



KERJA PRAKTIK – EF234603

Implementasi Model Klasifikasi Menggunakan Metode Transfer Learning untuk Pemilihan *Slice* Citra 3D CBCT

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember

Kampus ITS Sukolilo Surabaya 60111

Periode: 1 Mei 2024 - 31 Juli 2024

Oleh:

Katarina Inezita Prambudi 5025211148

Pembimbing Departemen

Dini Adni Navastara, S.Kom, M.Sc.

Pembimbing Lapangan

Prof. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024



KERJA PRAKTIK – EF234603

Implementasi Model Klasifikasi Menggunakan Metode Transfer Learning untuk Pemilihan *Slice* Citra 3D CBCT

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya 60111
Periode: 1 Mei 2024 - 31 Juli 2024

Oleh:

Katarina Inezita Prambudi 5025211148

Pembimbing Departemen

Dini Adni Navastara, S.Kom, M.Sc.

Pembimbing Lapangan

Prof. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|-------------------------------------------|------|
| DAFTAR ISI..... | ii |
| DAFTAR GAMBAR | viii |
| DAFTAR TABEL | xi |
| DAFTAR PSEUDOCODE..... | xiv |
| LEMBAR PENGESAHAN..... | xvi |
| KATA PENGANTAR..... | xx |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Tujuan..... | 2 |
| 1.3. Manfaat..... | 2 |
| 1.4. Rumusan Masalah | 3 |
| 1.5. Lokasi dan Waktu Kerja Praktik | 3 |
| 1.6. Metodologi Kerja Praktik | 3 |
| 1.6.1. Perumusan Masalah..... | 3 |
| 1.6.2. Studi Literatur..... | 4 |
| 1.6.3. Analisis dan Perancangan..... | 4 |
| 1.6.4. Implementasi Sistem | 4 |
| 1.6.5. Hasil dan Pembahasan..... | 4 |
| 1.6.6. Kesimpulan dan Saran..... | 4 |
| 1.7. Sistematika Laporan | 4 |

| | | |
|-------------------------------|------------------------------------------|----|
| 1.7.1. | Bab I Pendahuluan..... | 4 |
| 1.7.2. | Bab II Profil Instansi | 5 |
| 1.7.3. | Bab III Tinjauan Pustaka | 5 |
| 1.7.4. | Bab IV Analisis dan Perancangan | 5 |
| 1.7.5. | Bab V Implementasi Sistem | 5 |
| 1.7.6. | Bab VI Hasil dan Pembahasan | 5 |
| 1.7.7. | Bab VII Kesimpulan dan Saran..... | 5 |
| BAB II PROFIL INSTANSI..... | | 7 |
| 2.1. | Sejarah Instansi..... | 7 |
| 2.2. | Visi Instansi..... | 8 |
| 2.3. | Misi Instansi | 8 |
| 2.4. | Struktur Instansi | 8 |
| 2.5. | Laboratorium..... | 9 |
| 2.6. | Lokasi | 12 |
| BAB III TINJAUAN PUSTAKA..... | | 14 |
| 3.1. | Penelitian Terkait | 14 |
| 3.2. | Implan Gigi..... | 18 |
| 3.3. | CBCT..... | 21 |
| 3.4. | <i>Image Classification</i> | 22 |
| 3.5. | <i>Histogram Equalization (HE)</i> | 23 |
| 3.6. | Augmentasi..... | 25 |
| 3.7. | CNN | 26 |
| 3.7.1. | Convolutional Layer..... | 26 |

| | | |
|---------------------------------------------|-------------------------------------------|-----------|
| 3.7.2. | Active Function ReLU | 27 |
| 3.7.3. | Pooling Layer | 28 |
| 3.7.4. | Fully Connected Layer | 29 |
| 3.7.5. | Softmax Classifier | 29 |
| 3.8. | Transfer Learning | 29 |
| 3.9. | VGG-16 | 30 |
| 3.10. | ResNet | 31 |
| 3.11. | MobileNet..... | 32 |
| 3.12. | K Fold Cross Validation..... | 33 |
| 3.13. | Hyperparameters..... | 34 |
| 3.13.1. | Hyperparameters pada Proses Training..... | 34 |
| 3.13.2. | Hyperparameters Regularisasi..... | 34 |
| 3.13.3. | Loss Function | 35 |
| 3.14. | Metrik Evaluasi | 35 |
| BAB IV ANALISIS DAN PERANCANGAN..... | | 38 |
| 4.1. | Desain Sistem | 38 |
| 4.2. | Desain Model | 40 |
| 4.3. | Dataset | 41 |
| 4.4. | Pre-Processing Data | 42 |
| 4.5.1. | Histogram Equalization | 42 |
| 4.5.2. | LabelEncoder..... | 43 |
| 4.5.3. | Augmentasi Data | 44 |
| 4.5. | Proses Training dan Testing | 46 |

| | | |
|--------|-------------------------------------------------------------------------------|----|
| 4.6.1. | KFold Cross Validation | 46 |
| 4.6.2. | Inisialisasi Hyperparameter | 46 |
| 4.6.3. | Skenario Pengujian | 47 |
| 4.6. | Evaluasi Model | 48 |
| BAB V | IMPLEMENTASI | 50 |
| 5.1. | Pre-Processing Data | 50 |
| 5.1.1. | Histogram Equalization | 50 |
| 5.1.2. | Encoding | 51 |
| 5.1.3. | Augmentasi Data | 51 |
| 5.2. | Proses Training dan testing | 52 |
| 5.2.1. | KFold Cross Validation | 52 |
| 5.2.2. | Inisialisasi Hyperparameter | 53 |
| 5.2.3. | Kompilasi Model | 54 |
| BAB VI | HASIL DAN PEMBAHASAN | 57 |
| 6.1. | Hasil Evaluasi | 57 |
| 6.1.1. | <i>Histogram Equalization dan Layer</i> <i>Trainable False</i> | 57 |
| 6.1.2. | <i>Tanpa Histogram Equalization dan Layer</i> <i>Trainable False</i> | 60 |
| 6.1.3. | <i>Histogram Equalization dan Layer</i> <i>Trainable True</i> | 64 |
| 6.1.4. | <i>Tanpa Histogram Equalization dan Layer</i> <i>Trainable True</i> | 67 |
| 6.2. | Pembahasan | 70 |

| | |
|------------------------------------|----|
| BAB VII KESIMPULAN DAN SARAN | 82 |
| 7.1. Kesimpulan..... | 82 |
| 7.2. Saran..... | 83 |
| DAFTAR PUSTAKA..... | 85 |
| BIODATA PENULIS..... | 92 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 2.1 Struktur Instansi Departemen Teknik Informatika, ITS..... | 9 |
| Gambar 3.1 Tempat Pemasangan Implan Gigi: (A) Transdental Implan, (B) Transmucosal Implan, (C) Subperiosteal Implan, (D) Endosseous Implant..... | 18 |
| Gambar 3.2 Bentuk Implan Gigi: (A) Blade Form, (B) Root Form | 19 |
| Gambar 3.3 Bagian Implan Gigi | 20 |
| Gambar 3.4 Penerapan Histogram Equalization..... | 23 |
| Gambar 3.5 Hasil Augmentasi Data : (A) Gambar Asli, (B) Rotasi, (C) Width Shift, (D) Height Shift, (E) Shear, (F) Zoom, (G) Horizontal flip..... | 25 |
| Gambar 3.6 Arsitektur CNN | 26 |
| Gambar 3.7 Max Pooling dan Average Pooling..... | 28 |
| Gambar 3.8 Arsitektur VGG-16..... | 30 |
| Gambar 3.9 Blok Residual Network | 31 |
| Gambar 3.10 MobileNet..... | 32 |
| Gambar 3.11 Confusion Matrix..... | 35 |
| Gambar 4.1 Diagram Alir Sistem..... | 38 |
| Gambar 4.2 Desain Model..... | 41 |
| Gambar 4.3 (A) Area Representatif dan (B) Area tidak Representatif..... | 41 |
| Gambar 4.4 Penerapan Histogram Equalization..... | 43 |
| Gambar 4.5 Hasil Augmentasi Data : (A) Gambar Asli, (B) Rotasi, (C) Width Shift, (D) Height Shift, (E) Shear, (F) Zoom, (G) Horizontal Flip | 45 |
| Gambar 6.1 Rata-Rata Confusion Matrix VGG-16 pada Skenario Histogram Equalization dan Layer Trainable False..... | 77 |

Gambar 6.2 Hasil Citra 3D CBCT Berdasarkan *Confusion Matrix*
Menggunakan Model VGG-16.....78

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | |
|----------------------------------------------------------------------------------------------------------------|----|
| Tabel 6.1 Hasil Evaluasi VGG-16 pada Skenario Histogram Equalization dan Layer Trainable False | 57 |
| Tabel 6.2 Hasil Evaluasi ResNet50 pada Skenario Histogram Equalization dan Layer Trainable False | 58 |
| Tabel 6.3 Hasil Evaluasi ResNet101 pada Skenario Histogram Equalization dan Layer Trainable False | 58 |
| Tabel 6.4 Hasil Evaluasi ResNet152 pada Skenario Histogram Equalization dan Layer Trainable False | 59 |
| Tabel 6.5 Hasil Evaluasi MobileNet pada Skenario Histogram Equalization dan Layer Trainable False | 60 |
| Tabel 6.6 Hasil Evaluasi VGG-16 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 61 |
| Tabel 6.7 Hasil Evaluasi ResNet50 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 61 |
| Tabel 6.8 Hasil Evaluasi ResNet101 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 62 |
| Tabel 6. 9 Hasil Evaluasi ResNet152 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 62 |
| Tabel 6.10 Hasil Evaluasi MobileNet pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 63 |
| Tabel 6.11 Hasil Evaluasi VGG-16 pada Skenario Histogram Equalization dan Layer Trainable True | 64 |
| Tabel 6.12 Hasil Evaluasi ResNet50 pada Skenario Histogram Equalization dan Layer Trainable True | 64 |
| Tabel 6.13 Hasil Evaluasi ResNet101 pada Skenario Histogram Equalization dan Layer Trainable True | 65 |
| Tabel 6.14 Hasil Evaluasi ResNet152 pada Skenario Histogram Equalization dan Layer Trainable True | 65 |

| | |
|----------------------------------------------------------------------------------------------------------------------------|----|
| Tabel 6.15 Hasil Evaluasi MobileNet pada Skenario Histogram Equalization dan Layer Trainable True | 66 |
| Tabel 6.16 Hasil Evaluasi VGG-16 pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 67 |
| Tabel 6.17 Hasil Evaluasi ResNet50 pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 68 |
| Tabel 6.18 Hasil Evaluasi ResNet101 pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 68 |
| Tabel 6.19 Hasil Evaluasi ResNet152 pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 69 |
| Tabel 6.20 Hasil Evaluasi MobileNet pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 69 |
| Tabel 6.21 Hasil Evaluasi Rata-Rata Semua Model pada Skenario Histogram Equalization dan Layer Trainable False | 70 |
| Tabel 6.22 Hasil Evaluasi Rata-Rata Semua Model pada Skenario Tanpa Histogram Equalization dan Layer Trainable False | 72 |
| Tabel 6.23 Hasil Evaluasi Rata Rata Semua Model pada Skenario Histogram Equalization dan Layer Trainable True | 74 |
| Tabel 6.24 Hasil Evaluasi Rata-Rata Semua Model pada Skenario Tanpa Histogram Equalization dan Layer Trainable True | 76 |

[Halaman ini sengaja dikosongkan]

DAFTAR PSEUDOCODE

| | |
|--------------------------------------------------------|----|
| Pseudocode 5.1 Penggunaan Histogram Equalization | 51 |
| Pseudocode 5.2 Penggunaan LabelEncoder | 51 |
| Pseudocode 5.3 Augmentasi Data | 52 |
| Pseudocode 5.4 K Fold Cross Validation | 53 |
| Pseudocode 5.5 Arsitektur Model | 54 |
| Pseudocode 5.6 Kompilasi Model | 54 |
| Pseudocode 5.7 Evaluasi Model..... | 55 |

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Implementasi Model Klasifikasi Menggunakan Metode
Transfer Learning untuk Pemilihan *Slice* Citra 3D
CBCT**

Oleh:

Katarina Inezita Prambudi

5025211148

Disetujui oleh Pembimbing Kerja Praktik:

1. Dini Adni Navastara, S.Kom,
M.Sc.
NIP. 198510172015042001


(Pembimbing Departemen)

2. Prof. Dr. Eng. Chastine
Faticah, S.Kom., M.Kom.
NIP. 197512202001122002


(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Implementasi Model Klasifikasi Menggunakan Metode Transfer Learning untuk Pemilihan *Slice* Citra 3D CBCT

Nama Mahasiswa : Katarina Inezita Prambudi
NRP : 5025211148
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dini Adni Navastara, S.Kom, M.Sc.
Pembimbing Lapangan : Prof. Dr. Eng. Chastine Fatichah,
S.Kom., M.Kom.

ABSTRAK

Pemasangan implan gigi merupakan salah satu pilihan dalam menggantikan gigi yang hilang. Rencana perawatan yang tepat bisa didapat dari hasil interpretasi yang berasal dari alat CBCT 3D Artificial Intelligence (AI) digunakan dalam dunia kedokteran gigi untuk mendukung diagnostik pencitraan medis dan gigi digunakan pendekatan deep learning dalam mengklasifikasi ukuran implan gigi. Pemodelan klasifikasi ini memanfaatkan metode transfer learning dengan menggunakan algoritma Convolutional Neural Network (CNN).

Model transfer learning yang digunakan untuk klasifikasi terdiri dari VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet. Model VGG-16 memiliki akurasi tertinggi sebesar 98,20% pada skenario menggunakan histogram equalization dan layer trainable false.

Kata Kunci : Implan Gigi, Citra 3D CBCT, Transfer Learning, Histogram Equalization, Layer Trainable

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Implementasi Model Klasifikasi Menggunakan Metode Transfer Learning untuk Pemilihan *Slice* Citra 3D CBCT.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Ibu Dini Adni Navastara, S.Kom, M.Sc. selaku dosen pembimbing kerja praktik.
3. Ibu Prof. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 3 Juli 2024
Katarina Inezita Prambudi

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Salah satu hal yang mempengaruhi kualitas hidup manusia secara signifikan adalah kehilangan gigi. Kehilangan gigi dapat menyebabkan para penderitanya mengalami kesulitan dalam mengunyah makanan dan tentunya kehilangan kepercayaan diri.[1] Pemasangan implan gigi merupakan salah satu pilihan dalam menggantikan gigi yang hilang.[2] Permintaan masyarakat akan kebutuhan pemasangan dan perawatan implan gigi mulai meningkat seiring dengan permasalahan diatas.[3]

Meskipun keberhasilan implan gigi semakin meningkat dari tahun ke tahun, tidak akan menutup kemungkinan adanya kegagalan dan komplikasi pemasangan implan.[2] Kegagalan dalam pemasangan implan gigi tidak hanya menyebabkan kerugian finansial, melainkan juga mempengaruhi kondisi tulang rahang pasien dan juga berdampak langsung pada psikologis pasien. Kegagalan implan gigi dapat diantisipasi dengan melakukan rencana perawatan yang tepat dan akurat. Rencana perawatan yang tepat bisa didapat dari hasil interpretasi yang berasal dari alat *Cone beam computed tomography 3-dimension (CBCT 3D)*. [4]

Alat radiografi yang digunakan saat pemasangan implan gigi adalah CBCT 3D. Dalam bidang kedokteran, alat ini digunakan untuk menghasilkan pencitraan secara tiga dimensi yang mencakup bidang aksial, koronal dan sagital serta dapat mengukur densitas tulang. Keakuratan dari perhitungan kualitas dan kuantitas tulang rahang menjadi salah satu faktor penentu keberhasilan dari pemasangan implan gigi.[4]

Artificial Intelligence (AI) digunakan dalam dunia kedokteran gigi untuk mendukung diagnostik pencitraan medis dan gigi. Dalam hal ini, AI telah terbukti meningkatkan akurasi, efisiensi, dan presisi setara dengan pakar medis secara lebih efektif.[5] Digunakan pendekatan *deep learning* dalam

mengklasifikasi citra 3D CBCT. Pemodelan klasifikasi ini memanfaatkan metode *transfer learning* dengan menggunakan algoritma *Convolutional Neural Network* (CNN).[6]

Model *transfer learning* yang digunakan untuk klasifikasi terdiri dari VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet. VGG-16 merupakan model klasifikasi yang terdiri dari 13 convolutional layers, 2 fully connected layers, dan 1 softmax classifier.[7] ResNet digunakan untuk menangani masalah *vanishing gradient* karena memperkenalkan *shortcut connection* dan arsitektur *residual blocks*. [8] MobileNet adalah model arsitektur CNN yang telah dilatih sebelumnya menggunakan konvolusi yang dapat dipisahkan secara mendalam untuk membangun bobot yang ringan pada jaringan saraf.[9]

Berdasarkan latar belakang di atas kontribusi kerja praktik adalah melakukan implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT. Implementasi model klasifikasi tersebut menggunakan *pretrained* model VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan untuk berkontribusi pada penelitian yang dilakukan oleh dosen dalam implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT.

1.3. Manfaat

Manfaat yang diperoleh adalah terciptanya model klasifikasi yang lebih akurat dan valid untuk digunakan dalam rekomendasi ukuran implan gigi melalui evaluasi dan pengujian model yang dapat meningkatkan kualitas perawatan implan gigi.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara melakukan preprocessing pada implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT ?
2. Bagaimana performa model transfer learning (VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet) pada klasifikasi untuk pemilihan *slice* citra 3D CBCT?
3. Bagaimana pengaruh *histogram equalization* pada performa model transfer learning ?
4. Bagaimana pengaruh *layer trainable* pada performa model *transfer learning* ?

