

**DETEKSI KANKER KULIT MELANOMA MENGGUNAKAN  
MOBILENETV2 CONVOLUTIONAL NEURAL NETWORK  
BERBASIS ANDROID**

**LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH:**

**MUHAMMAD FARIS FAHRU ROZI**

**NIM 32601900021**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2023**

**FINAL PROJECT**

**MELANOMA SKIN CANCER DETECTION  
USING MOBILENETV2 CONVOLUTIONAL NEURAL NETWORK  
ON ANDROID PLATFORM**

Proposed to complete the requirement to obtain a bachelor's degree (S1)  
at Informatics Engineering Department of Industrial Technology Faculty  
Sultan Agung Islamic University



Arranged By:

**MUHAMMAD FARIS FAHRU ROZI**

**NIM 32601900021**

**MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG**

**2023**

**LEMBAR PENGESAHAN PEMBIMBING**

Laporan Tugas Akhir dengan judul **“Deteksi Kanker Kulit Melanoma Menggunakan MobileNetV2 Convolutional Neural Network Berbasis Android”** ini disusun oleh :

Nama : Muhammad Faris Fahru Rozi

NIM : 32601900021

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari : Fabu

Tanggal : 16 Agustus 2023

Mengesahkan,

Pembimbing I



Ir. Sri Mulyono, M.Eng  
NIDN. 0626066601

Pembimbing II



Ghufro, ST, M.Kom  
NIDN. 0602079005

Mengetahui,

Ketua Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Islam Sultan Agung



  
Ir. Sri Mulyono, M.Eng  
NIDN. 0626066601

## LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “**Deteksi Kanker Kulit Melanoma Menggunakan MobileNetV2 Convolutional Neural Network Berbasis Android**” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Kamis

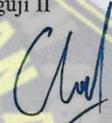
Tanggal : 10 Agustus 2023

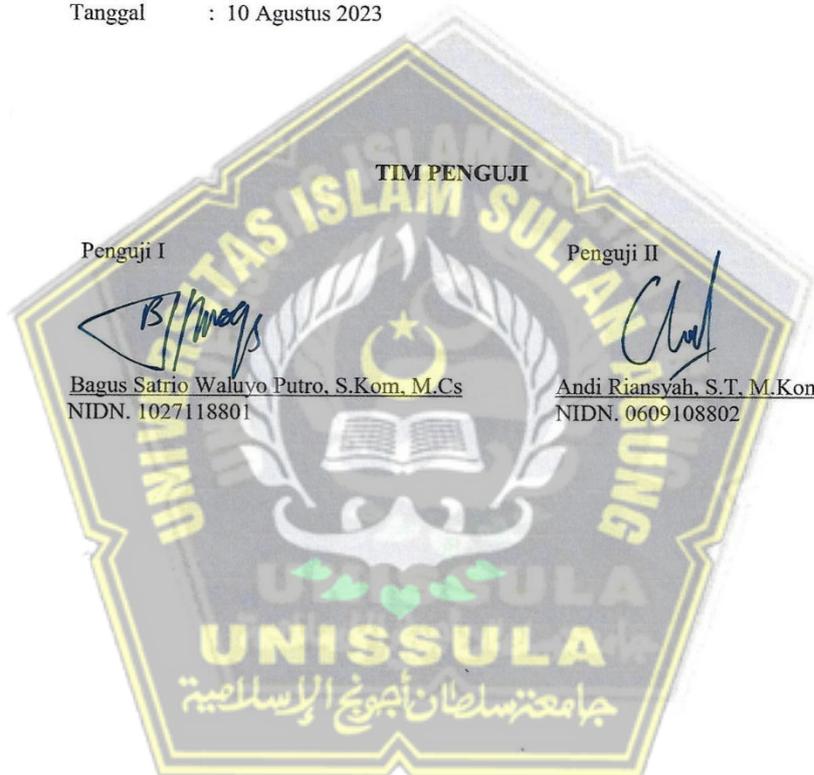
### TIM PENGUJI

Penguji I

  
Bagus Satrio Waluyo Putro, S.Kom, M.Cs  
NIDN. 1027118801

Penguji II

  
Andi Riansyah, S.T, M.Kom  
NIDN. 0609108802



## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Muhammad Faris Fahru Rozi

NIM : 32601900021

Judul Tugas Akhir : Deteksi Kanker Kulit Melanoma Menggunakan  
MobileNetV2 Convolutional Neural Network Berbasis  
Android

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 30 Agustus 2023

Yang Menyatakan,



Muhammad Faris Fahru Rozi

## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Faris Fahru Rozi

NIM : 32601900021

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Alamat Asal : Klitih, RT 03 RW 04, Desa Bolo, Kec Demak, Kab Demak

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **Deteksi Kanker Kulit Melanoma Menggunakan MobileNetV2 Convolutional Neural Network Berbasis Android.**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 30 Agustus 2023

Yang menyatakan,



Muhammad Faris Fahru Rozi

## KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Deteksi Kanker Kulit Melanoma Menggunakan *MobileNetV2 Convolutional Neural Network* Berbasis Android” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Bapak Ir. Sri Mulyono, M.Eng yang telah meluangkan waktu, memberi ilmu, nasehat dan saran.
4. Dosen pembimbing II penulis Bapak Ghufron, S.T., M.Kom yang telah meluangkan waktu, memberi ilmu, nasehat dan saran.
5. Orang tua penulis yang telah merestui penulis untuk menimba ilmu di kampus ini.
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis, menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan in dan masa mendatang.

Semarang, 02 Agustus 2023



Muhammad Faris Fahru Rozi

## DAFTAR ISI

<b>COVER .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN PEMBIMBING .....</b>	<b>Error! Bookmark not defined.</b>
<b>LEMBAR PENGESAHAN PENGUJI.....</b>	<b>Error! Bookmark not defined.</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR ....</b>	<b>Error! Bookmark not defined.</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....</b>	<b>Error! Bookmark not defined.</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>ABSTRAK .....</b>	<b>xiii</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah .....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II .....</b>	<b>5</b>
<b>TINJAUAN PUSTAKA DAN DASAR TEORI.....</b>	<b>5</b>
<b>2.1 Tinjauan Pustaka .....</b>	<b>5</b>
<b>2.2 Dasar Teori .....</b>	<b>8</b>
2.2.1 Kanker Kulit <i>Melanoma</i> .....	8
2.2.2 <i>Machine Learning</i> .....	9
2.2.3 <i>Deep Learning</i> .....	10
2.2.4 <i>Convolutional Neural Network</i> .....	10
2.2.5 <i>MobileNetV2</i> .....	13
2.2.6 <i>Tensorflow</i> .....	14

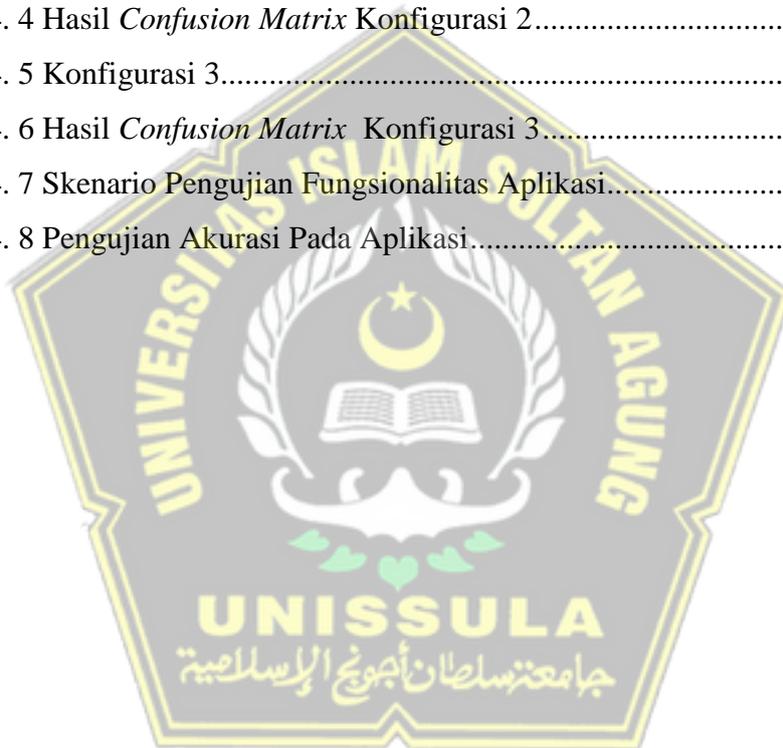
2.2.7	<i>Keras</i>	16
2.2.8	<i>Tensorflow Lite</i>	16
2.2.9	Android	17
2.2.10	<i>Room Database</i>	17
<b>BAB III.....</b>		<b>18</b>
<b>METODOLOGI PENELITIAN .....</b>		<b>18</b>
3.1	Metode Penelitian.....	18
3.1.1	<i>Data Collecting</i> .....	18
3.1.2	<i>Augmentasi Data</i> .....	18
3.1.3	Pelatihan Model .....	19
3.1.4	Evaluasi Model.....	21
3.1.5	Konversi Model.....	22
3.1.6	Pengembangan Aplikasi Android .....	22
3.1.7	Pengujian <i>Black Box</i> .....	22
3.2	Analisis Kebutuhan .....	23
3.3	Analisis Sistem .....	23
3.4	Perancangan Antarmuka.....	25
3.4.1	Halaman <i>Home</i> .....	25
3.4.2	Menu Tambah Gambar .....	26
3.4.3	Halaman Verifikasi Lanjutan .....	27
3.4.4	Halaman Hasil Deteksi Kanker Kulit.....	28
3.4.5	Halaman Riwayat Deteksi.....	29
3.4.6	Halaman Petunjuk Melakukan Deteksi.....	30
<b>BAB IV .....</b>		<b>31</b>
<b>HASIL DAN ANALISIS PENELITIAN.....</b>		<b>31</b>
4.1	Implementasi Antarmuka .....	31
4.2.1	Halaman Home.....	31
4.2.2	Menu Tambah Gambar .....	32
4.2.3	Halaman Verifikasi Lanjutan .....	33
4.2.4	Halaman Hasil Deteksi Kanker Kulit.....	34
4.2.5	Halaman Riwayat Deteksi.....	35
4.2.1	Halaman Petunjuk Melakukan Deteksi.....	36

4.2	Evaluasi Model.....	37
4.3	Pengujian Aplikasi .....	43
<b>BAB V.....</b>		<b>50</b>
<b>KESIMPULAN DAN SARAN .....</b>		<b>50</b>
5.1	Kesimpulan.....	50
5.2	Saran.....	50
<b>DAFTAR PUSTAKA .....</b>		<b>51</b>
<b>LAMPIRAN.....</b>		<b>54</b>



## DAFTAR TABEL

Tabel 2. 1 Tinjauan Pustaka .....	7
Tabel 3. 1 <i>Confusion Matrix</i> .....	21
Tabel 3. 2 Analisis Kebutuhan .....	23
Tabel 4. 1 Konfigurasi 1 .....	37
Tabel 4. 2 Hasil <i>Confusion Matrix</i> Konfigurasi 1 .....	38
Tabel 4. 3 Konfigurasi 2.....	39
Tabel 4. 4 Hasil <i>Confusion Matrix</i> Konfigurasi 2.....	40
Tabel 4. 5 Konfigurasi 3.....	41
Tabel 4. 6 Hasil <i>Confusion Matrix</i> Konfigurasi 3.....	42
Tabel 4. 7 Skenario Pengujian Fungsionalitas Aplikasi.....	43
Tabel 4. 8 Pengujian Akurasi Pada Aplikasi.....	44



## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur <i>CNN</i> .....	11
Gambar 2. 2 Proses Konvolusi.....	12
Gambar 2. 3 Proses <i>Max Pooling</i> .....	12
Gambar 2. 4 Struktur <i>Fully Connected Layer</i> .....	13
Gambar 2. 5 Arsitektur <i>MobileNetV2</i> .....	14
Gambar 3. 1 Sampel Citra <i>Melanoma</i> .....	18
Gambar 3. 2 Sampel Citra <i>Non-Melanoma</i> .....	18
Gambar 3. 3 Model <i>Summary MobileNetV2</i> .....	19
Gambar 3. 4 <i>Flowchart Sistem</i> .....	24
Gambar 3. 5 Desain Halaman <i>Home</i> .....	25
Gambar 3. 6 Desain Menu Tambah Gambar .....	26
Gambar 3. 8 Desain Menu Verifikasi Lanjutan .....	27
Gambar 3. 9 Desain Halaman Hasil Deteksi Kanker Kulit.....	28
Gambar 3. 10 Desain Halaman Riwayat Deteksi.....	29
Gambar 3. 11 Desain Halaman Petunjuk .....	30
Gambar 4. 4 Antarmuka Halaman <i>Home</i> .....	31
Gambar 4. 5 Antarmuka Menu Tambah Gambar.....	32
Gambar 4. 7 Antarmuka Halaman Verifikasi Lanjutan .....	33
Gambar 4. 8 Antarmuka Halaman Hasil Deteksi Kanker Kulit.....	34
Gambar 4. 9 Antarmuka Halaman Riwayat Deteksi.....	35
Gambar 4. 10 Antarmuka Petunjuk Melakukan Deteksi .....	36
Gambar 4. 1 <i>Confusion Matrix</i> Konfigurasi 1 .....	37
Gambar 4. 2 <i>Confusion Matrix</i> Konfigurasi 2 .....	39
Gambar 4. 3 <i>Confusion Matrix</i> Konfigurasi 3 .....	41

## ABSTRAK

*Melanoma* adalah jenis kanker kulit yang paling ganas dan berpotensi menyebabkan kematian jika tidak segera terdeteksi dan ditangani sejak dini. Oleh karena itu, pengembangan sistem untuk mendeteksi kanker kulit *melanoma* sangat penting dalam upaya deteksi dini penyakit tersebut. Pengembangan sistem deteksi penyakit kanker kulit *melanoma* menggunakan arsitektur *MobileNetV2 Convolutional Neural Network* yang direpresentasikan pada sistem berbasis android. Arsitektur *MobileNetV2* dipilih karena kemampuannya untuk mencapai akurasi tinggi dalam klasifikasi citra dengan ukuran model yang relatif kecil. Untuk mendapatkan performa model terbaik, dilakukan beberapa percobaan konfigurasi yang berbeda. Beberapa konfigurasi yang dieksplorasi meliputi penggunaan *optimizer SGD, Adadelta, dan RMSprop*, serta variasi jumlah *epoch* dalam setiap konfigurasi. Berdasarkan evaluasi model menggunakan *confusion matrix*, Performa model terbaik didapat pada konfigurasi 3 dengan jumlah *epoch* 20 dan menggunakan *optimizer SGD* dengan nilai *accuracy* sebesar 0.9309, *precision* sebesar 0.9805, *recall* sebesar 0.8792, dan *f1-score* sebesar 0.9270. Pengujian aplikasi menggunakan metode *black box testing* dengan menggunakan data pengujian dari *dataset* mendapatkan akurasi tertinggi mencapai 100% baik *melanoma* maupun *non-melanoma*.

