



**DETEKSI AWAL KLASIFIKASI JENIS PENYAKIT KANKER KULIT
DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS MOBILE APPS**

SKRIPSI

**DIMAS PUTRA MAWARDI
NPM 20670021**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG
2024**



**DETEKSI AWAL KLASIFIKASI JENIS PENYAKIT KANKER KULIT
DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS MOBILE APPS**

**Diajukan kepada Fakultas Teknik dan Informatika Universitas PGRI
Semarang untuk Penyusunan Skripsi**

SKRIPSI

DIMAS PUTRA MAWARDI

NPM 20670021

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024

LAPORAN TUGAS AKHIR

**DETEKSI AWAL KLASIFIKASI JENIS PENYAKIT KANKER
KULIT DENGAN ALGORITMA CONVOLUTIONAL NEURAL
NETWORK (CNN) BERBASIS MOBILE APPS**

Disusun dan diajukan oleh

DIMAS PUTRA MAWARDI

NPM 20670021

Telah disetujui oleh pembimbing untuk dilanjutkan untuk
menempuh sidang laporan Tugas Akhir pada tanggal 04 Juni

2024

Pembimbing Utama,

Pembimbing Pendamping



**Mega Novita, S.Si., M.Si., M.Nat.Sc.,
Ph.D**

NIDN 0615118801



**Nugroho Dwi Saputro, S.Kom.,
M.Kom.**

NIDN 0623058802

SKRIPSI
DETEKSI AWAL KLASIFIKASI JENIS PENYAKIT KANKER KULIT
DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK
(CNN) BERBASIS MOBILE APPS

Disusun dan diajukan oleh

DIMAS PUTRA MAWARDI

NPM 20670021


Telah dipertahankan di depan Dewan Penguji pada tanggal 01 Juli 2024 dan
Dinyatakan telah memenuhi syarat Dewan Penguji



Ketua,


Ibnu Toto Husodo, S.T., M.T
NIDN. 0602126902

Sekretaris,

Bambang Agus H, S.Kom., M.Kom
NIDN. 0601088201

Penguji I,

Mega Novita, Ph.D
NIDN. 0615118801

Penguji II,

Nugroho Dwi S, S.Kom., M.Kom
NIDN. 0623058802

Penguji III,

Bambang Agus H, S.Kom., M.Kom
NIDN. 0601088201

MOTTO DAN PEMBAHASAN

MOTTO

"Dengar, Semua orang memiliki gilirannya masing-masing. Bersabarlah dan tunggulah! Itu akan datang dengan sendirinya."

- Gol D. Roger -

"Menangis lalu bangkitlah!!!"

- Akagami No Shanks -

"Shanks pernah berkata, Jika jalannya terlihat terlalu mudah, mungkin kamu berada di jalan yang salah!!!"

- Monkey D. Luffy –

Persembahkan:

Saya persembahkan tugas akhir ini untuk :

1. Kedua orang tua saya.
2. Kakak dan adik saya.
3. Para guru yang telah memberikan ilmu kepada saya.
4. Pacar saya.
5. Teman temanku yang memberikan semangat kepada saya.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Dimas Putra Mawardi
NPM : 20670021
Program Studi : Informatika
Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa tugas akhir yang saya buat ini benar-benar merupakan hasil karya sendiri, bukan plagiarisme.

Apabila pada kemudian hari tugas akhir ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 11 Juni 2024

Yang membuat pernyataan

Dimas Putra Mawardi,
20670021

ABSTRAK

Kulit, sebagai organ luar yang melapisi tubuh manusia, memiliki peran penting dalam menjaga kesehatan dan keutuhan tubuh. Salah satu penyakit yang dapat mengancam kulit adalah kanker kulit, yang dapat menyebabkan kerusakan serius bahkan kematian. Proses diagnosis tradisional untuk kanker kulit sering melibatkan biopsi, yang mahal dan invasif. Faktor risiko untuk kanker kulit termasuk paparan sinar ultraviolet, faktor genetik, pola hidup yang tidak sehat, dan infeksi human papillomavirus. Deteksi dini sangat penting untuk mengatasi kanker kulit, namun diagnosa sering kali sulit karena kesulitan membedakan antara lesi kulit yang berpotensi ganas dan yang tidak. Untuk mengatasi tantangan ini, metode *deep learning* seperti *Convolutional Neural Network* (CNN) menjadi perhatian, karena kemampuannya dalam mengklasifikasikan citra kulit dengan akurasi yang tinggi, memfasilitasi diagnosa kanker kulit secara lebih cepat dan efisien. Penelitian ini menginvestigasi pemanfaatan algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasikan penyakit kanker kulit berdasarkan citra dermoskopi. Dengan dataset berjumlah 10.015 citra yang terbagi menjadi data latih (75%) dan data uji (25%), model yang dikembangkan menggunakan teknik *transfer learning* mampu mencapai tingkat akurasi sebesar 97,38%, dengan akurasi validasi mencapai 97,39% setelah melalui 200 *epoch*. Pengujian yang dilakukan, *white box* yang mendapatkan hasil 2 *independent path* yang berarti tidak kompleks dan mudah untuk perbaikan pada *code* program sehingga dikatakan baik. Pengujian *black box*, maupun UAT (*User Acceptance Testing*), menunjukkan tingkat keberhasilan yang sangat tinggi, mencapai 100%. Selain itu, hasil klasifikasi menunjukkan rata-rata nilai *precision*, *recall*, dan *f1_score* sebesar 97%, menandakan peningkatan signifikan dari penelitian sebelumnya. Penelitian ini juga berhasil mengatasi beberapa kelemahan penelitian sebelumnya dengan meningkatkan jumlah kelas dan dataset yang digunakan, dari dua kelas dan 5.000 citra pada penelitian sebelumnya menjadi tujuh kelas dan 10.015 citra, yang menghasilkan diagnosis yang lebih akurat dan representatif dalam deteksi penyakit kanker kulit.

Kata kunci: Kanker kulit, CNN, *deep learning*, dermoskopi, *transfer learning*, diagnosis awal, deteksi kanker kulit.

PRAKARTA

Puji syukur kami haturkan kepada Allah SWT atas berkat dan rahmat-Nya, sehingga kami berhasil menyelesaikan tugas akhir ini dengan lancar. Tugas akhir berjudul "Klasifikasi Jenis Penyakit Kanker Kulit Dengan Algoritma *Convolutional Neural Network* (CNN) Berbasis *Mobile Apps* " ini kami susun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer.

Kami mengakui bahwa proses penyusunan tugas akhir ini tidaklah mudah dan penuh dengan berbagai hambatan serta rintangan. Namun, berkat bimbingan, dukungan, nasehat, dan dorongan dari berbagai pihak, terutama dari para pembimbing, kami berhasil mengatasi segala kesulitan tersebut. Oleh karena itu, kami ingin mengucapkan terima kasih kepada:

1. Dr. Sri Suciati, M Hum., selaku Rektor Universitas PGRI Semarang.
2. Bapak Ibnu Toto Husodo, ST., MT., selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang, yang telah memberikan izin untuk kami melakukan penelitian.
3. Bapak Bambang Agus Herlambang, S.Kom., M.Kom., selaku Ketua Program Studi Informatika Universitas PGRI Semarang.
4. Ibu Mega Novita, Ph.D., dan Bapak Nugroho Dwi Saputro, S.Kom., M.Kom. selaku Dosen Pembimbing I dan II Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang, yang telah membimbing kami dengan penuh dedikasi.
5. Seluruh Dosen dan Staf di Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang atas ilmu dan bimbingan yang telah diberikan selama masa studi kami.
6. Kedua orang saya Bapak Aris Mawardi dan Ibu Suharni, terima kasih atas doa dan dukungan yang tak henti-hentinya selama proses penyelesaian tugas akhir ini. Dukungan moril dan semangat yang diberikan telah menjadi pendorong utama kami untuk menyelesaikan tugas ini dengan baik.
7. Teruntuk pacar saya selalu memberi dukungan dan semangat tanpa henti kepada saya.

8. Teman-teman dekat kami yang memberikan semangat dalam proses penyusunan tugas akhir ini.
9. Teruntuk teman-teman dekat saya yang tidak bisa disebutkan satu persatu yang telah memberikan semangat dalam penyusunan tugas akhir ini.
10. Teman-teman di Program Studi Informatika Universitas PGRI Semarang.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan memberikan dukungan. Penulis berharap tugas akhir ini dapat memberikan manfaat bagi banyak orang khususnya di bidang Informatika.

Semarang, 11 Juni 2024

Penulis

Dimas Putra Mawardi,

20670021

DAFTAR ISI

SAMPUL LUAR.....	i
SAMPUL DALAM.....	ii
HALAMAN PERSETUJUAN.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
MOTTO DAN PEMBAHASAN	iv
PERNYATAAN KEASLIAN TULISAN	vi
ABSTRAK	vii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	2
C. Tujuan Penelitian	3
D. Rumusan Masalah	3
E. Batasan Masalah.....	3
F. Manfaat Penelitian	4
BAB II KAJIAN TEORI.....	5
A. Tinjauan Pustaka	5
B. Landasan Teori.....	10
C. Kerangka Berfikir.....	27
BAB III METODE PENELITIAN.....	29

A. Pendekatan Penelitian	29
B. Fokus Penelitian	29
C. Desain Penelitian.....	30
D. Teknik Pengumpulan Data.....	32
E. Teknik Analisis Data.....	34
BAB IV HASIL PENGEMBANGAN DAN PEMBAHASAN.....	35
A. Hasil	35
1. Analisis Kebutuhan	35
2. Desain	39
3. Implementasi Perhitungan Model CNN	54
4. Pengujian	79
B. Pembahasan	93
1. Analisis Kebutuhan	93
2. Desain	93
3. Implementasi	94
4. Pengujian	94
BAB V KESIMPULAN DAN SARAN.....	96
A. Kesimpulan	96
B. Saran.....	96
DAFTAR PUSTAKA	98
LAMPIRAN	101

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	5
Tabel 2. 2 Jenis dan Deskripsi Kanker Kulit	10
Tabel 2. 3 <i>Use Case Diagram</i>	20
Tabel 2. 4 Activity Diagram.....	21
Tabel 2. 5 Squence Diagram	22
Tabel 2. 6 Class Diagram	23
Tabel 4. 1 <i>Folder Data Training</i>	35
Tabel 4. 2 Folder Data Testing.....	36
Tabel 4. 3 Kebutuhan Perangkat Keras <i>Mobile Smartphone</i>	36
Tabel 4. 4 Kebutuhan Perangkat Keras <i>Computer</i>	37
Tabel 4. 5 Kebutuhan Perangkat Lunak Computer	38
Tabel 4. 6 Encode Class atau Label	55
Tabel 4. 7 Perbandingan Penelitian.....	62
Tabel 4. 8 Confusion matrix.....	66
Tabel 4. 9 White Box Testing	80
Tabel 4. 10 Black Box Testing.....	84
Tabel 4. 11 Hasil Black Box Testing	86
Tabel 4. 12 User Acceptance Testing (UAT).....	89
Tabel 4. 13 Hasil User Acceptance Testing (UAT)	91

DAFTAR GAMBAR

Gambar 2. 1 Metode Waterfall	18
Gambar 2. 2 Kerangka Berfikir.....	28
Gambar 3. 1 Skema Desain Penelitian.....	31
Gambar 3. 2 Contoh Isi Citra Dataset	33
Gambar 4. 1 Use Case Diagram.....	40
Gambar 4. 2 Activity Diagram Menu Prediksi	41
Gambar 4. 3 <i>Activity Diagram Menu Artikel</i>	42
Gambar 4. 4 Activity Diagram Menu Tentang	43
Gambar 4. 5 Sequence Diagram Menu Prediksi	44
Gambar 4. 6 Sequence Diagram Menu Artikel.....	45
Gambar 4. 7 Sequence Diagram Menu Tentang	46
Gambar 4. 8 <i>Class Diagram</i>	48
Gambar 4. 9 Desain User Interface Halaman Splash Screen	49
Gambar 4. 10 Desain User Interface Halaman Prediksi	50
Gambar 4. 11 Desain User Interface Halaman Navigation.....	51
Gambar 4. 12 Desain User Interface Halaman Artikel	52
Gambar 4. 13 Desain Interface Halaman Tentang.....	53
Gambar 4. 14 Data Preparation.....	54
Gambar 4. 15 <i>Data Visualization</i>	56
Gambar 4. 16 Ringkasan Arsitektur CNN	57
Gambar 4. 17 Pelatihan Model.....	59
Gambar 4. 18 Hasil Pelatihan Model	59
Gambar 4. 19 Evaluasi Model.....	62
Gambar 4. 20 Requiremenst.txt	68
Gambar 4. 21 Pengujian API Localhost di Postman.....	72
Gambar 4. 22 Halaman Splash Screen.....	74
Gambar 4. 23 Halaman Prediksi	75
Gambar 4. 24 Halaman Hasil Prediksi.....	76
Gambar 4. 25 Halaman Navigation.....	77

Gambar 4. 26 Halaman Artikel	78
Gambar 4. 27 Halaman Tentang	79
Gambar 4. 28 Flowgraph Basis Path.....	83

BAB I

PENDAHULUAN

A. Latar Belakang

Kulit merupakan salah satu organ tubuh manusia yang sangat penting karena kulit merupakan lapisan luar yang menutupi tubuh manusia. Kulit memiliki 3 lapisan yang terdiri dari lapisan kulit paling luar (*Epidermis*), lapisan kulit dibawah *Epidermis* dan lapisan kulit dalam (*Hipodermis*) [1]. Kanker kulit salah satu jenis kanker yang dapat menimbulkan kecacatan dengan biaya pengobatan yang tinggi bahkan penyakit ini dapat menyebabkan kematian. Secara umum dokter kulit mendiagnosa kanker kulit menggunakan biopsi. Proses penggunaan biopsi dengan mengambil sampel potongan kecil pada jaringan kulit lalu diperiksa di laboratorium. Biaya yang dikeluarkan untuk biopsi cukup mahal serta dapat melukai atau menggores kulit manusia [2].

Terdapat berbagai faktor penyebab timbulnya kanker kulit yaitu diantaranya faktor peningkatan radiasi sinar *ultraviolet*, faktor genetik, pola hidup yang tidak sehat, dan infeksi *human papillomavirus* [3]. Penentuan penyakit kulit tidak boleh dilakukan secara sembarangan, karena penyakit kulit bisa sangat berbahaya bila terjadi kesalahan dalam perawatan dan penanganannya [4]. Maka melalui fakta tersebut, deteksi dini merupakan salah satu cara untuk mengatasi penyakit kanker kulit. Akan tetapi, pakar dermatologis mengatakan bahwa adanya kesulitan dalam membedakan antara luka bakar dan tahi lalat. Dokter dermatologis dapat mendiagnosis kanker kulit dengan melalui proses biopsi. Biopsi adalah pengambilan sejumlah kecil jaringan tubuh manusia untuk pemeriksaan 2 laboratorium yang bertujuan untuk mendeteksi adanya suatu penyakit, Masalah ini menimbulkan ketertarikan dalam mengklasifikasikan citra kanker kulit untuk memudahkan diagnosa secara klinis. Metode *deep learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah

convolutional neural network (CNN). Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada *visual cortex* manusia sehingga memiliki kemampuan mengolah

Dalam penelitian ini, Teknik *deep learning* (DL) dalam medis memiliki arsitektur pembelajaran mesin, yang didorong dengan kompetensi untuk menangani kumpulan data besar dari perhitungan kompleks dan menghasilkan penilaian yang cukup akurat untuk menangani masalah klasifikasi gambar terutama dalam menganalisis penyakit. Untuk klasifikasi gambar, *convolutional neural network* (CNN) merupakan sebuah metode dalam *deep learning* yang biasa digunakan dalam klasifikasi gambar dalam jumlah besar [5]. CNN memiliki kerangka kerja konseptual yang signifikan, termasuk pembagian bobot, persepsi dan ruang pengambilan sampel domain, yang menjamin perpindahan relative, distorsi dan karakteristik penskalaan. Kemajuan dalam perangkat keras komputasi dan ketersediaan dataset *dermoscopic open source* menjadikan *neural networks* sebagai metode yang efisien untuk klasifikasi gambar.