1.5. Lokasi dan Waktu Kerja Praktik

Pengerjaan kerja praktik ini lakukan secara *hybrid*. Adapun kerja praktik dimulai pada tanggal 1 Mei 2024 hingga 31 Juli 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi:

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan dari implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT, saya mengikuti rapat bersama dengan pembimbing lapangan, yaitu Prof. Chastine. Dalam rapat tersebut, Prof. Chastine memaparkan hasil pertemuan beliau dengan Pak Naufal (Mahasiswa S3 yang sedang melakukan disertasi). Pada saat rapat, saya dijelaskan bagaimana konsep dan proses dalam melakukan pemilihan *slice* citra 3D CBCT. Dalam rapat tersebut dibahas juga mengenai model klasifikasi apa saja yang akan digunakan dalam implementasi model klasifikasi tersebut.

1.6.2. Studi Literatur

Setelah mendapat gambaran bagaimana model klasifikasi yang akan digunakan, saya diberitahu model apa saja yang akan diimplementasikan untuk membuat model klasifikasi tersebut. Model yang dipakai meliputi VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet. Selain itu, saya diberikan dataset untuk dilakukan pengujian pada dataset tersebut.

1.6.3. Analisis dan Perancangan

Setelah model diberikan, saya melakukan analisis pada model tersebut untuk mengetahui bagaimana model tersebut bekerja. Setelah selesai melakukan analisis, saya merancang model dengan berbagai parameter.

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini saya melakukan *hyperparameter tuning* pada model.

1.6.5. Hasil dan Pembahasan

Setelah melakukan *hyperparameter tuning*, perlu adanya evaluasi untuk mengetahui apakah model dapat melakukan klasifikasi. Kemudian dilakukan evaluasi menggunakan dataset testing untuk mengetahui performa model klasifikasi.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

Sistematika laporan kerja praktik meliputi:

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Instansi

Bab ini berisi sejarah instansi, visi instansi, misi instansi, struktur instansi, laboratorium, dan lokasi instansi

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan

Bab ini berisi uraian perancangan yang dilakukan untuk proses implementasi model klasifikasi.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi implementasi yang dilakukan untuk proses implementasi model klasifikasi.

1.7.6. Bab VI Hasil dan Pembahasan

Bab ini berisi hasil evaluasi dan pembahasan dari model klasifikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL INSTANSI

2.1. Sejarah Instansi

Saat ini, invasi teknologi informasi sudah terasa di berbagai aspek kehidupan manusia, dan ini akan menjadi tulang punggung pertumbuhan ekonomi negara, baik sekarang maupun di masa depan. Pemerintah sangat menyadari hal ini, sehingga sejak Repelita V yang lalu, mereka telah menetapkan bahwa pendidikan tinggi di bidang komputer dan informatika harus diprioritaskan, bersama dengan kursus di bidang lain seperti rekayasa, perilaku, manajemen, akuntansi, dan kesenian. Pendidikan tinggi dirancang untuk mempersiapkan Indonesia untuk era pembangunan industri dan teknologi informasi. Dengan demikian, pada tahun 1985, Direktorat Jenderal Pendidikan Tinggi pemerintah memutuskan untuk membuka program studi S1 baru di bidang ilmu teknologi komputer di empat universitas atau lembaga pendidikan yang memiliki program ITS. Pada awalnya, program di ITS disebut Program Studi Teknik Komputer. Namun, pada tahun 1993, namanya diubah menjadi Jurusan Teknik Komputer. Pada tahun 1996, Surat Keputusan Direktur Jenderal Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996 dikeluarkan. Menurut Surat Keputusan Nomor 003/BAN-PT/Ak-X/S1/V/2006 dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) yang dikeluarkan pada tanggal 18 Mei 2006, Jurusan Teknik Informatika menerima nilai akreditasi A. Selain program Sarjana (S1), Jurusan Teknik Informatika juga menawarkan program Pasca Sarjana (S2). Program ini dimulai sejak tahun 1994 dan diizinkan oleh surat keputusan Direktur Jenderal Pendidikan Tinggi No. 2851/D/T/2001. Jurusan Teknik Informatika juga memulai program Doktor (S3) pada tahun 2011.[10]

2.2. Visi Instansi

Sejalan dengan visi ITS untuk menjadi perguruan tinggi dengan reputasi internasional dalam ilmu pengetahuan, teknologi, dan seni, terutama dalam membangun industri dan kelautan yang berwawasan lingkungan, Visi Departemen Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa. Visi Departemen Teknik Informatika adalah menjadi lembaga pendidikan dan penelitian di bidang informatika yang unggul di tingkat nasional dan memiliki reputasi internasional.[11]

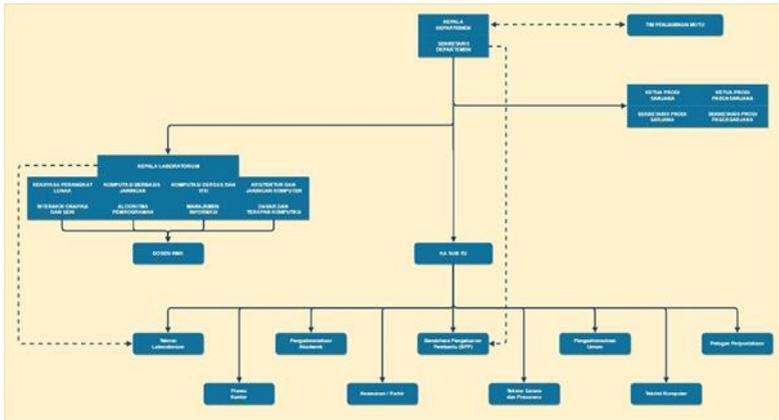
2.3. Misi Instansi

Berikut adalah misi-misi yang dimiliki oleh Departemen Teknik Informatika, ITS [12].

1. Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalin kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

2.4. Struktur Instansi

Gambar 2.1 merupakan gambar struktur instansi yang terdapat di Departemen Teknik Informatika, ITS [12].



Gambar 2.1 Struktur Instansi Departemen Teknik Informatika, ITS [12]

2.5. Laboratorium

Mahasiswa Departemen Teknik Informatika ITS memiliki akses ke fasilitas laboratorium yang terdiri dari laboratorium bidang minat dan laboratorium workshop. Jenis laboratorium bidang minat mencakup berbagai mata kuliah yang dapat dipelajari oleh mahasiswa Departemen Teknik Informatika, seperti berikut: [13]

1. Laboratorium Rekayasa Perangkat Lunak
Keahlian dalam melakukan pengujian perangkat lunak, mengelola proyek perangkat lunak, mengurangi resiko kesalahan perangkat lunak, dan membuat perangkat lunak game adalah bidang minat di laboratorium ini.
2. Laboratorium Komputasi Cerdas dan Visi
Kemampuan lulusan dalam memanipulasi dan menganalisis data citra dalam berbagai bidang aplikasi (AI, biomedika, industri), menerapkan metode sistem cerdas dalam berbagai bidang aplikasi, memodelkan dan mengoptimalkan

sistem nyata adalah bidang keahlian yang ditekankan di laboratorium ini.

3. Laboratorium Grafika, Interaksi, dan Game

Di bidang minat ini, laboratorium menawarkan keahlian yang menekankan kemampuan lulusan dalam mendesain, mengembangkan, mencatat proses pembuatan game yang standar, serta menggunakan game engine untuk membuat model tiga dimensi, pemrograman, dan aplikasi realitas virtual tiga dimensi.

4. Laboratorium Manajemen Cerdas Informasi

Laboratorium di bidang minat ini menawarkan bidang keahlian yang ditekankan pada kemampuan lulusan dalam menganalisis, mensintesis, dan mengevaluasi proses bisnis dan sistem informasi, memasukkan rekayasa pengetahuan ke dalam aplikasi, melakukan investigasi, pengujian, dan evaluasi kematangan dan kepatutan prosedur dan tata kelola teknologi informasi.

5. Laboratorium Komputasi Berbasis Jaringan

Di laboratorium ini, bidang keahlian yang ditekankan adalah kemampuan sarjana, magister, dan doktor dalam membangun infrastruktur jaringan yang aman, sistem grid, aplikasi standar jaringan, dan aplikasi multimedia berbasis jaringan.

6. Laboratorium Teknologi Jaringan dan Keamanan Siber Cerdas

Laboratorium bidang minat ini menawarkan bidang keahlian yang menekankan kemampuan lulusan dalam menerapkan keamanan jaringan dan berbagai arsitektur jaringan yang sesuai dengan standar teknologi terbaru.

7. Laboratorium Algoritma dan Pemrograman

Kemampuan untuk merancang dan menganalisa algoritma berdasarkan kaidah pemrograman yang kuat untuk menyelesaikan masalah secara efektif dan efisien, kemampuan untuk menggunakan model pemrograman yang ada mendasari berbagai bahasa pemrograman yang ada, kemampuan untuk

memilih bahasa pemrograman untuk menghasilkan aplikasi yang sesuai, seperti mengembangkan sistem atau aplikasi berbasis kerangka kerja, dan peran yang sesuai di laboratorium ini adalah bidang keahlian yang ditekankan.

8. Laboratorium Pemodelan dan Komputasi Terapan

Pemodelan dan simulasi, peramalan sains, optimasi, dan komputasional saintifik adalah semua bidang di mana laboratorium ini menerima penelitian dan kerja sama industri.

Laboratorium workshop yang disediakan oleh Departemen Teknik Informatika adalah sebagai berikut :

1. Workshop Pemrograman 1

Laboratorium ini memiliki fasilitas unggulan dalam hal teknologi manajemen komputer dan fasilitas fisik representatif (PC, sistem suara video, jaringan). Selain itu, ruangnya dapat menampung sebanyak 54 orang.

2. Workshop Pemrograman 2

Laboratorium ini memiliki fasilitas unggulan dalam hal teknologi manajemen komputer dan fasilitas fisik representatif (PC, sistem suara video, jaringan). Selain itu, ruangnya dapat menampung sebanyak 54 orang.

3. Laboratorium Pascasarjana S2

Laboratorium ini memungkinkan mahasiswa program master untuk menyelesaikan tugas kuliah dan tesis, seperti studi literatur, uji coba aplikasi dan data, dan penulisan tesis. Mereka juga dapat menggunakan laboratorium bidang minat mereka sendiri.

4. Laboratorium Pascasarjana S3

Mahasiswa program doktor dapat menggunakan laboratorium ini untuk menyelesaikan tugas kuliah dan disertasi, seperti studi literatur, uji coba aplikasi dan data, dan penulisan disertasi. Laboratorium Pascasarjana S3 terdiri dari tiga lantai. Laboratorium S3 di lantai 1 digunakan oleh mahasiswa S3 yang belum mendaftar, Laboratorium Pascasarjana di lantai 3 digunakan oleh mahasiswa S3 yang sudah mendaftar, dan Laboratorium Kerjasama di lantai 3

digunakan oleh mahasiswa S3 yang bekerja sama. Mahasiswa program doktor juga dapat menggunakan laboratorium bidang minat mereka sesuai dengan bidang penelitian mereka.

2.6. Lokasi

Departemen Teknik Informatika, Kampus ITS
Sukolilo, Surabaya 60111.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Penelitian Terkait

Pada penelitian yang berjudul “Diagnosa Covid-19 Chest X-Ray dengan Convolutional Neural Network Arsitektur Resnet-152” tahun 2021, yang ditulis oleh Widi Hastomo dan Adhitio Satyo Bayangkari Karno bertujuan untuk mendiagnosa karakteristik penyakit secara otomatis dan menyediakan alat pendukung pengambilan keputusan yang berharga bagi ahli radiologi untuk deteksi dan diagnosis yang lebih akurat atau efisien dari pneumonia yang terinfeksi Covid-19. Convolution Neural Network (CNN) dengan arsitektur ResNet152 menjadi pilihan karena dapat menghasilkan akurasi yang baik, cukup memerlukan komputasi yang minimum serta dapat diaplikasikan oleh komputer sederhana, hal ini berarti setiap rumah sakit dan komunitas tidak perlu melakukan investasi yang besar dalam penerapannya. Semua gambar sampel memiliki berbagai ukuran. Oleh karena itu, gambar - gambar itu ada yang diubah ukurannya menjadi 224 x 224 piksel. Normalisasi data dilakukan untuk pembelajaran sistem yang lebih baik serta untuk mencegah *overfitting* dan memudahkan generalisasi. Dataset dipartisi menjadi tiga set berbeda, yaitu set pelatihan, set validasi, dan set pengujian. Dengan set pelatihan dan validasi, model CNN yang diusulkan dilatih. Eksperimen dijalankan hingga epoch yang berbeda seperti 50, 60, dan 100. Setelah menjalankan jaringan selama 50 epoch, akurasi jaringan mulai jenuh. Dengan 50 epoch, model mencapai akurasi pelatihan dan validasi yang diharapkan. *Fully Connected Layer* (FCL) dari model dipilih untuk mengekstrak fitur. Fitur vektor diekstraksi dari setiap gambar pelatihan dengan lapisan ini. Fitur vektor dari semua

gambar ini dimasukkan ke dalam empat pengklasifikasi arsitektur ResNet152. Untuk menyesuaikan hyperparameter pengklasifikasi yang terbaik. Terakhir, semua pengklasifikasi ResNet152 digabungkan untuk mengembangkan ansambel pengklasifikasi yang memprediksi label kelas berdasarkan suara mayoritas pengklasifikasi ResNet152. Kinerja sistem yang diusulkan dievaluasi dalam hal *confusion matrix*, *precision*, *accuracy*, dan *F1-score*. Dengan demikian, dataset siap untuk dimasukkan ke dalam jaringan CNN dan untuk melatih model.[14]

Penelitian yang memiliki judul “Penerapan Transfer Learning pada Convolutional Neural Networks dalam Deteksi Covid-19” tahun 2021, yang ditulis oleh Buyut Khoirul Umri dan Visq Delica bertujuan untuk menggunakan metode transfer learning yang akan diuji dalam dataset skala besar dan kecil. Beberapa tahap seperti menyiapkan data gambar dengan preprocessing sebagai tahap training dan evaluasi sebelum dijalankan skenario yang direncanakan. Untuk mengetahui performa terbaik pada penerapan transfer learning yang dilakukan, maka dilakukan 2 skenario yang menerapkan dataset besar dan kecil pada penerapan CNN. Terdapat lima model *pre-trained*, yaitu VGG19, VGG16, MobileNet, DenseNet201, dan ResNet50 yang masing-masing diuji menggunakan dataset besar maupun kecil. Skenario satu adalah penerapan dataset kecil dengan transfer learning, sedangkan skenario dua adalah penerapan dataset besar dengan transfer learning. Berdasarkan hasil analisis dan pengujian yang telah dilakukan pada dua skenario yang berisi penerapan transfer learning pada kasus dataset besar dan kecil dalam mendapatkan performa terbaik dalam mendeteksi Covid-19 menggunakan gambar Chest X-Ray (CXR) dapat disimpulkan bahwa semua penerapan kasus dataset besar dapat meningkatkan performa dari semua model yang diujikan. Penerapan dataset besar dalam deteksi

ini sangat disarankan karena memiliki performa yang terbukti lebih baik.[15]

Penelitian yang berjudul “*Convolutional Neural Network (CNN) Based on Artificial Intelligence in Periodontal Diseases Diagnosis*” tahun 2024, yang ditulis oleh Nurul Huda Danial dan Dian Setiawati berfokus pada pengembangan alat kecerdasan buatan untuk membantu peran diagnostik dan terapeutik. Penggunaan sistem CNN dalam diagnosis gigi dan perencanaan perawatan yaitu memungkinkan dokter gigi untuk mengurangi kesalahan diagnostik yang timbul dari ketegangan atau kelelahan. Sistem CNN juga dapat menangkap detail yang terlewatkan oleh dokter dalam diagnosis radiografi dan pemeriksaan radiografinya dapat direkam untuk membuat database. Penelitian ini mengevaluasi kegunaan sistem deteksi berbantuan komputer berdasarkan algoritma *deep* CNN dalam mengkategorikan tingkat keparahan kehilangan tulang akibat penyakit periodontal. VGG16 berguna untuk mendeteksi kehilangan tulang alveolar pada radiografi periapikal dan memiliki kemampuan yang memuaskan untuk mendeteksi tingkat keparahan kehilangan tulang pada gigi. Hasilnya menunjukkan bahwa mesin dapat bekerja lebih baik berdasarkan klasifikasi level dan karakteristik diagnosis citra yang ditangkap. Penelitian ini menunjukkan bahwa CNN memiliki akurasi diagnostik sebesar 73,04% dalam mendeteksi kehilangan tulang alveolar pada gigi anterior. Analisis *deep learning* menggunakan radiografi dapat membantu dalam mendiagnosis dan merencanakan perawatan penyakit periodontal dengan deteksi dini perubahan periodontal.[5]

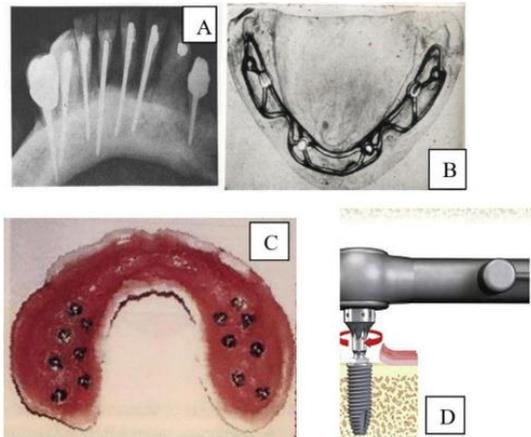
Penelitian yang berjudul “*ResNet-50 based Deep Neural Network using Transfer Learning for Brain Tumor Classification*” tahun 2022, yang ditulis oleh Madona Sahaai dan kolega bertujuan pada pembagian dan klasifikasi tumor otak menggunakan model