Kata kunci: Kanker Kulit *Melanoma*, *MobileNetV2*, *Convolutional Neural Network*, *Confusion Matrix*, *Black Box Testing*.

## ABSTRACT

*Melanoma is the most violent and potentially fatal type of skin cancer if not detected and treated early. Therefore, the development of a system to detect melanoma skin cancer is crucial in the early detection of the disease. The development of the system for detecting melanoma skin cancer uses the MobileNetV2 Convolutional Neural Network architecture that is represented on an android-based system. The MobileNETV2 architecture was chosen because of its ability to high accuracy in image classification with relatively small model sizes. To get the best model performance, perform several different configuration experiments. Some of the configurations explored included the use of SGD optimizers, Adadelta, and RMSprop, as well as variations in the number of epochs in each configuration. Based on the model evaluation using a confusion matrix, the best model performance was achieved in configuration 3 with the epoch number of 20 and using SGD Optimizers with an accuracy value of 0.9309, precision of 0.9805, recall of 0.8792, and f1-score of 0.9270. Application testing using the black box method using test data from datasets achieved the highest accuracy of achieving 100% both melanoma and non-melanoma.*

*Keywords: Melanoma Skin Cancer, MobileNetV2, Convolutional Neural Network, Confusion Matrix, Black Box Testing.*

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Saat ini, teknologi telah digunakan secara luas dalam berbagai kegiatan sehari-hari manusia. Kegiatan manusia terasa lebih cepat dan mudah dengan memanfaatkan teknologi. Salah satu bidang yang telah terbantu oleh teknologi adalah bidang kesehatan. Teknologi pada bidang kesehatan telah memberikan dampak positif seperti memudahkan akses layanan kesehatan, meningkat kualitas hidup, membantu dalam diagnosis penyakit, mencegah penularan penyakit, dan masih banyak lagi dampak positif perkembangan teknologi pada bidang kesehatan.

Dari pemaparan singkat diatas, peneliti berencana untuk membuat sistem yang dapat membantu meningkatkan kualitas layanan kesehatan. Sistem yang akan dikembangkan adalah sebuah sistem yang bisa mendeteksi penyakit kanker kulit *melanoma*. Diharapkan bahwa sistem ini akan membantu para dokter dan masyarakat mendeteksi kanker kulit *melanoma* pada tahap awal. Sistem ini akan dikembangkan untuk platform android agar mudah diakses oleh berbagai kalangan mulai dari tenaga medis sampai masyarakat umum hanya dengan menggunakan perangkat android.

Dalam tugas akhir ini, peneliti akan berfokus pada studi kasus deteksi penyakit kanker kulit *melanoma*. Kanker kulit *melanoma* merupakan salah satu jenis kanker kulit yang jarang terjadi meskipun begitu jenis kanker ini termasuk yang paling mematikan dan berpotensi menyebabkan kematian. Orang dengan banyak tahi lalat atau memiliki keluarga yang pernah menderita kanker kulit melanoma memiliki risiko lebih tinggi terjangkit *melanoma maligna*. *Melanoma* bisa dimulai dari tahi lalat baru yang muncul atau perubahan pada tahi lalat yang sudah ada. Pertumbuhan dan perkembangan melanoma terjadi karena kerusakan DNA pada sel-sel kulit yang umumnya disebabkan oleh paparan radiasi *ultraviolet* dari sinar matahari. Ketika seseorang mengalami melanoma, biasanya area tubuh seperti wajah, tangan, punggung, dan kaki menjadi lokasi kanker yang umum. Namun, perlu dicatat bahwa melanoma juga bisa muncul di bagian tubuh yang

jarang terpapar sinar matahari, seperti telapak kaki dan telapak tangan. (Ritonga, 2019).

Data dari RSUP Sanglah Denpasar pada tahun 2017 menunjukkan bahwa terdapat 564 kasus kanker kulit. Dari jumlah tersebut, 25,7% atau sekitar 145 kasus adalah kasus *melanoma*, sementara sisanya, yaitu 74,3% atau sekitar 419 kasus, merupakan kasus *non-melanoma*. Pada tahun 2018, jumlah kasus kanker kulit mencapai 425 kasus. Dari jumlah ini, sekitar 30,2% atau sekitar 129 kasus adalah kasus *melanoma*, dan sekitar 69,8% atau sekitar 296 kasus adalah kasus *non-melanoma*. Dengan demikian, data tersebut menunjukkan tren peningkatan jumlah kasus kanker kulit pada tahun 2018 dibandingkan dengan tahun 2017. Selain itu, proporsi kasus *melanoma* juga mengalami sedikit peningkatan dari tahun sebelumnya (Yogiswara dkk., 2021). Meskipun demikian, penting untuk tidak mengabaikan jenis kanker ini karena memiliki potensi kematian yang tinggi bagi penderitanya. Oleh karena itu, penelitian terus dilakukan untuk mengembangkan metode deteksi *melanoma* yang lebih baik guna membantu dalam mengidentifikasi kasus lebih awal dan meningkatkan peluang kesembuhan. Oleh karena itu, peneliti bertujuan untuk mengembangkan sebuah sistem yang mampu mendeteksi dini penyakit kanker kulit *melanoma* menggunakan arsitektur *MobileNetV2 Convolutional Neural Network*. Nantinya diharapkan sistem dapat membantu tenaga medis dan masyarakat dalam deteksi dini potensi mengidap kanker kulit *melanoma*, sehingga penanganan penyakit dapat segera dilakukan.

## 1.2 Perumusan Masalah

Bagaimana *algoritma Convolutional Neural Network* yang diimplementasikan dalam sistem berbasis android mampu melakukan klasifikasi citra kulit yang memiliki probabilitas terjangkit kanker kulit *melanoma*.

### 1.3 Pembatasan Masalah

Berikut adalah beberapa batasan masalah yang diberlakukan dalam pembuatan tugas akhir:

1. Dataset berasal dari website *kaggle* yaitu <https://www.kaggle.com/datasets/drscarlat/melanoma>. Pada dataset tersebut terdapat 2 kategori kanker kulit yaitu *melanoma* dan *non-melanoma*.
2. Jumlah data yang digunakan sebanyak 17.805 citra yaitu 10.682 citra pelatihan, 3.562 citra validasi, 3.561 citra pengujian.
3. Data uji hanya berasal dari dataset dan pengujian pada aplikasi hanya menggunakan *gallery*.
4. Metode yang digunakan adalah *Convolutional Neural Network* dengan arsitektur *MobileNetV2*.
5. Sistem dikembangkan hanya pada *platform android*.
6. Luaran dari sistem yang dibuat adalah menampilkan hasil deteksi kanker kulit dari probabilitas terjangkit kanker kulit *melanoma*.

### 1.4 Tujuan

Tujuan tugas akhir ini adalah mengembangkan sebuah sistem untuk mendeteksi penyakit kanker kulit *melanoma* berbasis android dengan menggunakan arsitektur *MobileNetV2 Convolutional Neural Network*.

### 1.5 Manfaat

Berikut manfaat dari penelitian tugas akhir ini:

1. Membantu pasien dalam mendeteksi dini penyakit kanker kulit *melanoma*.
2. Memudahkan pasien dalam mengakses layanan kesehatan khususnya dalam deteksi penyakit kanker kulit *melanoma*.

## 1.6 Sistematika Penulisan

Berikut adalah sistematika penelitian dalam pembuatan laporan tugas akhir:

### BAB 1 : PENDAHULUAN

Bab 1 berfokus pada memberikan latar belakang yang menjelaskan urgensi dari penelitian, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penelitian yang akan diikuti dalam tugas akhir ini.

### BAB 2 : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab 2 akan menyajikan tinjauan pustaka dan dasar teori yang merujuk pada penelitian-penelitian sebelumnya sebagai referensi utama dalam penelitian ini. Arsitektur *MobileNetV2 Convolutional Neural Network* akan digunakan dalam penelitian untuk menganalisis dan mengolah

### BAB 3 : METODE PENELITIAN

Bab 3 akan menyajikan pemaparan tentang tahapan metode dalam proses penelitian, dimulai dari perancangan sistem hingga proses klasifikasi data. Rincian mengenai setiap tahap, termasuk pengumpulan data, akan dijelaskan secara mendetail dalam bab ini.

### BAB 4 : HASIL DAN ANALISIS PENELITIAN

Bab 4 akan menyajikan pemaparan mengenai hasil dari proses penelitian, termasuk hasil klasifikasi menggunakan arsitektur *MobileNetV2 Convolutional Neural Network*, beserta hasil dari pengujian yang telah dilakukan. Informasi mengenai hasil klasifikasi dan evaluasi performa dari metode *CNN* akan disajikan secara mendalam dalam bab ini.

### BAB 5 : KESIMPULAN DAN SARAN

Bab 5 akan berisi rangkuman kesimpulan serta saran dari seluruh proses penelitian yang telah dilakukan.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Metode *Convolutional Neural Network (CNN)* telah terbukti efektif dan akurat dalam klasifikasi citra gambar dan sering digunakan dalam penelitian sebelumnya. Seperti penelitian yang dilakukan (Riva dkk., 2022) mengenai klasifikasi tulisan tangan dengan membandingkan metode *CNN* dan *ANN* dengan *Projection Histogram*. Hasil percobaan menggunakan *CNN* menunjukkan nilai akurasi pada data validasi adalah 0,9752 dan pada data pengujian adalah 0,9689. Sementara itu, metode *projection histogram + ANN* memperoleh nilai akurasi pada data validasi 0,8904 dan pada data pengujian 0,8962. Dari hasil pengujian ini, terlihat bahwa *CNN* memiliki akurasi yang lebih tinggi dibandingkan dengan metode *projection histogram + ANN*.

Pada penelitian yang dilakukan (Naufal, 2021) mengenai klasifikasi citra cuaca yang membandingkan metode *CNN*, *SVM*, dan *KNN*. Dalam algoritma *KNN*, peneliti memilih metode perhitungan jarak *Euclidean* dan memilih 5 tetangga karena dianggap menghasilkan kinerja paling optimal dibandingkan dengan pengaturan lainnya. Sementara itu, pada *SVM*, peneliti memutuskan untuk menggunakan kernel *RBF* karena dinilai memberikan kinerja terbaik dibandingkan dengan opsi kernel lainnya. Dari hasil perbandingan dengan matriks *accuracy*, *precision*, *recall* dan *f1-score*, disimpulkan bahwa algoritma *CNN* memiliki nilai rata-rata performa yang terbaik dengan nilai sebesar 0,9420, *KNN 5 euclidean* mendapat nilai sebesar 0,7660 dan *SVM RBF* mendapat nilai sebesar 0,8600. pada penelitian ini tidak hanya dilakukan pengujian performa tapi juga perbandingan pada waktu pelatihan dan pengujian. Dari hasil pengamatan waktu eksekusi *CNN* memiliki waktu paling lama dibandingkan *SVM* dan *KNN*. Meskipun begitu, *CNN* menghasilkan performa yang jauh lebih unggul dibandingkan dengan *SVM* dan *KNN*, dengan perbedaan yang cukup signifikan.

Pada penelitian oleh (Prasetyo dkk., 2023) mengenai analisa perbandingan algoritma *SVM* dan *CNN* untuk sistem deteksi katarak. Dari hasil penelitian yang

dilakukan didapatkan hasil dari penggunaan *algoritma SVM* dengan menggunakan rasio data 80:20, menggunakan *linear kernel*, menggunakan parameter *C (Cost)* berjumlah 1 dan *gamma* sebesar 0.001 mendapatkan hasil *accuracy* sebesar 96,67%, *precision* 96,84%, *recall* 96,67% dan *f1-score* 96,62%. Hasil performa terbaik berdasarkan seluruh skenario pengujian didapat dengan menggunakan *algoritma CNN* dengan menggunakan rasio data 80:20, menggunakan *optimizer Adam*, dengan *learning rate* 0.001 dan jumlah *epoch* 25. Hasil pengujian yang didapat adalah *accuracy* sebesar 98,89%, *precision* sebesar 98,92%, *recall* sebesar 98,92% dan *f1-score* sebesar 98,89%.

Pada penelitian oleh (Zaelani & Miftahuddin, 2022) mengenai perbandingan *algoritma EfficientNetB3* dan *MobileNetV2* yang digunakan untuk identifikasi jenis-jenis buah menggunakan fitur daun. Penelitian tersebut menggunakan *algoritma Convolutional Neural Network* yang membandingkan dua model yaitu *EfficientNetB3* dan *MobileNetV2*. Selama proses pelatihan, dilakukan percobaan modifikasi parameter telah diterapkan pada model untuk mencapai akurasi optimal. Kedua metode yang diuji merupakan *pre-trained model CNN* yang telah sebelumnya dilatih dengan dataset yang cukup besar, yaitu *ImageNet*. Beberapa parameter juga telah dicoba untuk mendapat hasil terbaik, variasi jumlah *epoch*, variasi *optimizer* dan *bathsize*. Pada model *EfficientNetB3* dengan 20 *epoch*, dengan *optimizer SGD* menghasilkan akurasi sebesar 0,2370, sementara model *EfficientNetB3* dengan 50 *epoch* dan *optimizer Adamax* mencapai akurasi sebesar 0,3051. kemudian model *MobileNetV2* dengan *epoch* 20 dan *optimizer Adam*, akurasi yang tercapai adalah 0,9914, dan pada 50 *epoch* dengan *optimizer Adamax*, akurasi mencapai 0,9860.

