,02 & 0,06 \end{bmatrix}$$

- Vektor bias :  $b_j = \begin{bmatrix} 0,03 \\ 0,02 \\ 0,04 \end{bmatrix}$

Menghitung output lapisan tersembunyi selanjutnya

$$z_j = \sum_{i=1}^c W_{i,j} h_i + b_j \quad (3.3)$$

$$z_j = \begin{bmatrix} 0,04 & 0,02 & 0,06 & 0,03 \\ 0,05 & 0,04 & 0,01 & 0,07 \\ 0,03 & 0,05 & 0,02 & 0,06 \end{bmatrix} \begin{bmatrix} 15,83 \\ 23,97 \\ 22,82 \\ 25,81 \end{bmatrix} + \begin{bmatrix} 0,03 \\ 0,02 \\ 0,04 \end{bmatrix}$$

$$z_1 = \left( (0,04 * 15,83) + (0,02 * 23,97) + (0,06 * 22,82) \right) + 0,03 = 3,29$$

$$z_2 = \left( (0,05 * 15,83) + (0,04 * 23,97) + (0,01 * 22,82) \right) + 0,02 = 3,81$$

$$z_3 = \left( (0,03 * 15,83) + (0,05 * 23,97) + (0,02 * 22,82) \right) + 0,04 = 3,72$$

Langkah selanjutnya adalah perhitungan *softmax* dengan rumus eksponensial  $z_1$  dibagi dengan total eksponensial  $z_1$ , eksponensial  $z_2$ , dan eksponensial  $z_3$ . Tahapan ini dilakukan 3 kali baik pada  $z_1, z_2, z_3$ . Perhitungan aktivasi *softmax* dengan rumus pada Persamaan (3.4) sebagai berikut:

*Aktivasi Softmax*

$$s(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (3.4)$$

$$s(z_1) = \frac{e^{3,29}}{e^{3,29} + e^{3,81} + e^{3,72}} = \frac{26,75}{112,95} \approx 0,24$$

$$s(z_2) = \frac{e^{3,81}}{e^{3,29} + e^{3,81} + e^{3,72}} = \frac{44,97}{112,95} \approx 0,40$$

$$s(z_3) = \frac{e^{3,72}}{e^{3,29} + e^{3,81} + e^{3,72}} = \frac{41,23}{112,95} \approx 0,36$$

Jadi, probabilitas output setelah aktivasi *Softmax* untuk memprediksi kelas dari model adalah sebagai berikut :