Oleh sebab itu, penulis melakukan penelitian untuk meningkatkan program klasifikasi jenis penyakit kanker kulit dengan menggunakan teknologi AI. Melalui berbagai penelitian ini, dapat dilihat bahwa algoritma CNN secara konsisten menunjukkan akurasi yang baik dalam klasifikasi citra, termasuk dalam domain medis. Adapun juga judul yang diangkat untuk merancang sebuah aplikasi dan dipaparkan pada penulisan skripsi ini berjudul “DETEKSI AWAL KLASIFIKASI JENIS PENYAKIT KANKER KULIT DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS MOBILE APPS”.

B. Identifikasi Masalah

1. Kanker kulit, terutama melanoma, bisa sangat ganas dan menyebar dengan cepat jika tidak terdeteksi sejak dini. Deteksi awal sangat krusial untuk meningkatkan angka kesembuhan dan mengurangi risiko kematian.

2. Pemeriksaan kulit secara mandiri seringkali kurang akurat karena masyarakat umum tidak memiliki pengetahuan atau alat yang memadai untuk mengenali tanda-tanda kanker kulit.
3. Tidak semua orang memiliki akses ke dokter spesialis kulit atau fasilitas medis untuk pemeriksaan rutin. Aplikasi mobile yang dapat digunakan untuk deteksi awal akan sangat membantu, terutama di daerah terpencil.

C. Tujuan Penelitian

1. Pada penelitian bertujuan guna membangun sebuah sistem yang bisa mengklasifikasi penyakit kanker kulit dengan semaksimal mungkin.
2. Penelitian ini bertujuan mengimplementasi algoritma CNN untuk mencari akurasi yang baik pada klasifikasi kanker kulit.
3. Memberikan informasi mengenai jenis kanker kulit kepada pengguna.

D. Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, rumusan masalah dalam penelitian ini terdiri dari pertanyaan utama mengenai pengimplementasian algoritma *Convolutional Neural Network* (CNN). Penelitian ini berfokus pada bagaimana mengembangkan algoritma CNN untuk mendeteksi dan mengklasifikasikan jenis kanker kulit dengan akurasi tinggi. Deteksi ini dilakukan berdasarkan gambar kulit yang diambil melalui perangkat mobile.

E. Batasan Masalah

1. Sistem pakar ini hanya dapat mengidentifikasi penyakit kanker kulit yakni Solar keratosis (*Actinic Keratos*), Kanker Sel Basal (*Basal Cell Carcinoma*), Keratosis Seboroik (*Benign Keratos-Like Lesions*), Histiositoma Fibrosa Jinak (*Dermatofibroma*), Tahi Lalat (*Melanocytic Nevi*), Kelainan Pembuluh Darah (*Vascular Lesions*), Kanker Kulit Hitam (*Melanoma*).
2. Sistem pakar ini hanya menggunakan menggunakan akses galeri saja dan belum mendukung akses ambil foto.

3. Validasi *convolutional neural network* sebagai metode untuk mengidentifikasi penyakit kulit.
4. Penelitian ini akan menggunakan metode pengembangan perangkat lunak *waterfall*, namun hanya akan dibatasi hingga tahapan pengujian. Tahapan *waterfall* yang akan dilalui meliputi analisis kebutuhan, desain, implementasi, dan pengujian.
5. Pada penelitian ini, terdapat beberapa batasan yang mempengaruhi hasil deteksi penyakit kanker kulit. Dua jenis penyakit yang sulit dideteksi adalah *Dermatofibroma* dan *Vascular Lesions*. Kesulitan ini terutama disebabkan oleh terbatasnya jumlah data dalam dataset yang tersedia.

F. Manfaat Penelitian

1. Membantu pengguna aplikasi untuk lebih memahami jenis kanker kulit mereka dan langkah-langkah perawatan yang sesuai, sehingga dapat meningkatkan kesadaran akan pentingnya merawat kulit dengan baik.
2. Mengatasi kendala biaya tinggi untuk konsultasi dan perawatan di klinik dengan menyediakan opsi yang lebih terjangkau dan mudah diakses oleh masyarakat umum.
3. Memvalidasi penggunaan *Convolutional Neural Network* sebagai metode efektif dalam mengidentifikasi jenis kanker kulit pada tubuh, yang dapat menjadi dasar bagi pengembangan teknologi untuk deteksi dan diagnosis penyakit kulit secara lebih luas.

BAB II

KAJIAN TEORI

A. Tinjauan Pustaka

Tinjauan pustaka yang digunakan dalam penelitian ini disajikan melalui analisa penelitian terdahulu. Namun pada setiap penelitian mempunyai beberapa perbedaan antara satu sama lain yang meliputi dari berbagai sumber penelitian-penelitian terdahulu. Pada Tabel 2.1 menunjukkan tinjauan pustaka dari penelitian terdahulu.

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti dan Tahun Penelitian	Judul Penelitian	Metode	Kesimpulan
1.	E. Rasywir, R. Sinaga, dan Y. Pratama, 2020	Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode <i>Convolutional Neural Network</i> (CNN).	<i>Convolutional Neural Network</i> (CNN).	Melakukan klasifikasi pada penyakit sawit yang menggunakan metode CNN (<i>Convolutional Neural Network</i>) dengan hasil 0,98 yang merupakan ketepatan terbesar dan 0,83 yang merupakan ketepatan

				terendah dengan ketepatan 0,87 sebagai rata-rata.
2.	Y. A. Hasma and W. Silfianti, 2018	Implementasi <i>Deep Learning</i> Menggunakan <i>Tensorflow</i> Dengan Metode <i>Faster Regional Convolutional Neural Network</i> Untuk Pendeteksian Jerawat.	<i>Convolutional Neural Network</i>	melakukan penelitian tentang mendeteksi kondisi kulit wajah dengankamera laptop menggunakan <i>framework TensorFlow Object Detection GUI</i> . Dalam pengenalan bekas luka, jerawat, pus (nanah) yang merupakan kondisi kulit dari model <i>Faster R-CNN</i> . Didapatkan ketepatan keberhasilan deteksi objek 72,4%.
3.	G. Wicaksono, S. Andryana,	Aplikasi Pendeteksi Penyakit Pada Daun	<i>Convolutional Neural Network</i>	melakukan penelitian tentang sistem pendeteksi penyakit apel

	and B, 2020	Tanaman Apel Dengan Metode <i>Convolutional Neural Network.</i>		dengan menggunakan citra daun apel. data yang disiapkan berasal dari <i>SP Mohanty CEO & Cofounder CrowdAI</i> membuat 3151 citra daun dan mendapatkan ketepatan model 99,4%, ketepatan validasi 97,8% dari dataset <i>PlantVillage</i> [6].
4.	Luqman Hakim, Zamah Sari, Handhajani , 2021	Klasifikasi Citra Pigmen Kanker Kulit Menggunakan <i>Convolutional Neural Network</i>	<i>Convolutional Neural Network</i>	Dapat dilihat bahwa model yang diusulkan pada klasifikasi pigmen kanker kulit memperoleh akurasi sebesar 75%. Dengan nilai <i>precision</i> dan <i>recall</i> tertinggi terdapat pada kelas <i>benign</i> yaitu masing-

				masing sebesar 0,80 dan 0,82 serta nilai <i>f1_score</i> sebesar 0,81 [7].
5.	Alfitra Salam, Febi Yanto, Surya Agustian, 2023	Perbandingan Klasifikasi Citra CT-Scan Kanker Paru-Paru Menggunakan <i>Contrast Stretching</i> Pada CNN dengan <i>EfficientNet-B0</i>	<i>Convolutional Neural Network</i> dengan <i>EfficientNet-B0</i>	Hasil klasifikasi tertinggi menggunakan optimizer ADAM, dan learning rate 0.01, di mana akurasi mencapai 72.48%, presisi 71.52%, <i>recall</i> 64.2%, dan skor F1 64.76%. Sementara itu, hasil yang diperoleh dari dataset asli menunjukkan perbedaan. Hasil klasifikasi menggunakan optimizer ADAM dan <i>learning rate</i> 0.01, dengan akurasi mencapai 64.22%, presisi 52.69%, <i>recall</i>

				50.52%, dan skor F1 43.51% [8].
--	--	--	--	---------------------------------

Errissya Rasywir (2020) melakukan klasifikasi penyakit pada tanaman sawit menggunakan metode CNN (*Convolutional Neural Network*). Hasil penelitian menunjukkan tingkat akurasi tertinggi sebesar 0,98 dan tingkat akurasi terendah sebesar 0,83. Rata-rata akurasi yang diperoleh dari penelitian ini adalah 0,87.

Yunita Aulia Hasma (2018) melakukan penelitian tentang mendeteksi kondisi kulit wajah dengan kamera laptop menggunakan framework TensorFlow *Object Detection* GUI. Dalam penelitian ini, bekas luka, jerawat, dan pus (nanah) dikenali sebagai kondisi kulit menggunakan model Faster R-CNN. Hasil penelitian menunjukkan ketepatan keberhasilan deteksi objek sebesar 72,4% [6].

Guntur Wicaksono (2020) melakukan penelitian tentang sistem pendeteksi penyakit apel dengan menggunakan citra daun apel. Data yang digunakan berasal dari SP Mohanty, CEO & *Co-founder* CrowdAI, yang membuat 3151 citra daun. Penelitian ini memperoleh ketepatan model sebesar 99,4% dan ketepatan validasi sebesar 97,8% dari dataset Plant Village.

Luqman Hakim (2021) melakukan klasifikasi citra pigmen kanker kulit menggunakan *Convolutional Neural Network*. Hasil penelitian menunjukkan bahwa model yang diusulkan pada klasifikasi pigmen kanker kulit memperoleh akurasi sebesar 75%. Nilai precision dan recall tertinggi terdapat pada kelas benign, yaitu masing-masing sebesar 0,80 dan 0,82, dengan nilai f1_score sebesar 0,81 [7].

Alfitra Salam (2023) melakukan perbandingan klasifikasi citra CT-Scan kanker paru paru menggunakan *Contrast Stretching* pada CNN dengan *EfficientNet-B0*. Hasil klasifikasi tertinggi menggunakan optimizer ADAM, dan *learning rate* 0.01, di mana akurasi mencapai 72.48%, presisi 71.52%, recall 64.2%, dan skor F1 64.76%. Sementara itu, hasil yang diperoleh dari

dataset asli menunjukkan perbedaan. Hasil klasifikasi menggunakan optimizer ADAM dan *learning rate* 0.01, dengan akurasi mencapai 64.22%, presisi 52.69%, recall 50.52%, dan skor F1 43.51% [8].

B. Landasan Teori

1. Kanker Kulit

Kanker kulit merupakan suatu penyakit yang disebabkan oleh perubahan sifat-sifat penyusun sel kulit yang normal menjadi ganas, dimana nantinya sel-sel akan terus membelah diri menjadi bentuk yang abnormal secara terus menerus tumbuh dan tidak terkendali. Dilihat dari segi histopatologik memiliki struktur yang tidak teratur dengan diferensiasi sel dalam berbagai tingkatan pada *kromatin*, *nukleus*, dan *sitoplasma* [3]. Pada tabel 2. 2 dapat dilihat klasifikasi terhadap jenis kanker kulit beserta deskripsinya.

Tabel 2. 2 Jenis dan Deskripsi Kanker Kulit

Penyakit	Deskripsi
<i>Actinic keratoses</i>	<i>Actinic keratosis</i> atau sering juga disebut solar keratosis, adalah gangguan yang menyebabkan kulit kasar dan bersisik akibat paparan <i>ultraviolet</i> dari sinar matahari dalam waktu lama (umumnya selama bertahun-tahun). Biasanya, kondisi ini terjadi pada wajah, bibir, telinga, lengan, kulit kepala, leher, atau kaki.
<i>Basal cell carcinoma</i>	<i>Basal cell carcinoma</i> salah satu jenis kanker kulit yang ditandai dengan kemunculan benjolan yang mudah berdarah. Benjolan ini dapat bertambah besar seiring berjalannya waktu. Benjolan tersebut timbul pada area tubuh yang

	sering terpapar sinar matahari dan umumnya tidak terasa nyeri.
<i>Benign keratosis-like lesions</i>	<i>Benign keratosis</i> atau sering juga disebut Keratosis seboroik adalah benjolan seperti kutil yang tumbuh di permukaan kulit dan bisa bertambah seiring waktu. Kondisi ini lebih sering terjadi pada orang dewasa, terutama yang lanjut usia. Keratosis seboroik dapat tumbuh di permukaan kulit mana pun.
<i>Dermatofibroma</i>	<i>Dermatofibroma</i> adalah jenis tumor kulit non-kanker yang sangat umum. Tumor ini terdiri dari kombinasi sel fibroblastik, histiosit, dan jenis jaringan ikat yang disebut kolagen. Tumor berkembang di lapisan kulit disebut dermis. Nama lain dari <i>dermatofibroma</i> adalah histiositoma fibrosa jinak.
<i>Melanocytic Nevi</i>	<i>Melanocytic Nevi</i> adalah bercak berwarna yang terdapat di atas permukaan kulit. Bercak berwarna ini berasal dari sel-sel melanosit, yakni pembentuk warna kulit dan rambut. Kondisi ini umumnya muncul saat lahir atau segera setelah lahir, terbanyak pada dewasa muda, dan menurun pada orang tua.
<i>Vascular lesions</i>	<i>Vascular lesions</i> adalah kelainan yang relatif umum terjadi pada kulit dan jaringan di bawahnya, yang lebih dikenal sebagai tanda lahir. Ada tiga kategori

	utama lesi <i>vascular Hemangioma</i> , Malformasi Vaskular, dan Granuloma Piogenik.
<i>Melanoma</i>	<i>Melanoma</i> adalah salah satu jenis kanker kulit yang terjadi akibat pertumbuhan sel melanosit secara tidak normal. Penyakit ini termasuk jenis kanker kulit ganas yang dapat menyebar dengan cepat ke organ tubuh lain, seperti mata, hidung, tenggorokan, otak, dan paru-paru.

2. Deep Learning

Deep Learning adalah bagian dari kecerdasan buatan dan *machine learning* yang merupakan pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa dan lain-lain. *Deep Learning* berbeda dari teknik *machine learning* yang tradisional, karena *deep learning* secara otomatis melakukan representasi dari data seperti gambar, video atau text tanpa memperkenalkan aturan kode atau pengetahuan domain manusia [9]. Selain itu *deep learning* merupakan subbidang *machine learning* yang algoritmanya terinspirasi dari strukturotak manusia, struktur tersebut dinamakan *Artificial Neural Networks* atau disingkat ANN. Pada dasarnya, ia merupakan jaringan saraf yang memiliki tiga atau lebih lapisan ANN. Ia mampu belajar dan beradaptasi terhadap sejumlah besar data serta me-nyelesaikan berbagai permasalahan yang sulit diselesaikan dengan algoritma *machine learning* lainnya.

3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu algoritma paling populer digunakan untuk *Deep Learning*, sebuah *machine learning* yang model pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada media dua dimensi seperti gambar, video, teks atau suara. Algoritma CNN akan sangat berguna khususnya ketika digunakan untuk mencari pola pada suatu gambar kemudian mengenali objek pada gambar tersebut [10]. Bukan hanya pada objek atau benda saja, CNN ini sebenarnya juga bisa digunakan untuk mengenali jenis kanker kulit yang selama ini perlu segmentasi untuk meningkatkan akurasi. Penelitian awal yang mendasari CNN ini pertama kali dilakukan oleh Hubel dan Wiesel mengenai *visual cortex* pada indera penglihatan kucing. Pada dasarnya klasifikasi citra menggunakan MLP sudah bisa dilakukan, akan tetapi ketika digunakan untuk melakukan klasifikasi data dalam jumlah banyak, akurasi yang didapatkannya pun menurun. Oleh karena itu, algoritma CNN ini dikembangkan karena algoritma ini mampu untuk mempelajari langsung data yang ada pada gambar, kemudian menggunakan pola yang didapatkan untuk mengklasifikasi.