Pada penelitian oleh (Winnarto dkk., 2022) mengenai klasifikasi jenis tumor otak dengan menggunakan arsitektur *MobileNetV2*. Penelitian tersebut melalui beberapa tahapan, diantaranya yaitu *data collecting*, *preprocessing data*, pelatihan, evaluasi model, dan implementasi. Dalam penelitian ini, model *MobileNetV2* berhasil melakukan klasifikasi jenis tumor otak dengan performa yang baik, terbukti dengan akurasi mencapai 88,64% dan nilai loss sebesar 0,3424. Hasil dari *confusion matrix* juga menunjukkan bahwa lebih banyak citra telah diklasifikasikan

dengan benar daripada yang salah, dan dari 32 citra sampel yang diamati, 38 citra berhasil terdeteksi dengan hasil yang sesuai atau benar.

Berdasarkan beberapa penelitian diatas dapat ditarik kesimpulan bahwa penggunaan *MobileNetV2 Convolutional Neural Network* merupakan *algoritma* yang paling tepat dalam penelitian tugas akhir deteksi kanker kulit *melanoma*. *CNN* menunjukkan performa yang baik pada klasifikasi citra jika dibandingkan dengan metode lainnya seperti *SVM*, *KNN* dan *ANN*. Dimana *CNN* rata-rata memiliki tingkat akurasi diatas 90%. Tingkat akurasi dapat terus ditingkat dengan memepertimbangkan beberapa hal diantaranya jumlah dan kualitas citra, pemilihan *pre-trained model*, pemilihan *optimizer*, jumlah *epoch*, besar dari *learning rate* dan masih banyak parameter lainnya.

Berikut beberapa rujukan lain yang digunakan oleh peneliti dalam proses penyusunan tugas akhir:

Tabel 2. 1 Tinjauan Pustaka Penelitian

No	Judul	Pembahasan
1	Implementasi Model Arsitektur <i>VGG16</i> dan <i>MobileNetV2</i> Untuk Klasifikasi Citra Kupu-Kupu (Ramayanti dkk., 2022).	Tujuan dari penelitian ini adalah membandingkan performa model <i>VGG16</i> dengan <i>MobileNetV2</i> menggunakan 4955 citra kupu-kupu yang diberi label 50 spesies. Hasil penelitian menunjukkan bahwa <i>MobileNetV2</i> tanpa <i>fine tuning</i> mencapai akurasi terbaik, yaitu sebesar 96%. Selain itu, <i>MobileNetV2</i> menghasilkan nilai <i>precision</i> , <i>recall</i> , <i>f1-score</i> , dan <i>cohens kappa</i> yang lebih tinggi, menunjukkan bahwa model tersebut memiliki keseimbangan yang lebih baik dalam akurasi pada setiap kelasnya.
2	<i>Comparative Study of VGG16 and MobileNetV2 for Masked Face</i>	Pada penelitian menggunakan <i>pre-trained model transfer learning</i> untuk mengenali wajah yang mengenakan masker. Penelitian ini menganalisis dua model yaitu <i>VGG16</i> dan <i>MobileNetV2</i> . Model terbaik diperoleh dengan ukuran <i>batch 32</i> dan

	<i>Recognition</i> (Adhinata dkk., 2021).	jumlah <i>epoch</i> 50 pada setiap model. Hasil yang didapatkan menunjukkan model <i>MobileNetV2</i> lebih akurat dibandingkan <i>VGG16</i> , dengan nilai akurasi sebesar 95,42%.
3	Deteksi Penyakit pada Tanaman Padi Menggunakan <i>MobileNet Transfer Learning</i> Berbasis Android (Kahfi dkk., 2022).	Pada penelitian ini memiliki tujuan untuk mengembangkan sebuah aplikasi yang mampu mendeteksi penyakit yang menyerang daun padi. Penelitian ini menganalisa performa dari dua model yaitu <i>MobileNetV1</i> dan <i>MobileNetV2</i> . Pada <i>MobileNetV1</i> dengan menggunakan <i>optimizer SGD</i> mendapat akurasi sebesar 76% dan <i>optimizer Adam</i> mendapat akurasi sebesar 94%. Pada <i>MobileNetV2</i> dengan menggunakan <i>optimizer SGD</i> mendapat akurasi sebesar 74% dan <i>optimizer Adam</i> sebesar 95%. Sehingga dapat disimpulkan model <i>MobileNetV2</i> dengan <i>fine tuning</i> memiliki performa yang lebih baik dalam deteksi penyakit yang menyerang daun padi.

## 2.2 Dasar Teori

### 2.2.1 Kanker Kulit *Melanoma*

Kanker kulit adalah jenis kanker yang sering didiagnosis di seluruh dunia, terutama pada populasi dengan kulit berwarna putih. Angka kematian akibat kanker kulit terus meningkat dalam dekade terakhir. Di Indonesia, kanker kulit berada di posisi ketiga dalam urutan kasus terbanyak setelah kanker payudara dan kanker rahim. Setiap tahun, kanker kulit menyumbang sekitar 5,9% hingga 7,8% dari total kasus kanker. Di Indonesia, kasus kanker kulit paling umum adalah *karsinoma sel basal* dengan persentase sebesar 65,5%, kemudian *karsinoma sel skuamosa* sebesar 23%, *melanoma maligna* sebesar 7,9%, dan jenis kanker kulit lainnya. *Melanoma* merupakan jenis kanker kulit dengan angka kematian tertinggi, terutama jika tidak terdeteksi pada tahap awal. Sementara itu, kanker kulit jenis *non-melanoma* seperti

*sel skuamosa* dan *sel basal* lebih umum, namun memiliki tingkat *metastasis* yang lebih rendah, sehingga hanya sebagian kecil dari kasus tersebut yang berakhir dengan kematian (Wilvestra dkk., 2018).

*Melanoma* adalah satu dari tiga kanker kulit yang memiliki angka kematian rendah, namun menyumbang sekitar 75% dari total kasus kematian akibat kanker kulit. *Melanoma* berasal dari *melanosit* dan ditandai dengan munculnya lesi berwarna hitam pada kulit. Sangat penting untuk mendeteksi keganasan sel melanosit sejak dini agar diagnosis dapat ditingkatkan, terutama pada lesi baru yang berpigmen atau perubahan pada tahi lalat seperti perubahan warna, permukaan, konsistensi, atau ukuran, serta munculnya gejala seperti panas, rasa terbakar, dan sakit. Sangat penting untuk mendapatkan perawatan medis segera jika lesi yang sebelumnya datar menjadi lebih buruk atau lesi satelit ditemukan di sekitarnya. (Ariesta dkk., 2018).

Paparan matahari, kondisi kongenital, trauma seperti luka bakar, paparan bahan kimia, riwayat lesi kulit berulang, tipe kulit, usia, dan jenis kelamin adalah beberapa faktor risiko yang dapat menyebabkan penyakit kanker kulit. Data dari rekam medik Rumah Sakit Umum Pusat (RSUP) Dr. M. Djamil Padang menunjukkan bahwa selama periode tahun 2015–2022, 114 kasus kanker kulit tercatat, termasuk *karsinoma sel basal*, *karsinoma sel skuamosa*, dan *melanoma maligna*. Paparan sinar matahari yang memiliki sifat *karsinogenik* merupakan salah satu faktor risiko utama kanker kulit. Karena lokasinya di peta, orang Indonesia sering terpapar sinar *ultraviolet*. Karena mayoritas orang Indonesia sering beraktivitas di bawah sinar matahari langsung, mereka lebih rentan terhadap kanker kulit karena paparan sinar matahari. (Tarisa dkk., 2022).

### **2.2.2 Machine Learning**

*Machine Learning* adalah salah satu cabang ilmu dari *artificial intelligence* yang bisa belajar sendiri tanpa harus diprogram terus menerus. Melalui pembelajaran tersebut mesin mampu memiliki perilaku layaknya manusia seperti mampu beradaptasi dengan hal baru dan mengambil keputusan.

Untuk menganalisis kumpulan data besar dan menemukan pola tertentu, *machine learning* membutuhkan data. *Supervised learning* dan *unsupervised*

*learning* adalah dua teknik utama pengajaran mesin yang paling umum. (Jeremia dkk., 2021).

Dari pendekatan *Machine Learning*(*pre-training*) menghasilkan sebuah *pre-trained model*. *Pre-trained model* menyediakan fondasi pengetahuan dan fitur yang bisa digunakan untuk tugas lanjutan, menghemat waktu dan sumber daya serta membantu mengatasi masalah pembelajaran mesin. *Pre-trained model* ini digunakan untuk mengklasifikasikan kanker kulit *melanoma* atau *non-melanoma*

### **2.2.3 Deep Learning**

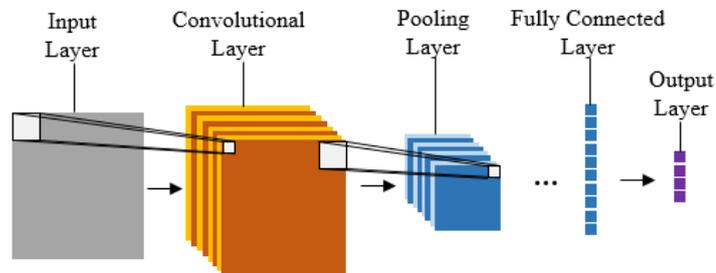
Salah satu cabang *machine learning* adalah *deep learning*, yang terinspirasi dari struktur otak manusia. Mirip dengan *neuron-neuron* yang saling terkoneksi membentuk jaringan saraf yang kompleks. Dengan menggunakan fungsi transformasi *non-linear* berlapis-lapis, *deep learning* menggunakan model pemodelan abstraksi tingkat tinggi pada data. Jumlah dan kompleksitas jaringan saraf dalam *deep learning* bisa mencapai ratusan lapisan. *Algoritma* ini memiliki kemampuan untuk secara otomatis mengekstraksi fitur dari data. Dengan demikian, *Deep Learning* sangat penting untuk kecerdasan buatan karena dapat membantu pemrograman mengurangi beban pemilihan fitur secara eksplisit. (Lorentius dkk., 2020).

Pada penelitian ini, memanfaatkan salah satu arsitektur dari *deep learning* yaitu *MobileNetV2* yang digunakan untuk tugas pengolahan gambar kanker kulit. *MobileNetV2* menggunakan prinsip-prinsip penting dari *deep learning* untuk menyelesaikan masalah dalam bidang *computer vision*.

### **2.2.4 Convolutional Neural Network**

Tidak jauh berbeda dengan jenis *neural network* lainnya, *Convolutional Neural Network (CNN)* sangat populer digunakan untuk mendeteksi dan mengenali objek pada data gambar. Pada dasarnya, *CNN* terdiri dari *neuron-neuron* yang memiliki fungsi aktivasi, bobot, dan bias. *CNN* sering digunakan untuk memproses data gambar karena memiliki banyak lapisan dan termasuk dalam kategori *Deep Neural Networks*. *CNN* menggunakan dua pendekatan: klasifikasi dengan *feedforward* dan pembelajaran dengan *algoritma backpropagation*. (Peryanto dkk., 2020). Manfaat metode *Convolutional Neural Network (CNN)* akan lebih terlihat

ketika digunakan untuk klasifikasi gambar yang memiliki jumlah data yang lebih besar. Karena *CNN* memiliki lebih banyak kelas daripada metode klasifikasi lainnya, manfaatnya akan lebih terlihat.

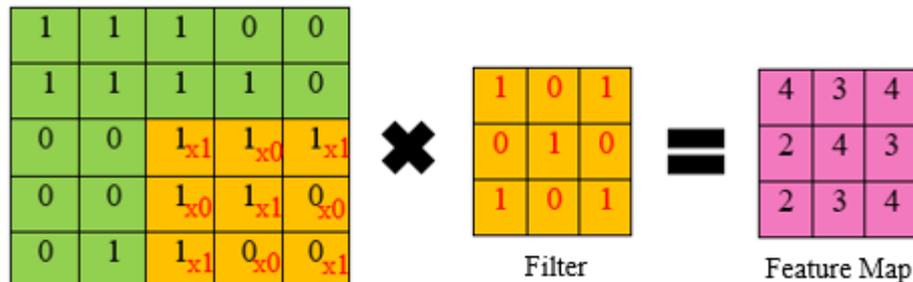


Gambar 2. 1 Arsitektur *CNN*

Dua komponen utama dari struktur *Convolutional Neural Network (CNN)* adalah *fully-connected layer* dan *feature extraction layer*. Pada *convolutional layer* dan *pooling layer* terdapat *feature extraction layer*. *Convolutional layer* menggunakan konsep *weight sharing* untuk mengurangi kompleksitas perhitungan dan menggunakan *sliding window* untuk mendeteksi fitur-fitur penting pada data gambar. Sementara itu, *pooling layer* berfungsi untuk merangkum informasi dari hasil konvolusi, sehingga mengurangi dimensi data. Selanjutnya, hasil vektor dari operasi konvolusi dan *pooling* pada *feature extraction layer* akan dihubungkan ke *fully-connected layer*, yang merupakan bagian dari *multilayer perceptron* yang digunakan untuk melakukan klasifikasi data.

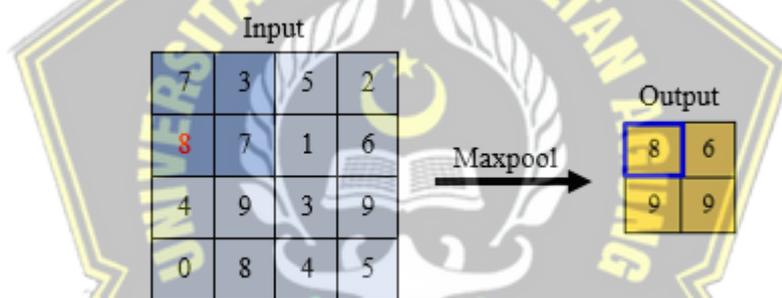
*Convolutional layer* adalah lapisan utama dan paling penting pada *CNN* yang bertanggung jawab untuk operasi konvolusi. Lapisan ini menghasilkan citra baru yang menampilkan fitur dari citra *input*. Untuk memproses citra yang dimasukkan, *convolution layer* menggunakan filter-filter yang dipelajari secara acak. *Convolution layer* terdiri dari banyak *neuron* yang membentuk sebuah filter dengan panjang, tinggi, dan kedalaman tertentu. Setiap filter kemudian digeser dan dioperasikan menggunakan operasi “dot” antara data masukan dengan nilai-nilai pada filter. (Nugroho dkk., 2020). *Convolution layer* berperan penting dalam jaringan saraf *CNN* untuk mendeteksi objek berdasarkan atribut yang dimiliki. Proses konvolusi melibatkan perkalian matriks antara filter dengan area citra yang bersesuaian. Perhitungan ini dapat diilustrasikan pada Gambar 2.2, di mana

dilakukan perkalian piksel gambar dengan piksel pada filter untuk menghasilkan output yang memetakan fitur-fitur tertentu dari citra masukan.