ResNet50 yang telah dilatih sebelumnya. Penelitian ini mencapai validasi 95,3% pada berbagai kelas data tumor otak. Untuk meningkatkan akurasi klasifikasi, sistem yang disarankan akan menggunakan metode augmentasi data dan transfer learning. Untuk meningkatkan efisiensi, penelitian tentang arsitektur *deep learning* lainnya mungkin diperlukan. Model ini dapat digunakan dalam pemrosesan gambar medis karena dapat membantu dalam diagnosis tumor dan perencanaan pengobatan. Penelitian ini melibatkan 233 orang yang menderita tumor otak dari tiga jenis yang berbeda.[8]

Penelitian yang berjudul “Automated Detection of Missing Tooth Regions in CBCT Slices for Dental Implant Planning Using CNN with Transfer Learning” tahun 2024, yang ditulis oleh Mohammad Farid Naufal dan kolega bertujuan untuk mengimplementasikan CNN dengan transfer learning untuk deteksi otomatis area gigi hilang dalam irisan CBCT. Arsitektur ini termasuk Inceptionv3, VGG16, MobileNetV2, ResNet50, dan ResNet101. Dataset terdiri dari 3.883 gambar CBCT yang dikumpulkan dari Rumah Sakit Pendidikan Universitas Airlangga Surabaya. Untuk meningkatkan kontras dilakukan preprocessing menggunakan *histogram equalization*. Untuk menghindari *overfitting*, teknik augmentasi data digunakan. Dengan *F1-score* rata rata 98,20%, ResNet101 adalah metode terbaik untuk perencanaan implan gigi, mengurangi waktu dan tenaga, meningkatkan efisiensi dan akurasi proses, dan meningkatkan hasil perawatan pasien. Ini ditunjukkan oleh evaluasi dengan k fold cross validation dengan nilai k sebesar lima. [16]

3.2. Implan Gigi

Penyebab umum kehilangan gigi pada orang dewasa terdiri dari penyakit periodontal, trauma, dan cacat perkembangan. [17] Selain mengurangi estetika, kehilangan gigi memiliki konsekuensi yang besar dan nyata pada kesehatan pasien. [18] Implan gigi merupakan salah satu perawatan yang digemari saat ini untuk menggantikan gigi yang hilang. Hal ini dikarenakan implan gigi dapat mencapai tujuan dari kedokteran gigi, yaitu mengembalikan pasien ke kontur, fungsi, kenyamanan, estetika, fonasi, dan kesehatan yang normal.[19]



Gambar 3.1 Tempat Pemasangan Implan Gigi: (A) Transdental Implan, (B) Transmucosal Implan, (C) Subperiosteal Implan, (D) Endosseous Implant [20]

Implan gigi digunakan untuk meniru akar gigi asli atau sebagai akar pengganti yang ditanamkan secara bedah di rahang pasien. Untuk menahan gigi, implan ditanamkan ke dalam jaringan lunak dan keras rongga mulut. Ini memberikan dukungan dan

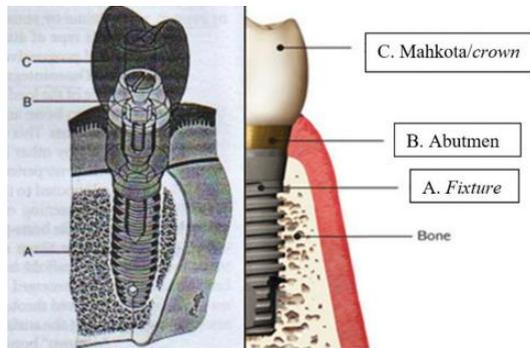
retensi untuk protesa gigi lepasan dan cekat. [19] Terdapat beberapa jenis implan gigi bergantung pada klasifikasinya. Menurut bentuk dan posisinya pada rahang, implan gigi dapat dibagi menjadi tiga jenis, yaitu implan subperiosteal, implan endosseous, dan implan transosteal. Jenis implan juga dapat dibagi berdasarkan bentuknya. Terdapat dua jenis implan, yaitu *threaded* implan dan implan halus. [19]



Gambar 3.2 Bentuk Implan Gigi: (A) Blade Form, (B) Root Form [20]

Oseointegrasi merupakan proses penyatuan tulang dengan implan. Konsep ini pertama kali dikenalkan oleh Branemark pada tahun 1952. Keberhasilan implan gigi juga berkaitan dengan keterampilan operator, kualitas dan kuantitas tulang, serta kebersihan mulut pasien. Berdasarkan struktur kimianya, bahan implan gigi dapat berupa titanium, keramik atau zirconia. Sedangkan, berdasarkan respon biologis dan aktivitas biodinamik, bahan implan dapat dibedakan menjadi material *biotoleran*, *bionert*, dan *bioaktif*. Bagian implan gigi dibagi menjadi *fixture* atau badan implan, abutmen, dan protesa. *Fixture* atau badan implan merupakan bagian yang tertanam pada tulang rahang dan bertindak sebagai infrastruktur karena menyangga struktur

diatasnya, yaitu abutmen dan protesa. Abutmen merupakan bahan yang dipasang pada badan implan yang biasanya dikencangkan dengan sekrup dan berada di bagian intraoral setelah melewati gingiva. Fungsi dari abutmen adalah menghubungkan infrastruktur dengan suprastruktur. Protosa merupakan restorasi akhir yang dipasang pada abutmen. Protosa bisa berupa mahkota, bridge atau denture. Protosa dikenal sebagai suprastruktur. Implan gigi dapat dikatakan berhasil bila terjadi oseointegrasi. [20]



Gambar 3.3 Bagian Implan Gigi [20]

Di bidang kedokteran gigi, implan gigi memiliki pengaruh yang sangat besar. Di Amerika Serikat, data menunjukkan peningkatan jumlah perawatan gigi dengan implan lebih dari sepuluh kali lipat dari tahun 1983 hingga 2002 dan lima kali lipat lagi dari tahun 2000 hingga 2005. Pada tahun 2020, industri ini diperkirakan akan mencapai \$10 miliar dengan lebih dari 1 juta implan gigi dipasang setiap tahun. [21]

3.3. CBCT

Bidang kedokteran gigi telah sangat dipengaruhi oleh kemajuan teknologi. Cara dokter gigi merencanakan dan merancang kasus implan telah berubah karena penggunaan tomografi terkomputerisasi, terutama *Cone Beam Computerized Tomography* (CBCT). Peralatan pemindaian yang lebih cepat, efektif dan memancarkan radiasi yang lebih rendah memungkinkan dokter untuk merencanakan kasus implan secara virtual dengan tingkat akurasi yang lebih tinggi. Kemajuan besar dalam teknologi desain atau manufaktur berbantuan komputer (CAD/CAM) dan gambar CBCT memungkinkan dokter merencanakan, merancang, dan memproses kasus di lingkungan klinik mulai dari pembuatan sementara hingga prostesis akhir. [21]

CBCT mengatasi keterbatasan pencitraan gigi dua dimensi tradisional dan memungkinkan penggambaran yang akurat dari detail multiplanar struktur tulang maksilofasial serta jaringan lunak yang mengelilingi struktur tersebut. Geometri dan ukuran sistem CBCT sangat beragam tergantung pada metode yang digunakan untuk memposisikan pasien. Kebanyakan perangkat memungkinkan pemindaian dalam posisi berdiri dan duduk, tetapi beberapa sistem hanya memungkinkan pemindaian dalam posisi terlentang. Berat sistem berkisar antara 67 kg dan 1050 kg. Saat ini, sebagian besar sistem CBCT yang tersedia menggunakan sudut rotasi 360 derajat untuk akuisisi data. Sudut rotasi ini dapat berkisar antara 180 derajat dan 540 derajat tergantung pada sistem. Untuk mengurangi paparan radiasi pasien, beberapa perangkat memungkinkan pemindaian dengan dua atau tiga sudut rotasi yang berbeda. Waktu pemindaian CBCT saat ini berkisar antara 2 hingga 45 detik. [22]

Penggunaan CBCT dalam implantologi telah mengurangi atau bahkan menghilangkan kebutuhan untuk prosedur. Ini terjadi karena CBCT memungkinkan pengukuran yang tepat terhadap jarak, area, dan volume tulang yang akan ditempatkan implan. Gambar irisan melintang dalam beberapa bidang diberikan oleh CBCT sehingga dapat membantu penilaian akurat terhadap tinggi, lebar, dan sudut tulang. CBCT juga memungkinkan visualisasi posisi foramen mentalis dan kanal alveolar inferior di mandibula, serta sinus di maksila.[23]

3.4. *Image Classification*

Selama bertahun - tahun, klasifikasi gambar telah menjadi fokus penelitian di seluruh dunia. *Deep learning* dalam klasifikasi gambar telah mendorong kemajuan bidang ini. Sejak tahun 2012, *Convolutional neural network* (CNN) menjadi algoritma utama untuk klasifikasi gambar. [24]

Ekstraksi fitur gambar menjadi fokus penelitian pada *image classification*. Untuk mendeskripsikan seluruh gambar, klasifikasi gambar biasanya menggunakan metode pembelajaran fitur atau ekstraksi fitur secara manual, kemudian menggunakan *classifier* untuk menentukan kategori objek. [25]

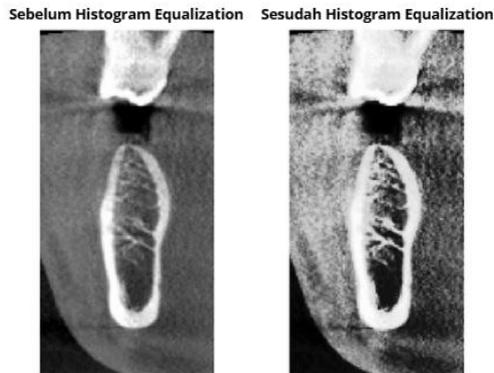
Metode klasifikasi berdasarkan CNN adalah proses pembelajaran *end-to-end* dan berbeda dengan metode klasifikasi gambar tradisional. Dalam metode ini, hanya gambar asli yang dimasukkan, proses pelatihan dan prediksi dilakukan dalam jaringan, dan hasil akhir dikeluarkan. Metode ini meninggalkan cara manual untuk mengekstraksi fitur gambar tertentu dan memecahkan kebuntuan dalam metode klasifikasi tradisional. [25]

Klasifikasi gambar medis melibatkan pemilihan dan labeling gambar medis berdasarkan daftar label yang konsisten.

Tugas ini mencakup mengekstraksi fitur - fitur dari gambar dan menggunakannya untuk memberikan label yang sesuai. [26]

3.5. *Histogram Equalization (HE)*

Histogram equalization (HE) merupakan proses untuk meratakan distribusi nilai piksel sehingga kontras dari sebuah citra dapat diperbaiki. HE berpotensi membantu proses klasifikasi. [27] Keunggulan menerapkan metode pengaturan histogram pada gambar medis sebelum menggunakan algoritma *machine learning* (juga dikenal sebagai *deep learning*) adalah bahwa metode ini memberikan kontras yang ditingkatkan pada gambar yang membuat lebih mudah dilihat oleh mesin. Akibatnya, metode ini dapat mengekstraksi fitur yang lebih menonjol untuk tugas klasifikasi. *Classifier* dapat mendiagnosis gambar medis dengan tingkat akurasi yang lebih tinggi dengan bantuan metode ini. [28]



Gambar 3.4 Penerapan *Histogram Equalization*

HE adalah teknik yang terkenal untuk meningkatkan gambar global dan banyak digunakan dalam bidang pemrosesan gambar, pembelajaran mesin, dan diagnosis gambar medis otomatis sebagai langkah pra-pemrosesan. Menurut metode global,

transformasi HE diterapkan untuk setiap piksel citra secara bersamaan. Untuk meningkatkan rentang dinamis citra, metode HE biasanya meregangkan *Probability Density Function* (PDF).

HE digunakan untuk meratakan histogram berdasarkan kontras gambar dan rentang tingkat intensitas gambar. Tujuannya adalah untuk meningkatkan kontras gambar dengan mendistribusikan ulang nilai intensitas piksel. $H(i)$ merupakan histogram gambar yang diperoleh dari jumlah piksel n_i dengan tingkat intensitas i .

$$H(i) = n_i$$

untuk $i = 0, 1, 2, \dots, (L - 1)$ (3.1)

Dimana L adalah rentang maksimum dari tingkat keabuan (untuk gambar 8-bit, $L = 256$ dengan rentang 0 -255). Berdasarkan persamaan 3.1, histogram dibagi menurut jumlah piksel dengan intensitas tertentu. Jumlah total piksel dapat dihitung melalui persamaan 3.2

$$n = \sum_{i=0}^{L-1} H(i)$$
 (3.2)

Hasil dari perhitungan jumlah total piksel akan digunakan untuk menghitung probabilitas piksel. Probabilitas piksel ini nantinya digunakan untuk menghitung *cumulative distribution function* (CDF) sesuai dengan persamaan 3.4. CDF sendiri memiliki rentang nilai antara 0 - 255 sebagai distribusi dari HE

$$p_x(i) = p(x = i) = \frac{n_i}{n}$$
 (3.3)

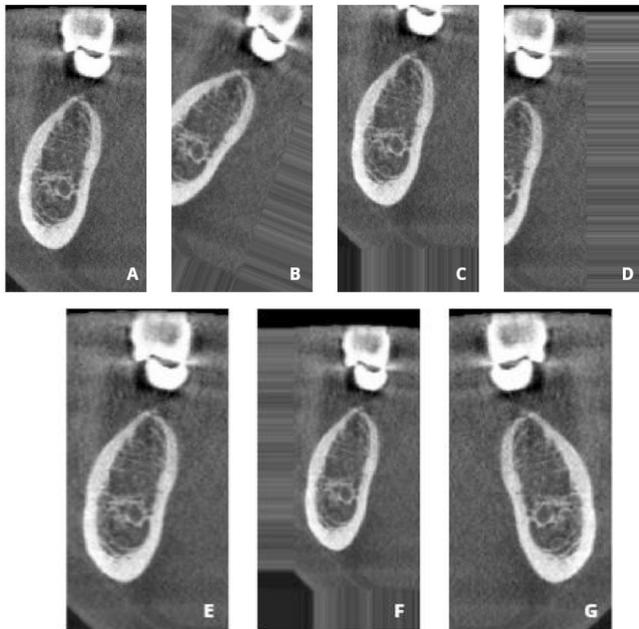
$$cdf_x(i) = \sum_{j=0}^i p_x(x = j)$$
 (3.4)

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{n - cdf_{min}}\right)$$
 (3.5)

Histogram yang telah melalui proses ekualisasi akan dihitung sesuai dengan persamaan 3.5 [29]

3.6. Augmentasi

Dalam *deep learning*, augmentasi data sering digunakan untuk meningkatkan ukuran dan keragaman set pelatihan. Ini dapat dianggap sebagai bentuk teknik regularisasi yang mengurangi kesalahan generalisasi model.[30]

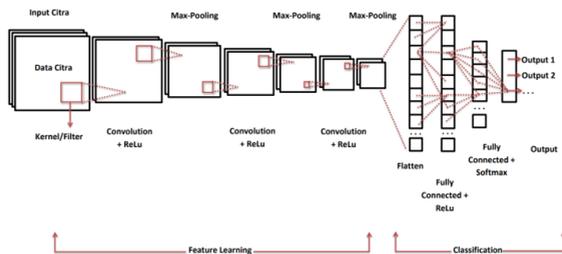


Gambar 3.5 Hasil Augmentasi Data : (A) Gambar Asli, (B) Rotasi, (C) Width Shift, (D) Height Shift, (E) Shear, (F) Zoom, (G) Horizontal flip

3.7. CNN

Convolutional Neural Network (CNN) termasuk dalam *deep learning* yang digunakan untuk memproses data format dua dimensi seperti gambar dan suara. CNN termasuk dalam metode *supervised learning* karena mengklasifikasikan data yang telah memiliki label atau kategori. [32]

Arsitektur CNN terdiri dari *input layer*, *hidden layer*, dan *output layer*. *Hidden layers* sendiri dibagi menjadi dua tahap, yaitu *feature learning* dan *classification*. Pada tahap *feature learning* terdapat *convolutional layer*, *ReLU*, dan *pooling layer*. Pada tahap *classification* terdapat *flatten*, *fully connected layer*, dan *loss layer*. [33]



Gambar 3.6 Arsitektur CNN [33]

3.7.1. Convolutional Layer

Korteks visual otak hewan terdiri dari sel - sel neuron yang mengambil fitur dari gambar. Setiap sel neuron mengambil fitur unik yang membantu pemahaman gambar. Lapisan konvolusi yang dimodelkan berdasarkan sel - sel neuron ini bertujuan untuk mengekstraksi fitur seperti tepi, warna, tekstur, dan orientasi gradien. Lapisan konvolusi terdiri dari filter yang dapat dipelajari. Filter ini dikenal sebagai kernel atau filter konvolusi. Filter ini memiliki

ukuran $n \times m \times d$, dimana d adalah kedalaman gambar. *Dot product* dihitung antara entri filter dan input. CNN secara alami mempelajari filter yang teraktivasi ketika menemukan tepi, warna, tekstur, dan atribut lainnya. Selanjutnya, output lapisan konvolusi dimasukkan ke dalam lapisan fungsi aktivasi. [26]

Input dari setiap node dalam lapisan konvolusi hanyalah blok kecil matriks jaringan di lapisan sebelumnya dan biasanya berukuran 3×3 atau 5×5 . [34]

3.7.2. Active Function ReLU

Setiap elemen dari lapisan konvolusi diaktifkan dengan menggunakan fungsi aktivasi. Ini membuat input dan output jaringan menjadi proses pemetaan nonlinear tanpa mengubah ukuran matriks.[34] Dalam beberapa tahun terakhir, fungsi aktivasi yang paling banyak digunakan adalah *rectified linear unit* (ReLU). ReLU menyelesaikan masalah hilangnya gradien yang hilang disebabkan oleh fungsi *sigmoid* dan *tanh* yang meningkat seiring dengan bertambahnya jumlah lapisan jaringan. [24]. Jika x negatif, fungsi nonlinier ini mengambil angka bernilai real x dan mengubah x menjadi 0. [26]

Rumus matematis dari fungsi ReLU dapat dilihat pada persamaan 3.6.