4. Pemrosesan Citra

Untuk membantu dalam diagnosis dan pengenalan kanker kulit, teknologi pemrosesan citra telah berkembang pesat, termasuk penggunaan CNN sebagai metode yang efektif. CNN merupakan algoritma *deep learning* yang telah terbukti efektif dalam pengenalan pola pada gambar. CNN adalah algoritma *deep learning* yang sangat efektif dalam memproses dan menganalisis data citra. Dalam konteks kanker kulit, penggunaan metode CNN dapat membantu membedakan antara kanker kulit benign dan malignant secara otomatis melalui analisis citra kulit. Hal ini dapat memberikan bantuan penting bagi para profesional medis dalam mengambil keputusan yang tepat mengenai

pengobatan dan tindak lanjut pasien [11]. Langkah pengolahan citra pada klasifikasi citra menggunakan *convolutional neural network* (CNN) meliputi beberapa langkah utama. Pertama, gambar dimasukkan ke dalam model CNN yang terdiri dari lapisan konvolusional, lapisan aktivasi, dan lapisan pengumpulan untuk mengekstrak fitur-fitur penting. Hasilnya kemudian ditarik ke lapisan yang terhubung sepenuhnya untuk klasifikasi. Setelah model dilatih menggunakan data pelatihan dengan label yang diketahui, akurasi diukur menggunakan data pengujian yang tidak terlihat. Akurasi model dicapai dengan membandingkan prediksi lapisan model dengan label sebenarnya pada data pengujian.

5. Machine Learning

Machine learning atau pembelajaran mesin, cabang dari AI-kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengem-bangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data [12]. Hal Ini merupakan teknik yang digunakan untuk mengembangkan mesin otomatis berdasarkan eksekusi pada algoritma dan kumpulan aturan yang terdefiniskan. Pada *Machine learning* dilengkapi sejumlah aturan program yang dijalankan oleh algoritma, oleh karena itu pada teknik mesin belajar dapat dikategorikan sebagai instruksi yang dijalankan dan dipelajari secara otomatis untuk menghasilkan *output* yang optimal, hal ini dilakukan secara otomatis tanpa ada campur tangan manusia sedikitpun. Semua dilakukan secara otomatis untuk mengubah data menjadi beberapa pola dan diinputkan jauh ke dalam sistem untuk mendeteksi masalah otomatis.

Data-data yang masuk ke dalam mesin akan dianalisis, yang kemudian menghasilkan prediksi, saran, maupun keputusan. *Machine learning* yang lebih dalam nantinya disebut sebagai *Deep learning*.

Beberapa contoh *machine learning* yang sering kita temui. Optimasi iklan dalam strategi digital mar-keting, penerjemah tulisan tangan menjadi teks *software* pengecekan terjemahan dan tata bahasa. Aplikasi teknologi *machine learning* ternyata ada dalam berbagai bentuk yang sangat akrab dengan aktivitas sehari-hari, mulai dari transportasi, teknologi, finansial, pendidikan, kesehatan, dan juga media sosial yang sering di kunjungi.

6. Pengolahan Data

Pengolahan citra digital (*Digital Image Processing*) merupakan disiplin ilmu yang mempelajari teknik dalam mengolah citra, citra yang dimaksud adalah merupakan gambar diam (foto) [13]. Sedangkan arti digital adalah pengolahan citra/gambar dilakukan menggunakan komputer secara digital. RGB adalah singkatan dari *Red-Green-Blue*, merupakan tiga warna dasar (*primary colors*) yang secara umum dijadikan acuan warna lainnya. Dari basis RGB, kita dapat mengkonversi warna menjadi kode-kode angka yang membuat warna tersebut akan tampil universal. Komputer sudah mengemas informasi warna menjadi model warna yang sama sehingga membuat pengolahan warna RGB dapat dilakukan dengan mudah.

7. Dataset

Dataset memegang peran sentral dalam pengembangan dan evaluasi model pembelajaran mesin. Kualitas dataset tidak hanya memengaruhi kinerja model yang dihasilkan, tetapi juga dapat mempengaruhi generalisasi model terhadap situasi dunia nyata. Oleh karena itu, dataset yang baik harus mencakup variasi yang memadai dari data yang ditemui dalam kehidupan nyata, mengurangi sebanyak mungkin bias yang mungkin terjadi, dan memberikan representasi yang seimbang dari kelas atau label yang berbeda untuk memastikan model dapat mempelajari dengan baik berbagai fitur dan pola yang mungkin ada. Dataset diperoleh dari kumpulan data *International Skin Imaging*

Collaboration (ISIC) berupa citra digital yang terdiri dari beberapa kategori penyakit kanker kulit [14].

8. Python

Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido van Rossum pada tahun 1991. Python adalah Interpreter yang memiliki kemampuan *exception handling* serta agar sintaksnya yang mudah dipahami Python menggunakan indentasi untuk membedakan *blok kode*, tidak seperti bahasa pemrograman lain yang menggunakan simbol sebagai pembeda *blok kode*. Secara desain pun Python dibuat agar mudah dipahami, salah satunya yaitu tidak diwajibkan untuk menggunakan ‘**semicolon(;)**’ [15]. Pengadopsian *dynamic typing*, yaitu variable yang ingin dibuat tidak diketahui tipe datanya sampai ia dipanggil untuk pertama kali atau dieksekusi, tidak perlu untuk mendeklarasikan variable dan memungkinkan perubahan tipe data dalam proses eksekusi program.

9. Android

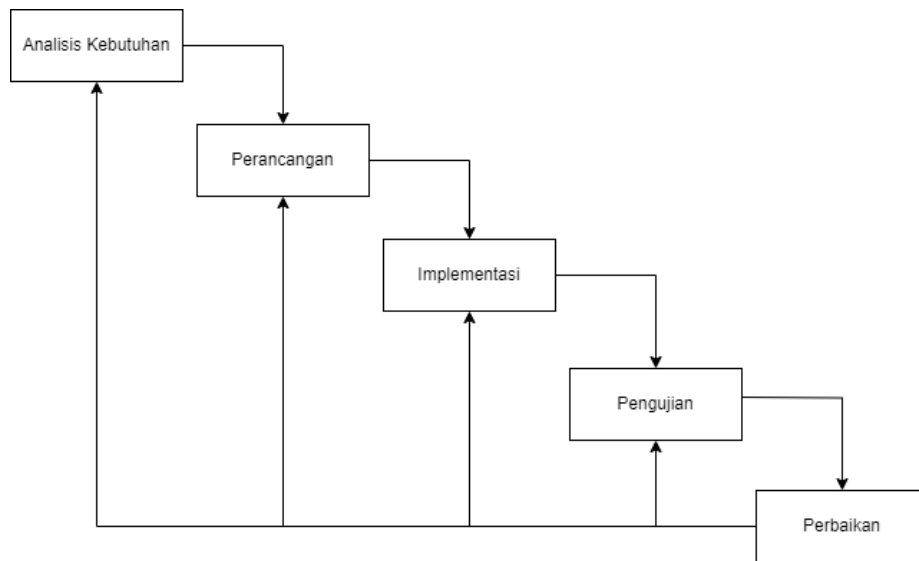
Android merupakan sistem operasi mobile yang dikembangkan oleh Google, dirancang khusus untuk perangkat bergerak seperti *smartphone*, tablet, dan perangkat *wearable*. Didasarkan pada kernel Linux, Android menawarkan lingkungan pengembangan terbuka yang memungkinkan para pengembang untuk membuat aplikasi dan perangkat lunak yang berjalan pada berbagai perangkat dengan berbagai ukuran layar dan fitur. Android sangat populer karena fleksibilitasnya, kemampuan kustomisasi yang tinggi, dan integrasinya dengan berbagai layanan Google, seperti Google Play Store, Gmail, Google Maps, dan lainnya. Pengembang aplikasi Android menggunakan bahasa pemrograman Java atau Kotlin, serta SDK (*Software Development Kit*) Android yang disediakan oleh Google untuk membuat aplikasi yang beraneka ragam, mulai dari aplikasi produktivitas hingga game yang

canggih. Android untuk menerapkan model *TensorFlow lite* dan membuat aplikasi [16].

Java adalah bahasa pemrograman yang populer, dikembangkan oleh Sun *Microsystems*. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi native untuk android [17]. Bahasa pemrograman ini bersifat multiplatform yakni bahasa ini dapat digunakan di berbagai platform, seperti desktop, android dan bahkan untuk sistem operasi Linux.

10. Metode *Waterfall*

Metode *waterfall* merupakan salah satu pendekatan dalam pengembangan perangkat lunak yang mengikuti proses yang linear dan terstruktur. Dalam metode ini, pengembangan perangkat lunak dilakukan secara bertahap dari awal hingga selesai, dengan tahapan-tahapan yang terdefinisi dengan jelas dan dilakukan secara berurutan seperti air terjun (*waterfall*). Penelitian ini dikembangkan dengan metode *Software Development Life Cycle waterfall* untuk membuat aplikasi. Dimulai dengan mengumpulkan studi literatur mengenai penelitian sebelumnya, analisis kebutuhan, perancangan, implementasi aplikasi, pengujian dan perbaikan [18]. Dimulai dari Analisis Kebutuhan, di mana kebutuhan pengguna atau klien dikumpulkan dan dianalisis secara menyeluruh. Tahap berikutnya adalah Perancangan, di mana arsitektur dan tata letak sistem dirancang berdasarkan kebutuhan yang telah diidentifikasi. Setelah desain selesai, tahap Implementasi dimulai dengan penulisan kode dan pembangunan perangkat lunak. Kemudian, perangkat lunak tersebut diuji dalam tahap Pengujian untuk memastikan tidak ada bug dan semua fungsi bekerja sesuai dengan spesifikasi. Tahap terakhir adalah Perbaikan, di mana bug dan masalah yang ditemukan selama pengujian diperbaiki. Setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya.



Gambar 2. 1 Metode *Waterfall* [18]

11. Unified Modeling Language (UML)

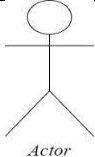


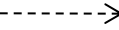


Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek [19]. Dengan menggunakan UML, pengembang dapat menggambarkan struktur statis dari sistem menggunakan diagram kelas, objek, dan komponen, sementara perilaku dinamis sistem dapat diilustrasikan melalui diagram aktivitas, urutan, dan keadaan. Interaksi antar komponen dalam sistem dapat dijelaskan dengan jelas menggunakan diagram interaksi seperti diagram sekuen dan kolaborasi. Selain itu, UML juga memfasilitasi dokumentasi yang komprehensif dan terstruktur, yang sangat berguna dalam memahami dan menjelaskan detail sistem kepada pemangku kepentingan yang berbeda. Dengan demikian, UML tidak hanya membantu dalam perancangan sistem yang lebih efisien, tetapi juga meningkatkan komunikasi dan pemahaman antara berbagai pihak yang terlibat dalam pengembangan perangkat lunak. UML terdiri dari berbagai jenis diagram, termasuk diagram kelas,

diagram sekuen, diagram aktivitas, diagram kasus pengguna, dan banyak lagi, yang masing-masing memiliki tujuan dan notasi yang spesifik.

A. *Use Case Diagram*

Use Case Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna atau aktor eksternal. Diagram ini memberikan gambaran tentang interaksi antara sistem yang akan dibangun dengan aktor-aktor eksternal yang berinteraksi dengannya. *Use case diagram* adalah sebuah diagram yang menunjukkan hubungan antara *actors* dan *use cases* [20]. *Aktor* dalam *use case* diagram mewakili pengguna sistem atau entitas luar yang berinteraksi dengan sistem, sedangkan *use case* menggambarkan fungsi-fungsi atau tujuan yang ingin dicapai oleh aktor tersebut dalam interaksi dengan sistem. *Use case* diagram membantu dalam pemahaman kebutuhan fungsional sistem dan membantu dalam merancang solusi yang memenuhi kebutuhan pengguna dengan jelas dan terstruktur.

Tabel 2. 3 Use Case Diagram







Simbol	Keterangan
	Aktor : Mengambarkan seseorang yang berinteraksi dengan system atau alat ketika berkomenikasi dengan <i>use case</i> .
	<i>Usecase</i> : Menggambarkan fungsional sistem yang akan dibuat, agar <i>user</i> lebih memahami penggunaan sistem.
	<i>Association</i> : Penghubung antara aktor dengan <i>use case</i> .
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
<< <i>include</i> >> 	<i>Include</i> : Hubungan antara <i>use case</i> dengan <i>usecase</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
<< <i>extend</i> >> 	<i>Extend</i> : Hubungan antara <i>usecase</i> dengan <i>usecase extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

B. Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari

Use Case yang memiliki alur aktivitas. Alur atau aktivitas berupa bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. Dalam buku *Rekayasa Perangkat Lunak* karangan Rosa A.S mengatakan [21]. Diagram aktivitas tidak menjelaskan kelakuan aktor. Dapat diartikan bahwa dalam pembuatan *Activity Diagram* hanya dapat dipakai untuk menggambarkan alur kerja atau aktivitas sistem saja.

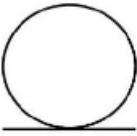
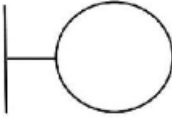
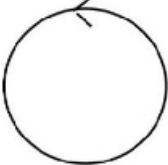
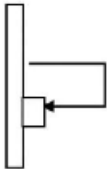
Tabel 2. 4 *Activity Diagram*



Simbol	Keterangan
	Status awal: Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas: Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i> : Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / <i>Join</i> : Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir: Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i> : <i>Swimlane</i> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

C. Sequence Diagram

Sequence diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, display, dan sebagainya berupa message yang digambarkan terhadap waktu [22]. Diagram ini memberikan representasi visual tentang urutan pesan atau panggilan yang dikirimkan antara objek-objek selama eksekusi suatu skenario. Objek-objek yang terlibat dalam interaksi direpresentasikan sebagai kotak dengan nama di dalamnya, sedangkan pesan atau panggilan antar objek direpresentasikan sebagai garis-garis vertikal yang menghubungkan objek-objek tersebut.

Tabel 2. 5 *Sequence Diagram*


Simbol	Keterangan
	<i>Entity Class</i> : Gambaran sistem sebagai landasan dalam menyusun basis data.
	<i>Boundary Class</i> : Menangani komunikasi antar lingkungan sistem .
	<i>Control Class</i> : Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
	<i>Recursive</i> : Pesan untuk dirinya.


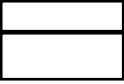

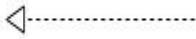


	<p><i>Activation</i>: Mewakili proses durasi aktivasi sebuah operasi.</p>
	<p><i>Life Line</i>: Komponen yang digambarkan garis putus terhubung dengan objek.</p>

D. Class Diagram

Class diagram adalah salah satu pemodelan yang cukup penting dalam UML, fungsinya adalah untuk membuat sebuah logical models dari sebuah *system*. Sebuah *class* diagramakan menunjukkan bagaimanaskema dari arsitektur sebuah sistem yang sedang dirancang. *Class diagram* digambarkan dengan class yang berisi atribut dan method, setiap *class* akan dihubungkan dengan sebuah garis disebut asosiasi [23]. Diagram ini memberikan representasi visual tentang struktur objek dalam sistem dan bagaimaimana mereka berinteraksi satu sama lain. Setiap kelas direpresentasikan sebagai sebuah kotak dengan tiga bagian, yaitu bagian atas untuk nama kelas, bagian tengah untuk atribut-atribut kelas, dan bagian bawah untuk metode-metode kelas.