Gambar 2. 2 Proses *Convolution*

Setelah proses konvolusi, langkah selanjutnya dalam *Convolutional Neural Network* adalah proses *max pooling*. *Pooling* merupakan teknik untuk menghasilkan gambar dengan piksel yang lebih kecil, tetapi tetap mempertahankan informasi penting dari gambar sebelumnya.

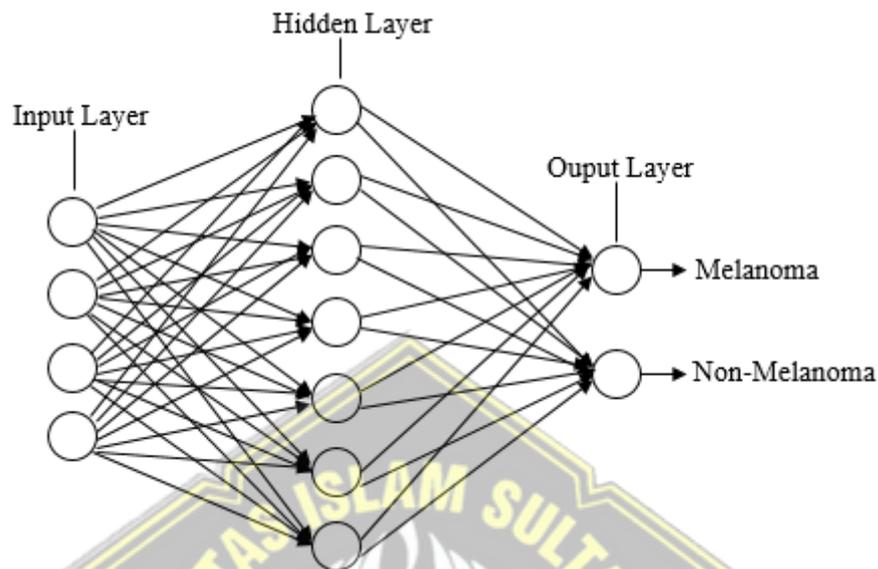


Gambar 2. 3 *Max Pooling*

Gambar 2.3 merupakan proses *max pooling* yang dilakukan pada *pooling layer*. Proses ini digunakan untuk mengurangi dimensi data dari suatu citra dan dapat mempercepat proses klasifikasi. Terdapat beberapa jenis *pooling* yang dapat digunakan dalam *Convolutional Neural Network*. Namun, *max pooling* dan *average pooling* adalah yang paling umum. *Max pooling* menyimpan nilai terbesar dari setiap matriks *pooling*, sementara *average pooling* mengambil nilai rata-rata dari area *pooling kernel*. Biasanya, setiap model *CNN* hanya menggunakan satu jenis *pooling* untuk memproses data gambar (Rozaqi dkk., 2021).

Langkah berikutnya adalah proses *fully connected layer*. Seluruh *neuron* dari layer *convolutional layer* dan *pooling layer* dihubungkan ke setiap *neuron* tunggal yang ada pada *fully connected layer* (Yohannes dkk., 2020). Proses ini biasa disebut dengan "*flatten*". Hasil akhir dari *fully connected layer* adalah *output* yang berupa

label kelas *melanoma* dan *non-melanoma*. Gambar 2.4 merupakan struktur lapisan *fully connected layer*.

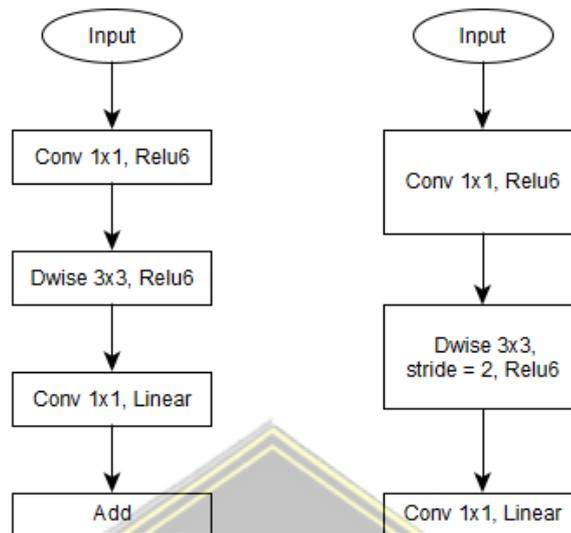


Gambar 2. 4 Struktur *Fully Connected Layer*

### 2.2.5 *MobileNetV2*

*MobileNetV2* merupakan sebuah model jaringan yang merupakan hasil pengembangan dari model sebelumnya yaitu *MobileNets*. Tujuan dari pengembangan model ini adalah untuk menciptakan model *computer vision* yang dapat diadaptasi secara optimal pada perangkat *mobile*, yang merupakan sebuah tantangan dalam *State of The Art*. Dibandingkan dengan model sebelumnya, *MobileNetV2* berhasil mengurangi secara signifikan jumlah operasi dan memori yang dibutuhkan tanpa mengorbankan tingkat akurasi yang dihasilkan. Arsitektur *MobileNetV2* dan arsitektur *CNN* lainnya memiliki perbedaan dalam penggunaan *convolutional layer*. *Convolutional layer* pada *MobileNetV2* menggunakan filter dengan ketebalan yang sesuai dengan ketebalan citra masukan. Arsitektur *MobileNetV2* mengimplementasikan beberapa teknik konvolusi, termasuk *linear bottleneck*, *depthwise convolutional*, dan *pointwise convolution*. (Hariman dkk., 2023).

*MobileNetV2* telah dilatih pada *dataset* yang besar, seperti *dataset ImageNet*, sebagai dasar untuk mendeteksi kanker kulit melanoma. Gambar 2.5 merupakan gambaran arsitektur *MobileNetV2*.



Gambar 2. 5 Arsitektur *MobileNetV2*

### 2.2.6 Tensorflow

*Tensorflow* adalah *library machine learning open source* yang digunakan untuk *dataflow programming*. *Library* tersebut dikembangkan oleh *Google Brain Team*. *Tensorflow* dirilis pada 11 februari 2017. Salah satu keunggulan utama dari *Tensorflow* kemampuannya untuk dijalankan pada berbagai jenis *CPU* dan *GPU*. *TensorFlow* bisa dijalankan di banyak sistem operasi seperti *Windows*, *Linux 64 bit*, *macOS*, serta pada platform komputasi seluler seperti *Android* dan *iOS*. (Lorentius dkk., 2020).

*Tensorflow* memiliki beberapa fungsi selama proses pelatihan model *CNN* pada penelitian ini, diantaranya:

1. *Convolution 2D*

Lapisan *convolution 2D* digunakan untuk mengekstraksi fitur dari gambar. Dapat juga menentukan jumlah filter, ukuran filter, dan fungsi aktivasi.

2. *Max Pooling 2D*

Lapisan ini digunakan untuk *max pooling*, yaitu mengurangi dimensi spasial dari fitur-fitur yang diperoleh dari lapisan konvolusi.

3. *Dense*

Lapisan ini digunakan untuk menghubungkan fitur-fitur yang diperoleh dari lapisan sebelumnya ke lapisan *output*.

4. *Flatten*

Lapisan ini digunakan untuk meratakan data, sehingga dapat dihubungkan ke lapisan *dense*.

5. *Dropout*

Lapisan ini digunakan untuk mengurangi *overfitting* secara acak “menonaktifkan” beberapa unit selama pelatihan.

6. *Optimizer*

*Tensorflow* menyediakan berbagai optimasi yang bisa digunakan untuk mengoptimalkan fungsi kesalahan selama proses pelatihan, seperti *SGD*, *Adadelta*, *RMSprop*

7. *Model*

Kelas ‘Model’ merupakan definisi dari model *CNN*. Kelas tersebut menggabungkan berbagai lapisan menjadi satu model lengkap.

8. *Metrics*

*Metrics* menyediakan berbagai matrik evaluasi yang bisa digunakan selama dan setelah proses pelatihan, seperti *accuracy*, *precision*, *recall*, dan *f1-score*.

a. *Accuracy*

*Accuracy* digunakan untuk mengukur sejauh mana model dapat melakukan klasifikasi dengan benar.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

b. *Precision*

*Precision* digunakan untuk mengukur perbandingan antara jumlah data yang diprediksi benar sebagai positif dengan total jumlah data yang diprediksi oleh sistem sebagai positif.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

c. *Recall*

*Recall* digunakan untuk mengukur perbandingan antara jumlah data yang diprediksi benar sebagai positif dengan total keseluruhan data aktual yang sebenarnya positif.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

d. *F1-Score*

*F1-Score* digunakan untuk mengukur nilai rata-rata dari perbandingan antara *precision* dan *recall*.

$$F1-Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

### 2.2.7 Keras

*Keras* merupakan sebuah *API neural network* tingkat tinggi yang bisa berjalan di berbagai *framework machine learning* seperti *Tensorflow*, *CNTK*. *Keras* dikembangkan dengan menggunakan bahasa *Python*. *Keras Applications* merupakan model *deep learning* dengan pra-pelatihan *weights*. Model ini dapat digunakan untuk melakukan prediksi, ekstraksi fitur, atau *fine-tuning*. (Pangestu & Bunyamin, 2018).

Sejak *Tensorflow 2.0*, *library Keras* menjadi bagian *integral* dari *Tensorflow* yang digunakan sebagai antarmuka tingkat tinggi untuk membangun dan melatih model *CNN* untuk tugas klasifikasi kanker kulit *melanoma* dan *non-melanoma*.

### 2.2.8 Tensorflow Lite

*Tensorflow* adalah sebuah *library open source* yang biasanya digunakan untuk *dataflow programming*. *Tensorflow* dikembangkan oleh *Google Brain Team* yang dirilis pada Februari 2017. *Tensorflow* juga digunakan untuk *research* oleh Google sendiri. Kemudian pada bulan Mei 2017 Google merilis versi ringan dari *Tensorflow* yang diberi nama *Tensorflow lite* yang mampu di eksekusi lebih cepat pada perangkat *mobile*. Pada *Tensorflow lite* terdapat *model maker* yaitu *software stack* yang digunakan untuk mengubah model menjadi format *FlatBuffer* supaya bisa dieksekusi oleh *Tensorflow Lite Interpreter* (Permana & Budayawan, 2020).

Format *Tensorflow Lite* merupakan format yang dioptimalkan untuk penerapan pada perangkat *mobile* atau *environment* yang terbatas sumber daya. Format tersebut berguna untuk mengintegrasikan model *CNN* yang telah dilatih sebelumnya dengan aplikasi deteksi kanker kulit *melanoma* berbasis android.

### 2.2.9 Android

Android adalah sebuah sistem operasi *open source* berbasis *Linux*, sehingga memungkinkan para developer untuk membuat aplikasi sesuai dengan keinginan mereka sendiri. Awalnya, platform ini didirikan oleh Android Inc, tetapi selanjutnya diakuisisi oleh Google tahun 2007 dan merilis *Android Open source Project (AOSP)*. *OHA (Open Handset Alliance)* kemudian mendukung pengembangan dan distribusi sistem operasi android. Android adalah *platform* yang relatif baru, namun proses pengembangannya sangat cepat. Setiap versi sistem operasi android diberi nama berdasarkan urutan huruf alfabet (Budiawan dkk., 2023).

Aksebilitas *platform* android menjadi alasan peneliti dalam memilih platform ini untuk mengembangkan sistem deteksi kanker kulit *melanoma*. Android tersedia berbagai kisaran harga, menjadikannya lebih mudah diakses oleh berbagai lapisan masyarakat. Ini penting untuk menjangkau sebanyak mungkin pengguna yang membutuhkan deteksi kanker kulit *melanoma*.

### 2.2.10 Room Database

*Room Database* berasal dari pengembangan *SQLite* yang merupakan lapisan abstraksi di atasnya yang memungkinkan akses *database* lebih mudah dan lancar. *Room* memungkinkan penyimpanan *database* secara lokal. Hal tersebut sangat menguntungkan untuk mengani jumlah data terstruktur yang besar dengan mempertahankannya secara lokal. salah satu kegunaan umum dari *Room* adalah menyimpan data yang relevan agar pengguna dapat mengakses konten tersebut ketika *offline* saat perangkat tidak terhubung dengan jaringan. Setiap perubahan yang dilakukan oleh pengguna akan disimpan secara lokal dan kemudian disinkronkan dengan *server* ketika perangkat terhubung kembali ke jaringan. *Room database* pada aplikasi deteksi kanker kulit *melanoma* digunakan untuk menyimpan riwayat deteksi yang dilakukan oleh pengguna. (Tandiari dkk., 2019).

## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

#### 3.1.1 Data Collecting

Tahap *data collecting* merupakan tahapan pengumpulan data yang untuk penelitian dalam deteksi penyakit kanker kulit *melanoma*. Dalam penelitian ini, menggunakan dataset berupa gambar kanker kulit yang telah diklasifikasikan ke dalam 2 kategori utama, yaitu *melanoma* dan *non-melanoma*. *Dataset* terdiri dari 17.805 citra kulit yang telah dikelompokkan menjadi dua kelas. *Dataset* berasal dari *kaggle* yaitu <https://www.kaggle.com/datasets/drscarlat/melanoma>. Penelitian ini, menggunakan *dataset HAM10K* yang telah diawasi secara cermat dan telah melawati proses normalisasi. Diagnosis dataset telah divalidasi *histopatologi* pada lebih 50% kasus yang didasarkan pada konsensus dokter kulit.



Gambar 3. 1 Sampel Citra *Melanoma*



Gambar 3. 2 Sampel Citra *Non-Melanoma*

#### 3.1.2 Augmentasi Data

Pada tahap ini akan dilakukan *augmentasi*. tahap ini digunakan untuk meningkatkan kualitas dari citra masukan supaya mencegah *overfitting*. Tahap pertama menggunakan objek *ImageDataGenerator()*. Dengan objek tersebut menggunakan beberapa paramater diantaranya *rescale* untuk mengubah rentang piksel menjadi 0-1, *rotation range* yang digunakan untuk memudahkan model

mempelajari objek dari berbagai sudut pandang, *horizontal flip* yang digunakan untuk mempelajari objek dari berbagai arah dan orientasi, *shear range* digunakan untuk menggeser piksel dalam arah vertikal atau horizontal, *fill mode* untuk menentukan mode yang digunakan untuk mengisi piksel yang kosong akibat proses *augmentasi*.