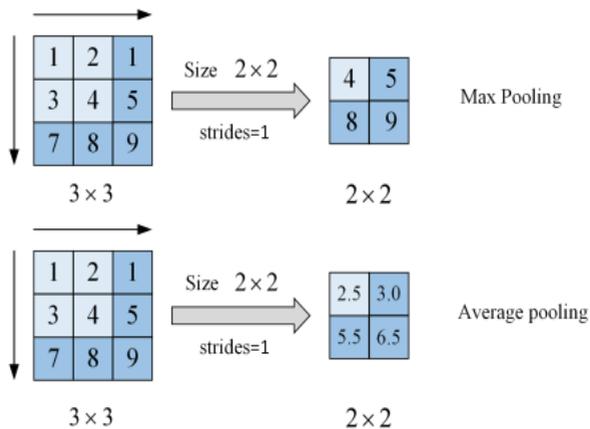
$$ReLU = \max(0, x) \tag{3.6}$$

Salah satu kekurangan dari fungsi aktivasi ReLU adalah beberapa neuron dalam jaringan mungkin tidak teraktivasi setiap saat sehingga bobotnya tidak dapat diperbarui. [34]

3.7.3. Pooling Layer

Metode down sampling digunakan pada lapisan pooling untuk mendapatkan fitur-fitur utama pada matriks fitur lokal. Fitur-fitur ini kemudian digambarkan dengan mengurangi dimensi matriks fitur dan mengurangi jumlah parameter dalam jaringan pada *fully connected layer* yang terakhir. Hal ini mengurangi kompleksitas model, mengurangi kemungkinan masalah *overfitting* dalam pelatihan, dan meningkatkan ketahanan model. Pada saat yang sama, perhitungan model menjadi lebih cepat. [34]

Dua teknik yang paling umum digunakan dalam CNN adalah *average pooling* dan *maximum pooling*. [25] *Maximum pooling* dapat mengurangi kesalahan konvolusi dan menjaga informasi struktur gambar, sedangkan metode *average pooling* secara efektif mengurangi dampak *noise* pada gambar. [34]



Gambar 3.7 Max Pooling dan Average Pooling [25]

3.7.4. Fully Connected Layer

Neuron antara berbagai lapisan terhubung sepenuhnya di lapisan *fully connected* yang berada di belakang lapisan konvolusi dan lapisan pooling.[25] Lapisan *fully connected* dapat menerima output dari lapisan sebelumnya sebagai input. Ada juga probabilitas skor untuk setiap kelas dimana gambar input dapat dihitung. Ketika gambar input diklasifikasikan, kelas dengan nilai tertinggi dihasilkan. [14]

3.7.5. Softmax Classifier

Salah satu jenis algoritma *logistic regression* dikenal sebagai fungsi aktivasi *softmax*. Fungsi ini dapat digunakan untuk mengklasifikasikan lebih dari dua kelas. Tugas klasifikasi kelas biner adalah standar klasifikasi yang biasa dilakukan oleh algoritma ini. [35] *Softmax* dapat menghitung probabilitas untuk semua label dengan mengambil vektor nilai bernilai riil dari label yang ada dan mengubahnya menjadi vektor dengan nilai antara nol dan satu yang akan bernilai satu ketika semua label dijumlahkan.[36]

3.8. Transfer Learning

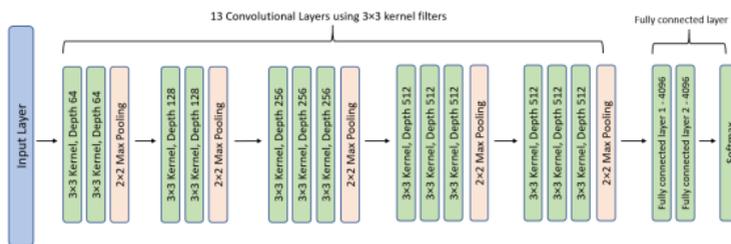
Transfer learning merupakan salah satu pendekatan alternatif untuk *training* model CNN. *Transfer learning* menggunakan model yang sudah di *training* pada domain dataset tertentu, seperti ImageNet. [37] *Transfer learning* juga dapat mempersingkat waktu pelatihan karena model telah menyelesaikan tahap pelatihan pertama dengan menggunakan kumpulan data yang sangat besar. [38] *Transfer learning* dapat digunakan untuk meningkatkan kinerja dan akurasi model *deep learning* yang sedang dibuat serta mempercepat proses pembuatan model itu sendiri tanpa harus memulainya dari awal. [36] Secara khusus, dalam *transfer learning* dengan CNN untuk klasifikasi gambar

medis, tugas klasifikasi gambar medis (target tugas) dapat dipelajari dengan menggunakan karakteristik umum yang dipelajari dari klasifikasi gambar alami (sumber tugas), dimana label tersedia di kedua domain.[39]

3.9. VGG-16

VGG Group dari Universitas Oxford pertama kali mengusulkan VGG-16. VGG-16 menggunakan beberapa kernel konvolusi 3x3 berturut turut daripada kernel konvolusi yang lebih besar, seperti 11x11 dan 5x5 pada AlexNet. Efek penggunaan beberapa kernel konvolusi kecil untuk rentang bidang reseptif tertentu lebih baik daripada menggunakan kernel konvolusi yang lebih besar karena lapisan nonlinier berlapis - lapis dapat meningkatkan kedalaman jaringan yang memungkinkan untuk mempelajari pola yang lebih kompleks dan mengurangi biaya komputasi. [24]

VGG-16 merupakan salah satu arsitektur transfer learning yang terdiri dari 13 *convolutional layers*, 2 *fully connected layers*, dan 1 *SoftMax classifier*. [7]



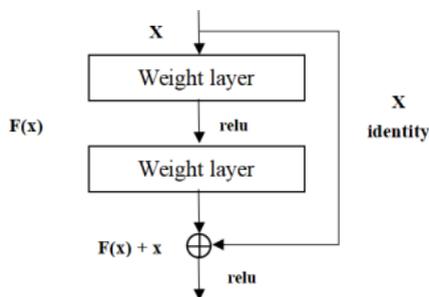
Gambar 3.8 Arsitektur VGG-16 [7]

Arsitektur VGG-16 dimulai dengan dua *convolutional layers* pada lapisan pertama dan kedua. Masing - masing terdiri dari 64 fitur *kernel filters* berukuran 3x3. Dimensi dari gambar input akan berubah menjadi 224x224x64. Hasil output diteruskan

ke layer *max pooling* dengan *stride 2*. Pada lapisan *convolutional* ketiga dan keempat terdapat 128 fitur *kernel filters* berukuran 3×3 . Hasil output akan diteruskan ke layer *max pooling* dengan *stride 2*. Dimensi output pada *convolutional layer* ketiga akan menjadi $112 \times 112 \times 128$. Lapisan *convolutional* kelima hingga ketujuh menggunakan 256 *feature map* dengan ukuran 3×3 dan diikuti dengan *max pooling* dengan *stride 2*. Lapisan kedelapan hingga ketiga belas terdiri dari dua lapisan *convolutional* dengan ukuran 3×3 dimana masing masing layer memiliki 512 *kernel filters*. Pada akhir masing - masing *convolutional* terdapat *max pooling* dengan *stride 1*. Lapisan keempat belas dan kelima belas merupakan *fully connected hidden layers* dengan 4096 unit yang diakhiri dengan *softmax* yang merupakan layer output dengan 1000 unit. [7]

3.10. ResNet

ResNet adalah salah satu algoritma *Convolutional Neural Network* yang digunakan untuk mencegah terjadinya *vanishing* atau *exploding gradients*. [40] ResNet menerapkan teknik *skip connections* dengan membuat jalur langsung yang menghubungkan output dari satu lapisan ke lapisan yang lebih jauh ke depan melewati satu atau lebih lapisan diantaranya.[8]

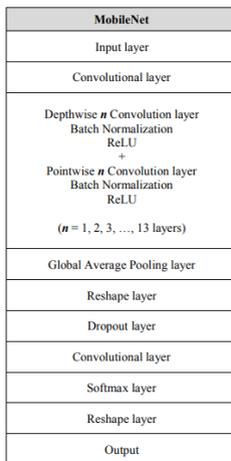


Gambar 3.9 Blok Residual Network [40]

Arsitektur pada ResNet50, ResNet101, dan ResNet152 menggunakan desain bottleneck dengan tiga lapisan *convolutional*. ResNet50 memiliki sekitar 3,8 miliar FLOPs, ResNet101 memiliki sekitar 7,6 miliar FLOPs, dan ResNet152 memiliki sekitar 11,3 miliar FLOPs. Meskipun ResNet152 memiliki kedalaman yang terus meningkat, tetapi model ini memiliki kompleksitas lebih rendah jika dibandingkan dengan VGG-16 yang memiliki sekitar 15,3 miliar FLOPs. [41]

3.11. MobileNet

MobileNet merupakan salah satu algoritma yang menggunakan *Convolutional Neural Network*. MobileNet dikembangkan oleh Google untuk efisiensi komputasi pada perangkat *mobile* tanpa mengorbankan akurasi hasil secara signifikan. [42]



Gambar 3.10 MobileNet [43]

MobileNet didasarkan pada *depthwise separable convolutions* yang terdiri dari dua layer inti, yaitu *depthwise*

convolutions dan *pointwise convolutions*. *Depthwise convolutions* melakukan filtering pada setiap input tanpa menciptakan fitur baru. Sedangkan *pointwise convolutions* akan menghubungkan informasi dari output *depthwise convolutions* untuk menciptakan fitur baru. Arsitektur pada MobileNet terdiri dari *input layer*, *convolutional layer*, *depthwise convolution layer* yang diikuti dengan *batch normalization layer* dan *ReLU layer*, *pointwise convolution layer* yang diikuti dengan *batch normalization layer* dan *ReLU layer*, *global average pooling layer*, *reshape layer*, *dropout layer*, *convolutional layer*, *softmax layer*, dan *reshape layer*. [43]

3.12. K Fold Cross Validation

Untuk meningkatkan kinerja metode, proses k fold cross validation dapat menguji set data yang diacak untuk setiap set data.[44] K fold cross validation dapat mengikuti langkah-langkah dasar berikut :

- a. Tetapkan jumlah lipatan k
- b. Bagi dataset menjadi k bagian
- c. Tetapkan lipatan k-1 untuk set pelatihan dan pengujian
- d. Latih model pada set pelatihan
- e. Verifikasi hipotesis pada set uji
- f. Mempertahankan hasil validasi
- g. Langkah c-f harus diulang sebanyak k kali menggunakan lipatan terakhir sebagai set pengujian. Terakhir, pastikan model divalidasi pada setiap lipatan
- h. Rata - rata hasil dari langkah f digunakan untuk mendapatkan skor akhir.

Kelebihan terbesar dari k fold cross validation adalah bahwa metode ini tidak peduli bagaimana data dibagi. Dalam set

pelatihan, setiap titik data muncul “k-1” kali, tetapi dalam set uji, mereka muncul tepat satu kali.[45]

3.13. Hyperparameters

Hyperparameter merupakan variabel yang sangat mempengaruhi proses pelatihan model CNN. Variabel hyperparameter dapat ditentukan sebelum proses pelatihan. [35]

3.13.1. Hyperparameters pada Proses Training

Hyperparameters untuk proses training terdiri dari epoch, *early stopping*, dan adam *optimizer*. Dalam proses pelatihan, epoch adalah hyperparameter yang menentukan berapa lama model algoritma pembelajaran akan bekerja untuk mengolah dataset pelatihan secara keseluruhan. [35]

Untuk mencapai *early stopping*, estimasi kesalahan generalisasi diperiksa selama proses pelatihan dan dihentikan segera setelah estimasi kesalahan generalisasi mulai meningkat.[46] Adam adalah algoritma pengoptimalan yang dapat digunakan sebagai pengganti metode penurunan gradien *stochastic* klasik untuk memperbarui ulang jaringan berat berdasarkan data pelatihan. Adam dapat mencapai hasil yang baik dengan cepat sehingga algoritma ini populer di bidang *deep learning*. [47]

3.13.2. Hyperparameters Regularisasi

Hyperparameters untuk regulasi dapat terdiri dari dropout dan L2 kernel regularisasi. Dalam metode regulasi jaringan syaraf yang dikenal sebagai dropout, beberapa neuron dipilih secara acak dan tidak akan digunakan selama proses pelatihan. Kontribusi neuron yang dibuang akan dihentikan, sementara jaringan dan bobot baru tidak

diterapkan pada neuron saat *backpropagation*. [48] Nilai dropout biasanya berkisar antara 20% dan 50% dari neuron. [35]

Metode regularisasi L2 menambahkan L2 norm ke vektor parameter sebagai istilah regularisasi pada *loss function*. Metode ini membatasi kompleksitas model selama pelatihan dan dengan demikian mengurangi overfitting.[49]

3.13.3. Loss Function

Binary cross entropy adalah kelas khusus cross entropy dimana target prediksi adalah 1 atau 0. Binary cross entropy digunakan sebagai tujuan pelatihan untuk meminimalkan rata - rata kesalahan probabilitas antara label prediksi dan target untuk setiap piksel. Ini sangat berguna untuk melatih model menyelesaikan banyak masalah klasifikasi sekaligus, jika setiap klasifikasi dapat direduksi menjadi pilihan biner.[50]

3.14. Metrik Evaluasi

Parameter pengukuran performanya dapat digunakan untuk menentukan seberapa baik model klasifikasi bekerja. Untuk menghitung faktor-faktor tersebut, diperlukan matriks yang dikenal sebagai *confusion matrix*. [48]

| | | Kejadian Sebenarnya | |
|-----------------------|---|---------------------|-------------------|
| | | P | N |
| Hipotesis Kejadian | P | True Positive | False Positive |
| | N | False Negative | True Negative |

Gambar 3.11 Confusion Matrix [48]

Pada *confusion matrix* terdapat beberapa nilai, yaitu “*True Positive*” (TP), “*True Negative*” (TN), “*False Positive*” (FP), “*False Negative*” (FN). [48]

Terdapat empat parameter untuk mengevaluasi model, antara lain *accuracy*, *precision*, *recall*, dan *f1 score*. [51] Parameter tersebut dapat dirumus sebagai berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (3.6)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.7)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.8)$$

$$F1\ Score = \frac{2 (Recall \times Precision)}{(Recall + Precision)} \quad (3.9)$$

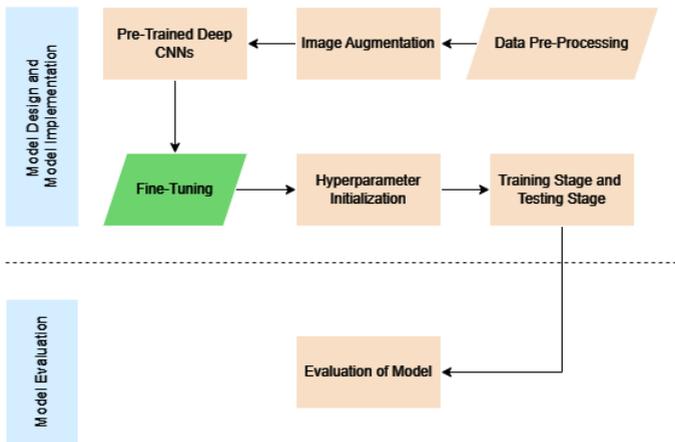
[Halaman ini sengaja dikosongkan]

BAB IV ANALISIS DAN PERANCANGAN

Pada bagian ini akan dijelaskan beberapa tahapan, yaitu kerangka berpikir dan tahapan penelitian yang dilakukan, perencanaan dan skenario model yang akan dibuat serta solusi dan rencana evaluasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT.

4.1. Desain Sistem

Untuk mempermudah implementasi model maka dijabarkan langkah-langkah apa saja yang akan diambil. Kerangka pikir dari implementasi model ini direpresentasikan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Sistem

Dalam implementasi ini tahap pertama adalah menentukan tahap penelitian. Dalam tahap penelitian yang pertama adalah menentukan sebuah ide dari pertemuan bersama pembimbing lapangan dan melakukan analisis literatur berupa *paper* / makalah dari jurnal berkualitas yang berkaitan dengan model klasifikasi

citra 3D CBCT menggunakan metode *transfer learning* untuk rekomendasi ukuran implan gigi. Kemudian dari analisis tersebut, didapatkan sebuah masalah dari model klasifikasi citra 3D CBCT untuk rekomendasi ukuran implan gigi. Dari masalah tersebut, ditemukan sebuah solusi untuk menyelesaikan atau memperbaiki masalah tersebut sehingga dibuatlah rumusan masalah penelitian atau *research question*.

Tahap selanjutnya adalah tahap implementasi model klasifikasi citra 3D CBCT untuk rekomendasi ukuran implan gigi menggunakan metode *transfer learning* yang terdiri dari model VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet. Langkah pertama adalah melakukan pengumpulan data berupa citra 3D CBCT yang merepresentasikan anatomi rahang dari pasien. Dataset telah dikategorikan ke dalam dua kelas, yaitu ‘area representatif’ dan ‘area tidak representatif’. Dataset akan dibagi menggunakan metode k fold cross validation dengan nilai k sebesar lima. Kemudian sebelum citra masuk ke dalam sebuah arsitektur model, akan dilakukan preprocessing berupa penerapan *histogram equalization* dan proses augmentasi.