Tabel 2. 6 *Class Diagram*

Simbol	Keterangan
	<p><i>Generalization</i>: Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).</p>

	<p><i>Nary Association</i>: Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.</p>
	<p><i>Class</i>: Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.</p>
	<p><i>Collaboration</i>: Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.</p>
	<p><i>Realization</i>: Operasi yang benar-benar dilakukan oleh suatu objek.</p>
	<p><i>Dependency</i>: Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.</p>
	<p><i>Association</i>: Apa yang menghubungkan antara objek satu dengan objek lainnya.</p>

12. Pengujian *White Box*

Pengujian *White Box* adalah metode pengujian perangkat lunak yang memeriksa struktur internal atau kode sumber aplikasi. Metode ini digunakan untuk mendeteksi kesalahan logis dalam kode program, membantu *debugging*, menemukan kesalahan ketik, dan mengungkap asumsi pemrograman yang keliru. Pengujian ini berfokus pada verifikasi jalur, kondisi, dan aliran data dalam kode [24]. Berikut adalah

beberapa metode dan rumus yang digunakan dalam pengujian white box.

1. Coverage Metrics

a. Statement Coverage (C1)

Mengukur persentase pernyataan (*statement*) dalam kode yang telah dieksekusi setidaknya sekali selama pengujian.

Statement Coverage =

$$\left(\frac{\text{Number of Statements Executed}}{\text{Total Number of Statements}} \right) \times 100\% \quad (2.1)$$

b. Branch Coverage (C2)

Mengukur persentase cabang (*branch*) atau keputusan (*decision*) yang telah dieksekusi dalam pengujian.

Branch Coverage =

$$\left(\frac{\text{Number of Executed Branches}}{\text{Total Number of Branches}} \right) \times 100\% \quad (2.2)$$

c. Condition Coverage (C3)

Mengukur persentase kondisi boolean dalam pernyataan yang telah dievaluasi menjadi benar dan salah.

Condition Coverage =

$$\left(\frac{\text{Number of Executed Conditions}}{\text{Total Number of Conditions}} \right) \times 100\% \quad (2.3)$$

2. Cyclomatic Complexity (McCabe's Complexity)

Mengukur kompleksitas logis dari sebuah program. Rumusnya adalah:

$$V(G) = E - N + 2P \quad (2.4)$$

di mana:

- 1) $V(G)$ adalah kompleksitas siklomatik.
- 2) E adalah jumlah *edge* dalam grafik kontrol aliran.
- 3) N adalah jumlah *node* dalam grafik kontrol aliran.

4) P adalah jumlah komponen terhubung.

3. *Path Coverage* (C4)

Mengukur persentase jalur eksekusi dalam program yang telah diuji.

Path Coverage =

$$\left(\frac{\text{Number of Executed Paths}}{\text{Total Number of Paths}} \right) \times 100\% \quad (2.5)$$

4. *Multiple Condition Coverage* (C5)

Mengukur persentase dari semua kombinasi kondisi yang dievaluasi selama pengujian.

Multiple Condition Coverage =

$$\left(\frac{\text{Number of Condition Combinations Executed}}{\text{Total Number of Condition Combinations}} \right) \times 100\% \quad (2.6)$$

5. *Basis Path Testing*

Menggunakan rumus kompleksitas siklomatik untuk menentukan jumlah jalur uji dasar (*basis paths*) yang mencakup setiap kemungkinan jalur eksekusi.

- 1) Identifikasi semua jalur independen dalam kode.
- 2) Uji setiap jalur independen.

13. Pengujian *Black Box*

Pengujian *Black Box* adalah metode penting dalam pengujian perangkat lunak yang dilakukan tanpa memerhatikan struktur internal sistem atau komponen yang diuji. Dalam pengujian ini, sistem atau komponen dianggap sebagai sebuah "kotak hitam", di mana pengujian dilakukan berdasarkan pada spesifikasi fungsional atau persyaratan pengguna. Sebelum dirilis untuk digunakan oleh pengguna, aplikasi ini harus menjalankan Pengujian *black box* disini akan berfokus pada output yang dikeluarkan ketika setiap elemen pada halaman antarmuka dijalankan [25].

Metode ini melibatkan penggunaan teknik-teknik seperti pengujian fungsional, pengujian *boundary*, pengujian kasus uji negatif, dan pengujian kasus uji acak. Pendekatan ini sangat penting karena memungkinkan penguji untuk mengevaluasi sistem dari perspektif pengguna atau pemangku kepentingan, memastikan bahwa perilaku sistem sesuai dengan harapan dan persyaratan yang telah ditetapkan. Dengan memanfaatkan metode pengujian *black box* ini secara efektif, tim pengembangan dapat mengidentifikasi dan memperbaiki cacat sistem sebelum diperkenalkan ke lingkungan produksi, meningkatkan kualitas dan keandalan sistem secara keseluruhan.

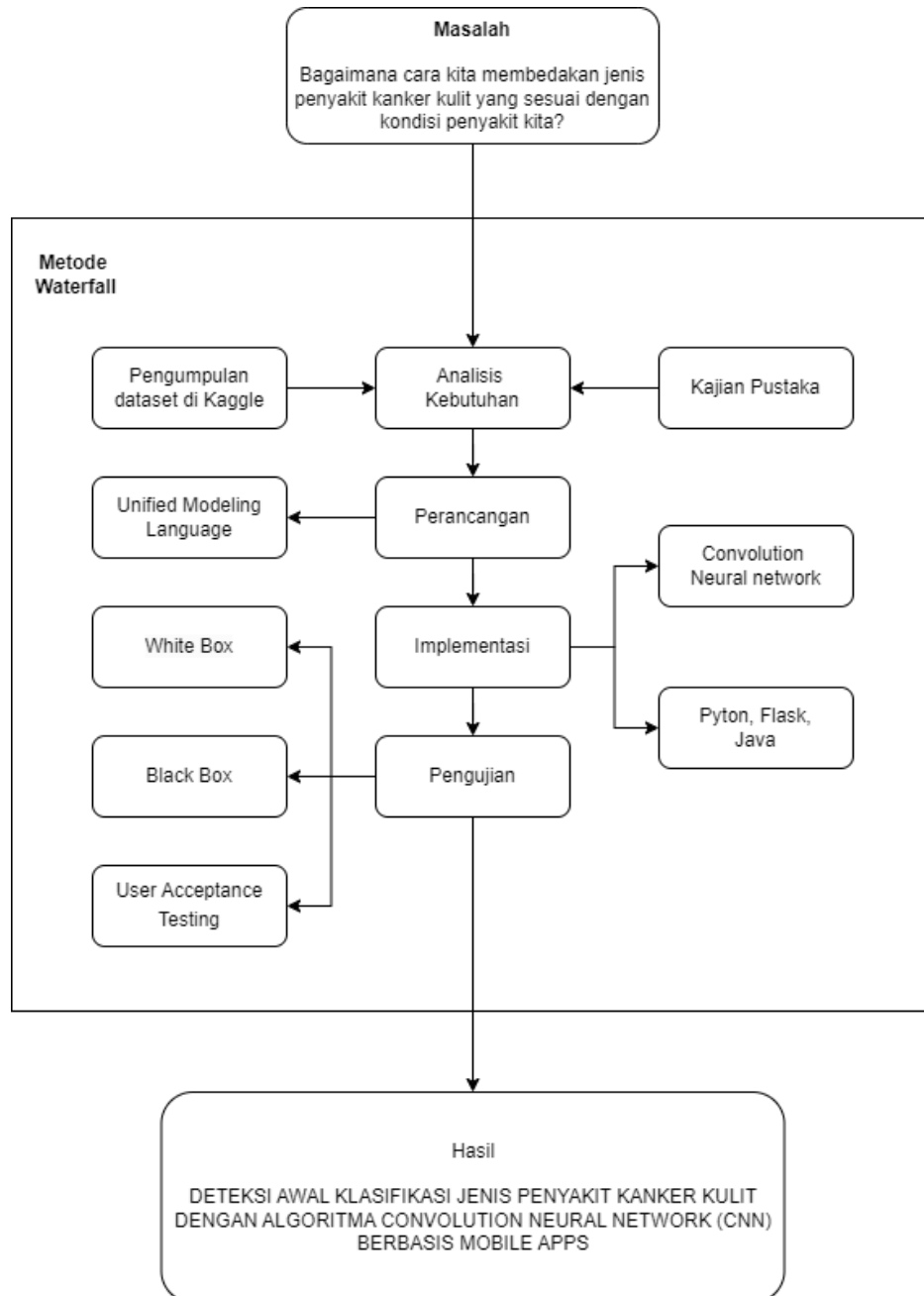
14. *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah tahap penting dalam siklus pengembangan perangkat lunak di mana produk akhir diuji oleh pengguna akhir untuk memastikan bahwa produk tersebut memenuhi persyaratan dan berfungsi seperti yang diharapkan dalam situasi nyata. berfungsi sebagai pos pemeriksaan terakhir untuk memvalidasi perangkat lunak terhadap kebutuhan bisnis dan pengguna. Selama UAT, perangkat lunak diuji dalam lingkungan yang meniru lingkungan produksi menggunakan data dan skenario dunia nyata, yang membantu mengidentifikasi masalah yang mungkin terlewatkan selama fase pengujian sebelumnya [26].

C. Kerangka Berfikir

Kerangka berfikir untuk klasifikasi jenis kanker kulit dimulai dengan pemilihan dataset yang mencakup berbagai jenis kanker kulit dan pra-pemrosesan data, termasuk normalisasi dan augmentasi. Selanjutnya, model CNN dirancang dan dilatih menggunakan dataset pelatihan, diikuti oleh evaluasi kinerja menggunakan dataset validasi dan penyetelan parameter model. Setelah model teruji, dilakukan pengujian pada dataset pengujian terpisah untuk validasi akhir. Akhirnya, model yang telah terlatih dan

divalidasi diterapkan dalam aplikasi atau sistem yang relevan untuk penggunaan praktis dalam mengidentifikasi dan mengklasifikasikan jenis kanker kulit secara akurat.



Gambar 2. 2 Kerangka Berfikir

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Pendekatan penelitian yang digunakan dalam studi ini adalah metode *waterfall*, yang terbukti efektif dalam pengembangan sistem secara sistematis dan terstruktur. *Waterfall* dipilih karena kesesuaiannya dengan langkah-langkah yang diperlukan dalam pengembangan sistem klasifikasi jenis penyakit kanker kulit menggunakan *Convolutional Neural Network* (CNN). Metode ini mengikuti alur yang jelas, dimulai dari pemahaman atas kebutuhan sistem, kemudian melanjutkan ke tahap analisis data citra dermatoskopi, desain arsitektur CNN, implementasi kode program, pengujian model klasifikasi, hingga pemeliharaan model untuk memastikan kehandalan dan performa yang optimal.

Analogi 'air terjun' dalam metode *waterfall* mencerminkan proses berurutan yang terjadi dalam pengembangan sistem ini. Seperti air yang mengalir secara bertahap dari satu tingkat ke tingkat berikutnya, penelitian ini mengalami perjalanan yang sama, dimulai dari identifikasi masalah kanker kulit, kemudian menuju tahap pelabelan data, pelatihan model CNN, evaluasi performa model, dan akhirnya pemeliharaan model untuk memperbarui dan meningkatkan kemampuannya. Dengan pendekatan ini, diharapkan penelitian ini dapat memberikan kontribusi yang berarti dalam upaya deteksi dini dan diagnosis yang akurat terhadap penyakit kanker kulit.

B. Fokus Penelitian

Pada penelitian ini, fokus utama adalah pengembangan aplikasi berbasis Android yang dapat beroperasi dengan baik dan efisien. Aplikasi ini bertujuan untuk mengklasifikasikan jenis penyakit kanker kulit berdasarkan foto yang diambil oleh pengguna.

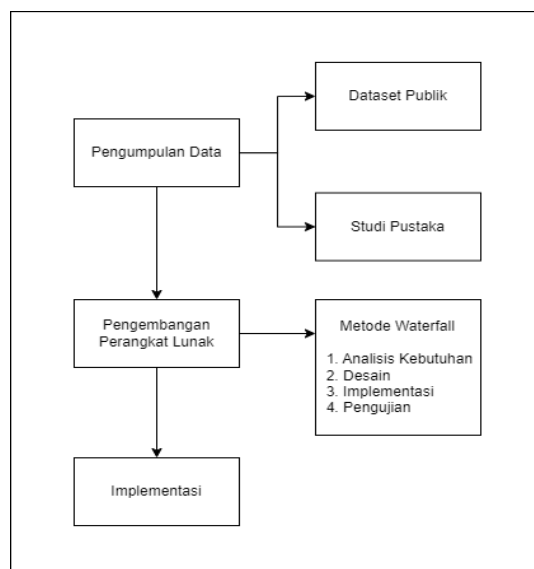
1. Aplikasi berjalan dengan baik dan berbasis android
Penelitian ini akan memastikan bahwa aplikasi yang dikembangkan dapat berjalan dengan baik dan lancar pada platform android. Hal ini termasuk dalam aspek desain antarmuka pengguna yang mudah digunakan dan responsif.
2. Aplikasi mengklasifikasikan jenis penyakit kanker kulit
Fokus utama dari aplikasi adalah kemampuannya dalam mengenali dan mengklasifikasikan jenis penyakit kanker kulit berdasarkan foto yang diambil oleh pengguna. Proses klasifikasi ini akan menggunakan teknologi pengolahan citra dan kecerdasan buatan, termasuk penggunaan algoritma *Convolutional Neural Network* (CNN) untuk analisis gambar.
3. Target pengguna aplikasi adalah mahasiswa dan masyarakat umum
Aplikasi ini ditujukan untuk pengguna dari berbagai latar belakang, terutama mahasiswa dan masyarakat umum yang peduli terhadap kesehatan kulit dan pencegahan penyakit kanker kulit. Desain aplikasi akan memperhatikan kebutuhan dan preferensi pengguna yang beragam.

C. Desain Penelitian

Desain penelitian untuk Aplikasi Pendeteksi Penyakit Kanker Kulit menggunakan Algoritma *Convolutional Neural Network* (CNN) berbasis *Mobile Apps* akan mengikuti pendekatan *Waterfall* dengan tahapan- tahapan berikut.

1. **Tahap Analisis:** Pada tahap ini, peneliti akan mengumpulkan dataset dari kaggle dan studi pustaka persyaratan untuk pengembangan sistem klasifikasi jenis penyakit kanker kulit. Peneliti akan mengidentifikasi jenis data yang diperlukan, termasuk data citra dermatoskopi yang relevan dan berkualitas tinggi untuk pelatihan model klasifikasi. Selain itu, peneliti akan melakukan analisis tentang jenis-jenis penyakit kanker kulit yang perlu diidentifikasi oleh sistem.

2. **Tahap Perancangan:** Pada tahap ini, peneliti akan merancang arsitektur desain *Unified Modeling Language* (UML), termasuk pemilihan dan penyesuaian algoritma *Convolutional Neural Network* (CNN) yang tepat untuk klasifikasi jenis penyakit kanker kulit. Selain itu, akan dirancang pula desain antarmuka pengguna yang intuitif untuk memfasilitasi pengguna dalam mengambil dan memasukkan foto kulit untuk analisis.
3. **Tahap Implementasi:** Tahap ini melibatkan pembuatan kode pemrogram sistem seperti Java dan Python, klasifikasi berdasarkan desain yang telah dibuat sebelumnya. Peneliti akan mengimplementasikan algoritma CNN dan membangun sistem berbasis android yang mampu menerima input foto kulit pengguna dan mengeluarkan output berupa klasifikasi jenis penyakit kanker kulit.
4. **Tahap Pengujian:** Tahap ini akan dilakukan pengujian *white box*, *black box* dan uat klasifikasi yang dikembangkan sesuai dengan desainnya dan berfungsi dengan baik. Sistem akan diuji dengan menggunakan dataset yang relevan dan diverifikasi oleh tim penguji untuk memeriksa akurasi dan keandalannya dalam mengklasifikasikan jenis penyakit kanker kulit dari foto yang diambil oleh pengguna.