*Augmentasi* selanjutnya menggunakan objek *flow\_from\_directory()*. Objek ini membutuhkan data untuk dibagi menjadi beberapa direktori, yaitu */train*, */validation* dan */test*. Pada setiap direktori didalamnya terdapat sub direktori untuk tiap kelas, contohnya *train/positif\_melanoma*, *train/negatif\_melanoma*. Kemudian semua gambar dimasukkan dalam sub direktori sesuai dengan nama kelasnya. Selanjutnya gambar akan diubah dimensinya menjadi 224x224 piksel. Setiap gambar akan diubah menjadi *tensor* dan disesuaikan dengan rentang nilai 0-1.

### 3.1.3 Pelatihan Model

Input	Operator	t	c	n	S
224x224x3	Conv2d	-	32	1	2
112x112x32	bottleneck	1	16	1	11
112x112x16	bottleneck	6	24	2	2
56x56x24	bottleneck	6	32	3	2
28x28x32	bottleneck	6	64	4	2
14x14x64	bottleneck	6	96	3	1
14x14x96	bottleneck	6	160	3	2
7x7x160	bottleneck	6	320	1	1
7x7x320	Conv2d 1x1	-	1280	1	1
7x7x1280	globalavgpoll	-	1280	1	-
1x1x1280	Dense	-	2	-	-

Gambar 3. 3 Model Summary MobileNetV2

Lapisan pertama merupakan sebuah *input layer* yang akan menerima gambar dengan resolusi 224x224 dan 3 kanal warna *RGB*. Pada layer pertama ini dapat menerima input dengan jumlah sampel yang berbeda-beda karena nantinya akan didefinisikan sebagai *none*, tergantung pada *batch size* yang diberikan. Setiap urutan

atau kelompok lapisan dalam arsitektur terdiri dari satu atau lebih lapisan yang memiliki konfigurasi yang sama dan diulang sebanyak  $n$  kali. Setiap urutan dimulai dengan lapisan pertama yang memiliki langkah (*stride*)  $s$ .

*Linear Residual Block (Bottleneck)* adalah sebuah struktur yang terdiri dari tiga lapisan konvolusi yang berperan penting dalam proses jaringan. Pertama, ada *Expansion Layer*, yang berupa lapisan konvolusi  $1 \times 1$ . Fungsinya adalah untuk meningkatkan jumlah saluran data sebelum data tersebut masuk ke *Depthwise Convolution*. Dengan menggunakan faktor ekspansi ( $t$ ), lapisan ini dapat memperluas jumlah saluran keluaran ( $c$ ) dibandingkan dengan jumlah saluran masukan. Selanjutnya, *Depthwise Convolution* berperan dalam menyaring data yang penting pada tahap jaringan ini. Lapisan ini melakukan konvolusi terhadap setiap saluran data secara terpisah, memungkinkan untuk menangkap informasi yang lebih spesifik dan efisien. Terakhir, *Projection Layer* bertanggung jawab untuk menge mbalikan data ke dimensi semula atau mengurangi dimensi data menjadi lebih kecil, sesuai dengan kebutuhan pada saat itu. Secara keseluruhan, *Linear Residual Block (Bottleneck)* dengan tiga lapisan konvolusi ini memainkan peran krusial dalam meningkatkan efisiensi dan kualitas representasi data dalam jaringan. Nilai *default* dari faktor ekspansi adalah 6.

Setelah melewati jaringan dasar, citra akan memiliki resolusi *output* sebesar  $7 \times 7 \times 1280$ . Untuk melakukan klasifikasi pada citra tersebut, digunakan teknik *Global Average Pooling* yang bertujuan untuk mengurangi ukuran citra dari  $7 \times 7$  menjadi  $1 \times 1$ . Teknik ini dilakukan pada tahap pengklasifikasian pertama. Dalam lapisan *dense* dengan menggunakan fungsi aktivasi *softmax*, akan dilakukan dua jenis klasifikasi pada *dataset* yaitu klasifikasi kanker kulit *melanoma* dan kanker kulit *non-melanoma*.

Selanjutnya melakukan kompilasi atau persiapan model sebelum tahap pelatihan model. Pada tahap kompilasi menggunakan *optimizer*. *Loss function* pada model menggunakan *categorical\_crossentropy*. Fungsi tersebut cocok digunakan pada label yang tidak saling *overlap* atau eksklusif. Dan menggunakan metrik *accuracy* sebagai evaluasi performa model nantinya.

Selanjutnya dilakukan proses pelatihan model dengan menggunakan fungsi *fit()* dari objek model. Pada fungsi tersebut, dilakukan beberapa hal seperti *steps per epoch* untuk mengatur jumlah *batch* yang akan diambil dari generator pada setiap *epoch*, serta menambahkan objek *callback* yang telah dibuat sebelumnya sebagai argumen dalam parameter *callbacks*. Dengan menggunakan objek *callback* pada proses pelatihan model digunakan untuk memantau dan mengontrol proses pelatihan dengan lebih efektif, termasuk monitoring nilai akurasi, mengurangi *learning rate*, dan memberhentikan proses pelatihan model jika diperlukan.

### 3.1.4 Evaluasi Model

Pada tahap evaluasi model, model akan dievaluasi dengan menggunakan *confusion matrix*. Evaluasi dengan *confusion matrix* akan menghasilkan nilai *accuracy*, *precision*, *recall*, dan *F1-Score* terhadap dua kategori kanker kulit, yaitu *melanoma* dan *non-melanoma*. Rincian tabel *confusion matrix* dapat ditemukan pada Tabel 3.1.

Tabel 3. 1 *Confusion Matrix*

		Nilai Prediksi	
		Negatif	Positif
Nilai Aktual	Negatif	TN	FP
	Positif	FN	TP

Ada 4 kombinasi berbeda antara nilai prediksi dan nilai aktual di confusion matrix, dan istilah yang menunjukkan hasil klasifikasi adalah sebagai berikut:

1. *True Positif* (TP) adalah data yang diprediksi benar dan hasil aktualnya juga benar. Ini berarti model telah berhasil mengidentifikasi kelas positif dengan tepat.
2. *True Negatif* (TN) adalah data yang diprediksi salah dan hasil aktualnya juga salah. Model secara tepat mengidentifikasi kelas negatif.
3. *False Positif* (FP) adalah data yang diprediksi salah, tetapi hasil aktualnya sebenarnya benar. Model "salah" memprediksi kelas positif padahal seharusnya negatif.

4. *False Negatif* (FN) adalah Data yang diprediksi salah, tetapi hasil aktualnya sebenarnya benar. Model "salah" memprediksi kelas negatif padahal seharusnya positif.

Dalam konteks *confusion matrix*, TP, FP, TN, dan FN adalah representasi penting untuk mengukur performa dan keakuratan dari model klasifikasi. Nilai metrik tersebut digunakan untuk mencari nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

### 3.1.5 Konversi Model

Tahap ini merupakan tahap lanjutan setelah proses pelatihan model. Supaya model bisa diintegrasikan pada *platform* android, sebelumnya model harus di konversi terlebih dahulu menjadi format *TensorFlow Lite*. Konversi ini bertujuan untuk mengoptimalkan ukuran model dan meningkatkan efisiensi saat dijalankan pada perangkat android dengan menggunakan *TensorFlow Lite Runtime*.

Proses konversi model dengan *TensorFlow Lite* akan memastikan bahwa model tetap mempertahankan akurasi dan kualitas klasifikasi yang tinggi, sambil mengurangi penggunaan sumber daya perangkat android secara optimal. Dengan menggunakan *TensorFlow Lite*, model dapat dijalankan diberbagai perangkat android dengan performa yang seragam dan konsisten.

### 3.1.6 Pengembangan Aplikasi Android

Tahap pengembangan aplikasi adalah tahap selanjutnya setelah proses konversi model menjadi format *TensorFlow Lite*. Sistem dikembangkan pada *platform* android dengan bahasa pemrograman *kotlin* dan memanfaatkan teknologi kecerdasan buatan yang telah diimplementasikan dalam model *CNN*. Proses pengembangan akan mencakup integrasi model *CNN* ke dalam aplikasi android agar dapat melakukan klasifikasi gambar dengan akurasi yang tinggi. Pada tahap ini juga dilakukan tahap pengamatan hasil klasifikasi pada aplikasi untuk memastikan keandalan dan akurasi model *CNN* pada platform android. Diharapkan aplikasi mampu melakukan klasifikasi dengan tepat dan akurasi yang tinggi.

### 3.1.7 Pengujian *Black Box*

Pada tahap ini, aplikasi yang telah dibuat kemudian dilakukan pengujian dengan metode *black box* untuk memverifikasi bahwa fungsionalitasnya berjalan dengan baik dan sesuai dengan yang diharapkan. Pengujian *black box* adalah

pendekatan pengujian, di mana pengujian dilakukan berdasarkan pengamatan pada masukan dan keluaran dari perangkat lunak tanpa harus mengetahui keseluruhan kode program internal dari aplikasi tersebut. Dengan menggunakan metode ini, pengujian dapat memfokuskan pada perilaku eksternal aplikasi untuk memastikan bahwa aplikasi berfungsi dengan benar sesuai dengan persyaratan dan kebutuhan yang telah ditetapkan.

Pengujian *black box* memberikan berbagai jenis masukan yang berbeda ke aplikasi dan mengamati respon atau *ouput* yang dihasilkan. Dengan mencoba berbagai skenario penggunaan dalam beragam situasi yang mungkin terjadi saat aplikasi berinteraksi dengan pengguna atau sistem lain.

### 3.2 Analisis Kebutuhan

Pada analisis kebutuhan, peneliti melakukan analisis *tools* yang dibutuhkan dalam pengembangan sistem. *Tools* yang dibutuhkan dijabarkan pada tabel 3.2 dibawah.

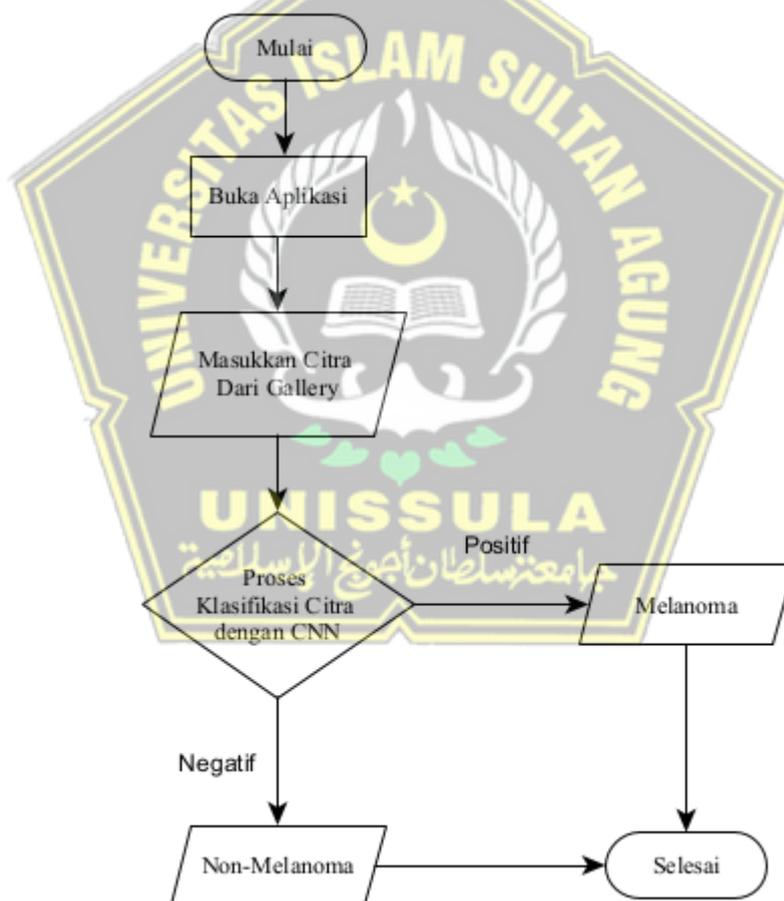
Tabel 3. 2 Analisis Kebutuhan

No	Tools	Version
1	Laptop Lenovo A514-51G-59HF	Window 11
2	Smartphone Redmi Note 10 Pro	Android 11
3	Python	3.8.0
4	TensorFlow	2.10.0
5	Keras	2.10.0
6	TensorFlow Lite	2.10.0
7	Jupyter Notebook	6.5.3
8	Android Studio	Dolphin, 2021.3.1
9	Gradle	7.4
10	Kotlin	1.7.20
11	Room Database	2.3.0

### 3.3 Analisis Sistem

Pada tahap ini peneliti melakukan analisis untuk menentukan alur kerja dari sistem yang dibuat dalam bentuk *flowchart*. Pertama-tama pengguna membuka aplikasi. Selanjutnya pengguna akan diarahkan ke halaman *home*. Pengguna dapat

menekan tombol ‘Tambah Gambar’ untuk menambahkan gambar. Terdapat pilihan mengambil gambar dari *gallery*. Pengguna dapat menekan tombol *gallery*. Selanjutnya pengguna akan diarahkan menuju *gallery*, pengguna dapat memilih gambar yang akan di masukkan kedalam aplikasi sesuai keinginan. Setelah pengguna memilih gambar, dilakukan proses *cropping* supaya gambar lebih fokus pada area kulit yang akan dilakukan deteksi. Kemudian klik potong. Selanjutnya klik ‘Lanjut’ untuk melanjutkan proses deteksi. Kemudian aplikasi akan melakukan klasifikasi penyakit berdasarkan masukan dari pengguna dan menampilkan hasil klasifikasi serta *presentase* dari hasil deteksi berupa positif atau negatif. Gambar 3.3 merupakan gambaran *flowchart* sistem pada penelitian ini.