$$\text{Probabilitas kelas 1} = 0,24$$

$$\text{Probabilitas kelas 2} = 0,40$$

$$\text{Probabilitas kelas 3} = 0,36$$

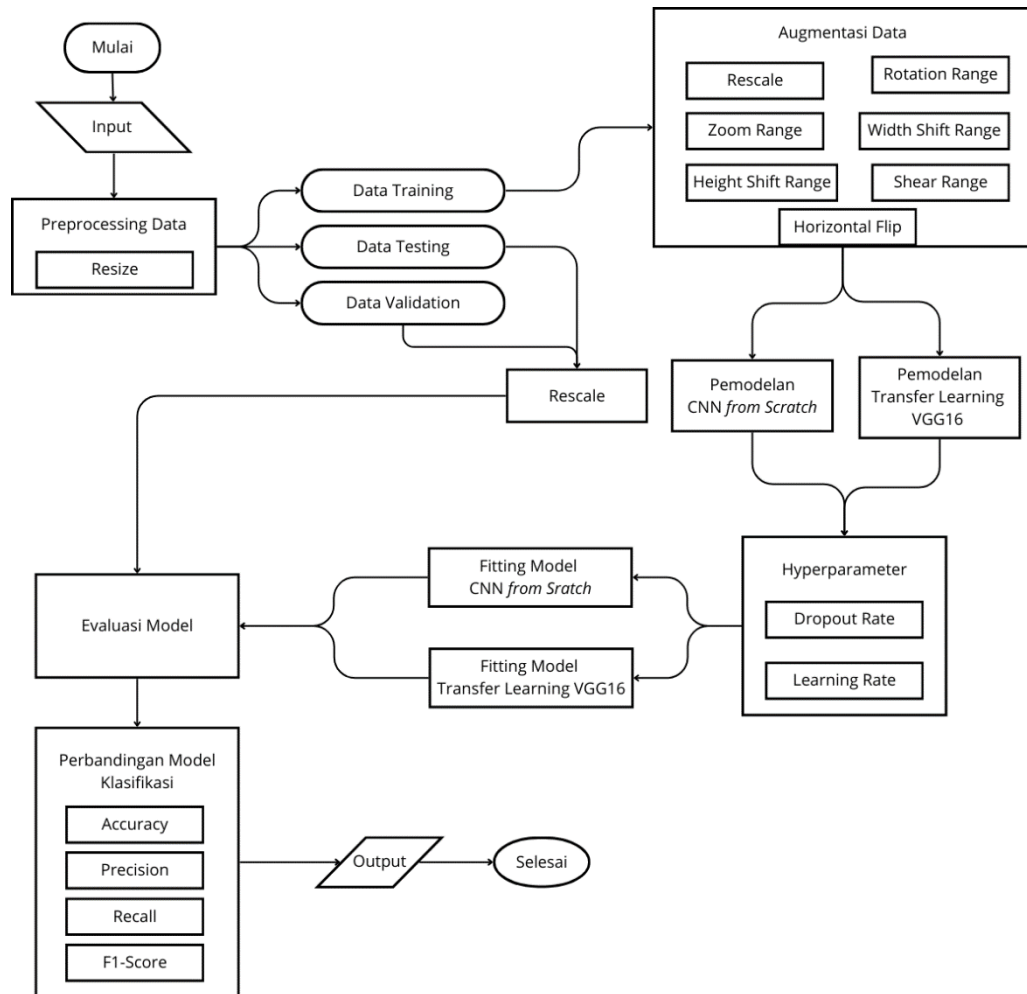
Dari hasil perhitungan *softmax* diatas bahwa kelas 2 memiliki probabilitas tertinggi (0,40), yang berarti input citra yang dimasukkan pada model diprediksi adalah skoliosis.



Berikut ini langkah-langkah yang dilakukan dalam penelitian ini adalah :

1. Kumpulan data yang digunakan pada penelitian ini adalah kumpulan data sekunder dari internet, yang diperoleh dari *website Kaggle*. Kumpulan data tersebut merupakan data gambar pasien penderita penyakit tulang belakang yang mencakup 338 gambar dikelompokkan menjadi 3 kelas, yaitu normal, skoliosis, dan spondylolisthesis.
2. Dengan menggunakan bantuan Jupyter notebook kumpulan data diinput sesuai dengan struktur direktori yang telah diatur. Setiap kelas dalam kumpulan data ini direpresentasikan oleh subdirektori yang mengandung gambar-gambar yang termasuk dalam kelas tersebut. Hasil input data menunjukkan jumlah gambar tiap 3 kelas, yaitu 71 gambar berlabel NormalFinal, 188 gambar berlabel ScolFinal, dan 79 gambar berlabel SpondFinal.
3. *Data Preprocessing*. Dengan menyeragamkan data dilakukan proses *Resize* bertujuan untuk memperkecil ukuran gambar menjadi 224x224.
4. Data dikelompokkan menjadi 3 kelompok yaitu data *training*, *validation*, dan *testing*. Penelitian ini menggunakan 3 komposisi pembagian data yang berbeda, rincian proporsi yaitu komposisi 1 (60% data *training*, 20% data *validation*, dan 20 data *testing*), komposisi 2 (80% data *training*, 10% data *validation*, dan 10 data *testing*), dan komposisi 3(90% data *training*, 5% data *validation*, dan 5% data *testing*).
5. *Data Augmentasi*. Data latih diperbanyak sehingga menghindari *overfitting* dengan melakukan manipulasi data. Penelitian ini menggunakan teknik augmentasi meliputi *Rescale* = 1/225, *Rotation range* = 15, *Zoom range* = 0.2, *Width shift range* = 0.1, *Height shift range* = 0.1, *Shear range* = 0.2, *horizontal flip* = True, dan mengisi piksel kosong setelah transformasi dengan menerapkan *Fill Mode* = nearest.
6. Membangun model klasifikasi pada algoritma CNN dan *Transfer Learning* VGG16. Pada model CNN hanya terdiri lapisan dasar seperti Conv2D, MaxPooling2D, Dropout, Flatten, dan Dense. Sedangkan, pada model *Transfer Learning* VGG16 terdiri dari lapisan basis menggunakan VGG16 yang sudah dilatih, kemudian ditambahkan lapisan tambahan di atasnya seperti Conv2D, MaxPooling2D, Dropout, Flatten, dan Dense.