Gambar 3. 1 Skema Desain Penelitian

D. Teknik Pengumpulan Data

1. Dataset Publik

Pada penelitian ini, peneliti mengumpulkan data berupa citra kulit dari dataset HAM10000 yang tersedia secara publik di *Kaggle.com*. Dataset ini terdiri dari berbagai jenis lesi kulit yang diklasifikasikan ke dalam 7 kelas yakni *Melanocytic nevi* (nv), *Melanoma* (mel), *Benign keratosis-like lesions* (bkl), *Basal cell carcinoma* (bcc), *Actinic keratoses* (akiec), *Vascular lesions* (vasc), dan *Dermatofibroma* (df). Data tersebut terdiri dari 10.015 sampel citra dan metadata yang terkait, seperti jenis lesi, jenis kelamin pasien, dan lokasi lesi pada tubuh. Setelah mengunduh dataset, peneliti menyimpannya di *Google Drive* untuk penyimpanan yang aman.

Selanjutnya, peneliti melakukan pra-pemrosesan data, termasuk konversi label teks menjadi nilai numerik menggunakan *LabelEncoder*, visualisasi distribusi kelas dan fitur lainnya, serta penyeimbangan distribusi kelas menggunakan teknik resampling. Setelah itu, peneliti membagi dataset menjadi data *training* (75%) dan data *testing* (25%) menggunakan fungsi '**train_test_split**' dari *library Scikit-learn*. Setelah pra-pemrosesan selesai, peneliti memuat citra dari file yang sesuai menggunakan path yang telah ditentukan. Citra-citra ini kemudian diubah menjadi larik numpy dan dinormalisasi agar nilai pikselnya berada dalam rentang 0-1. Peneliti membangun dan melatih model *Convolutional Neural Network* (CNN) menggunakan *library keras*. Model ini memiliki beberapa lapisan konvolusi dan lapisan tersembunyi lainnya, seperti lapisan *dropout* untuk mencegah *overfitting*. Untuk contoh citra dataset dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Contoh Isi Citra Dataset

2. Studi Pustaka

Studi pustaka merupakan fase yang krusial dalam proses penelitian. Dimana peneliti memperdalam pemahaman mereka tentang topik yang diteliti serta memperkuat dasar teoritis yang telah ada sebelumnya. Dalam konteks penelitian tentang klasifikasi lesi kulit menggunakan dataset HAM10000, studi pustaka dilakukan dengan melakukan pencarian, pengumpulan, dan analisis literatur serta karya ilmiah yang berkaitan dengan pemrosesan citra medis dan pembelajaran mesin. Beberapa sumber yang digunakan dalam studi pustaka mencakup.

- a) Buku-buku yang fokus pada dermatologi dan teknologi pemrosesan citra medis, yang memberikan landasan teoritis tentang berbagai

jenis lesi kulit, ciri visual yang penting, dan metode analisis citra yang relevan untuk penelitian ini.

- b) Jurnal-jurnal ilmiah yang menerbitkan penelitian terkait klasifikasi lesi kulit menggunakan teknologi pemrosesan citra dan pembelajaran mesin. Di dalam jurnal-jurnal tersebut, peneliti dapat menemukan informasi mengenai perkembangan terkini dalam teknik analisis, dataset yang digunakan, serta evaluasi kinerja model klasifikasi.
- c) Sumber-sumber online seperti website dan internet dimanfaatkan untuk mencari informasi terkait dataset HAM10000 dan penelitian-penelitian sebelumnya yang menggunakan dataset tersebut. Platform seperti forum online, blog, dan Kaggle juga memberikan wawasan tambahan mengenai penggunaan dataset secara efektif dan hasil-hasil penelitian yang relevan.

E. Teknik Analisis Data

Berikut adalah teknik analisis data yang dilakukan setelah semua data yang dibutuhkan terkumpul dalam penelitian ini.

1. Pengumpulan seluruh data citra yang diperlukan dilakukan dengan mengunduhnya dari sumber data yang tersedia, yaitu Kaggle.
2. Dilakukan analisis kebutuhan pengguna untuk memastikan bahwa aplikasi yang akan dibuat memenuhi kebutuhan mereka.
3. Data yang telah terkumpul diolah dan diproses menggunakan metode atau algoritma yang telah ditentukan agar dapat diimplementasikan menjadi aplikasi sesuai dengan perencanaan yang telah disusun sebelumnya.

BAB IV

HASIL PENGEMBANGAN DAN PEMBAHASAN

A. Hasil

1. Analisis Kebutuhan

a. Analisis Kebutuhan Data

Dalam penelitian ini, penulis bertujuan untuk melatih model klasifikasi citra dengan menggunakan dataset yang diunduh dari Kaggle, sebuah platform yang menyediakan beragam dataset untuk proyek-proyek penelitian. Setelah penulis mengunduh data citra tersebut, penulis memindahkannya ke *Google Drive* untuk penyimpanan yang lebih aman dan mudah diakses. Data akan dibagi menjadi dua *folder*, yaitu *folder data training* dan *data testing*, dengan proporsi 75% untuk data *training* dan 25% untuk data *testing*.

1) *Data Training*

Folder Data Training digunakan untuk tempat penyimpanan dataset citra yang dipakai dalam proses pelatihan model. Perincian jumlah file citra yang disimpan dalam folder data pelatihan tercantum pada Tabel 4.1. Informasi tersebut membantu dalam memahami distribusi dan volume dataset yang digunakan dalam penelitian ini.

Tabel 4. 1 *Folder Data Training*

No.	Nama Folder	Jumlah File Citra
1.	Actinic keratoses	245
2.	Basal cell carcinoma	385
3.	Benign keratosis-like lesions	824
4.	Dermatofibroma	86
5.	Melanocytic Nevi	5029
6.	Vascular lesions	107
7.	Melanoma	835

2) *Data Testing*

Folder Data Testing digunakan untuk menyimpan dataset citra yang digunakan untuk menguji performa akhir model setelah proses pelatihan selesai. Detail pembagian jumlah file citra dalam folder data uji pada penelitian ini dapat disimak dalam Tabel 4.2.

Tabel 4. 2 *Folder Data Testing*

No.	Nama Folder	Jumlah File Citra
1.	Actinic keratoses	82
2.	Basal cell carcinoma	128
3.	Benign keratosis-like lesions	275
4.	Dermatofibroma	29
5.	Melanocytic Nevi	1676
6.	Vascular lesions	36
7.	Melanoma	278

b. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem melibatkan pemahaman, dokumentasi, dan pengaturan persyaratan yang harus dipenuhi oleh suatu sistem perangkat lunak. Tujuan dari analisis kebutuhan sistem adalah untuk memahami permasalahan yang akan dihadapi atau kebutuhan yang harus diterapkan pada sistem. Dalam konteks ini, berikut adalah kebutuhan yang akan digunakan oleh peneliti untuk membuat aplikasi.

1) Kebutuhan Perangkat Keras *Mobile Smartphone*.

Tabel 4. 3 Kebutuhan Perangkat Keras *Mobile Smartphone*

RAM	GPU	CPU	Penyimpanan	Resolusi
4GB (Minimal)	-	-	3GB (Minimal Tersedia)	1280x720 (Minimal)

Tabel 4.3 menunjukkan spesifikasi perangkat keras yang diperlukan untuk menjalankan aplikasi pada *smartphone*. Kolom pertama mencantumkan kebutuhan RAM minimal sebesar 4GB, sementara kolom penyimpanan adalah 3GB. Resolusi layar yang disarankan adalah 1280x720 piksel, namun tidak ada spesifikasi khusus untuk GPU dan CPU.

2) Kebutuhan Perangkat Lunak *Mobile Smartphone*.

Kebutuhan perangkat lunak untuk *smartphone*. Sistem operasi yang diperlukan adalah android, namun tidak mencantumkan aplikasi spesifik yang diperlukan. Hal ini menunjukkan bahwa perangkat lunak minimal yang dibutuhkan adalah sistem operasi android tanpa rincian aplikasi tambahan.

3) Kebutuhan Perangkat Keras *Computer*.

Tabel 4. 4 Kebutuhan Perangkat Keras *Computer*

RAM	GPU	CPU	Penyimpanan	Resolusi
8GB (Minimal)	Nvidia GeForce GTX 10 series / yang lebih baru, atau AMD Radeon RX 500 series atau yang	Intel Core i5 / setara AMD atau yang lebih baru	16GB (Minimal Tersedia)	1280x720 (Minimal)

	lebih baru			
--	---------------	--	--	--

Tabel 4.4 menunjukkan kebutuhan perangkat keras minimal untuk laptop. Kolum pertama mencantumkan RAM minimal sebesar 8GB dan kolum GPU mencantumkan Nvidia GeForce GTX 10 series atau AMD Radeon RX 500 series atau yang lebih baru. CPU minimal yang diperlukan adalah interl Core i5 atau setara dari AMD, dengan penyimpanan minimal 16GB dan resolusi layer minimal 1280x720 piksel.

4) Kebutuhan Perangkat Lunak *Computer*.

Tabel 4. 5 Kebutuhan Perangkat Lunak *Computer*

Sistem Operasi	Aplikasi
Windows 10 / 11	<ul style="list-style-type: none"> - Google Colaboratory - Visual Studio Code - Android Studio - Postman - Figma - Draw.io - Kaggle - Chrome Browser - Google Drive

Tabel 4.5 menunjukkan berbagai aplikasi yang kompatibel dengan sistem operasi Windows 10 dan 11. Aplikasi yang tercantum termasuk alat pengembangan seperti *Google Colaboratory*, Visual Studio Code, Android Studio, dan Postman. Selain itu, terdapat juga aplikasi desain dan kolaborasi seperti Figma, Draw.io, Kaggle, Chrome Browser, dan Google

Drive yang mendukung produktivitas dan pengembangan proyek.

c. Analisis Kebutuhan Fungsional

1. Halaman awal pada aplikasi yaitu splash screen yang berfungsi untuk memberikan informasi bahwa aplikasi siap digunakan.
2. Aplikasi dapat menampilkan halaman menu prediksi yang didalamnya berfungsi menampilkan halaman *upload* gambar dan prediksi.
3. Aplikasi dapat menampilkan halaman didalam menu prediksi untuk mendeteksi penyakit kanker kulit menggunakan gambar yang sudah dimiliki oleh pengguna dan meng *upload* lalu memilih prediksi.
4. Aplikasi dapat menampilkan halaman menu artikel yang dimana pengguna dapat memahami tentang penyakit kanker kulit.
5. Aplikasi dapat menampilkan halaman menu tentang yang dimana pengguna dapat mengetahui tentang informasi pencipta.

2. Desain

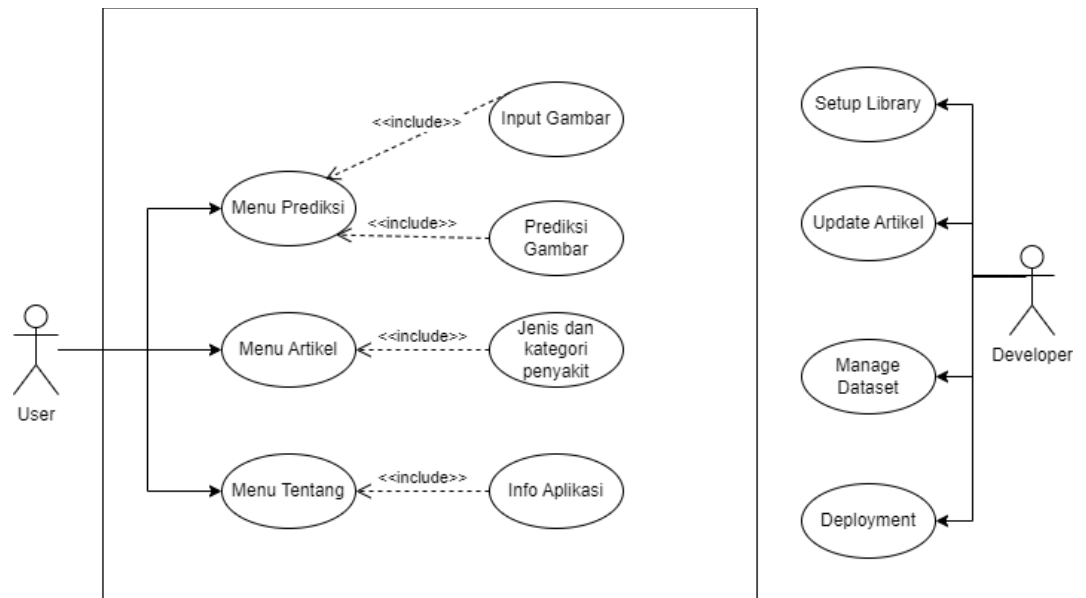
Desain sistem untuk pengembangan aplikasi jenis penyakit kanker kulit dibuat dengan tujuan menyediakan panduan yang jelas sebelum proses implementasi dimulai. Tujuannya adalah mempermudah penulisan kode dengan memberikan panduan yang terstruktur dan mudah diikuti. Berikut adalah desain yang disusun oleh penulis untuk pengembangan aplikasi terkait penyakit kanker kulit.

a) Desain *Unified Modeling Language* (UML)

Dalam desain *Unified Modeling Language* (UML), penulis menggunakan berbagai jenis diagram untuk menggambarkan berbagai aspek aplikasi atau sistem yang akan dikembangkan. Beberapa jenis diagram UML yang digunakan penulis meliputi.

1) *Use Case Diagram*

Use case diagram merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case diagram* mendefinisikan apa yang akan diproses oleh sistem dan komponen- komponennya. *Use case diagram* bekerja dengan menggunakan scenario yang merupakan deskripsi dari urutan atau langkah-langkah yang menjelaskan apa yang dilakukan oleh user terhadap sistem maupun sebaliknya [27]. *Use case diagram* dapat dilihat pada Gambar 4.1.



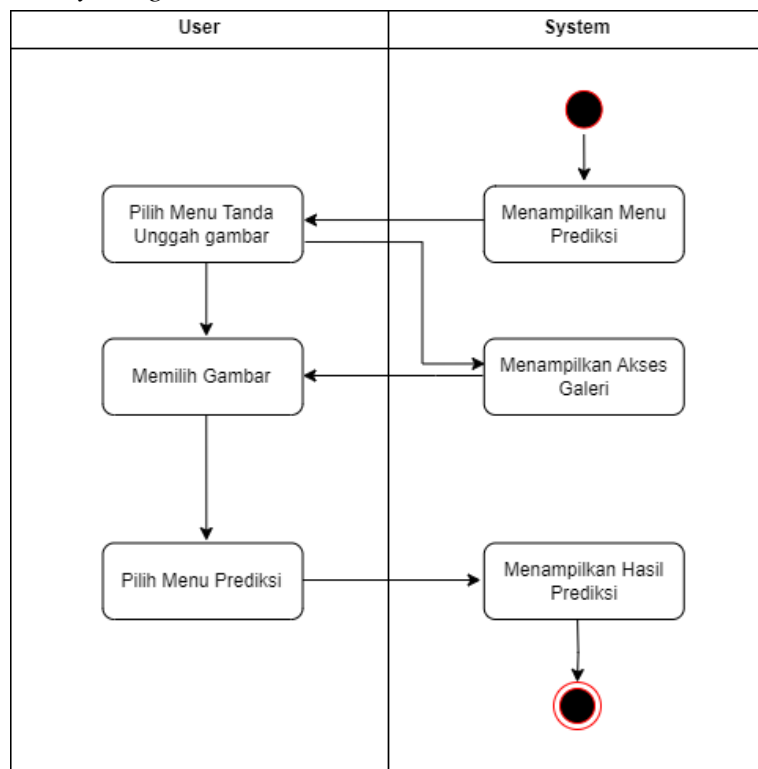
Gambar 4. 1 *Use Case Diagram*

Gambar 4.1 adalah *Use Case Diagram*, sistem memiliki dua peran utama, yakni pengguna (*User*) dan pengembang (*Developer*). Pengguna memiliki hak akses penuh terhadap aplikasi, termasuk kemampuan untuk membaca artikel tentang penyakit kanker kulit dan melakukan prediksi dengan mengunggah gambar melalui galeri. Pengembang memiliki otoritas untuk memperbarui data artikel dan mengelola *model dataset*.