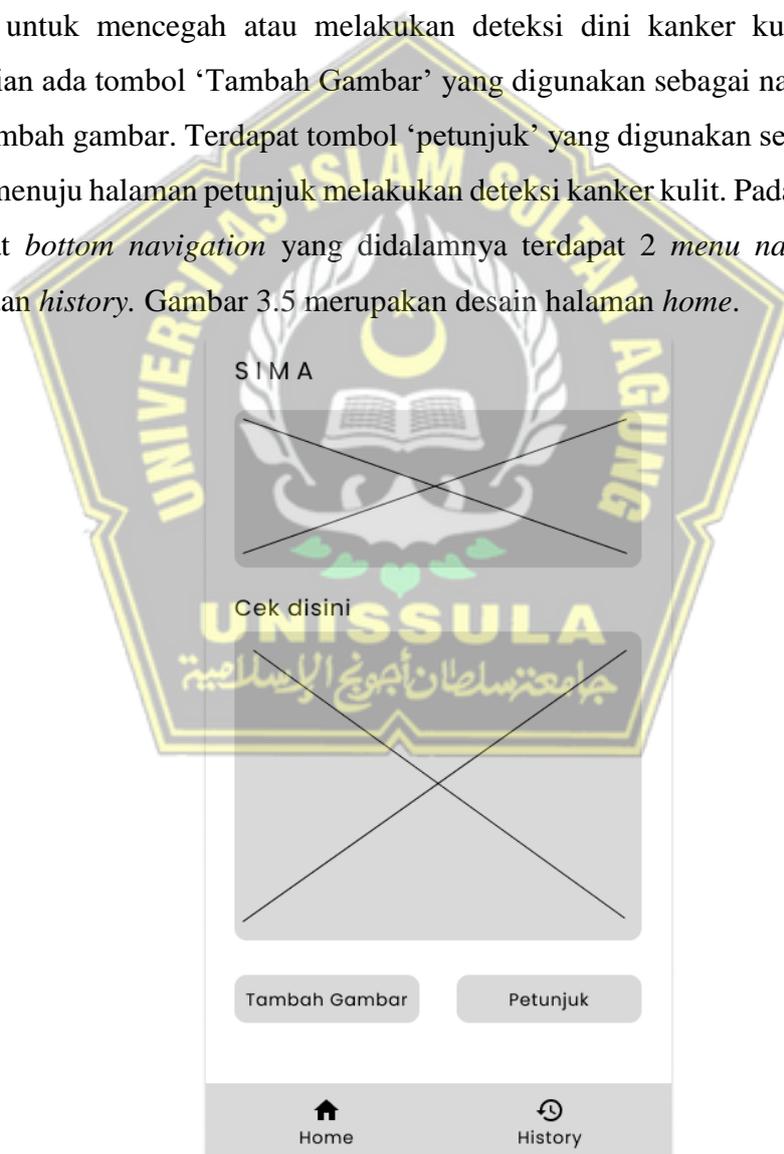
Gambar 3. 4 *Flowchart* Sistem

### 3.4 Perancangan Antarmuka

Perancangan antarmuka digunakan untuk mempermudah dalam tahap pengembangan aplikasi. Pengaturan tata letak yang optimal akan mempermudah interaksi pengguna dengan aplikasi. Tujuan tahap perancangan atarmuka adalah menciptakan antarmuka aplikasi yang menarik, efisien dan intuitif.

#### 3.4.1 Halaman *Home*

Perancangan desain antarmuka dari halaman *home*. Pada bagian *appbar* terdapat *title* dari aplikasi. Bagian *body* terdapat *banner* berisi ajakan terhadap pasien untuk mencegah atau melakukan deteksi dini kanker kulit *melanoma*, kemudian ada tombol ‘Tambah Gambar’ yang digunakan sebagai navigasi menuju fitur tambah gambar. Terdapat tombol ‘petunjuk’ yang digunakan sebagai navigasi untuk menuju halaman petunjuk melakukan deteksi kanker kulit. Pada bagian *footer* terdapat *bottom navigation* yang didalamnya terdapat 2 *menu navigation* yaitu *home* dan *history*. Gambar 3.5 merupakan desain halaman *home*.



Gambar 3. 5 Desain Halaman *Home*

### 3.4.2 Menu Tambah Gambar

Desain antarmuka dari menu tambah gambar. Tampilan menu menggunakan *bottom sheet dialog*. Pada halaman *sheet dialog* terdapat *title*, *icon* dan deskripsi dari *icon* sebagai tombol navigasi untuk menuju aplikasi *third-party* yaitu *gallery*. Gambar 3.6 merupakan desain menu tambah gambar.



Gambar 3. 6 Desain Menu Tambah Gambar

### 3.4.3 Halaman Verifikasi Lanjutan

Desain antarmuka halaman verifikasi lanjutan. Tampilan menu menggunakan *bottom sheet dialog*. Pada menu ini terdapat *review* gambar yang akan digunakan untuk deteksi kanker kulit. Terdapat tombol ‘Lanjut’ untuk melanjutkan proses deteksi dan tombol ‘Ubah’ untuk merubah gambar. Gambar 3.8 merupakan desain halaman verifikasi lanjutan



Gambar 3. 7 Desain Menu Verifikasi Lanjutan

### 3.4.4 Halaman Hasil Deteksi Kanker Kulit

Desain antarmuka halaman hasil deteksi kanker kulit. Tampilan halaman menggunakan *bottom sheet dialog*. Pada halaman ini menampilkan beberapa *property* seperti gambar yang dideteksi, informasi kategori kanker kulit dan *presentase* positif dan negatif dari terjangkit kanker kulit. Gambar 3.9 merupakan desain halaman hasil deteksi.



Gambar 3. 8 Desain Halaman Hasil Deteksi Kanker Kulit

### 3.4.5 Halaman Riwayat Deteksi

Desain antarmuka halaman riwayat deteksi. Pada bagian *appbar* terdapat *title* halaman. Pada bagian *body* terdapat *recyclerview* yang berisi item hasil deteksi seperti tanggal deteksi, gambar, hasil deteksi dan *presentase* deteksi. Pada bagian *footer* terdapat *bottom navigation* yang didalamnya terdapat 2 *menu navigation* yaitu *home* dan *history*. Gambar 3.10 merupakan desain halaman *history*.



Gambar 3. 9 Desain Halaman Riwayat Deteksi

### 3.4.6 Halaman Petunjuk Melakukan Deteksi

Desain antarmuka halaman petunjuk. Tampilan halaman menggunakan *bottom sheet dialog*. Halaman ini berisi *textview* yang digunakan untuk menampilkan informasi langkah-langkah cara melakukan deteksi kanker kulit. Gambar 3.11 merupakan desain halaman petunjuk melakukan deteksi.



Gambar 3. 10 Desain Halaman Petunjuk

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Implementasi Antarmuka

Setelah tahap pemodelan selesai. Tahap selanjutnya adalah implementasi sistem pada *platform* android. Antarmuka yang dikembangkan merujuk dari desain *wireframe* sistem pada bab III.

##### 4.2.1 Halaman Home

Implementasi desain antarmuka halaman *home*. Pada bagian *AppBar* terdapat *title* dari aplikasi. Bagian *body* terdapat *banner* berisi ajakan terhadap pasien untuk mencegah kanker kulit melanoma, kemudian ada tombol ‘Tambah Gambar’ yang digunakan sebagai navigasi menuju fitur tambah gambar. Terdapat tombol ‘petunjuk’ yang digunakan sebagai navigasi untuk menuju halaman petunjuk melakukan deteksi kanker kulit. Pada bagian *footer* terdapat *bottom navigation* yang didalamnya terdapat 2 *menu navigation* yaitu *home* dan *history*. Menu *home* digunakan untuk navigasi menuju halaman *home* dan menu *history* digunakan untuk navigasi menuju halaman *history*. Gambar 4.1 merupakan tampilan implementasi halaman *home*.



Gambar 4. 1 Antarmuka Halaman *Home*

#### 4.2.2 Menu Tambah Gambar

Implementasi dari desain antarmuka Menu Tambah Gambar. Tampilan menu dibuat dengan *bottom sheet dialog*. Pada halaman sheet dialog terdapat *title* ‘ambil gambar’. Terdapat *icon gallery* dan deskripsi yang digunakan tombol navigasi menuju aplikasi *gallery*. Setelah pengguna menekan tombol tersebut, aplikasi SIMA akan mengakses aplikasi *third party* yaitu *gallery*. *Gallery* disini digunakan untuk mengambil gambar dari perangkat pengguna. Gambar 4.5 merupakan tampilan implementasi Menu Tambah Gambar.



Gambar 4. 2 Antarmuka Menu Tambah Gambar

### 4.2.3 Halaman Verifikasi Lanjutan

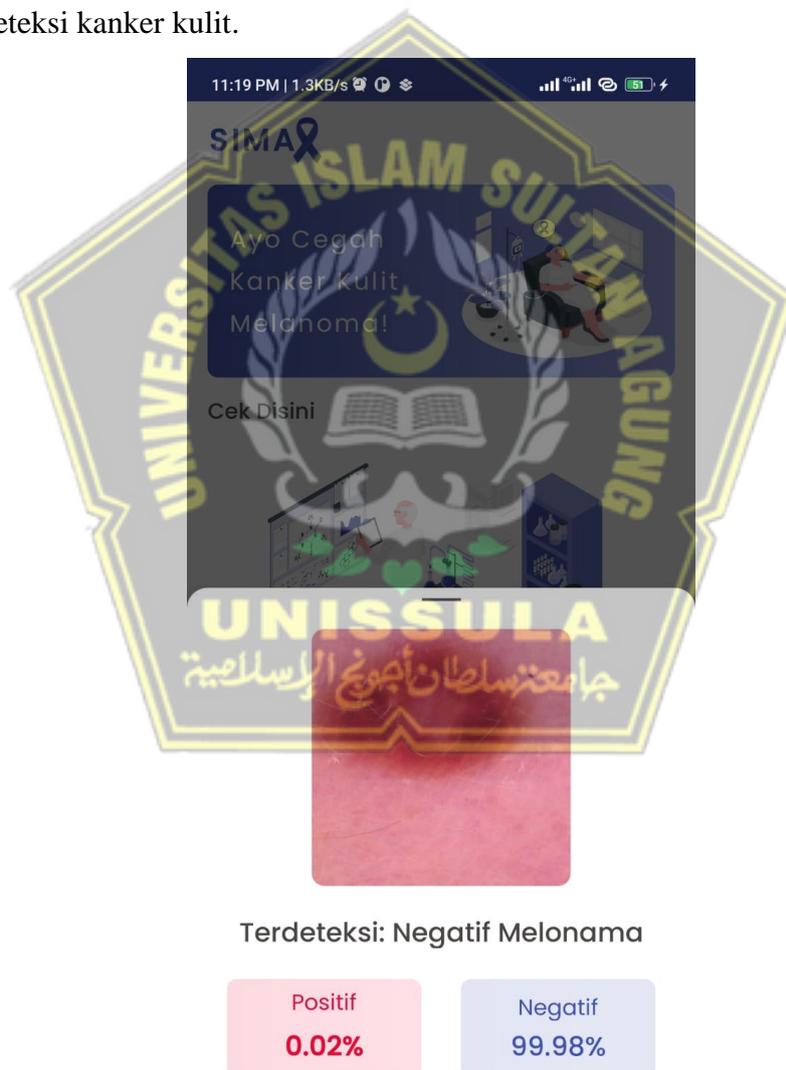
Implementasi desain antarmuka halaman verifikasi lanjutan. Tampilan menu dibuat menggunakan *bottom sheet dialog*. Pada menu ini terdapat *review* gambar yang akan digunakan untuk deteksi kanker kulit. Terdapat tombol ‘Lanjut’ digunakan untuk melanjutkan proses deteksi dan tombol ‘Ubah’ digunakan untuk merubah gambar. Ketika pengguna menekan tombol ‘Lanjut’ aplikasi melakukan deteksi gambar masukan dengan menggunakan model *CNN* yang telah dilatih sebelumnya. Ketika pengguna menekan tombol ‘Ubah’ pengguna akan diarahkan kembali ke aplikasi *gallery* untuk merubah gambar. Gambar 4.7 merupakan tampilan implementasi halaman verifikasi lanjutan.



Gambar 4. 3 Antarmuka Halaman Verifikasi Lanjutan

#### 4.2.4 Halaman Hasil Deteksi Kanker Kulit

Implementasi desain antarmuka halaman hasil deteksi kanker kulit. Tampilan halaman dibuat menggunakan *bottom sheet dialog*. Pada halaman ini menampilkan beberapa *property* seperti *imageview* yang digunakan untuk menampilkan gambar hasil deteksi, *textview* untuk menampilkan informasi kategori kanker kulit dan kemudian ada *presentase* dari terjangkit kanker kulit, *textview* dengan *backgroud* warna merah merupakan hasil *presentase* positif dan yang warna biru merupakan hasil *presentase* negatif. Gambar 4.8 merupakan tampilan implementasi halaman hasil deteksi kanker kulit.



Gambar 4. 4 Antarmuka Halaman Hasil Deteksi Kanker Kulit

#### 4.2.5 Halaman Riwayat Deteksi

Implementasi desain antarmuka halaman riwayat deteksi. Bagian *AppBar* terdapat *title* halaman. Bagian *body* terdapat *recyclerview* yang berisi *item* hasil deteksi. Pada bagian atas terdapat *textview* yang digunakan untuk menampilkan waktu saat melakukan deteksi. *Imageview* digunakan untuk menampilkan gambar hasil deteksi. Dibawah *property imageview* terdapat sebuah *textview* yang digunakan untuk menampilkan hasil deteksi. Kemudian ada *textview* dengan *background* merah merupakan *textview* untuk menampilkan hasil *presentase* positif dan *background* biru merupakan *textview* untuk menampilkan hasil *presentase* negatif. Bagian *footer* terdapat *bottom navigation* yang didalamnya terdapat 2 menu navigation yaitu *home* dan *history*. Menu *home* digunakan untuk navigasi menuju halaman *home* dan menu *history* digunakan untuk navigasi menuju halaman *history*. Gambar 4.9 merupakan tampilan implementasi halaman riwayat deteksi.



Gambar 4. 5 Antarmuka Halaman Riwayat Deteksi

#### 4.2.1 Halaman Petunjuk Melakukan Deteksi

Implementasi desain antarmuka Halaman Petunjuk. Tampilan halaman dibuat menggunakan *bottom sheet dialog*. Halaman ini berisi *textview* yang digunakan untuk menampilkan informasi langkah-langkah cara melakukan deteksi kanker kulit. Gambar 3.10 merupakan tampilan implementasi Halaman Petunjuk Melakukan Deteksi.