7. Dalam penelitian ini, *hyperparameter* dipilih dan diuji untuk menemukan konfigurasi yang paling optimal dalam klasifikasi gambar *x-ray* tulang belakang adalah *dropout rate* dan *learning rate*. Menentukan *hyperparameter* yang paling sesuai sehingga dilakukan sebanyak 6 percobaan model arsitektur yang berbeda yaitu 3 arsitektur CNN *From Scratch* dan 3 arsitektur *Transfer Learning* VGG16.
8. Pelatihan model pada arsitektur CNN dilakukan untuk menghindari *overfitting* dan memastikan model terbaik disimpan berdasarkan kinerja validasi menggunakan *callback* seperti *EarlyStopping* dan *ModelCheckpoint*. Sementara itu, model VGG16 dilatih tanpa *callback* untuk mengamati kinerja langsung dari proses pelatihan.
9. Mengevaluasi model hasil klasifikasi dengan *Confusion Matrix*. Acuan untuk penentuan baik atau tidaknya kinerja suatu model klasifikasi dapat dilihat dari parameter pengukuran kinerjanya seperti *Accuracy*, *Recall*, *Precision*, dan *F1-Score*.
10. Hasil penelitian dari 6 percobaan yang menunjukkan kinerja model terbaik dapat dijadikan sebuah acuan penerapan model klasifikasi gambar X-ray tulang belakang.



Gambar 33. *Flowchart* Metode Penelitian

## V. KESIMPULAN

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan diperoleh kesimpulan sebagai berikut:

1. Menerapkan metode *Transfer Learning* VGG16 pada klasifikasi citra *x-ray* tulang belakang terdiri dari beberapa tahap, mulai dari *preprocessing* data, pembagian komposisi data, *data augmentation*, penambahan lapisan pada arsitektur seperti lapisan konvolusi, lapisan *max pooling*, lapisan *dropout*, lapisan *flatten*, dan lapisan *fully connected*. Pelatihan model *Transfer Learning* melakukan pelatihan sebanyak 50 epoch.
2. Model *Transfer Learning* VGG16 diimplementasikan pada percobaan D, E, dan F. Pada penelitian ini diketahui bahwa Percobaan F memiliki kinerja terbaik dengan nilai akurasi (*Accuracy*) sebesar 97,06%, Presisi (*Precision*) sebesar 0,97, Sensitivitas (*Recall*) sebesar 0,95, dan skor-F1 (*F1-Score*) sebesar 0,96. Kinerja unggul ini terjadi karena perubahan komposisi data, penambahan lapisan *dropout*, dan penyesuaian *learning rate* yang berhasil meningkatkan kinerja model.

## DAFTAR PUSTAKA

- Albani, M., dan Andhi, R. R. 2023. Klasifikasi Uang Rupiah Kertas Tidak Layak Edar Menggunakan CNN Xception *Transfer Learning* Berbasis Website. *Jurnal Inovtek Polbeng - Seri Informatika*. **8**(2): 394–406.
- Ambarwari, A., Husni, M., dan Mahayana, D. 2023. Perkembangan Metode Klasifikasi Citra Penginderaan Jauh dalam Perspektif Revolusi Ilmiah Thomas Kuhn. *Jurnal Filsafat Indonesia*. **6**(3): 465–473.
- Batubara, N. A., dan Awangga, R. M. 2020. *Tutorial Object Detection Plate Number With Convolution Neural Network (CNN)*. Kreatif, Bandung.
- Berliani, T., Rahardja, E., dan Septiana, L. 2023. Perbandingan Kemampuan Klasifikasi Citra X-ray Paru-paru menggunakan *Transfer Learning* ResNet-50 dan VGG-16. *Journal of Medicine and Health*. **5**(2): 123–135.
- Cahya, F. N., Hardi, N., Riana, D., dan Hadianti, S. 2021. Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN). *Jurnal Sistem Informasi*. **10**(3): 618–626.
- Darmanto, H. 2019. Pengenalan Spesies Ikan Berdasarkan Kontur Otolith Menggunakan Convolutional Neural Network. *Joined Journal*. **2**(1): 41–59.
- Efrian, M. R., dan Latifa, U. 2022. Image Recognition Berbasis Convolutional Neural Network (CNN) untuk Mendeteksi Penyakit Kulit pada Manusia. *Jurnal POLEKTRO: Jurnal Power Elektronik*. **11**(1): 276–282.
- Ersyad, M. Z., Ramadhani, K. N., dan Arifianto, A. 2020. Pengenalan Bentuk Tangan dengan Convolutional Neural Network (CNN). *e-Proceeding of Engineering*. **7**(2): 8212–8222.
- Febrianti, A. S., Sardjono, T. A., dan Babgei, A. F. 2020. Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vektor Machine. *Jurnal Teknik ITS*. **9**(1): 118–123.
- Fraiwan, M., Audat, Z., Fraiwan, L., dan Manasreh, T. 2022. Using deep *Transfer Learning* to detect scoliosis and spondylolisthesis from x-ray images. *PLoS ONE*. **17**(5).