2) Activity Diagram

Diagram Aktivitas memvisualisasikan serangkaian langkah dalam proses secara terstruktur, menunjukkan bagaimana aktivitas saling berinteraksi untuk mencapai tujuan tertentu. Dalam diagram ini, setiap aktivitas diwakili oleh simbol khusus dan hubungan antar aktivitas digambarkan dengan panah. Hal ini membantu dalam pemahaman yang jelas tentang alur kerja sistem atau proses yang sedang dimodelkan.

a. Activity Diagram Menu Prediksi

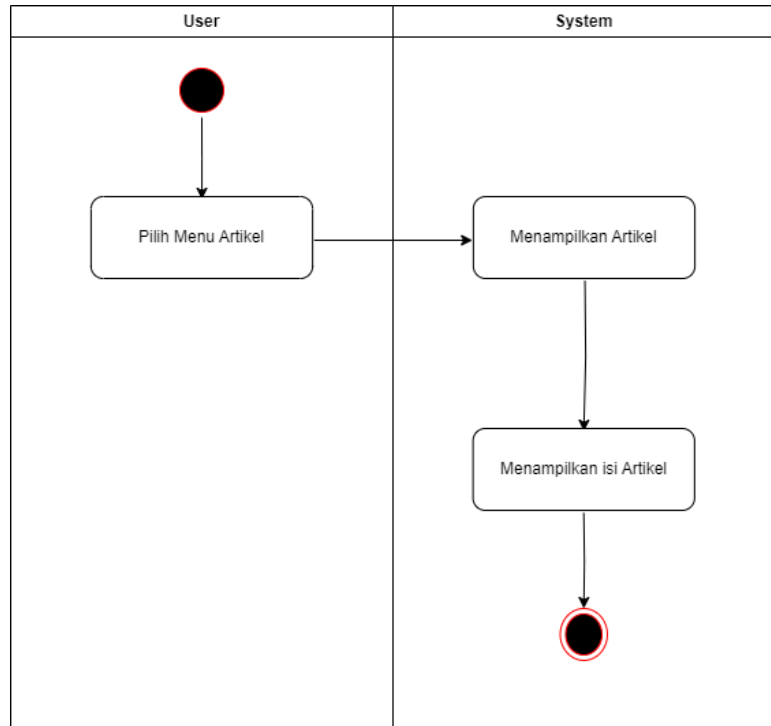


Gambar 4. 2 Activity Diagram Menu Prediksi

Gambar 4.2 adalah *Activity Diagram Menu Prediksi* digunakan untuk memprediksi jenis penyakit kanker kulit, yang merupakan salah satu menu penting dalam aplikasi yang sedang dikembangkan oleh penulis. Pada menu prediksi ini, pengguna diminta untuk memasukkan foto dengan cara mengunggah dari galeri. Jika foto telah dimasukkan maka sistem akan menampilkan foto yang telah

diinput selanjutnya pengguna diharuskan memilih menu prediksi, dan sistem akan menampilkan hasil prediksinya.

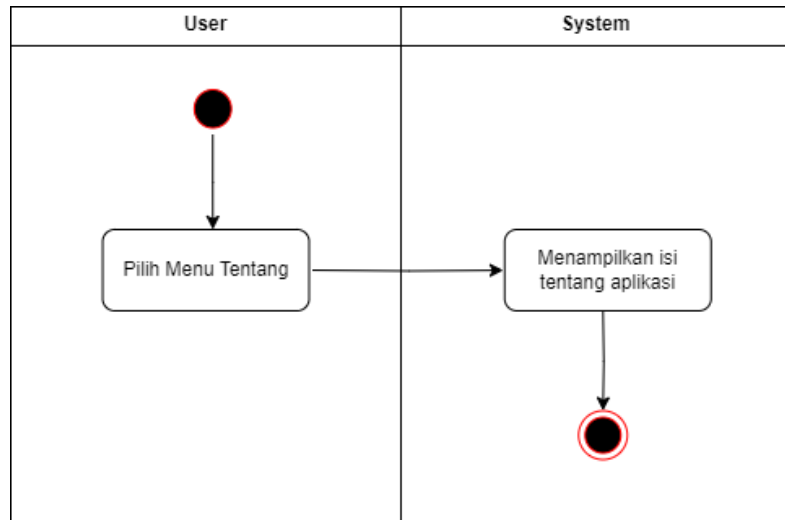
b. *Activity Diagram* Menu Artikel



Gambar 4. 3 *Activity Diagram* Menu Artikel

Gambar 4.3 adalah *Activity Diagram* Menu Artikel menjelaskan bahwa di menu artikel ini, pengguna dapat mengetahui dan membaca sejumlah artikel tentang jenis penyakit kanker kulit. Artikel-artikel ini memberikan informasi yang mendetail mengenai berbagai tipe kanker kulit. Dengan membaca artikel-artikel tersebut, pengguna akan mendapatkan pemahaman yang lebih baik tentang pentingnya deteksi dini dan perawatan yang tepat untuk kanker kulit.

c. *Activity Diagram Menu Tentang*



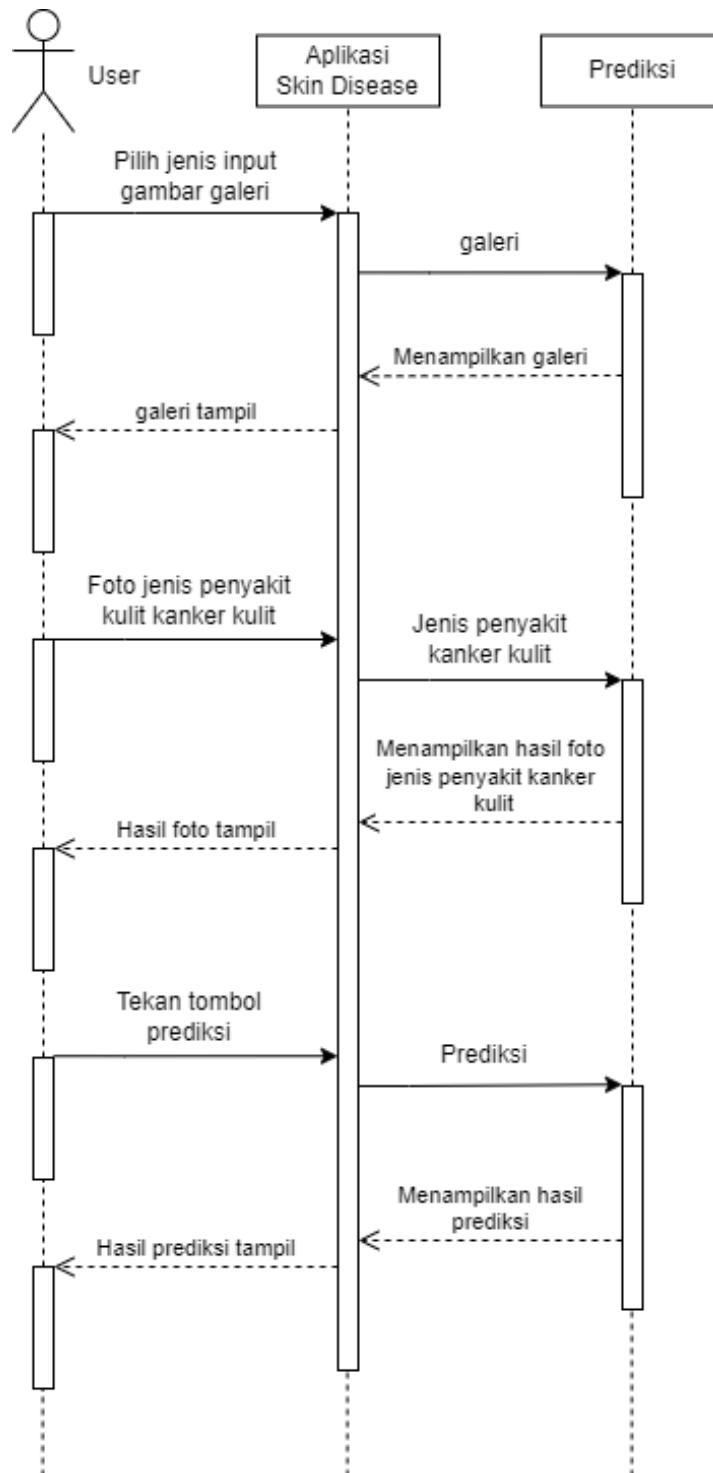
Gambar 4. 4 *Activity Diagram Menu Tentang*

Gambar 4.4 adalah *Activity Diagram Menu Tentang* menjelaskan bahwa di menu tentang, pengguna dapat mengetahui keterangan tentang aplikasi pendeteksi penyakit kanker kulit yang sedang dikembangkan oleh penulis. Aplikasi ini dirancang untuk membantu pengguna dalam mendeteksi tanda-tanda awal kanker kulit melalui analisis gambar kulit. Informasi di menu ini mencakup fitur-fitur utama aplikasi, cara kerjanya, dan manfaat yang dapat diperoleh pengguna dari menggunakan aplikasi tersebut.

3) *Sequence Diagram*

Sequence diagram digunakan untuk memvisualisasikan interaksi antara objek-objek dalam sebuah sistem dalam urutan waktu. Dalam diagram ini, setiap objek yang terlibat dalam interaksi direpresentasikan sebagai "*lifeline*", sebuah kotak vertikal yang menunjukkan waktu objek tersebut aktif selama proses berlangsung.

a. *Sequence Diagram Menu Prediksi*

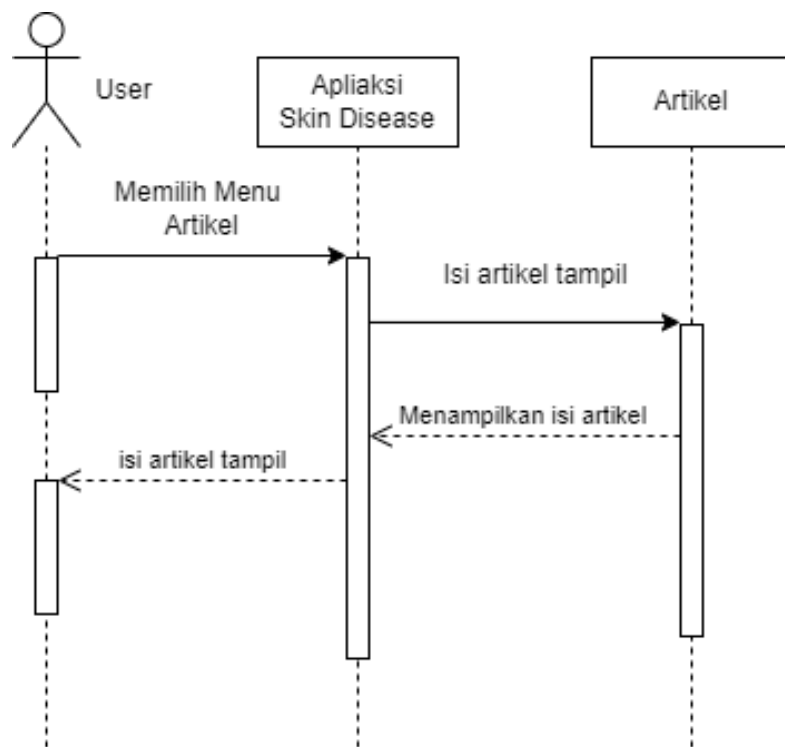


Gambar 4. 5 *Sequence Diagram Menu Prediksi*

Gambar 4.5 adalah *Sequence Diagram Menu Prediksi*, langkah awalnya adalah memilih jenis input

gambar, kemudian sistem akan membaca *user* memilih salah satu jenis foto dan akan menampilkan gambar di foto yang dipilih. Berikutnya *user* akan memilih foto yang telah dipilih oleh *user*. Selanjutnya *user* akan menekan tombol prediksi, lalu sistem akan mengirimkan jenis foto yang telah dipilih *user*. Setelah *user* mengirimkan hasil prediksi ke sistem, kemudian sistem akan menampilkan hasil prediksi ke *user*.

b. *Sequence Diagram Menu Artikel*

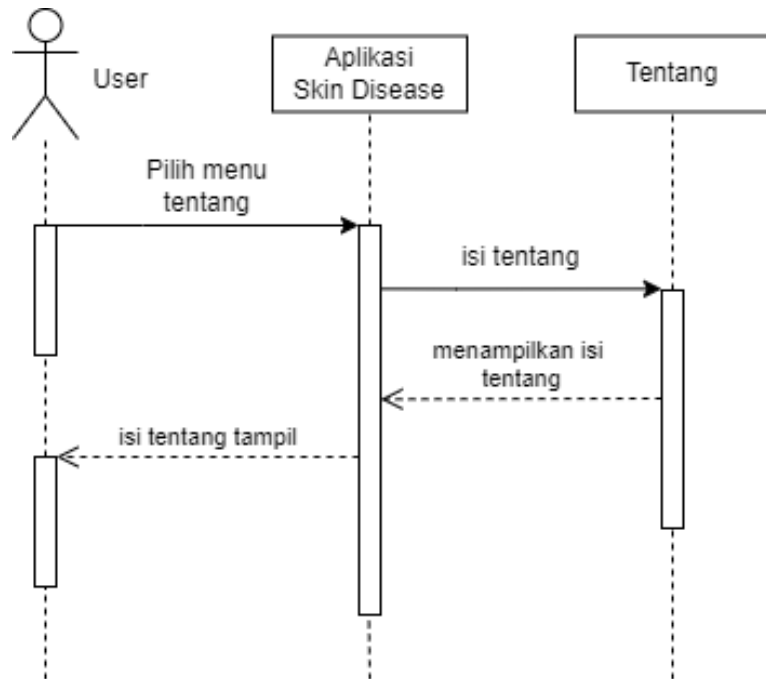


Gambar 4. 6 *Sequence Diagram Menu Artikel*

Gambar 4.6 adalah *Sequence Diagram Menu Artikel*, proses dimulai ketika pengguna memilih menu artikel. Permintaan ini kemudian dikirimkan ke sistem yang bertanggung jawab untuk menyediakan isi artikel yang diminta. Setelah sistem menerima permintaan tersebut, sistem akan menampilkan isi artikel kepada pengguna melalui menu artikel. Proses ini memastikan bahwa

pengguna dapat mengakses dan membaca artikel yang mereka inginkan tentang berbagai jenis penyakit kanker kulit.