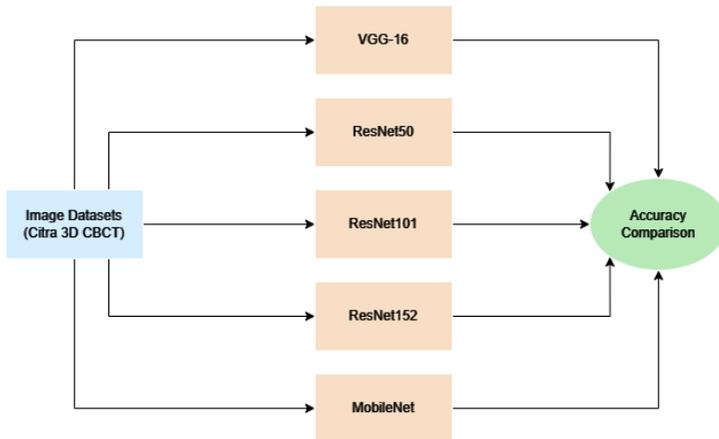
Dalam implementasi model, gambar akan masuk sebuah model dengan ukuran resolusi 224 x 224 piksel. Setiap model *pre-trained Deep CNN* yang padat pada layer bawahnya bekerja sebagai *feature extractor* untuk mengenali citra yang dimasukkan. Lapisan yang pada ini berisi banyak lapisan proses konvolusi pada tiap lapisannya yang berfungsi sebagai *dimension reduction* tanpa mengurangi informasi yang penting dari citra. Setelah gambar melalui proses *feature extractor* kemudian dilakukan *fine-tuning* dimana lapisan atas arsitektur *pre-trained* dicairkan sehingga sesuai dengan klasifikasi biner. Setelah itu akan dilakukan proses pelatihan dan pengujian dengan melakukan beberapa metode *hyperparameter tuning*.

Kemudian langkah selanjutnya adalah evaluasi. Dalam tahap ini digunakan *classification metrics* untuk mengukur akurasi performa dari setiap model.

4.2. Desain Model

Dalam hal ini digambarkan bagaimana desain model dari eksperimen setiap model arsitektur *pre-trained deep CNN*. Setiap dari arsitektur *pre-trained*, masing masing dilakukan sebagai *feature extraction* kemudian dalam fase ini akan melatih model dengan menggunakan 10 epoch pada setiap fold nya dengan *patience* lima pada *early stopping*. Proses ini diimplementasikan dengan menggunakan k fold cross validation untuk mendapatkan evaluasi model yang lebih akurat. Setiap model akan melalui skenario yang berbeda, yaitu dengan atau tanpa penggunaan *histogram equalization* serta dengan atau tanpa memperbolehkan layer yang telah dilatih sebelumnya untuk di latih ulang. Dalam implementasi model akan dilakukan pada lima arsitektur yang berbeda dalam *pre-trained deep CNN*, yaitu VGG-16, ResNet50, ResNet101, ResNet152, MobileNet. Pada bagian 'base_model', model *pre-trained* yang digunakan disesuaikan dengan model klasifikasi (VGG-16, ResNet50, ResNet101, ResNet152, dan MobileNet). Pada bagian ini *fully connected* yang terakhir tidak digunakan dan model *pre-trained* ini dilatih pada dataset *ImageNet*.

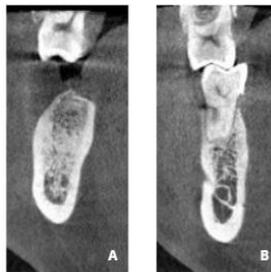
Lapisan tambahan yang digunakan terdiri dari lapisan *flatten*, *dropout*, dan *dense*. Lapisan *flatten* digunakan untuk mengubah output dari lapisan *convolutional* terakhir menjadi vektor satu dimensi. Lapisan *dropout* yang digunakan sebanyak dua lapisan dengan membuang 50% unit secara acak untuk mencegah overfitting. Lapisan *dense* pertama merupakan lapisan *fully connected* dengan 128 neuron dan menggunakan *ReLU* sebagai *activation function* dan ditambah dengan regularisasi L2 untuk mencegah overfitting. Lapisan *dense* terakhir merupakan lapisan output dengan satu neuron dan menggunakan *sigmoid* sebagai *activation function* untuk memprediksi klasifikasi biner.



Gambar 4.2 Desain Model

4.3. Dataset

Setiap model dilatih menggunakan dataset yang telah dikategorikan ke dalam dua kelas, yaitu ‘area representatif’ dan ‘area tidak representatif’.



Gambar 4.3 (A) Area Representatif dan (B) Area tidak Representatif

Area representatif adalah area yang tidak memiliki akar gigi dan akan dipasang implan. Sedangkan area tidak representatif adalah yang masih mengandung akar gigi, area rahang bagian depan, dan area rahang paling belakang. Dataset yang dilatih berupa slice CBCT yang mengikuti lengkung rahang. Kelas representatif memiliki jumlah gambar sebanyak 541, sedangkan kelas tidak representatif memiliki jumlah gambar sebanyak 3342.

4.4. Pre-Processing Data

Pada tahap ini ditunjukkan proses pengumpulan data yang akan digunakan untuk melakukan *training* dan *testing* pada model. Dataset yang akan digunakan digabungkan menjadi satu pada sebuah folder bernama ‘merged’ dimana di dalam folder tersebut telah dibagi menjadi dua folder yang masing masing mewakili nama kelas, yaitu ‘Non-Rep’ untuk dataset *non-representatif* dan ‘Rep’ untuk dataset *representative*.

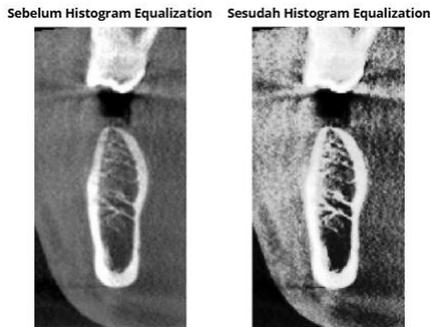
Preprocessing dilakukan untuk memastikan model dapat memahami gambar dalam hal ini. Langkah - langkah *preprocessing* termasuk mengimbangi histogram untuk meningkatkan kontras gambar, melakukan *label encoder* untuk mengubahnya menjadi format numerik yang dapat dipahami oleh model, dan meningkatkan data untuk meningkatkan variasi data dengan melakukan augmentasi data.

4.5.1. Histogram Equalization

Histogram equalization (HE) merupakan proses untuk meratakan distribusi nilai piksel sehingga kontras dari sebuah citra dapat diperbaiki. Selanjutnya data berupa gambar tersebut akan dibaca menggunakan *OpenCV* dan diubah ukurannya menjadi 224x224 piksel agar dapat digunakan sebagai input model. Terdapat dua skenario yang dijalankan. Skenario pertama adalah dataset gambar tidak melewati proses *histogram equalization*. Sedangkan

skenario kedua adalah dataset gambar akan melewati proses *histogram equalization*.

Pada saat gambar melewati proses *histogram equalization*, terdapat tiga tahapan yang akan dilewati. Pertama, gambar akan diubah menjadi *grayscale*. Kedua, *histogram equalization* akan diterapkan untuk meningkatkan kontras pada gambar. Ketiga, gambar akan diubah kembali menjadi mode *RGB* atau berwarna.



Gambar 4.4 Penerapan Histogram Equalization

Pada proses akhir terdapat fungsi untuk mengembalikan dua array, yaitu 'X' yang berisi gambar yang telah diproses dan 'y' yang berisi label kelas untuk setiap gambar.

4.5.2. LabelEncoder

Label encoding adalah proses mengubah label kategori menjadi format numerik yang dapat dipahami oleh model. Ini memungkinkan model untuk memproses dan belajar dari data dengan lebih baik. Proses ini akan mengubah area representatif menjadi label '1' dan area tidak representatif menjadi label '0'.

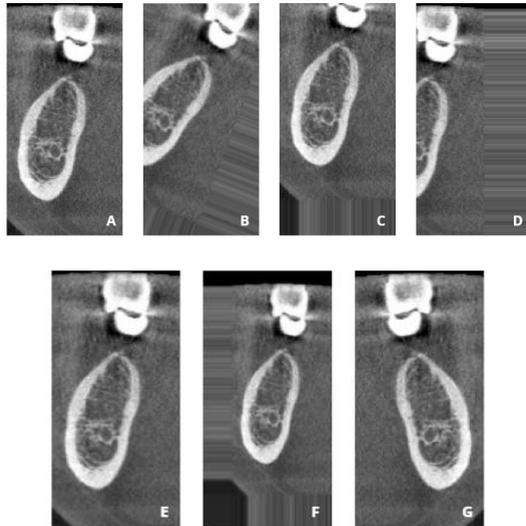
Label kelas yang awalnya berbentuk string akan diubah menjadi bentuk numerik agar dapat digunakan oleh model. Pada pengkodean label, saya menggunakan ‘LabelEncoder’ dari *scikit-learn*. Pada dataset yang menggunakan *histogram equalization*, akan digunakan ‘fit_transform’. Sedangkan pada dataset yang tidak menggunakan *histogram equalization*, akan digunakan ‘transform’. Hal ini dilakukan untuk memastikan konsistensi dalam proses pengkodean.

4.5.3. Augmentasi Data

Augmentasi data merupakan sebuah teknik untuk memanipulasi data tanpa kehilangan inti dari data tersebut. Augmentasi data biasanya digunakan dengan menambahkan data tambahan dari kelas yang kurang terwakili. Augmentasi data dilakukan untuk menambah variasi data *training*. Augmentasi data dilakukan dengan menciptakan variasi baru dari data yang sudah ada dengan melakukan transformasi gambar seperti rotasi, translasi, pergeseran, *zoom*, dan pembalikan horizontal. Untuk melakukan augmentasi data, digunakan ‘*ImageDataGenerator*’ dari Keras.

Pertama, rotasi gambar hingga 30 derajat dan rotasi dilakukan secara acak. Kedua, pergeseran gambar secara horizontal baik ke kiri atau ke kanan hingga 30% dari lebar gambar. Ketiga, pergeseran gambar secara vertikal baik ke atas atau ke bawah hingga 30% dari tinggi gambar. Keempat, transformasi geser pada gambar dengan sudut hingga 30 derajat. Kelima, memperbesar atau memperkecil secara acak pada gambar hingga 30%. Keenam, pembalikan acak pada gambar secara horizontal. Ketujuh, mengatur metode pengisian piksel yang kosong dengan nilai piksel terdekat untuk memastikan tidak ada area kosong yang

muncul setelah transformasi. Pada dataset terdapat kelas yang memiliki jumlah dataset lebih sedikit jika dibandingkan dengan kelas lainnya. Untuk mengatasi permasalahan ini, dilakukan augmentasi untuk kelas minoritas sehingga semua kelas memiliki jumlah dataset yang seimbang.



Gambar 4.5 Hasil Augmentasi Data : (A) Gambar Asli, (B) Rotasi, (C) *Width Shift*, (D) *Height Shift*, (E) *Shear*, (F) *Zoom*, (G) *Horizontal Flip*

Akan dilakukan penghitungan jumlah dataset pada masing masing kelas. Setelah mengetahui kelas mana yang memiliki jumlah dataset lebih sedikit, akan dilakukan pemilihan gambar yang nantinya akan diproses melalui teknik augmentasi data yang telah dijelaskan pada bagian sebelumnya. Proses augmentasi ini terus dilakukan hingga jumlah dataset pada kelas minoritas sudah mencapai target yang ditentukan. Gambar yang dihasilkan melalui proses

augmentasi akan digabungkan dengan dataset yang lain untuk membentuk dataset yang seimbang.

4.5. Proses Training dan Testing

Dalam pembuatan model, akan dilakukan proses *training* dan *testing*. Untuk bisa mendapatkan akurasi klasifikasi citra 3D CBCT menggunakan metode *transfer learning* untuk rekomendasi ukuran implan gigi, perlu dilakukan empat eksperimen seperti yang telah disebutkan sebelumnya. Penggunaan k fold cross validation, inisialisasi hyperparameter, serta pelatihan dan pengujian berulang adalah bagian dari proses ini.

4.6.1. KFold Cross Validation

Metode ini digunakan untuk mengevaluasi kinerja model. Metode ini membagi dataset menjadi k bagian yang sama besar. Pada proses ini k yang digunakan sebesar lima. Model dilatih dengan bagian sebanyak empat dan diuji dengan bagian yang tersisa. Proses ini akan diulang sebanyak lima kali sehingga setiap bagian digunakan hanya sekali sebagai data uji. Hal ini dilakukan untuk memastikan bahwa model tidak lebih - lebihkan bagian data tertentu dan akan memberikan gambaran yang lebih akurat tentang bagaimana kinerja model bekerja pada data yang tidak terlihat sebelumnya.

4.6.2. Inisialisasi Hyperparameter

Inisialisasi hyperparameter dilakukan untuk mengontrol dan mengoptimalkan performa model. Pada setiap bagian, model akan dilatih selama lima epoch. Epoch merupakan jumlah iterasi yang dilalui seluruh dataset oleh model selama pelatihan. Penggunaan lima epoch memungkinkan model untuk belajar dari data secara iteratif dan mengubah bobotnya untuk meminimalkan *loss function*.

Early stopping digunakan dengan nilai *patience* lima dan metric yang digunakan adalah nilai *validation loss*. Ini berarti pelatihan akan dihentikan jika metric tidak meningkat selama lima interval berturut - turut.

Optimizer *Adaptive Moment Estimation* (Adam) digunakan karena memiliki kemampuan untuk menyesuaikan tingkat pembelajaran secara adaptif selama pelatihan.

Dua lapisan dropout dengan *rate* 0.5 digunakan untuk mencegah overfitting. Lapisan pertama diletakkan di belakang lapisan *flatten* dan lapisan kedua diletakkan di belakang lapisan *dense*. Selama pelatihan, dropout akan mengabaikan sejumlah neuron secara acak. Ini membuat model lebih kuat karena tidak terlalu bergantung pada neuron tertentu.

Untuk mencegah nilai parameter model terlalu besar yang dapat menyebabkan overfitting, L2 kernel regularisasi ditambahkan. Regularisasi ini juga menambahkan penalti terhadap nilai parameter yang sangat besar dalam *loss function* yang membantu generalisasi model yang lebih baik.

Binary cross entropy digunakan sebagai *loss function*. Fungsi ini berguna untuk masalah klasifikasi biner dimana target prediksi adalah satu atau nol.

4.6.3. Skenario Pengujian

Pada pengujian akan dilakukan empat skenario yang terdiri dari :

- a. *Histogram equalization* dan *Layer Trainable False*
- b. Tanpa *Histogram equalization* dan *Layer Trainable False*
- c. *Histogram equalization* dan *Layer Trainable True*

d. Tanpa *Histogram equalization* dan *Layer Trainable True*

4.6. Evaluasi Model

Setelah proses training dan testing akan dilakukan evaluasi untuk menguji seberapa baik kinerja dari setiap model. Digunakan perhitungan terhadap *performance metrics* yang terdiri dari *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil ini didapatkan dari perhitungan manual berdasarkan *confusion matrix* yang ditampilkan di akhir setelah proses training dan training dilakukan.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi dari model yang dibuat. Implementasi ini akan dibagi menjadi preprocessing data, proses training dan testing, dan evaluasi model.