- Harani, N. H., Prianto, C., dan Hasanah, M. 2019. Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python. *Jurnal Teknik Informatika*. **11**(3): 48–53.
- Hasan, Moh. A., Riyanto, Y., dan Riana, D. 2021. Grape leaf image disease classification using CNN-VGG16 model. *Jurnal Teknologi dan Sistem Komputer*. **9**(4): 218–223.
- Hermawati, F. A., dan Zai, R. A. 2021. Konferensi Nasional Ilmu Komputer (KONIK) 2021 Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-Jones dan Convolutional Neural Networks (CNN). *Konferensi Nasional Ilmu Komputer (KONIK)*. **5**: 182–187.
- Indrawan, G. T., Nilogiri, A., dan Faruq, H. A. Al. 2022. Diagnosis COVID-19 Berdasarkan Citra X-ray Paru-Paru Menggunakan Metode Convolutional Neural Network Diagnose Of COVID-19 Based On X-ray Image Of The Lungs Using Convolutional Neural Network. *Jurnal Smart Teknologi*. **3**(4): 340–349.
- Irfansyah, D., Mustikasari, M., dan Suroso, A. 2021. Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi. *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*. **6**(2): 87–92.
- Kamil, R., Anton, dan Widiastuti. 2021. Perancangan Aplikasi Bahasa Isyarat “Isyaratku” Dengan Deep Learning Serta Google Cloud Platform. *Jurnal Sistem Informasi dan Informatika*. **1**(2): 91–97.
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., dan Doulamis, N. 2021. Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*. **9**(4) : 1-22.
- Meliuwati, P., dan Kurniati, E. 2022. Ekstraksi Data Digital Menggunakan Teknik *Max Pooling* dan *Average Pooling*. *Jurnal Riset Matematika*. **2**(2): 137–144.
- Miftahuddin, Y., dan Adani, F. 2022. Sistem Klasifikasi Jenis Kupu-Kupu Menggunakan Visual Geometry Group 16. *e-Proceeding FTI*.
- Muhammad, S., dan Wibowo, A. T. 2021. Klasifikasi Tanaman Aglaonema Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network (CNN). *e-Proceeding of Engineering*. **8**(5): 10621–10636.
- Nurani, D., Yanuar, I. L., dan Putra, A. D. (2022). Klasifikasi Jenis Penyakit Pada Citra Daun Padi Menggunakan Algoritma Convolution Neural Network. *Jurnal Teknologi Informasi dan Komputer*. **8**(3): 199–213.