c. *Sequence Diagram Menu Tentang*



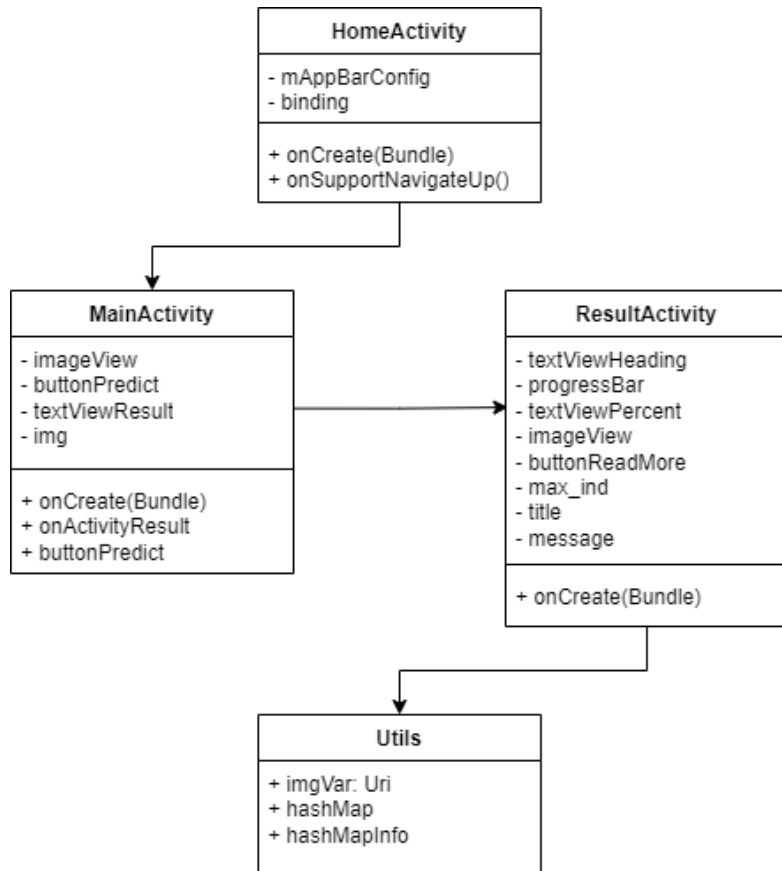
Gambar 4. 7 *Sequence Diagram Menu Tentang*

Gambar 4.7 adalah *Sequence Diagram Menu Tentang*, proses dimulai ketika pengguna memilih menu tentang. Permintaan ini kemudian dikirimkan ke sistem yang bertanggung jawab untuk menyediakan isi informasi tentang aplikasi pendeteksi kanker kulit. Setelah sistem menerima permintaan tersebut, sistem akan menampilkan isi informasi tersebut kepada pengguna melalui menu tentang. Proses ini memastikan bahwa pengguna dapat mengetahui keterangan lengkap mengenai aplikasi yang sedang dikembangkan, termasuk fitur-fitur dan manfaatnya.

d. *Class Diagram*

Class diagram adalah sebuah jenis diagram dalam *unified modeling language* (UML) yang berguna untuk memvisualisasikan struktur statis dari suatu sistem atau aplikasi. Diagram ini menggambarkan kelas-kelas dalam sistem, hubungan antara kelas-kelas tersebut, serta atribut dan metode yang dimiliki oleh setiap kelas. *Class diagram* membantu dalam memberikan gambaran yang jelas tentang struktur kelas sistem, sehingga mempermudah pemahaman tentang cara kerja sistem dan interaksi antar komponen sistem.

Dalam konteks khusus, class diagram dapat digunakan untuk menggambarkan klasifikasi jenis penyakit kanker kulit menggunakan algoritma *Convolutional Neural Network* (CNN) berbasis aplikasi *mobile*. Diagram ini membantu dalam memvisualisasikan struktur dan hubungan antara berbagai kelas yang terlibat dalam sistem. Contoh penggunaan *class diagram* ini dapat dilihat pada Gambar 4.8.



Gambar 4. 8 *Class Diagram*

4) Desain *User Interface*

Langkah awal dalam merancang antarmuka pengguna aplikasi klasifikasi jenis penyakit kanker kulit adalah membuat desain *wireframe*. *Wireframe* adalah representasi visual sederhana dari antarmuka pengguna yang menunjukkan struktur dasar aplikasi tanpa menggunakan warna, gambar, atau desain yang rumit. Tujuan dari *wireframe* adalah untuk menampilkan tata letak, susunan komponen utama, dan fungsi antarmuka pengguna tanpa terlalu memperhatikan detail visual yang kompleks. Dengan menggunakan *wireframe*, penulis dapat dengan jelas menentukan bagaimana setiap elemen antarmuka pengguna akan diatur dan berinteraksi dalam aplikasi tersebut. Beberapa desain *user interface* adalah desain halaman *splash*

screen, desain halaman prediksi, desain halaman *navigations*, desain halaman artikel, desain halaman tentang.

a. Desain Halaman *Splash Screen*



Gambar 4. 9 Desain *User Interface* Halaman *Splash Screen*

Gambar 4.9 adalah Desain *User Interface* Halaman *Splash Screen* adalah halaman pertama yang muncul saat membuka aplikasi. Pada splash screen ini, terdapat komponen utama yang mencakup logo GIF dan nama aplikasi tersebut. Logo GIF ditempatkan di bagian tengah halaman untuk memberikan identitas visual kepada pengguna. Di bawah logo GIF, terdapat nama aplikasi yang ditampilkan, dan setelah itu, pengguna akan diarahkan ke menu prediksi.

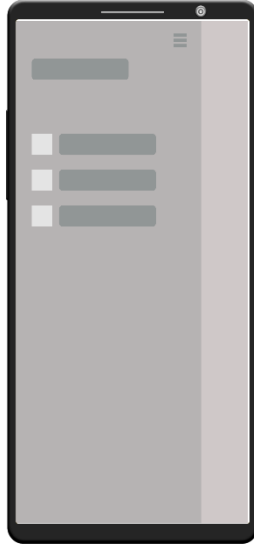
b. Desain Halaman Menu Prediksi



Gambar 4. 10 Desain *User Interface* Halaman Prediksi

Gambar 4.10 adalah Desain *User Interface* Halaman Prediksi yang menyajikan *tagline* prediksi dan dibawahnya ada *tagline* nama aplikasi. Terdapat tiga komponen yaitu bagian pertama adalah *navigations* dibagian pojok kiri atas yang berfungsi menampilkan menu pada aplikasi ini. Bagian kedua terdapat logo tambah gambar dibagian tengah yang berfungsi menambahkan gambar dari galeri yang berformat JPG atau PNG dan di bagian ketiga adalah tombol untuk memprediksi gambar yang sudah diinput.

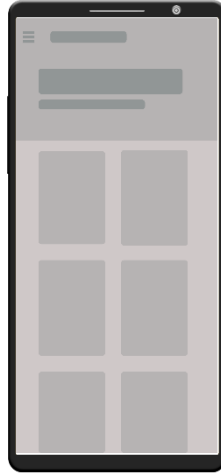
c. Desain Halaman Menu *Navigations*



Gambar 4. 11 Desain *User Interface* Halaman *Navigation*

Gambar 4.11 adalah Desain *User Interface* Halaman *Navigation* yang membagi akses ke halaman-halaman lainnya dalam aplikasi. Pada halaman ini, terdapat tagline nama aplikasi yang memberikan informasi singkat tentang aplikasi tersebut. Ada tiga pilihan menu untuk berpindah ke halaman lainnya, yaitu menu prediksi, menu artikel, dan menu tentang.

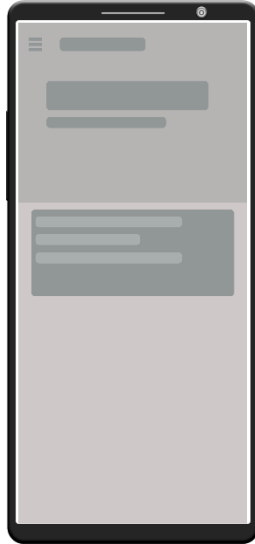
d. Desain Halaman Artikel



Gambar 4. 12 Desain *User Interface* Halaman Artikel

Gambar 4.12 adalah Desain *User Interface* Halaman Artikel dirancang untuk memfasilitasi pengguna dalam membaca artikel yang diminati. Pada bagian ini, terdapat judul artikel dengan keterangan di bawahnya yang menjelaskan isi artikel tersebut. Di pojok kiri atas, terdapat tombol navigasi, dan di bawahnya terdapat tagline yang menampilkan nama aplikasi.

e. Desan Halaman Tentang



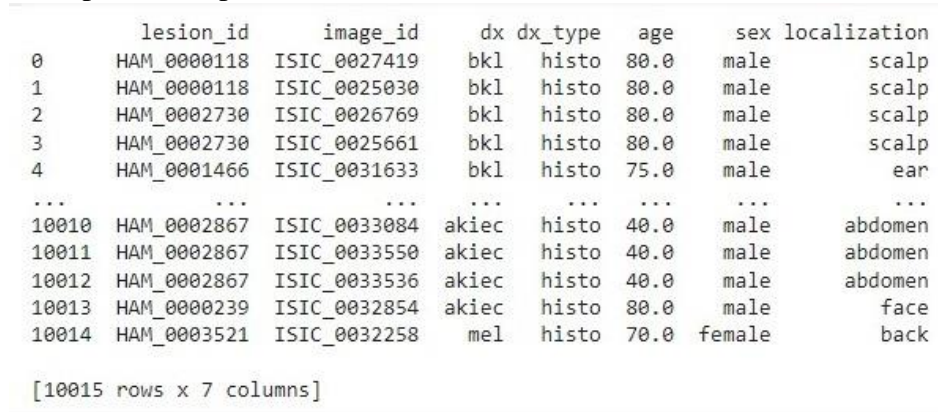
Gambar 4. 13 Desain Interface Halaman Tentang

Gambar 4.13 adalah Desain Interface Halaman Tentang menampilkan beberapa informasi penting kepada pengguna tentang informasi aplikasi tersebut. Dibagian pojok kiri atas terdapat tombol *navigation* yang disampingnya terdapat *tagline*. Serta dibagian tengah terdapat informasi aplikasi.

3. Implementasi Perhitungan Model CNN

a. Data Preparation

Penggunaan *Google Colab* menjadi langkah pertama dalam memudahkan akses dan pengelolaan data untuk proyek klasifikasi lesi kanker kulit. Melalui platform ini, proses pengunduhan dataset HAM10000 yang berisi jenis kanker kulit dari *Kaggle* dapat dilakukan dengan mudah. Setelah dataset terunduh, langkah berikutnya adalah membaca metadata dataset dari file CSV menggunakan pustaka *pandas*. Metadata ini berisi informasi penting tentang setiap citra, seperti jenis lesi kanker kulit, jenis kelamin pasien, lokasi lesi, dan usia pasien, yang menjadi dasar untuk analisis dan pemrosesan data lebih lanjut. Dengan demikian, penggunaan *Google Colab* dan proses unduhan serta pembacaan dataset menjadi langkah awal yang penting dalam implementasi proyek klasifikasi jenis kanker kulit. Untuk hasil *data preparation* dapat dilihat pada Gambar 4.14.



```

  lesion_id  image_id  dx dx_type  age  sex localization
0  HAM_0000118  ISIC_0027419  bkl  histo  80.0  male  scalp
1  HAM_0000118  ISIC_0025030  bkl  histo  80.0  male  scalp
2  HAM_0002730  ISIC_0026769  bkl  histo  80.0  male  scalp
3  HAM_0002730  ISIC_0025661  bkl  histo  80.0  male  scalp
4  HAM_0001466  ISIC_0031633  bkl  histo  75.0  male  ear
...
10010  HAM_0002867  ISIC_0033084  akiec  histo  40.0  male  abdomen
10011  HAM_0002867  ISIC_0033550  akiec  histo  40.0  male  abdomen
10012  HAM_0002867  ISIC_0033536  akiec  histo  40.0  male  abdomen
10013  HAM_0000239  ISIC_0032854  akiec  histo  80.0  male  face
10014  HAM_0003521  ISIC_0032258  mel  histo  70.0  female  back
[10015 rows x 7 columns]
```

Gambar 4. 14 *Data Preparation*

b. Pre-processing Data

Dalam tahap *Preprocessing Data*, langkah pertama yang dilakukan adalah *Label Encoding*, yang bertujuan untuk mengonversi label kelas dari dataset menjadi nilai numerik. Proses ini penting karena model *machine learning* memerlukan input yang

berupa angka, bukan teks, sehingga dengan melakukan *label encoding*, kita dapat mengubah label kelas menjadi representasi numerik yang dapat diproses oleh model. Setelah itu, dilakukan visualisasi distribusi data untuk memahami karakteristik dataset dengan lebih baik. Visualisasi ini membantu dalam mengidentifikasi sebaran data di antara berbagai kelas, memastikan keseimbangan kelas, serta memberikan wawasan tambahan tentang pola dan tren yang mungkin ada dalam dataset. Dengan memahami distribusi data, kita dapat mengambil keputusan yang tepat dalam merancang dan melatih model yang optimal untuk tujuan klasifikasi jenis kanker kulit.

Tabel 4. 6 *Encode Class* atau *Label*

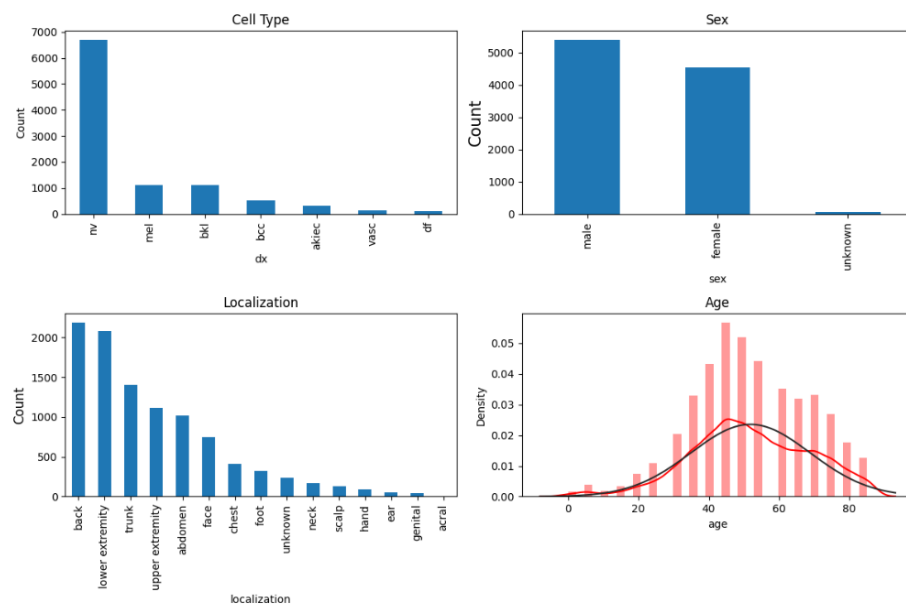
No.	Feature	Class
1	akiec	0
2	bcc	1
3	bkl	2
4	df	3
5	mel	4
6	nv	5
7	vasc	6

c. *Data Visualization*

Dalam tahap persiapan data dan visualisasi sebelum melatih model, Anda melakukan serangkaian langkah yang penting untuk memastikan keberagaman dataset dan memahami karakteristiknya dengan baik. Setelah memuat dataset, penulis melakukan visualisasi distribusi kelas, jenis kelamin, lokasi, dan distribusi umur dari sampel data. Ini memberikan pemahaman yang lebih baik tentang struktur dataset dan memungkinkan penulis untuk membuat keputusan yang tepat terkait dengan strategi pemrosesan data dan

pelatihan model. Selanjutnya, untuk memeriksa keberagaman dataset lebih lanjut, Penulis membuat visualisasi beberapa sampel citra dari setiap kelas. Dengan melakukan ini, penulis dapat melihat variasi dalam citra-citra yang mewakili setiap kelas lesi kanker kulit. Keberagaman dalam sampel citra sangat penting karena dapat mempengaruhi kemampuan model untuk mempelajari fitur-fitur yang relevan dan membuat prediksi yang akurat. Dengan demikian, langkah ini memastikan bahwa dataset yang penulis gunakan untuk melatih model mencakup representasi yang luas dari semua kelas yang ada.