5.1. Pre-Processing Data

Berikut adalah implementasi pada pre-processing data :

5.1.1. Histogram Equalization

Berikut adalah implementasi berupa pseudocode pada proses *histogram equalization*:

```
01 Procedure load_and_preprocess_images(dataset_dir, use_hist_eq):
02   Initialize X as empty list
03   Initialize y as empty list
04
05   For each class_name in list_files(dataset_dir) do:
06     class_dir = join_path(dataset_dir, class_name)
07
08     If is_directory(class_dir) then:
09       For each image_name in list_files(class_dir) do:
10         image_path = join_path(class_dir, image_name)
11         image = read_image(image_path)
12         resized_image = resize_image(image, (224, 224))
13
14         If use_hist_eq then:
15           gray_image = convert_to_grayscale(resized_image)
16           equalized_image = apply_histogram_equalization(gray_image)
17           equalized_image = convert_to_color_image(equalized_image)
18           Add equalized_image to X
19         Else:
20           Add resized_image to X
21
22         Add class_name to y
23
24   Return convert_to_array(X), convert_to_array(y)
```

```
25 End Procedure
```

Pseudocode 5.1 Penggunaan *Histogram Equalization*

5.1.2. Encoding

Berikut adalah implementasi berupa pseudocode pada proses encoding:

```
01 Procedure encode_labels(y_equalized, y_non_equalized)
02   Initialize encoder as LabelEncoder()
03
04   y_equalized_encoded = encoder.fit_transform(y_equalized)
05   y_non_equalized = encoder.transform(y_non_equalized)
06
07   Return y_equalized_encoded, y_non_equalized_encoded
08 End Procedure
```

Pseudocode 5.2 Penggunaan *LabelEncoder*

5.1.3. Augmentasi Data

Berikut adalah implementasi berupa pseudocode pada proses augmentasi:

```
01 Procedure augment_minority_classes(X_train, y_train, target_count)
02   Initialize datagen with:
03     rotation_range = 30
04     width_shift_range = 0.3
05     height_shift_range = 0.3
06     shear_range = 0.3
07     zoom_range = 0.3
08     horizontal_flip = True
09     fill_mode = 'nearest'
10
11   Initialize augmented_images as empty list
12   Initialize augmented_labels as empty list
13
14   class_counts = Counter(y_train)
```

```

15  for each class_label, count in class_counts.items() do:
16      if count < target_count then:
17          class_indices = find_indices(y_train == class_label)
18          class_images = X_train[class_indices]
19          class_labels = y_train[class_indices]
20
21          while count < target_count do:
22              for each (X_batch, y_batch) in datagen.flow(class_images,
class_labels, batch_size=32) do:
23                  Add X_batch to augmented_images
24                  Add y_batch to augmented_labels
25                  count += length(y_batch)
26                  if count >= target_count then:
27                      break
28
29          augmented_images = convert_to_array(augmented_images)
30          augmented_labels = convert_to_array(augmented_labels)
31
32  Return concatenate(X_train, augmented_images), concatenate(y_train,
augmented_labels)
33 End Procedure

```

Pseudocode 5.3 Augmentasi Data

5.2. Proses Training dan testing

Berikut adalah implementasi pada proses training dan testing :

5.2.1. KFold Cross Validation

Berikut adalah implementasi berupa pseudocode k fold cross validation :

```

01 Procedure k_fold_cross_validation(k, X, y)
02     Initialize k = 5
03
04     Initialize kfold as KFold with:
05     n_splits = k
06     shuffle = True

```

```

07  random_state = 42
08
09  Initialize early_stopping as EarlyStopping with:
10      monitor = 'val_loss'
11      patience = 3
12      restore_best_weights = True
13
14  for fold_num, (train_index, val_index) in enumerate(kfold.split(X)) do:
15      Print "Fold {fold_num + 1}"
16
17      X_train_fold = X[train_index]
18      y_train_fold = y[train_index]
19      X_val_fold = X[val_index]
20      y_val_fold = y[val_index]
21
22      target_count = max(Counter(y_train_fold).values())
23
24      X_train_augmented, y_train_augmented =
augment_minority_classes(X_train_fold, y_train_fold, target_count)
25      model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
26  End for
27  End Procedure
28 End Procedure

```

Pseudocode 5.4 K Fold Cross Validation

5.2.2. Inisialisasi Hyperparameter

Berikut adalah kode program yang digunakan untuk membangun model klasifikasi dan inisialisasi hyperparameter:

```

01 Procedure initialize_and_build_model(base_model)
02   Initialize base_model with selected pre-trained model
03
04   for each layer in base_model.layers do:
05       Set layer.trainable = False
06   End for
07
08   x = Flatten()(base_model.output)
09   x = Dropout(0.5)(x)

```

```

10 x = Dense(128, activation='relu', kernel_regularizer='l2')(x)
11 x = Dropout(0.5)(x)
12 predictions = Dense(1, activation='sigmoid')(x)
13
14 model = Model(inputs=base_model.input, outputs=predictions)
15 End Procedure

```

Pseudocode 5.5 Arsitektur Model

5.2.3. Kompilasi Model

Berikut adalah kode program yang digunakan untuk kompilasi model :

```

01 model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

```

Pseudocode 5.6 Kompilasi Model

5.3. Evaluasi Model

Hasil evaluasi yang digunakan terdiri dari *loss*, *accuracy*, *f1-score*, dan *confusion matrix*. Berikut adalah kode yang digunakan untuk evaluasi model :

```

01 Initialize lists to store results:
02 losses
03 accuracies
04 conf_matrices
05 training_times
06 testing_times
07 f1_scores
08
09 for each fold in KFold Cross Validation do:
10 start_time = time.time()
11 Train model with:
12 X_train_augmented
13 y_train_augmented
14 epochs = 10

```

```

15     validation_data = (X_val_fold, y_val_fold)
16     callbacks = [early_stopping]
17     training_time = time.time() - start_time
18     Append training_time / len(history.epoch) to training_times
19
20     start_time = time.time()
21     Evaluate model with:
22         X_val_fold
23         y_val_fold
24     val_loss, val_accuracy = model.evaluate()
25     testing_time = time.time() - start_time
26     Append testing_time to testing_times
27     Print "Validation Loss: {val_loss}, Validation Accuracy: {val_accuracy}"
28
29     Append history.history['val_loss'][-1] to losses
30     Append history.history['val_accuracy'][-1] to accuracies
31
32     Predict y_pred_fold with:
33         X_val_fold
34     Round y_pred_fold to y_pred_classes_fold
35
36     Append confusion_matrix(y_val_fold, y_pred_classes_fold) to conf_matrices
37     Append f1_score(y_val_fold, y_pred_classes_fold) to f1_scores
38
39     Print "Classification Report for Fold {fold_num + 1}:"
40     Print classification_report(y_val_fold, y_pred_classes_fold)
41
42     avg_loss = np.mean(losses)
43     avg_accuracy = np.mean(accuracies)
44     avg_training_time = np.mean(training_times)
45     avg_testing_time = np.mean(testing_times)
46
47     Print "Average Loss: {avg_loss}, Average Accuracy: {avg_accuracy}"
48     Print "Average Training Time per Epoch: {avg_training_time:.2f} seconds"
49     Print "Average Testing Time: {avg_testing_time:.2f} seconds"

```

Pseudocode 5.7 Evaluasi Model

[Halaman ini sengaja dikosongkan]

BAB VI HASIL DAN PEMBAHASAN

6.1. Hasil Evaluasi

Berikut adalah hasil evaluasi dari empat skenario yang telah dilakukan pada proses training dan testing

6.1.1. *Histogram Equalization dan Layer Trainable False*

Model VGG-16 menunjukkan performa yang sangat baik pada kelima lipatan dalam skenario *histogram equalization* dan *layer trainable false* seperti yang ditunjukkan pada Tabel 6.1. Lipatan kelima dan kedua memiliki akurasi terbaik dengan nilai 0,9871 atau 98,71%.

Tabel 6.1 Hasil Evaluasi VGG-16 pada Skenario
Histogram Equalization dan Layer Trainable False

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9717 | 0,9655 | 0,8155 | 0,8842 |
| 2 | 0,9871 | 0,9402 | 0,9735 | 0,9565 |
| 3 | 0,9858 | 0,9298 | 0,9725 | 0,9507 |
| 4 | 0,9755 | 0,9877 | 0,8163 | 0,8939 |
| 5 | 0,9871 | 0,9426 | 0,9746 | 0,9583 |

Seperti yang ditunjukkan dalam Tabel 6.2, hasil evaluasi menunjukkan bahwa model ResNet50 menunjukkan kinerja yang konsisten dengan tingkat akurasi yang tinggi pada kelima lipatan dalam skenario *histogram equalization* dan *layer trainable false*. Hasil evaluasi menunjukkan bahwa model ResNet50 memiliki performa

terbaik pada lipatan kelima dan akurasi tertinggi sebesar 0,9755 atau 97,55%.

Tabel 6.2 Hasil Evaluasi ResNet50 pada Skenario *Histogram Equalization* dan *Layer Trainable False*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9678 | 0,8545 | 0,9126 | 0,8826 |
| 2 | 0,9691 | 0,9495 | 0,8319 | 0,8868 |
| 3 | 0,9627 | 0,8077 | 0,9633 | 0,8787 |
| 4 | 0,9716 | 0,9872 | 0,7857 | 0,8750 |
| 5 | 0,9755 | 0,9160 | 0,9237 | 0,9190 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.3, model ResNet101 menunjukkan variasi performa yang signifikan pada kelima lipatan dalam skenario *histogram equalization* dan *layer trainable false*. Hasil evaluasi menunjukkan bahwa model ResNet101 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9832 atau 98,32%.

Tabel 6.3 Hasil Evaluasi ResNet101 pada Skenario *Histogram Equalization* dan *Layer Trainable False*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9627 | 0,7846 | 0,9903 | 0,8755 |
| 2 | 0,9434 | 0,9859 | 0,6195 | 0,7609 |

| | | | | |
|---|--------|--------|--------|--------|
| 3 | 0,9691 | 0,8512 | 0,9450 | 0,8957 |
| 4 | 0,9832 | 0,9885 | 0,8776 | 0,9297 |
| 5 | 0,9523 | 0,7647 | 0,9915 | 0,8635 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.4, model ResNet152 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9845 atau 98,45% dan akurasi terendah sebesar 0,9472 atau 94,72% pada fold kelima.

Tabel 6.4 Hasil Evaluasi ResNet152 pada Skenario *Histogram Equalization* dan *Layer Trainable False*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9730 | 0,9020 | 0,8932 | 0,8976 |
| 2 | 0,9704 | 0,9891 | 0,8053 | 0,8878 |
| 3 | 0,9794 | 0,9604 | 0,8899 | 0,9238 |
| 4 | 0,9845 | 0,9057 | 0,9796 | 0,9412 |
| 5 | 0,9472 | 0,7484 | 0,9831 | 0,8498 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.5, model MobileNet memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9794 atau 97,94% dan akurasi terendah sebesar 0,955 atau 95,50% pada fold kedua.

Tabel 6.5 Hasil Evaluasi MobileNet pada Skenario *Histogram Equalization* dan *Layer Trainable False*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9755 | 0,9200 | 0,8932 | 0,9064 |
| 2 | 0,9550 | 0,9643 | 0,7168 | 0,8223 |
| 3 | 0,9691 | 0,8571 | 0,9358 | 0,8947 |
| 4 | 0,9794 | 0,9457 | 0,8878 | 0,9158 |
| 5 | 0,9613 | 0,9000 | 0,8390 | 0,8684 |

Tabel 6.1 hingga Tabel 6.5 menyajikan hasil evaluasi yang dilakukan setiap model untuk skenario menggunakan *histogram equalization* dan *layer trainable false*. Hasil akurasi pada setiap model bila diurutkan dari yang terbaik, yaitu VGG-16 sebesar 98,71%, ResNet152 sebesar 98,45%, ResNet101 sebesar 98,32%, MobileNet memiliki akurasi sebesar 97,94%, dan ResNet50 memiliki akurasi sebesar 97,55%.

6.1.2. Tanpa *Histogram Equalization* dan *Layer Trainable False*

Model VGG-16 menunjukkan performa yang sangat baik pada kelima lipatan dalam skenario tanpa *histogram equalization* dan *layer trainable false* seperti yang ditunjukkan pada Tabel 6.6. Lipatan kedua memiliki akurasi terbaik dengan nilai 0,9858 atau 98,58%.

Tabel 6.6 Hasil Evaluasi VGG-16 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9730 | 0,9100 | 0,8835 | 0,8966 |
| 2 | 0,9858 | 0,9554 | 0,9469 | 0,9511 |
| 3 | 0,9794 | 0,8908 | 0,9725 | 0,9298 |
| 4 | 0,9845 | 0,9388 | 0,9388 | 0,9388 |
| 5 | 0,9832 | 0,9200 | 0,9746 | 0,9465 |

Seperti yang ditunjukkan dalam Tabel 6.7, hasil evaluasi menunjukkan bahwa model ResNet50 memiliki performa terbaik pada lipatan kedua dan kelima dengan akurasi tertinggi sebesar 0,9820 atau 98,20%.

Tabel 6.7 Hasil Evaluasi ResNet50 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9588 | 0,8087 | 0,9029 | 0,8532 |
| 2 | 0,9820 | 0,9381 | 0,9381 | 0,9381 |
| 3 | 0,9704 | 0,8707 | 0,9266 | 0,8978 |
| 4 | 0,9497 | 0,7153 | 1,000 | 0,8340 |
| 5 | 0,9820 | 0,9194 | 0,9661 | 0,9421 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.8, hasil evaluasi menunjukkan bahwa model

ResNet101 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9807 atau 98,07%.

Tabel 6.8 Hasil Evaluasi ResNet101 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9640 | 0,8099 | 0,9515 | 0,8750 |
| 2 | 0,9717 | 0,8421 | 0,9912 | 0,9106 |
| 3 | 0,9704 | 0,8981 | 0,8899 | 0,8940 |
| 4 | 0,9807 | 0,9560 | 0,8878 | 0,9206 |
| 5 | 0,9691 | 0,8561 | 0,9576 | 0,9040 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.9, model ResNet152 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9858 atau 98,58%

Tabel 6. 9 Hasil Evaluasi ResNet152 pada Skenario Tanpa Histogram Equalization dan Layer Trainable False

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9743 | 0,8807 | 0,9320 | 0,9057 |
| 2 | 0,9743 | 0,8974 | 0,9292 | 0,9130 |
| 3 | 0,9704 | 0,8981 | 0,8899 | 0,8940 |
| 4 | 0,9858 | 0,9677 | 0,9184 | 0,9424 |

| | | | | |
|---|--------|--------|--------|--------|
| 5 | 0,9639 | 0,8214 | 0,9746 | 0,8915 |
|---|--------|--------|--------|--------|

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.10, model MobileNet memiliki performa terbaik pada lipatan kelima dengan nilai akurasi tertinggi sebesar 0,9474 atau 94,74% dan akurasi terendah sebesar 0,8919 atau 89,19% pada lipatan kedua.

Tabel 6.10 Hasil Evaluasi MobileNet pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable False*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9151 | 0,9512 | 0,3786 | 0,5412 |
| 2 | 0,8919 | 1,000 | 0,2566 | 0,4085 |
| 3 | 0,9202 | 0,6424 | 0,9725 | 0,7737 |
| 4 | 0,9201 | 0,8214 | 0,4694 | 0,5974 |
| 5 | 0,9474 | 0,8812 | 0,7542 | 0,8128 |

Tabel 6.6 hingga Tabel 6.10 menyajikan hasil evaluasi yang dilakukan setiap model untuk skenario tanpa menggunakan *histogram equalization* dan *layer trainable false*. Hasil akurasi pada setiap model bila diurutkan dari yang terbaik, yaitu VGG-16 dan ResNet152 memiliki akurasi yang sama sebesar 98,58%, ResNet50 memiliki akurasi sebesar 98,20%, ResNet101 memiliki akurasi sebesar 98,07%, dan MobileNet memiliki akurasi sebesar 94,74%

6.1.3. *Histogram Equalization dan Layer Trainable True*

Model VGG-16 menunjukkan performa yang sangat baik pada kelima lipatan dalam skenario *histogram equalization* dan *layer trainable true* seperti yang ditunjukkan pada Tabel 6.11. Lipatan pertama memiliki akurasi terbaik dengan nilai 0,9627 atau 96,27%.

Tabel 6.11 Hasil Evaluasi VGG-16 pada Skenario *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9627 | 0,8491 | 0,8738 | 0,8612 |
| 2 | 0,9485 | 0,8174 | 0,8319 | 0,8246 |
| 3 | 0,9318 | 0,7333 | 0,8073 | 0,7686 |
| 4 | 0,9021 | 0,5705 | 0,9082 | 0,7008 |
| 5 | 0,9034 | 0,7722 | 0,5169 | 0,6193 |

Seperti yang ditunjukkan dalam Tabel 6.12, hasil evaluasi menunjukkan bahwa model ResNet50 memiliki performa terbaik pada lipatan pertama dengan akurasi tertinggi sebesar 0,9794 atau 97,94%.

Tabel 6.12 Hasil Evaluasi ResNet50 pada Skenario *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9794 | 0,9485 | 0,8932 | 0,9200 |
| 2 | 0,9318 | 0,8261 | 0,6726 | 0,7415 |

| | | | | |
|---|--------|--------|--------|--------|
| 3 | 0,9498 | 0,9730 | 0,6606 | 0,7869 |
| 4 | 0,9472 | 0,8519 | 0,7041 | 0,7709 |
| 5 | 0,8982 | 0,7910 | 0,4492 | 0,5730 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.13, hasil evaluasi menunjukkan bahwa model ResNet101 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9884 atau 98,84%.

Tabel 6.13 Hasil Evaluasi ResNet101 pada Skenario *Histogram Equalization dan Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9858 | 0,9894 | 0,9029 | 0,9442 |
| 2 | 0,1454 | 0,1454 | 1,0000 | 0,2539 |
| 3 | 0,9717 | 0,8595 | 0,9541 | 0,9043 |
| 4 | 0,9884 | 0,9588 | 0,9490 | 0,9538 |
| 5 | 0,9871 | 0,9355 | 0,9831 | 0,9587 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.14, model ResNet152 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9446 atau 94,46%.

Tabel 6.14 Hasil Evaluasi ResNet152 pada Skenario *Histogram Equalization dan Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
|---------------|-----------------|------------------|---------------|-----------------|

| | | | | |
|---|--------|--------|--------|--------|
| 1 | 0,8945 | 0,8889 | 0,2330 | 0,3692 |
| 2 | 0,7156 | 0,2545 | 0,4956 | 0,3363 |
| 3 | 0,9335 | 0,4444 | 0,8205 | 0,5766 |
| 4 | 0,9446 | 0,8571 | 0,6735 | 0,7543 |
| 5 | 0,3969 | 0,2003 | 0,9915 | 0,3333 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.15, model MobileNet memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9794 atau 97,94%.

Tabel 6.15 Hasil Evaluasi MobileNet pada Skenario *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9755 | 0,9200 | 0,8932 | 0,9064 |
| 2 | 0,9550 | 0,9643 | 0,7168 | 0,8223 |
| 3 | 0,9691 | 0,8571 | 0,9358 | 0,8947 |
| 4 | 0,9794 | 0,9457 | 0,8878 | 0,9158 |
| 5 | 0,9613 | 0,9000 | 0,8390 | 0,8684 |

Tabel 6.11 hingga Tabel 6.15 menyajikan hasil evaluasi yang dilakukan setiap model untuk skenario menggunakan *histogram equalization* dan *layer trainable true*. Hasil akurasi pada setiap model bila diurutkan dari yang terbaik, yaitu ResNet101 memiliki akurasi sebesar 98,84%, ResNet50 dan MobileNet memiliki akurasi yang

sama sebesar 97,74%, VGG-16 memiliki akurasi sebesar 96,27%, dan ResNet152 memiliki akurasi sebesar 94,46%

6.1.4. Tanpa *Histogram Equalization* dan *Layer Trainable True*

Model VGG-16 menunjukkan performa yang sangat baik pada kelima lipatan dalam skenario tanpa *histogram equalization* dan *layer trainable true* seperti yang ditunjukkan pada Tabel 6.16. Lipatan kedua memiliki akurasi terbaik dengan nilai 0,9678 atau 96,78%.