- Pangestu, M. A., dan Bunyamin, H. 2018. Analisis Kinerja dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika dan Sistem Informasi*. **4**(2): 337–344.
- Paraijun, F., Aziza, R. N., dan Kuswardani, D. 2022. Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah. *KILAT*. **11**(1): 1–9.
- Putro, E. C., Awangga, R. M., dan Andarsyah, R. 2020. *Tutorial Object Detection People with Faster Region-based Convolutional Neural Network (Faster R-CNN)*. Kreatif Industri Nusantara, Bandung.
- Ramadhan, M., Mulyana, D. I., dan Yel, M. B. 2022. Optimasi Algoritma Cnn Menggunakan Metode *Transfer Learning* untuk Klasifikasi Citra X-Ray Paru-Paru Pneumonia dan Non-Pneumonia. *Jurnal Teknik Informatika Kaputama (JTIK)*. **6**(2): 670–679.
- Rismiyati, dan Luthfiarta, A. 2021. VGG16 *Transfer Learning* Architecture for Salak Fruit Quality Classification. *Jurnal Informatika dan Teknologi Informasi*. **18**(1): 37–48.
- Roihan, A., Sunarya, P. A., dan Rafika, A. S. 2020. Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*. **5**(1): 75–82.
- Rozaqi, A. J., Sunyoto, A., dan Arief, M. R. 2021. Implementasi *Transfer Learning* pada Algoritma Convolutional Neural Network Untuk Identifikasi Penyakit Daun Kentang. *Procedia of Engineering and Life Science*. **1**(1).
- Rozi, M. I. F., Adiwijaya, N. O., dan Swasono, D. I. 2023. Identifikasi Kinerja Arsitektur *Transfer Learning* VGG16, ResNet-50, dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat Identification of VGG16, ResNet-50, and Inception-V3 Transfer Architecture Kinerja in Image Classification of Tomato Leaf Diseases. *Jurnal Riset Rekayasa Elektro*. **5**(2): 145–154.
- Sanjaya, U. P., Alawi, Z., Zayn, A. R., dan Dirgantara, G. 2023. Optimasi Convolutional Neural Network dengan Standard Deviasi untuk Klasifikasi Pneumonia pada Citra X-rays Paru. *Generation Journal*. **7**(3): 40–47.
- Shukla, N., & Fricklas, K. 2018. *Machine learning with TensorFlow*. Manning Publications
- Sidik, A. D., dan Ansawarman, A. 2022. Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning. *Formosa Journal of Multidisciplinary Research (FJMR)*. **1**(3): 559–568.
- Solehatin, N., Putri, A. A., Diana, P., dan Setiowati, D. 2022. Sosialisasi Nyeri Pinggang dan Perubahan Struktur Tulang Belakang Akibat Terlalu Lama

- Duduk dengan Posisi yang Salah pada Mahasiswa UIN Jakarta. *Jurnal Abdi Masyarakat Indonesia*. **2**(3): 1075–1080.
- Tangkelayuk, A., dan Mailoa, E. 2022. Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*. **9**(2): 1109–1119.
- Wahid, M. I., Lawi, A., Muh, D. A., dan Siddik, A. 2023. Perbandingan Kinerja Model Ensembled *Transfer Learning* Pada Klasifikasi Penyakit Daun Tomat. *Prosiding Seminar Nasional Teknik Elektro dan Informatika (SNTEI)*. **8**(1): 286–291.
- Wahyuni, S., dan Sulaeman, M. 2022. Penerapan Algoritma Deep Learning untuk Sistem Absensi Kehadiran Deteksi Wajah Di Pt Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*. **7**(1): 12–21.
- Wijaya, A. E., Swastika, W., dan Kelana, O. H. 2021. Implementasi *Transfer Learning* Pada Convolutional Neural Network Untuk Diagnosis Covid-19 dan Pneumonia Pada Citra X-Ray. *SAINSBERTEK Jurnal Ilmiah Sains dan Teknologi*. **2**(1): 10–15.
- Yadav, S. S., dan Jadhav, S. M. 2019. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*. **6**(1): 1–18.
- Yanto, B., Fimawahib, L., Supriyanto, A., Hayadi, B. H., dan Pratama, R. R. (2021). Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network. *Jurnal Inovtek Polbeng - Seri Informatika*. **6**(2): 260–268.
- Yenusi, Y. N., Suryasatriya Trihandaru, dan Setiawan, A. 2023. Comparison of Convolutional Neural Network (CNN) Models in Face Classification of Papuan and Other Ethnicities. *JST (Jurnal Sains dan Teknologi)*. **12**(1): 261–268.
- Zuhan, M., dan Kristian, Y. 2023. Detection of Porang Plant Diseases and Pests (Amorphophallus Muelleri) Based on Leaf Imagery Utilizing DCNN *Transfer Learning*. *JTECS : Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*. **3**(2): 129–140.