Gambar 4. 6 Antarmuka Petunjuk Melakukan Deteksi

## 4.2 Evaluasi Model

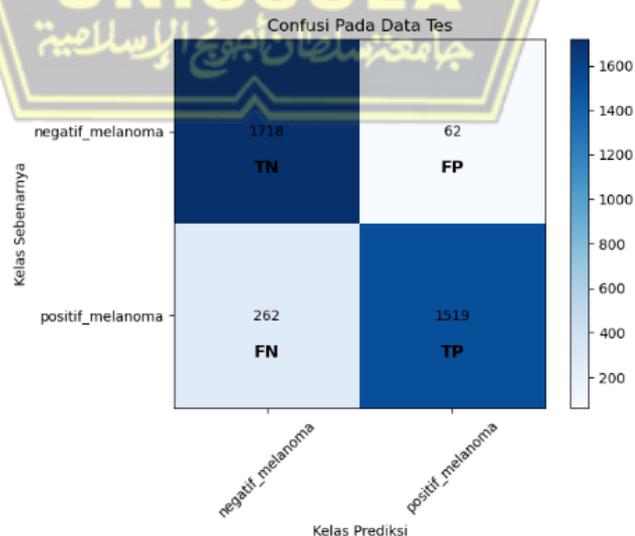
Untuk mengetahui sebuah model bekerja dengan baik atau tidak, perlu dilakukan evaluasi pada model tersebut. Proses evaluasi menggunakan *confusion matrix* untuk mendapatkan nilai akurasi (*accuracy*), presisi (*precision*), *racall*, dan *f1-score*.

### 1. Konfigurasi 1

Tabel 4. 1 Konfigurasi 1

<i>Epoch</i>	<i>Optimizer</i>	Learning Rate
10	<i>Adadelata</i>	2.5000e-04

Tabel 4.1 merupakan komposisi percobaan pada konfigurasi 1 dengan nilai *epoch* sebesar 10 dan menggunakan *optimizer Adadelata*. Dengan nilai *learning rate* sebesar 2.5000e-04. Pada pelatihan model konfigurasi 1 mendapatkan nilai *accuracy* 0,8956, *loss* 0,2627, validasi *accuracy* 0,9060, dan validasi *loss* 0,2459. Dengan hasil tersebut, terlihat model memiliki kinerja yang baik dalam memprediksi data pelatihan dan data validasi. Meskipun akurasi validasi sedikit lebih tinggi daripada akurasi pelatihan, tetapi perbedaannya tidak signifikan, yang menunjukkan bahwa model tidak mengalami *overfitting*. Selanjutnya dilakukan evaluasi dengan *confusion matrix* yang ditampilkan pada gambar 4.1 dibawah.



Gambar 4. 7 *Confusion Matrix* Konfigurasi 1

Berdasarkan nilai matriks pada gambar 4.1, nilai TP = 1519, TN = 1718, FN = 262, dan FP = 62. Berikut perhitungan akurasi(*accuracy*), presisi(*precision*), *recall*, dan *f1-score*.

$$Accuracy = \frac{1522 + 1720}{1518 + 59 + 263 + 1721} = 0,9104$$

$$Precision = \frac{1519}{1519 + 62} = 0,9607$$

$$Precision = \frac{1519}{1519 + 262} = 0,8528$$

$$F1-Score = \frac{2 \times 0,9607 \times 0,8528}{0,8528 + 0,9607} = 0,9036$$

Hasil perhitungan *confusion matrix* diatas ditampilkan pada tabel 4.2 dibawah.

Tabel 4. 2 Hasil *Confusion Matrix* Konfigurasi 1

Konfigurasi	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Konfigurasi 1	0,9104	0,9607	0,8528	0,9036

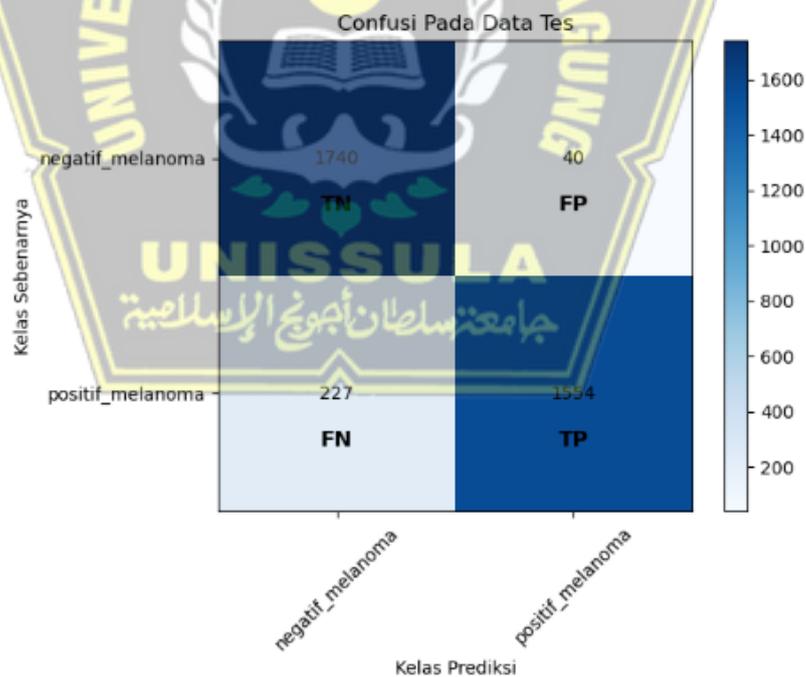
Setelah tahap evaluasi dengan menggunakan *confusion matrix*, dapat disimpulkan bahwa model memiliki akurasi yang tinggi, yang mengindikasikan kinerja keseluruhan yang baik. Selain itu, nilai *precision* yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik untuk mengidentifikasi data positif dengan benar. Namun, *recall* yang lebih rendah menandakan bahwa model dapat melewati beberapa kasus positif. Nilai *f1-score* yang tinggi mengindikasikan keseimbangan yang baik antara *precision* dan *recall*, yang menunjukkan performa yang bagus secara keseluruhan dalam klasifikasi data positif dan negatif.

## 2. Konfigurasi 2

Tabel 4. 3 Konfigurasi 2

<i>Epoch</i>	<i>Optimizer</i>	Learning Rate
10	RMSprop	6.2500e-05

Tabel 4.3 merupakan komposisi percobaan pada konfigurasi 2 dengan nilai *epoch* sebesar 10 dan menggunakan *optimizer RMSprop*. Dengan nilai *learning rate* sebesar 6.2500e-04. Pada pelatihan model konfigurasi 2 mendapatkan nilai *accuracy* 0,9206, *loss* 0,3637, validasi *accuracy* 0,9180, dan validasi *loss* 0,3572. Dengan hasil tersebut, terlihat model memiliki kinerja yang baik dalam memprediksi data pelatihan dan data validasi. Meskipun akurasi validasi sedikit lebih rendah daripada akurasi pelatihan, tetapi perbedaannya tidak signifikan, yang menunjukkan bahwa model tidak mengalami *overfitting*. Tahap selanjutnya evaluasi dengan *confusion matrix* ditampilkan pada gambar 4.2 dibawah.



Gambar 4. 8 *Confusion Matrix* Konfigurasi 2

Berdasarkan nilai matriks pada gambar 4.2, nilai TP = 1554, TN = 1740, FN = 227, dan FP = 40. Berikut perhitungan akurasi(*accuracy*), presisi(*precision*), *recall*, dan *f1-score*.

$$Accuracy = \frac{1554 + 1740}{1554 + 40 + 227 + 1740} = 0,9250$$

$$Precision = \frac{1554}{1554 + 40} = 0,9749$$

$$Recall = \frac{1554}{1554 + 227} = 0,8725$$

$$F1-Score = \frac{2 \times 0,9749 \times 0,8725}{0,8725 + 0,9749} = 0,9208$$

Hasil perhitungan *confusion matrix* diatas ditampilkan pada tabel 4.3 dibawah.

Tabel 4. 4 Hasil *Confusion Matrix* Konfigurasi 2

Konfigurasi	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Konfigurasi 2	0,9250	0,9749	0,8725	0,9208

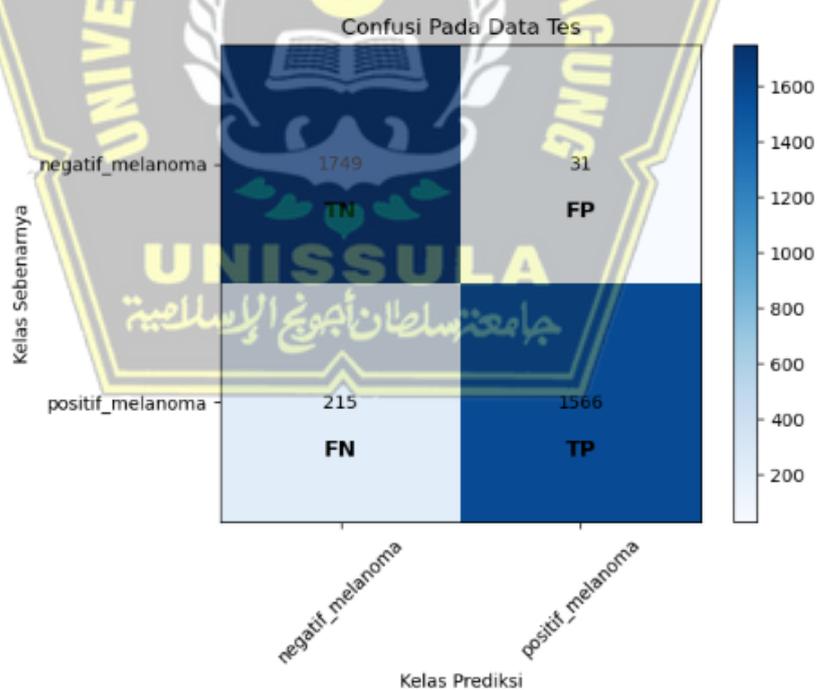
Setelah tahap evaluasi dengan menggunakan *confusion matrix*, dapat disimpulkan bahwa model memiliki akurasi yang tinggi, yang mengindikasikan kinerja keseluruhan yang baik. Selain itu, nilai *precision* yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik untuk mengidentifikasi data positif dengan benar. Namun, *recall* yang lebih rendah menandakan bahwa model dapat melewati beberapa kasus positif. Nilai *f1-score* yang tinggi mengindikasikan keseimbangan yang baik antara *precision* dan *recall*, yang menunjukkan performa yang bagus secara keseluruhan dalam klasifikasi data positif dan negatif.

### 3. Konfigurasi 3

Tabel 4. 5 Konfigurasi 3

<i>Epoch</i>	<i>Optimizer</i>	Learning Rate
20	<i>SGD</i>	3.9062e-05

Tabel 4.5 merupakan komposisi percobaan pada konfigurasi 3 dengan nilai *epoch* sebesar 20 dan menggunakan *optimizer SGD*. Dengan nilai *learning rate* sebesar 3.9062e-05. Pada pelatihan model konfigurasi 3 mendapatkan nilai *accuracy* 0,9503, *loss* 0,1236, validasi *accuracy* 0,9363, dan validasi *loss* 0,1588. Dengan hasil tersebut, terlihat model memiliki kinerja yang baik dalam memprediksi data pelatihan dan data validasi. Meskipun akurasi validasi sedikit lebih rendah daripada akurasi pelatihan, tetapi perbedaannya tidak signifikan, yang menunjukkan bahwa model tidak mengalami *overfitting*. Tahap selanjutnya evaluasi dengan *confusion matrix* yang ditampilkan pada gambar 4.3 dibawah.



Gambar 4. 9 *Confusion Matrix* Konfigurasi 3

Berdasarkan nilai matriks pada gambar 4.3, nilai TP = 1566, TN = 1749, FN = 215, dan FP = 31. Berikut perhitungan akurasi(*accuracy*), presisi(*precision*), *recall*, dan *f1-score*.

$$Accuracy = \frac{1566 + 1749}{1566 + 31 + 215 + 1749} = 0,9309$$

$$Precision = \frac{1566}{1566 + 31} = 0,9805$$

$$Recall = \frac{1566}{1566 + 215} = 0,8792$$

$$F1-Score = \frac{2 \times 0,9805 \times 0,8792}{0,8792 + 0,9805} = 0,9270$$

Hasil perhitungan *confusion matrix* diatas ditampilkan pada tabel 4.6 dibawah.

Tabel 4. 6 Hasil *Confusion Matrix* Konfigurasi 3

Konfigurasi	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Konfigurasi 3	0,9309	0,9805	0,8792	0,9270

Setelah tahap evaluasi dengan menggunakan *confusion matrix*, dapat disimpulkan bahwa model memiliki akurasi yang tinggi, yang mengindikasikan kinerja keseluruhan yang baik. Selain itu, nilai *precision* yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik untuk mengidentifikasi data positif dengan benar. Namun, *recall* yang lebih rendah menandakan bahwa model dapat melewatkan beberapa kasus positif. Nilai *f1-score* yang tinggi mengindikasikan keseimbangan yang baik antara *precision* dan *recall*, yang menunjukkan performa yang bagus secara keseluruhan dalam klasifikasi data positif dan negatif.

Dari tiga percobaan konfigurasi yang telah dilakukan, konfigurasi 3 merupakan konfigurasi terbaik. Dari hasil pelatihan model konfigurasi 3 mendapatkan nilai *accuracy* 0,9503, *loss* 0,1236, validasi *accuracy* 0,9363, dan validasi *loss* 0,1588. Setelah tahap pelatihan model selesai, kemudian model dilakukan evaluasi menggunakan *confusion matrix* dengan hasil *accuracy* 0,9309, *precision* 0,9805, *recall* 0,8792 dan *f1-score* 0,9270.