Tabel 6.16 Hasil Evaluasi VGG-16 pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9562 | 0,8632 | 0,7961 | 0,8283 |
| 2 | 0,9678 | 0,9151 | 0,8584 | 0,8858 |
| 3 | 0,8636 | 0,5789 | 0,1009 | 0,1719 |
| 4 | 0,8982 | 0,5597 | 0,9082 | 0,6926 |
| 5 | 0,8918 | 0,7742 | 0,4068 | 0,5333 |

Seperti yang ditunjukkan dalam Tabel 6.17, hasil evaluasi menunjukkan bahwa model ResNet50 memiliki performa terbaik pada lipatan kedua dengan akurasi tertinggi sebesar 0,9884 atau 98,84%.

Tabel 6.17 Hasil Evaluasi ResNet50 pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9717 | 0,9355 | 0,8447 | 0,8878 |
| 2 | 0,9884 | 0,9643 | 0,9558 | 0,9600 |
| 3 | 0,2085 | 0,1506 | 1,0000 | 0,2617 |
| 4 | 0,6430 | 0,2613 | 1,0000 | 0,4144 |
| 5 | 0,3492 | 0,1894 | 1,0000 | 0,3185 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.18, hasil evaluasi menunjukkan bahwa model ResNet101 memiliki performa terbaik pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9871 atau 98,71%.

Tabel 6.18 Hasil Evaluasi ResNet101 pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9807 | 0,9074 | 0,9515 | 0,9289 |
| 2 | 0,9871 | 0,9256 | 0,9912 | 0,9573 |
| 3 | 0,8456 | 0,4747 | 0,9450 | 0,6319 |
| 4 | 0,9356 | 0,6667 | 0,9796 | 0,7934 |
| 5 | 0,9768 | 0,8846 | 0,9746 | 0,9274 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.19, model ResNet152 memiliki performa terbaik

pada lipatan keempat dengan nilai akurasi tertinggi sebesar 0,9897 atau 98,97%

Tabel 6.19 Hasil Evaluasi ResNet152 pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9897 | 0,9524 | 0,9709 | 0,9615 |
| 2 | 0,9421 | 1,0000 | 0,6018 | 0,7514 |
| 3 | 0,1403 | 0,1403 | 1,0000 | 0,2460 |
| 4 | 0,4729 | 0,1921 | 0,9898 | 0,3217 |
| 5 | 0,5000 | 0,2332 | 1,0000 | 0,3782 |

Berdasarkan hasil evaluasi yang disajikan dalam Tabel 6.20, model MobileNet memiliki performa terbaik pada lipatan kelima dengan nilai akurasi tertinggi sebesar 0,9794 atau 97,94%.

Tabel 6.20 Hasil Evaluasi MobileNet pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| K Fold | Accuracy | Precision | Recall | F1-Score |
|---------------|-----------------|------------------|---------------|-----------------|
| 1 | 0,9151 | 0,9512 | 0,3786 | 0,5412 |
| 2 | 0,8919 | 1,000 | 0,2566 | 0,4085 |
| 3 | 0,9202 | 0,6424 | 0,9725 | 0,7737 |
| 4 | 0,9201 | 0,8214 | 0,4694 | 0,5974 |

| | | | | |
|---|--------|--------|--------|--------|
| 5 | 0,9474 | 0,8812 | 0,7542 | 0,8128 |
|---|--------|--------|--------|--------|

Tabel 6.16 hingga Tabel 6.20 menyajikan hasil evaluasi yang dilakukan setiap model untuk skenario tanpa menggunakan *histogram equalization* dan *layer trainable true*. Hasil akurasi pada setiap model bila diurutkan dari yang terbaik, yaitu ResNet152 memiliki akurasi sebesar 98,97%, ResNet50 dan MobileNet memiliki akurasi yang sama sebesar 98,84%, ResNet101 memiliki akurasi sebesar 98,71%, VGG-16 memiliki akurasi sebesar 96,78%, MobileNet memiliki akurasi sebesar 92,02%

6.2. Pembahasan

Pada bagian ini akan terdapat pembahasan mengenai gambaran menyeluruh tentang performa rata - rata dari kelima model yang direpresentasikan dalam bentuk Tabel 6.21 hingga 6.25

Tabel 6.21 Hasil Evaluasi Rata-Rata Semua Model pada Skenario *Histogram Equalization* dan *Layer Trainable False*

| Model | Accuracy | Precision | Recall | F1-Score |
|-----------|----------|-----------|--------|----------|
| VGG-16 | 0,9820 | 0,9519 | 0,9167 | 0,9340 |
| ResNet50 | 0,9703 | 0,8962 | 0,8879 | 0,8920 |
| ResNet101 | 0,9638 | 0,8559 | 0,8879 | 0,8716 |
| ResNet152 | 0,9700 | 0,8909 | 0,9200 | 0,9032 |
| MobileNet | 0,9690 | 0,9200 | 0,8519 | 0,8846 |

Tabel 6.21 memberikan gambaran menyeluruh tentang performa rata-rata berbagai model pada skenario *histogram equalization* dan *layer trainable false*. Kelima model tersebut diuji untuk *accuracy*, *precision*, *recall*, dan *f1-score*.

Performa terbaik ditunjukkan oleh VGG-16 dengan akurasi tertinggi sebesar 98,20% dan *f1-score* sebesar 93,40%. Model ini menunjukkan keseimbangan yang sangat baik antara akurasi dan *recall*. Akurasi ResNet50 sebesar 97,03% sedikit lebih rendah dari VGG-16 dan *f1-score* sebesar 89,20% yang menunjukkan kinerja yang kuat dan stabil. Dengan akurasi 96,38%, *precision* 85,59%, dan *recall* 88,79%, dan *f1-score* 87,16% model ResNet101 menunjukkan sedikit peningkatan dalam *recall* daripada *precision*. ResNet152 memiliki akurasi sebesar 97%, *precision* sebesar 89,09%, *recall* sebesar 92%, dan *f1-score* sebesar 90,32%. Ini menunjukkan bahwa model memiliki keseimbangan yang baik antara *precision* dan *recall* dengan kecenderungan lebih baik pada *recall*. MobileNet menunjukkan akurasi sebesar 96,9% dan *precision* yang tinggi sebesar 92%. Namun, *recall* sebesar 85,19% sedikit lebih rendah dan menghasilkan *f1-score* sebesar 88,46%. Ini menunjukkan bahwa meskipun model sangat baik dalam menemukan *True Positive* namun kurang dalam menangkap semua kasus kelas positif.

Secara keseluruhan, semua model pada skenario *histogram equalization* dan *layer trainable false* menunjukkan performa yang baik VGG-16 menunjukkan performa terbaik secara keseluruhan dengan *accuracy*, *precision*, *recall*, dan *f1-score* tertinggi. ResNet50 dan ResNet152 juga menunjukkan performa yang sangat baik dengan akurasi di atas 97% dan *f1-score* di atas 89%. ResNet101 dan MobileNet memiliki performa yang sedikit lebih rendah, namun masih menunjukkan hasil yang kuat dan dapat diandalkan.

Tabel 6.22 Hasil Evaluasi Rata-Rata Semua Model pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable False*

| Model | Accuracy | Precision | Recall | F1-Score |
|--------------|-----------------|------------------|---------------|-----------------|
| VGG-16 | 0,9819 | 0,9273 | 0,9444 | 0,9358 |
| ResNet50 | 0,9703 | 0,8500 | 0,9533 | 0,8987 |
| ResNet101 | 0,9729 | 0,8707 | 0,9439 | 0,9058 |
| ResNet152 | 0,9755 | 0,8929 | 0,9346 | 0,9132 |
| MobileNet | 0,9199 | 0,7922 | 0,5701 | 0,6630 |

Tabel 6.22 memberikan gambaran menyeluruh tentang performa rata-rata berbagai model pada skenario tanpa *histogram equalization* dan *layer trainable false*. Kelima model tersebut diuji untuk *accuracy*, *precision*, *recall*, dan *f1-score*.

Berdasarkan metrik yang disediakan, VGG-16 menunjukkan kinerja terbaik dari semua model. Dengan akurasi yang sangat tinggi sebesar 98,19%, VGG-16 sangat baik dalam mengklasifikasi data dengan benar. *Precision* dan *recall* yang tinggi menunjukkan bahwa model ini tidak hanya memprediksi sebagian besar kasus dengan benar tetapi juga memiliki keseimbangan yang baik antara hasil positif dan negatif yang benar. *F1-score* yang tinggi juga mendukung prediksi ini.

ResNet50 memiliki akurasi yang sedikit lebih rendah dibandingkan VGG-16 sebesar 97,03%. *Precision* yang lebih rendah menunjukkan bahwa meskipun model ini memiliki *recall* yang tinggi sebesar 95,33%, ia cenderung menghasilkan lebih banyak *False Positive* dibandingkan VGG-16. *F1-score* yang juga

lebih rendah dari VGG-16 mencerminkan ketidakseimbangan ini, tetapi tetap memberikan performa yang baik secara keseluruhan.

ResNet101 memiliki akurasi yang sangat baik sebesar 97,29% mendekati ResNet50. Akurasi dan *recall* berada di antara nilai ResNet50 dan VGG-16 dengan *f1-score* menunjukkan keseimbangan yang baik. Meskipun akurasi lebih tinggi dari ResNet50 dan *recall* sedikit lebih rendah, namun tetap menunjukkan kinerja yang sangat baik secara keseluruhan.

Akurasi tertinggi kedua setelah VGG-16 adalah ResNet152, dengan nilai sebesar 97,55%. *Precision* dan *recall* juga sangat baik dengan *f1-score* yang menunjukkan keseimbangan yang kuat antara keduanya. Performanya berada di antara ResNet101 dan VGG-16 tetapi tetap menunjukkan bahwa model ini efektif dalam klasifikasi dengan mengimbangi hasil *True Positive* dan *True Negative*.

MobileNet memiliki performa yang paling buruk dibandingkan model lainnya. Akurasinya sebesar 91,99%, *precision* dan *recall* yang sangat rendah sebesar 57,01% menunjukkan bahwa model ini tidak efektif dalam mengidentifikasi positif. *F1-score* yang rendah menunjukkan ketidakmampuan model untuk menyeimbangkan *precision* dan *recall*.

Secara keseluruhan, VGG-16 adalah model terbaik untuk akurasi, *precision*, *recall*, dan *f1-score*. Diikuti oleh ResNet152 dan ResNet101 yang menunjukkan kinerja yang baik dengan keseimbangan yang baik antara *precision* dan *recall* dan ResNet50 meskipun dirancang untuk efisiensi menunjukkan kinerja yang lebih rendah dalam hal metrik evaluasi ini.

Tabel 6.23 memberikan gambaran menyeluruh tentang performa rata-rata berbagai model pada skenario *histogram*

equalization dan *layer trainable true*. Kelima model tersebut diuji untuk *accuracy*, *precision*, *recall*, dan *f1-score*.

Tabel 6.23 Hasil Evaluasi Rata Rata Semua Model pada Skenario *Histogram Equalization* dan *Layer Trainable True*

| Model | Accuracy | Precision | Recall | F1-Score |
|--------------|-----------------|------------------|---------------|-----------------|
| VGG-16 | 0,9315 | 0,7368 | 0,7850 | 0,7602 |
| ResNet50 | 0,9432 | 0,8889 | 0,6729 | 0,7660 |
| ResNet101 | 0,8165 | 0,4274 | 0,9626 | 0,5920 |
| ResNet152 | 0,7603 | 0,3010 | 0,5463 | 0,3882 |
| MobileNet | 0,9690 | 0,9200 | 0,8519 | 0,8846 |

Dibandingkan dengan skenario tanpa *histogram equalization* dan *layer trainable false*, VGG-16 menunjukkan penurunan performa. *Precision* dan *recall* keduanya menurun menjadi 93,15% yang menunjukkan bahwa model ini kurang efisien dalam mengklasifikasi data dalam situasi ini. Selain itu, *f1-score* menurun yang menunjukkan ketidakseimbangan antara *precision* dan *recall*. Hal ini dapat disebabkan karena penggunaan *histogram equalization* pada distribusi data.

Dibandingkan dengan kasus sebelumnya, ResNet50 menunjukkan peningkatan akurasi sebesar 94,32%. Meskipun *precision* meningkat secara signifikan, *recall* telah menurun. Hal ini menunjukkan bahwa meskipun model ini lebih akurat dalam menemukan kelas positif, kemampuannya untuk menemukan semua kasus positif berkurang. *F1-score* tetap cukup baik, menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall* yang lebih baik dibandingkan VGG-16.

ResNet101 menunjukkan penurunan yang signifikan dalam akurasi sebesar 81,65% dan *precision* 42,74% tetapi peningkatan *recall* yang signifikan sebesar 96,26%. Hal ini menunjukkan bahwa meskipun model ini sangat baik untuk mendeteksi semua kasus positif, itu tidak begitu akurat dalam hal akurasi. Jumlah *f1-score* yang rendah menunjukkan kurangnya keseimbangan antara *precision* dan *recall*.

Dibandingkan dengan skenario sebelumnya, ResNet152 menurun dalam semua metrik evaluasi. Model ini tidak bekerja dengan baik dalam situasi ini karena memiliki akurasi terendah sebesar 76,03% serta *precision* dan *recall* yang sangat rendah. Jumlah *f1-score* yang rendah menunjukkan bahwa model ini tidak seimbang dalam hal *precision* dan *recall*. Pengaruh *histogram equalization* dapat menjadi penyebabnya.

MobileNet menunjukkan peningkatan dalam hal akurasi sebesar 96,90%, *precision* 92%, dan *recall* 85,19%. *F1-score* juga meningkat yang menunjukkan bahwa model ini mengimbangi akurasi dan *recall* dengan baik dalam situasi ini. Ini mungkin menunjukkan bahwa, dibandingkan dengan model lain, MobileNet mampu memanfaatkan *histogram equalization* dan *layer trainable true* dengan lebih baik.

Secara keseluruhan, VGG-16 dan ResNet101 mengalami penurunan performa dalam skenario ini. VGG-16 mengalami penurunan dalam semua metrik, sementara ResNet101 memiliki peningkatan *recall* tetapi penurunan *precision*. ResNet50 menunjukkan peningkatan akurasi dan *precision* tetapi penurunan *recall*. ResNet152 mengalami penurunan signifikan dalam semua metrik. MobileNet menunjukkan performa terbaik dalam skenario ini dengan akurasi, *precision*, *recall*, dan *f1-score* yang semuanya meningkat.

Tabel 6.24 memberikan gambaran menyeluruh tentang performa rata-rata berbagai model pada skenario tanpa *histogram equalization* dan *layer trainable true*. Kelima model tersebut diuji untuk *accuracy*, *precision*, *recall*, dan *f1-score*.

Tabel 6.24 Hasil Evaluasi Rata-Rata Semua Model pada Skenario Tanpa *Histogram Equalization* dan *Layer Trainable True*

| Model | Accuracy | Precision | Recall | F1-Score |
|--------------|-----------------|------------------|---------------|-----------------|
| VGG-16 | 0,9173 | 0,7471 | 0,6075 | 0,6701 |
| ResNet50 | 0,6327 | 0,2701 | 0,9630 | 0,4219 |
| ResNet101 | 0,9458 | 0,7273 | 0,9720 | 0,8320 |
| ResNet152 | 0,6098 | 0,2506 | 0,9159 | 0,3936 |
| MobileNet | 0,9199 | 0,7922 | 0,5701 | 0,663 |

VGG-16 memiliki kinerja yang cukup baik dengan akurasi 91,73% dan *precision* yang cukup tinggi sebesar 74,71% tetapi *recall* yang lebih rendah 60,75% menunjukkan bahwa model ini kehilangan beberapa kasus positif. *F1-score* berada di tengah menunjukkan ketidakseimbangan antara *recall* dan *precision*.

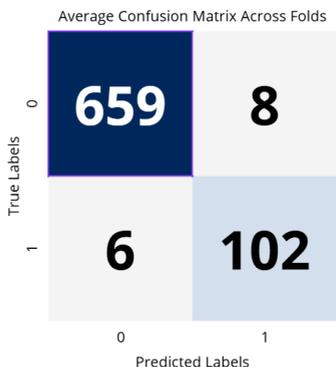
ResNet50 menunjukkan kinerja yang buruk dengan akurasi yang rendah sebesar 63,27%. *Precision* yang sangat rendah sebesar 27,01% menunjukkan bahwa model ini menghasilkan banyak *False Positive*. Namun, *recall* yang sangat tinggi sebesar 96,30% menunjukkan bahwa model ini hampir selalu berhasil mendeteksi kasus positif. *F1-score* yang rendah menunjukkan ketidakseimbangan besar antara *precision* dan *recall*.

ResNet101 menunjukkan performa terbaik diantara model lainnya dengan akurasi tinggi 94,58%. Dengan *recall* yang sangat

tinggi sebesar 97,20% menunjukkan bahwa model ini sangat baik dalam mendeteksi kasus positif. *F1-score* yang tinggi sebesar 83,20% menunjukkan keseimbangan yang baik antara *precision* dan *recall* yang menjadikan model ini sangat andal dalam klasifikasi.