### 4.3 Pengujian Aplikasi

Untuk memastikan aplikasi berjalan dengan baik dan benar sesuai dengan spesifikasi yang dibuat, maka harus dilakukan pengujian fungsionalitas pada aplikasi. Pengujian fungsionalitas aplikasi pada penelitian ini dilakukan dengan teknik *black box*. Berikut skenario dan hasil dari pengujian aplikasi dengan menggunakan teknik *black box*:

#### 1. Pengujian Fungsionalitas Aplikasi

Tabel 4. 7 Skenario Pengujian Fungsionalitas Aplikasi

Skenario	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Membuka aplikasi SIMA	Menampilkan halaman home dan tidak <i>foreclosed</i>	Sesuai	Berhasil
Menekan menu <i>home</i>	Menampilkan halaman home	Sesuai	Berhasil
Menekan tombol tambah gambar	Menampilkan menu tambah gambar	Sesuai	Berhasil
Menekan tombol kamera	Menampilkan Kamera	Sesuai	Berhasil
Mengambil gambar dari kamera	Gambar bisa di <i>import</i> ke halaman verifikasi lanjutan	Sesuai	Berhasil
Menekan tombol <i>gallery</i>	Menampilkan <i>gallery</i>	Sesuai	Berhasil
Mengambil gambar dari <i>gallery</i>	Gambar bisa di <i>import</i> ke halaman verifikasi lanjutan	Sesuai	Berhasil

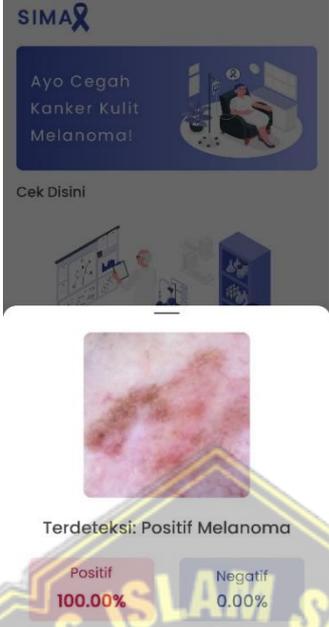
Menekan tombol lanjut	Menampilkan halaman hasil deteksi	Sesuai	Berhasil
Menampilkan halaman hasil deteksi	Data hasil deteksi berhasil disimpan ke <i>history</i>	Sesuai	Berhasil
Menekan tombol <i>history</i>	Menampilkan halaman <i>history</i>	Sesuai	Berhasil
Menekan tombol petunjuk	Menampilkan halaman petunjuk	Sesuai	Berhasil

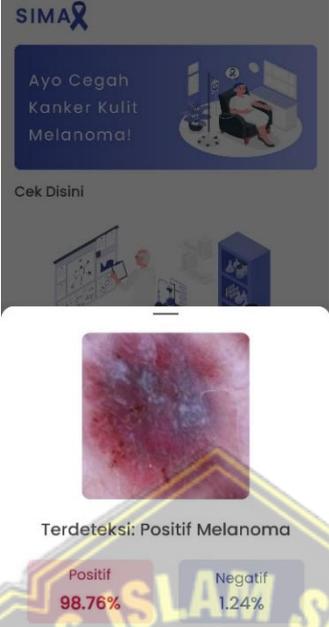
## 2. Pengujian Akurasi Pada Aplikasi

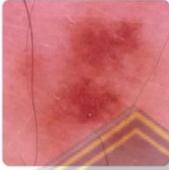
Untuk mengetahui tingkat akurasi pada aplikasi, maka dilakukan pengujian sebagai evaluasi apakah aplikasi dapat mendeteksi penyakit kanker kulit melanoma dengan benar. Pengujian ini menggunakan data pengujian yang berasal dari *kaggle*. Tabel 4.8 merupakan hasil pengujian akurasi pada aplikasi.

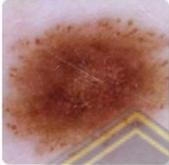
Tabel 4. 8 Pengujian Akurasi Pada Aplikasi

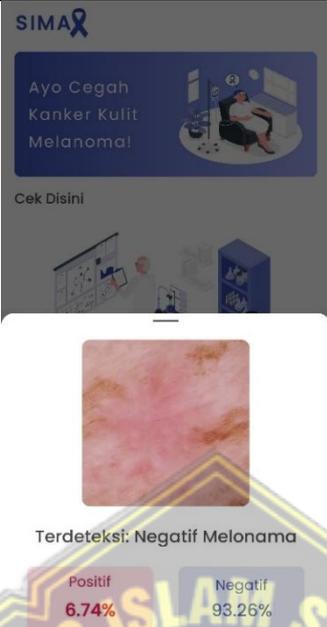
No	Kategori	Pengujian			Keterangan
		Gambar	Positif	Negatif	
1.	<i>Melanoma</i>		99,99%	0,01%	Sesuai

2.	<i>Melanoma</i>		100%	0%	Sesuai
3.	<i>Melanoma</i>		100%	0%	Sesuai

4.	<i>Melanoma</i>	 <p> <b>98,76%</b> Positif  <b>1,24%</b> Negatif     </p>	98,76%	1,24%	Sesuai
5.	<i>Melanoma</i>	 <p> <b>99,96%</b> Positif  <b>0,04%</b> Negatif     </p>	99,96%	0,04%	Sesuai

6.	<i>Non-melanoma</i>	 <p>Ayo Cegah Kanker Kulit Melanoma!</p> <p>Cek Disini</p>  <p>Terdeteksi: Negatif Melanoma</p> <p>Positif 1.14%    Negatif 98.86%</p>	1,14%	99,86%	Sesuai
7.	<i>Non-melanoma</i>	 <p>Ayo Cegah Kanker Kulit Melanoma!</p> <p>Cek Disini</p>  <p>Terdeteksi: Negatif Melanoma</p> <p>Positif 1.12%    Negatif 98.88%</p>	1,12%	98,88%	Sesuai

8.	<i>Non-melanoma</i>	  <p>Terdeteksi: Negatif Melanoma</p> <p>Positif 8.87%    Negatif 91.13%</p>	8,87%	91,13%	Sesuai
9.	<i>Non-melanoma</i>	  <p>Terdeteksi: Negatif Melanoma</p> <p>Positif 0.00%    Negatif 100.00%</p>	0%	100%	Sesuai

10.	<i>Non-melanoma</i>		6,74%	93,26%	Sesuai
-----	---------------------	---	-------	--------	--------

Berdasarkan tabel 4.8 terlihat bahwa aplikasi memiliki tingkat akurasi yang tinggi. Pada pengujian ini, diambil 5 sampel gambar data uji *melanoma* dan 5 sampel gambar data uji *non-melanoma*. Dari 2 kategori yang diuji yaitu *melanoma* dan *non-melanoma* aplikasi mampu mengklasifikasikan penyakit kanker kulit dengan tingkat akurasi tertinggi mencapai 100% dikedua kategori tersebut. Maka dapat disimpulkan bahwa aplikasi dapat diandalkan untuk mendeteksi secara dini penyakit kanker kulit *melanoma*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan oleh peneliti, dapat ditarik kesimpulan bahwa sistem deteksi kanker kulit *melanoma* menggunakan *MobileNetV2* yang diimplementasikan pada *platform* android menunjukkan kinerja yang baik dan dapat diambil kesimpulan penting sebagai berikut:

1. Konfigurasi terbaik didapat pada percobaan konfigurasi 3 dengan *accuracy* 0,9503, *loss* 0,1236, validasi *accuracy* 0,9363, dan validasi *loss* 0,1588. Dari hasil pelatihan model tersebut, dapat disimpulkan bahwa model tidak mengalami *overfitting*.
2. Model *MobileNetV2* menunjukkan kinerja yang sangat baik dalam deteksi kanker kulit melanoma. Dari hasil evaluasi menggunakan *confusion matrix* mendapat *accuracy* 0,9309, *precision* 0,9805, *recall* 0,8792, dan *f1-score* 0,9270. Berdasarkan evaluasi tersebut, model ini dapat diandalkan untuk memberikan hasil yang akurat dalam membedakan kanker kulit *melanoma* dan *non-melanoma*.
3. Aplikasi deteksi *melanoma* berbasis android yang diintegrasikan dengan model *MobileNetV2* yang telah dilatih sebelumnya, memberikan hasil deteksi yang akurat dengan akurasi tertinggi mencapai 100%. Menunjukkan bahwa aplikasi dapat diandalkan dalam mendeteksi dini penyakit kanker kulit *melanoma*.

#### 5.2 Saran

Meskipun hasil penelitian ini sudah menunjukkan performa yang baik, ada beberapa saran yang dapat dipertimbangkan untuk meningkatkan kualitas penelitian dan hasil yang lebih baik.

1. Untuk mengonfirmasi keefektifan model di dunia nyata, dapat dilakukan pengujian tambahan diluar dari *dataset* dengan validasi dari pakar terkait.
2. Untuk meningkatkan kinerja dan efisiensi, dapat memperbarui model dengan versi terbaru dari *MobileNet* atau arsitektur *deep learning* lainnya.

## DAFTAR PUSTAKA

- Adhinata, F. D., Tanjung, N. A. F., Widayat, W., Pasfica, G. R., & Satura, F. R. (2021). Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 7(2), 230. <https://doi.org/10.26555/jiteki.v7i2.20758>
- Ariesta, N., Musa, Z., & Septadina, I. S. (2018). Karakteristik Histopatologi Melanoma Maligna di Bagian Patologi Anatomi RSUP Dr. Moh. Hoesin Palembang Tahun 2009-2013. *Biomedical Journal of Indonesia: Jurnal Biomedik Fakultas Kedokteran Universitas Sriwijaya*, 4(1), 26–31. <https://doi.org/10.32539/bji.v5i1.7955>
- Budiawan, R. S., Hartono, B., Informatika, J. T., & Stikubank, U. (2023). *Pengembangan Sistem Pendeteksi Jenis Sayuran dengan Metode CNN Berbasis Android*. 5(1).
- Hariman, A. A., Mulyana, D. I., & Yel, M. B. (2023). Klasifikasi Jajanan Tradisional Jawa Tengah dengan Metode Transfer Learning dan MobileNetV2. *Jurnal Informasi Interaktif*, 8(1), 15–23. <http://e-journal.janabadra.ac.id/>
- Jeremia, G., Jacobus, A., Kambey, F. D., Elektro, T., Sam, U., Manado, R., & Manado, J. K. B. (2021). *34208-76836-2-Pb*. 16(4), 375–382.
- Kahfi, A., Shiddiq, A., & Dzikrullah, T. (2022). *Deteksi Penyakit pada Tanaman Padi Menggunakan MobileNet Transfer Learning Berbasis Android*. 2(2).
- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (2020). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network. *e-Proceeding of Engineering*, 7(1), 2558–2567.
- Naufal, M. F. (2021). Analisis Perbandingan *Algoritma SVM, KNN, dan CNN* untuk Klasifikasi Citra Cuaca. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(2), 311. <https://doi.org/10.25126/jtiik.2021824553>
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network ( CNN ) Pada Ekspresi Manusia. *Jurnal Algor*, Vol 2 No.1, 12–21.

- Pangestu, M. A., & Bunyamin, H. (2018). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika dan Sistem Informasi*, 4, 337–344.
- Permana, A., & Budayawan, K. (2020). Aplikasi Android Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah. *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, 8(4), 128. <https://doi.org/10.24036/voteteknika.v8i4.110349>
- Peryanto, A., Yudhana, A., & Umar, R. (2020). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format: Jurnal Ilmiah Teknik Informatika*, 8(2), 138. <https://doi.org/10.22441/format.2019.v8.i2.007>
- Prasetyo, A. R., Sussi, & Aditya, B. (2023). Analisis Perbandingan Algoritma Support Vector Machine ( Svm ) Dan Convolutional Neural Network ( Cnn ) Untuk Sistem Deteksi Katarak. 3(1), 1–10.
- Ramayanti, D., Sri Dianing Asri, & Lionie Lionie. (2022). Implementasi Model Arsitektur VGG16 dan MobileNetV2 Untuk Klasifikasi Citra Kupu-Kupu. *JSAI (Journal Scientific and Applied Informatics)*, 5(3), 182–187. <https://doi.org/10.36085/jsai.v5i3.2864>
- Ritonga, K. (2019). Sistem Pakar Mendiagnosa Penyakit Kanker Kulit Melanoma Menggunakan Metode Case Based Reasoning. *Jurnal Informasi dan Teknologi Ilmiah (INTI)*, 7(3), 5.
- Riva, L. S., Febriana, F., Panghurina, M., Maulida, R. A., Najah, B., Yasiin, A. D., & Chamidah, N. (2022). Perbandingan Algoritma Cnn Dan Ann Dengan Projection Histogram Untuk Klasifikasi Citra Tulisan Tangan Berupa Angka. 876–888.
- Rozaqi, A. J., Sunyoto, A., & Arief, M. rudyanto. (2021). Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network. *Creative Information Technology Journal*, 8(1), 22. <https://doi.org/10.24076/citec.2021v8i1.263>
- Tandiari, J., Yulia, Y., & Nathania, A. (2019). Implementasi Room Database pada Aplikasi SFA untuk PT . X berbasis Android. *Jurnal Infra*, 7(2), 2–6.

- Tarisa, R. E. D., Rustam, R., & Elmatris, E. (2022). Hubungan Jenis Pekerjaan dengan Kanker Kulit di RSUP Dr. M. Djamil Padang Tahun 2015 - 2020. *Jurnal Ilmu Kesehatan Indonesia*, 3(1), 67–73. <https://doi.org/10.25077/jikesi.v3i1.739>
- Wilvestra, S., Lestari, S., & Asri, E. (2018). Studi Retrospektif Kanker Kulit di Poliklinik Ilmu Kesehatan Kulit dan Kelamin RS Dr. M. Djamil Padang Periode Tahun 2015-2017. *Jurnal Kesehatan Andalas*, 7(Supplement 3), 47. <https://doi.org/10.25077/jka.v7i0.873>
- Winnarto, M. N., Mailasari, M., & Purnamawati, A. (2022). Klasifikasi Jenis Tumor Otak Menggunakan Arsitektur Mobilenet V2. *Jurnal SIMETRIS*, 13(2), 1–12.
- Yogiswara, I. G. A. I., Saputra, H., & Ekawati, N. P. (2021). Karakteristik Pasien Kanker Kulit Non-Melanoma Di RSUP Sanglah Pada Periode Tahun 2014 - 2018. *Intisari Sains Medis*, 12(2), 691–694. <https://doi.org/10.15562/ism.v12i2.1080>
- YohANNes, Y., Devella, S., & Arianto, K. (2020). Deteksi Penyakit Malaria Menggunakan Convolutional Neural Network Berbasis Saliency. *JUITA: Jurnal Informatika*, 8(1), 37. <https://doi.org/10.30595/juita.v8i1.6671>
- Zaelani, F., & Miftahuddin, Y. (2022). Perbandingan Metode EfficientNetB3 dan MobileNetV2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 9(1), 1–11. <https://doi.org/10.33197/jitter.vol9.iss1.2022.911>