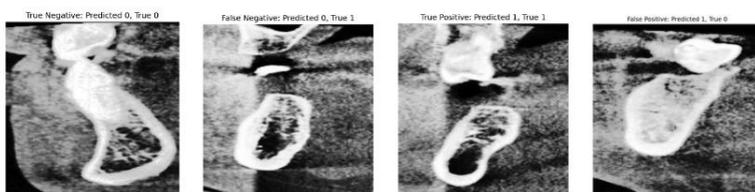
ResNet152 menunjukkan kinerja yang buruk dengan akurasi terendah sebesar 60,98%. *Precision* yang sangat rendah sebesar 25,06% menunjukkan bahwa model ini sering salah dalam prediksi kelas positif. Namun, *recall* yang tinggi sebesar 91,59% menunjukkan bahwa model ini baik dalam mendeteksi hampir semua kasus positif. *F1-score* yang rendah menunjukkan ketidakseimbangan yang signifikan antara *precision* dan *recall*.

MobileNet menunjukkan performa yang cukup baik dengan akurasi sebesar 91,99% yang hampir sama dengan VGG-16. *Precision* yang tinggi 79,22% menunjukkan bahwa model ini baik dalam mengidentifikasi kelas positif. Namun, *recall* yang lebih rendah 57,01% menunjukkan bahwa model ini kehilangan beberapa kasus positif. *F1-score* yang berada di tengah mencerminkan ketidakseimbangan antara *precision* dan *recall*.



Gambar 6.1 Rata-rata *Confusion Matrix* VGG-16 pada Skenario *Histogram Equalization* dan *Layer Trainable False*

Pada Gambar 6.1 terdapat *confusion matrix* yang merupakan hasil rata rata dari lima fold pada model VGG-16 dengan skenario *histogram equalization* dan *layer trainable false*. *Confusion matrix* tersebut memberikan gambaran mengenai performa model dalam menghasilkan klasifikasi citra 3D CBCT kedalam dua kelas. Jumlah *True Negatives* sebanyak 659 menunjukkan jumlah kasus dimana model dengan benar mengklasifikasikan citra sebagai kelas negatif atau non representatif. Jumlah *False Negatives* sebanyak 6 menunjukkan bahwa model secara keliru mengklasifikasikan citra sebagai negatif atau non representatif padahal sebenarnya adalah positif atau representatif. Jumlah *True Positive* sebanyak 302 menunjukkan bahwa model dengan benar mengklasifikasikan citra sebagai kelas positif atau representatif. Angka ini menunjukkan bahwa model sangat baik dalam mengenali citra positif atau representatif dan ini berkontribusi pada akurasi tinggi dari mode. Jumlah *False Positive* sebanyak 8 menunjukkan jumlah kasus dimana model gagal mengidentifikasi citra yang sebenarnya positif atau representatif padahal sebenarnya adalah negatif atau non representatif.



Gambar 6.2 Hasil Citra 3D CBCT Berdasarkan *Confusion Matrix* Menggunakan Model VGG-16

Pada Gambar 6.2 dapat dilihat beberapa gambar yang masuk pada masing-masing kategori *True Positive*, *False Positive*, *True Negative*, dan *False Negative*. Pada gambar *True Negative*,

dapat dilihat dengan jelas bahwa citra 3D CBCT menunjukkan kondisi non representatif dimana masih terdapat gigi pada bagian *alveolar bone* tersebut. Pada gambar *False Negative*, citra 3D CBCT seharusnya diklasifikasikan sebagai kelas representatif tetapi model memprediksi sebagai kelas non representatif. Padahal, dapat dilihat bahwa pada bagian *alveolar bone* sudah tidak terdapat gigi maupun akar gigi sehingga citra tersebut merupakan citra 3D CBCT yang siap untuk dilakukan proses implan. Pada gambar *True Positive*, terdapat bagian *alveolar bone* yang siap untuk dilakukan implan karena tidak terdapat gigi maupun akar gigi. Pada gambar *False Positive*, citra 3D CBCT seharusnya diklasifikasikan sebagai kelas non representatif tetapi model mengklasifikasikan sebagai kelas representatif. Padahal, pada citra tersebut masih terdapat gigi pada *alveolar bone* sehingga tidak dapat dilakukan implan.

Penggunaan *histogram equalization* pada model ini meningkatkan kontras dengan meratakan distribusi intensitas piksel sehingga membuat fitur yang penting lebih menonjol dan lebih mudah dipelajari oleh model. Model VGG-16 dapat memanfaatkan fitur yang sangat informatif dan umum yang telah dipelajari dari dataset ImageNet dengan menerapkan *layer trainable false*. Fitur ini telah terbukti efektif untuk tugas klasifikasi gambar secara keseluruhan. Dengan menerapkan *layer trainable false*, model akan terhindar dari *overfitting*. Tanpa melatih ulang semua lapisan, waktu komputasi yang diperlukan akan berkurang secara signifikan.

Hasil terbaik dari keempat tabel diatas dapat dilihat bahwa VGG-16 memiliki akurasi tertinggi sebesar 98,20% pada skenario menggunakan *histogram equalization* dan *layer trainable false*. Arsitektur VGG-16 memiliki 13 lapisan konvolusi dengan tiga lapisan *fully connected* yang memungkinkan model untuk menangkap fitur yang sangat kompleks dan beragam dari data.

Dengan menggunakan beberapa kernel 3x3 secara berurutan, VGG-16 dapat menangkap fitur lebih detail dan presisi serta meningkatkan efektivitas dalam mendeteksi tepi dan pola.

[Halaman ini sengaja dikosongkan]

BAB VII KESIMPULAN DAN SARAN

7.1. Kesimpulan

Dalam implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT, preprocessing data menjadi penting untuk memastikan kualitas dan konsistensi data. Preprocessing dimulai dengan mengumpulkan data dari direktori yang disusun berdasarkan kelas. Setiap kelas memiliki folder yang berisi gambar-gambar yang relevan. Untuk memastikan kompatibilitas dengan arsitektur model transfer learning yang digunakan, setiap gambar di resize menjadi ukuran seragam, yaitu 224x224 piksel. Untuk meningkatkan kontras gambar, *histogram equalization* digunakan. Ini dilakukan dengan mengubah gambar menjadi *grayscale* menggunakan *histogram equalization* dan kemudian mengubah gambar menjadi format 3-channel (BGR). Untuk mempermudah pelatihan model, *LabelEncoder* digunakan untuk mengkodekan label gambar menjadi format numerik. Terakhir, untuk menyeimbangkan distribusi kelas, augmentasi data minoritas dilakukan menggunakan *ImageDataGenerator*. Ini menggunakan teknik augmentasi seperti rotasi, pergeseran horizontal dan vertikal, shear, zoom, dan flip horizontal.

Hasil evaluasi menunjukkan bahwa model VGG-16 secara konsisten menunjukkan kinerja terbaik dengan akurasi tertinggi sebesar 98,20% pada skenario yang menggunakan *histogram equalization* dan *layer trainable false*. Model ini secara efektif memanfaatkan fitur yang telah dipelajari dari dataset *ImageNet*. Namun, ResNet50, ResNet101, dan ResNet152 menunjukkan bahwa kompleksitas model yang lebih besar tidak selalu

menghasilkan hasil yang lebih baik. Hal ini dapat disebabkan karena overfitting atau peningkatan waktu komputasi tanpa peningkatan performa yang signifikan. Meskipun MobileNet merupakan model yang lebih ringan, model ini memberikan hasil yang baik dalam beberapa kondisi.

Dengan meratakan distribusi intensitas piksel, penggunaan *histogram equalization* meningkatkan kontras gambar, memungkinkan model untuk lebih mudah mempelajari fitur penting gambar yang berkontribusi pada peningkatan akurasi model. Model yang menggunakan *histogram equalization* bekerja lebih baik dibandingkan dengan model yang tidak menggunakannya. Hal ini menunjukkan manfaat preprocessing dalam meningkatkan kualitas fitur yang dipelajari.

Dengan menggunakan *layer trainable false*, model dapat memanfaatkan fitur yang telah dipelajari dari kumpulan data *ImageNet* tanpa melatih ulang semua lapisan. Ini tidak hanya mengurangi kemungkinan overfitting, tetapi juga secara signifikan mengurangi waktu komputasi. Model tetap efektif dalam mendeteksi dan mengklasifikasikan gambar dengan *layer trainable false*.

7.2. Saran

Saran implementasi model klasifikasi menggunakan metode transfer learning untuk pemilihan *slice* citra 3D CBCT adalah dapat melakukan eksplorasi pada teknik peningkatan kontras lainnya, seperti *Contrast Limited Adaptive Histogram Equalization* (CLAHE). Selain itu, dapat melakukan *training* dan *testing* dengan menggunakan arsitektur model lainnya, seperti DenseNet atau EfficientNet.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Jubhari, E. H., & Pangiawan, W. Pentingnya perencanaan prostetik untuk gigi tiruan dukungan implan di zona estetika.
- [2] Riza, M. K. M., & Nugraheni, N. (2023). Tanggung Jawab Hukum Dokter Gigi Atas Terjadinya Risiko Terhadap Pemasangan Implan Gigi Pada Pasien. *Jurnal Ners*, 7(2), 1040-1054.
- [3] Yunus, B., & Murtala, B. (2010). Pemanfaatan hounsfield unit pada CT-scan dalam menentukan kepadatan tulang rahang untuk pemasangan implan gigi. *Journal of Dentomaxillofacial Science*, 9(1), 34-38.
- [4] Pramanik, F., & Firman, R. N. (2015). Interpretasi cone beam computed tomography 3-dimension dalam pemasangan implan dental di RumahSakit Gigi MulutFakultas Kedokteran Gigi Universitas Padjajaran (Interpretation of cone beam computed tomography 3-dimension in inserting dental implant at Dental Hospital of Faculty of Dentistry Padjajaran University). *Journal of Dentomaxillofacial Science*, 14(1), 50-54..
- [5] Danial, N. H., & Setiawati, D. (2024). CONVOLUTIONAL NEURAL NETWORK (CNN) BASED ON ARTIFICIAL INTELLIGENCE IN PERIODONTAL DISEASES DIAGNOSIS. *Interdental Jurnal Kedokteran Gigi (IJKG)*, 20(1), 139-148.
- [6] Ramadhani, T. S., Barkah, R. F., & Lubis, M. G. A. (2024, May). MODEL KLASIFIKASI CORAL HEALTH MENGGUNAKAN TRANSFER LEARNING RESNET-50. In *Indonesian Conference of Maritime* (Vol. 2, No. 1, pp. 150-161).
- [7] Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10), 143-150.

- [8] Sahaai, M. B., Jothilakshmi, G. R., Ravikumar, D., Prasath, R., & Singh, S. (2022, May). ResNet-50 based deep neural network using transfer learning for brain tumor classification. In *AIP Conference Proceedings* (Vol. 2463, No. 1). AIP Publishing.
- [9] Fuadi, A., & Suharso, A. (2022). Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit pada Citra Daun Kentang. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 7(3), 701-710.
- [10] “Sejarah,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/tentang-kami/sejarah/>
- [11] “Visi dan Misi,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/tentang-kami/visi-dan-misi/>
- [12] “Struktur Organisasi,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/dosen-staff/struktur-organisasi/>
- [13] “Laboratorium,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/fasilitas/laboratorium/>
- [14] Hastomo, W., & Karno, A. B. S. (2021). Diagnosa COVID-19 chest X-ray DENGAN convolution neural network Arsitektur ResNet-152. *KERNEL: Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika*, 2(1), 26-33.
- [15] Umri, B. K., & Delica, V. (2021). Penerapan transfer learning pada convolutional neural networks dalam deteksi covid-19. *JNANALOKA*, 53-61.
- [16] Naufal, M. F., Faticah, C., Astuti, E. R., Putra, R. H., & Prambudi, K. I. (2024). Automated detection of missing tooth regions in CBCT slices for dental implant planning using CNN with transfer learning. In *Proceedings of IEEE TENCON 2024*. Department of Informatics, Institut Teknologi Sepuluh Nopember.
- [17] W. Nicholson, J. (2020). Titanium alloys for dental implants: A review. *Prosthesis*, 2(2), 11.

- [18] Pandey, C., Rokaya, D., & Bhattarai, B. P. (2022). Contemporary concepts in osseointegration of dental implants: a review. *BioMed research international*, 2022(1), 6170452.
- [19] GIGI, D. P. D. DENTAL IMPLANT: A LITERATURE REVIEW.
- [20] Nilawati, N. (2024). *Pengenalan Implan Gigi: Teori Singkat dan Praktik Sederhana*. Indonesia Emas Group.
- [21] Resnik, R. (2020). *Misch's Contemporary Implant Dentistry E-Book: Misch's Contemporary Implant Dentistry E-Book*. Elsevier Health Sciences.
- [22] Kaasalainen, T., Ekholm, M., Siiskonen, T., & Kortetniemi, M. (2021). Dental cone beam CT: An updated review. *Physica Medica*, 88, 193-217.
- [23] Jain S, Choudhary K, Nagi R, Shukla S, Kaur N, Grover D. (2019). New evolution of cone beam computed tomography in dentistry: Combining digital technologies. *Imaging Sci Dent*, 49:179–90
- [24] Cai, L., Gao, J., & Zhao, D. (2020). A review of the application of deep learning in medical image classification and segmentation. *Annals of translational medicine*, 8(11).
- [25] Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 4712.
- [26] Sarvamangala, D. R., & Kulkarni, R. V. (2022). Convolutional neural networks in medical image understanding: a survey. *Evolutionary intelligence*, 15(1), 1-22.
- [27] Royce, E., Timotius, I. K., & Setyawan, I. (2012). Sistem pendeteksi senyum berdasarkan metode edge detection, histogram equalization, dan nearest neighbor. *Techné: Jurnal Ilmiah Elektroteknika*, 11(01), 75-82.
- [28] Roy, S., Bhalla, K., & Patel, R. (2024). Mathematical analysis of histogram equalization techniques for medical image

- enhancement: a tutorial from the perspective of data loss. *Multimedia Tools and Applications*, 83(5), 14363-14392.
- [29] Saifullah, S., & Dreżewski, R. (2023). Modified histogram equalization for improved CNN medical image segmentation. *Procedia Computer Science*, 225, 3021-3030.
- [30] Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5), 545-563.
- [31] Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2022). Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*.
- [32] Fahcruroji, A. R., Wijaya, M. Y., & Fauziah, I. (2024). IMPLEMENTASI ALGORITMA CNN MOBILENET UNTUK KLASIFIKASI GAMBAR SAMPAH DI BANK SAMPAH. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 11(1), 45-51.
- [33] ZAHARA, L. M. (2023). IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK (CNN) PADA KLASIFIKASI CITRA SERAT KAYU.
- [34] Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern recognition letters*, 141, 61-67.
- [35] ALAM, I. N. METODE TRANSFER LEARNING PADA DEEP CONVOLUTIONAL NEURAL NETWORK (DCNN) UNTUK PENGENALAN EKSPRESI WAJAH.
- [36] Ramadhan, F. E. (2020). *Penerapan image classification dengan pre-trained model mobilenet dalam client-side machine learning* (Bachelor's thesis, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta).
- [37] Widyaya, J. E., & Budi, S. (2021). Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan

- Pendekatan Transfer Learning Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1).
- [38] Setiawati, P., Karno, A. S. B., Hastomo, W., & Setiawan, I. (2024). Menggunakan Xception, Transfer Learning, dan Permutasi untuk Meningkatkan Klasifikasi Ketidakteraturan Permukaan Baja: Using Xception, Transfer Learning, and Permutation to Improve the Classification of Steel Surface Imperfections. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 350-358.
- [39] Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1), 69.
- [40] Ji, S. (2022). Multi-classification of Human Action Based on ResNet. *Highlights in Science, Engineering and Technology*, 23, 98-105.
- [41] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [42] Carlos, D., Herwindiati, D. E., & Lubis, C. (2024). Implementasi Algoritma Convolutional Neural Networks Untuk Klasifikasi Jenis Cat Tembok Menggunakan Arsitektur MobileNet. *Building of Informatics, Technology and Science (BITS)*, 6(1), 395-402.
- [43] Sae-Lim, W., Wettayaprasit, W., & Aiyarak, P. (2019, July). Convolutional neural networks using MobileNet for skin lesion classification. In *2019 16th international joint conference on computer science and software engineering (JCSSE)* (pp. 242-247). IEEE.
- [44] Cherradi, B., Terrada, O., Ouhmida, A., Hamida, S., Raihani, A., & Bouattane, O. (2021, July). Computer-aided diagnosis system for early prediction of atherosclerosis using machine

- learning and K-fold cross-validation. In *2021 international congress of advanced technology and engineering (ICOTEN)* (pp. 1-9). IEEE.
- [45] Prusty, S., Patnaik, S., & Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, 4, 972421.
- [46] Bentoumi, M., Daoud, M., Benaouali, M., & Taleb Ahmed, A. (2022). Improvement of emotion recognition from facial images using deep learning and early stopping cross validation. *Multimedia Tools and applications*, 81(21), 29887-29917.
- [47] Putri, O. N. (2020). Implementasi Metode CNN Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita).
- [48] Nurhikmat, T. (2018). Implementasi deep learning untuk image classification menggunakan algoritma Convolutional Neural Network (CNN) pada citra wayang golek.
- [49] Liu, T., Bao, J., Wang, J., & Zhang, Y. (2020). A coarse-grained regularization method of convolutional kernel for molten pool defect identification. *Journal of Computing and Information Science in Engineering*, 20(2), 021005.
- [50] Ruby, U., & Yendapalli, V. (2020). Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10).
- [51] Fuadi, A., & Suharso, A. (2022). Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit pada Citra Daun Kentang. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 7(3), 701-710.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Katarina Inezita Prambudi
Tempat, Tanggal Lahir : Sidoarjo, 16 Juni 2003
Jenis Kelamin : Perempuan
Telepon : +6282257902587
Email : inezitakatarina16@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)