Secara keseluruhan, visualisasi ini merupakan bagian integral dari proses pelatihan model penulis, karena membantu penulis memahami data yang akan digunakan oleh model penulis dan memastikan bahwa model dapat belajar dengan baik dari data tersebut. Dengan memeriksa keberagaman dataset dan memahami distribusi kelasnya, Penulis dapat meningkatkan peluang kesuksesan dalam melatih model klasifikasi jenis kanker kulit yang akurat dan andal. Untuk visualisasi data dapat dilihat pada Gambar 4.15.



Gambar 4. 15 *Data Visualization*

d. Pembuatan Model CNN

Dalam pengembangan model ini, penulis menggunakan *modul Sequential* dari *library Keras* untuk merancang secara berurutan arsitektur model. Penulis memasukkan serangkaian lapisan konvolusi (Conv2D) yang menerapkan fungsi aktivasi ReLU, yang diikuti oleh lapisan MaxPooling (MaxPool2D) untuk mengekstraksi fitur dan mengurangi dimensi gambar. Untuk mengurangi risiko overfitting yang umum terjadi pada model kompleks, penulis juga menambahkan lapisan *Dropout*. Setelah itu, penulis menggunakan lapisan *Flatten* untuk mengubah fitur-fitur yang diekstraksi menjadi vektor satu dimensi. Akhirnya, penulis menambahkan dua lapisan Dense yang bertujuan untuk melakukan klasifikasi. Berikut adalah ringkasan pada pembuatan model cnn dapat dilihat pada Gambar 4.16.

```
Model: "sequential"
-----
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 256)	7168
max_pooling2d (MaxPooling2D)	(None, 15, 15, 256)	0
dropout (Dropout)	(None, 15, 15, 256)	0
conv2d_1 (Conv2D)	(None, 13, 13, 128)	295040
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_1 (Dropout)	(None, 6, 6, 128)	0
conv2d_2 (Conv2D)	(None, 4, 4, 64)	73792
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 64)	0
dropout_2 (Dropout)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dense_1 (Dense)	(None, 7)	231

```
-----
Total params: 384455 (1.47 MB)
Trainable params: 384455 (1.47 MB)
Non-trainable params: 0 (0.00 Byte)
```

Gambar 4. 16 Ringkasan Arsitektur CNN

1) Pelatihan Model

Pada bagian pelatihan model ini, penulis menggunakan model yang telah didefinisikan sebelumnya untuk melakukan proses pembelajaran pada data latih. Penulis menetapkan ukuran batch sebesar 16 dan jumlah *epoch* sebanyak 200. Proses pelatihan dimulai dengan memanggil metode `fit()` dari objek model, yang menerima beberapa parameter. Anda menyediakan data latih (`x_train` dan `y_train`), jumlah epoch, ukuran batch, serta data validasi (`x_test` dan `y_test`) untuk memantau performa model selama pelatihan.

Selama pelatihan, model akan menyesuaikan bobotnya berdasarkan gradien dari fungsi kerugian terhadap data latih. Ini dilakukan dengan membagi dataset latih menjadi batch-batch kecil, di mana setiap batch akan digunakan untuk menghitung nilai loss dan gradien yang akan digunakan untuk memperbarui parameter-model. Setelah satu *epoch* selesai (yaitu, seluruh dataset latih telah melalui model satu kali), model akan dievaluasi menggunakan data validasi untuk memeriksa performanya pada data yang belum pernah dilihat sebelumnya.

Setelah proses pelatihan selesai, Anda mengevaluasi performa model pada data uji menggunakan metode `evaluate()`, dan mencetak akurasi model pada data uji. Ini membantu Anda memahami seberapa baik model penulis mampu melakukan klasifikasi pada data baru yang belum pernah dilihat selama pelatihan. Untuk rincian pada pelatihan model dapat dilihat pada Gambar 4.17.

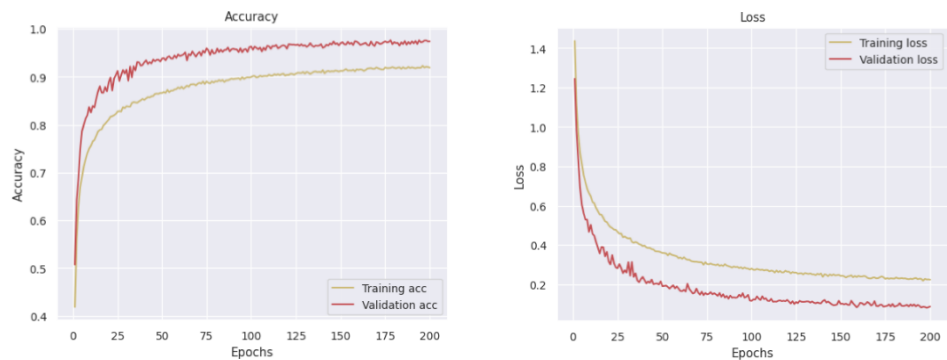

```

Epoch 188/200
2133/2133 - 13s - loss: 0.2254 - acc: 0.9207 - val_loss: 0.0867 - val_acc: 0.9726 - 13s/epoch - 6ms/step
Epoch 189/200
2133/2133 - 13s - loss: 0.2284 - acc: 0.9184 - val_loss: 0.0971 - val_acc: 0.9688 - 13s/epoch - 6ms/step
Epoch 190/200
2133/2133 - 13s - loss: 0.2257 - acc: 0.9198 - val_loss: 0.0889 - val_acc: 0.9740 - 13s/epoch - 6ms/step
Epoch 191/200
2133/2133 - 13s - loss: 0.2293 - acc: 0.9189 - val_loss: 0.0973 - val_acc: 0.9699 - 13s/epoch - 6ms/step
Epoch 192/200
2133/2133 - 13s - loss: 0.2319 - acc: 0.9178 - val_loss: 0.0872 - val_acc: 0.9747 - 13s/epoch - 6ms/step
Epoch 193/200
2133/2133 - 13s - loss: 0.2256 - acc: 0.9192 - val_loss: 0.0944 - val_acc: 0.9686 - 13s/epoch - 6ms/step
Epoch 194/200
2133/2133 - 13s - loss: 0.2282 - acc: 0.9191 - val_loss: 0.0896 - val_acc: 0.9762 - 13s/epoch - 6ms/step
Epoch 195/200
2133/2133 - 13s - loss: 0.2277 - acc: 0.9189 - val_loss: 0.0809 - val_acc: 0.9721 - 13s/epoch - 6ms/step
Epoch 196/200
2133/2133 - 12s - loss: 0.2179 - acc: 0.9229 - val_loss: 0.0853 - val_acc: 0.9728 - 12s/epoch - 6ms/step
Epoch 197/200
2133/2133 - 13s - loss: 0.2272 - acc: 0.9196 - val_loss: 0.0839 - val_acc: 0.9756 - 13s/epoch - 6ms/step
Epoch 198/200
2133/2133 - 13s - loss: 0.2240 - acc: 0.9210 - val_loss: 0.0795 - val_acc: 0.9757 - 13s/epoch - 6ms/step
Epoch 199/200
2133/2133 - 13s - loss: 0.2246 - acc: 0.9205 - val_loss: 0.0825 - val_acc: 0.9739 - 13s/epoch - 6ms/step
Epoch 200/200
2133/2133 - 13s - loss: 0.2234 - acc: 0.9193 - val_loss: 0.0864 - val_acc: 0.9739 - 13s/epoch - 6ms/step
356/356 [=====] - 1s 3ms/step - loss: 0.0864 - acc: 0.9739
Test accuracy: 0.9738901257514954

```

Gambar 4. 17 Pelatihan Model

2) Grafik Hasil Pelatihan Model



Gambar 4. 18 Hasil Pelatihan Model

Gambar 4.18 adalah Hasil Pelatihan Model yang menunjukan Grafik pertama menampilkan perubahan akurasi pada data pelatihan (*Training accuracy*) dan data validasi (*Validation accuracy*) sepanjang proses pelatihan model. Sama seperti pada grafik loss, kita ingin melihat kedua kurva ini meningkat seiring dengan berjalannya epoch, menunjukkan bahwa model kita semakin mampu memprediksi dengan tepat pada data yang digunakan untuk pelatihan maupun data yang tidak terlihat selama pelatihan (data validasi).

Grafik kedua menampilkan perubahan loss (kerugian) pada data pelatihan (*Training loss*) dan data validasi (*Validation loss*) selama proses pelatihan model. Tujuannya adalah untuk melihat bagaimana loss berubah seiring dengan jumlah *epoch*. Idealnya, kita ingin melihat kedua kurva ini berkurang seiring berjalannya *epoch*, menunjukkan bahwa model kita semakin baik dalam mempelajari pola-pola dalam data.

Namun, jika terjadi *overfitting*, grafik akurasi pada data validasi mungkin akan mulai turun atau tetap stabil, sementara akurasi pada data pelatihan terus meningkat. Sedangkan jika terjadi *underfitting*, kedua kurva mungkin akan tetap rendah atau tidak mengalami peningkatan yang signifikan seiring dengan berjalannya *epoch*. Oleh karena itu, grafik ini memberikan wawasan yang penting tentang performa dan pembelajaran model selama proses pelatihan.

3) Evaluasi Model

Pada langkah evaluasi model, pertama-tama dilakukan evaluasi terhadap performa model pada data uji. Evaluasi ini bertujuan untuk mengukur seberapa baik model yang telah dilatih dapat menggeneralisasi pola-pola yang telah dipelajari ke dalam data baru yang tidak pernah dilihat sebelumnya. Hasil evaluasi tersebut kemudian divisualisasikan dalam bentuk grafik *loss* dan akurasi pada setiap *epoch*, yang memberikan gambaran tentang bagaimana performa model berubah seiring berjalannya proses pelatihan.

Selanjutnya, pada langkah prediksi dan evaluasi lanjutan, model yang telah dilatih digunakan untuk melakukan prediksi terhadap data uji. Prediksi tersebut kemudian dievaluasi lebih detail menggunakan *classification report*, yang memberikan informasi tentang berbagai metrik evaluasi seperti presisi, *recall*,

dan *f1-score* untuk setiap kelas. *Classification report* ini membantu untuk memahami performa model secara lebih mendalam, khususnya dalam hal kemampuan model untuk mengklasifikasikan data pada setiap kelas dengan baik.

Selain itu, dilakukan pembuatan *confusion matrix* untuk menganalisis performa model pada setiap kelas. *Confusion matrix* adalah tabel yang menggambarkan jumlah data yang diklasifikasikan dengan benar dan yang salah untuk setiap kelas. Dari *confusion matrix* ini, kita dapat melihat seberapa baik model dapat mengklasifikasikan data pada setiap kelas, serta mengidentifikasi kelas mana yang mungkin memiliki performa yang kurang baik dan memerlukan perhatian lebih lanjut. Dengan demikian, langkah prediksi dan evaluasi lanjutan memberikan pemahaman yang lebih komprehensif tentang performa dan kekurangan model yang telah dibuat. Untuk melihat hasil evaluasi kinerja model dapat dilihat pada Gambar 4.19.

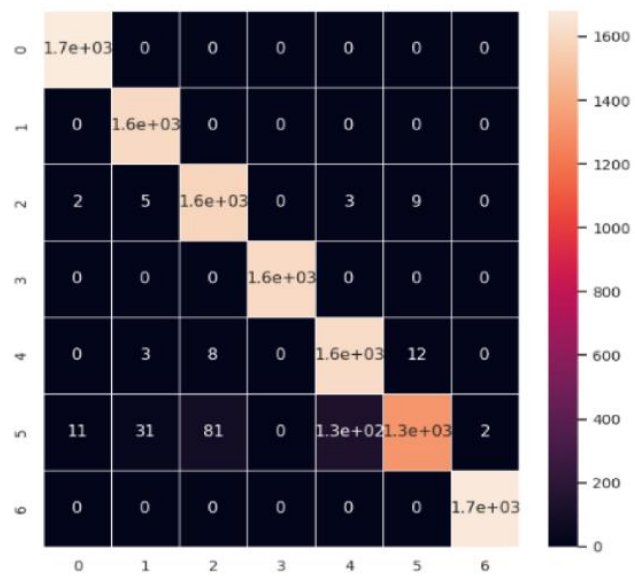
```

Classification Report:
              precision    recall  f1-score   support

     0:       0.99         1.00         1.00       1681
     1:       0.98         1.00         0.99       1596
     2:       0.95         0.99         0.97       1607
     3:       1.00         1.00         1.00       1608
     4:       0.92         0.99         0.95       1637
     5:       0.98         0.84         0.91       1573
     6:       1.00         1.00         1.00       1673

 accuracy          0.97         0.97         0.97       11375
 macro avg         0.97         0.97         0.97       11375
 weighted avg     0.97         0.97         0.97       11375

```



Gambar 4. 19 Evaluasi Model

Tabel 4. 7 Perbandingan Penelitian

Aspek	Penelitian Terdahulu [7]	Penelitian Penulis
Dataset	HAM10000	HAM10000
Jumlah Gambar	5,000	10,015
Jumlah Kelas	2 (Non-melanocytic malignant, Benign)	7 (Melanocytic nevi, Melanoma, Benign keratosis-like lesions, Basal cell carcinoma,

		Actinic keratoses, Vascular lesions, Dermatofibroma)
Akurasi Model	75%	97.38%
Precision (tertinggi)	80%	97%
Recall (tertinggi)	82%	97%
F1 Score	81%	97%
Penggunaan Metode	Classification_report	CNN dengan augmentasi data, preprocessing, dan normalisasi gambar
Keunggulan	Akurasi dan performa model lebih rendah dengan hanya dua kelas	Lebih banyak kelas, lebih detail pada jenis lesi

Pada Tabel 4.7 Menunjukkan perbandingan ini bahwa penelitian penulis lebih unggul dalam hal akurasi dan tingkat detail klasifikasi, dengan penggunaan 7 kelas dibandingkan dengan 2 kelas dalam penelitian terdahulu. Penelitian Anda menggunakan teknik augmentasi data dan *preprocessing* yang lebih komprehensif, yang berkontribusi pada hasil yang lebih baik.

a) *Precision*

Precision mengukur seberapa banyak dari data yang diprediksi sebagai positif yang sebenarnya positif.

Rumus:

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

dimana:

- a. TP adalah *True Positives* (data positif yang diprediksi benar)
- b. FP adalah *False Positives* (data negatif yang diprediksi salah sebagai positif)
- c. FN adalah *False Negatives* (data positif yang diprediksi salah)

b) *Recall*:

Recall mengukur seberapa banyak dari data yang sebenarnya positif yang diprediksi dengan benar.

Rumus:

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

c) *F1-score*:

F1-score adalah rata-rata harmonis antara *precision* dan *recall*. Ini memberikan keseimbangan antara *precision* dan *recall*.

Rumus:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.3)$$

Dalam rumus-rumus di atas, TP, FP, dan FN mewakili jumlah *True Positives*, *False Positives*, dan *False Negatives* berturut-turut. Setiap metrik ini memberikan informasi yang berbeda tentang performa model dalam

melakukan klasifikasi. Perhitungan ini biasanya dilakukan untuk setiap kelas dalam masalah klasifikasi multikelas.

Keterangan :

True Positive (TP) : Sampel data positif, prediksi negatif

False Positive (FP) : Sampel data positif, prediksi positif

False Negative (FN) : Sampel data positif, prediksi negatif

a. Perhitungan Classification Report:

1. *Macro Avg*

a) *Precision Macro Avg*

$$\begin{aligned} &= \frac{0,99 + 0,98 + 0,95 + 1,00 + 0,92 + 0,98 + 1,00}{7} \quad (4.4) \\ &= \frac{6,82}{7} = 0,97 \end{aligned}$$

b) *Recall Macro Avg*

$$\begin{aligned} &= \frac{1,00 + 1,00 + 0,99 + 1,00 + 0,99 + 0,84 + 1,00}{7} \quad (4.5) \\ &= \frac{6,81}{7} = 0,97 \end{aligned}$$

c) *F1-Score Macro Avg*

$$\begin{aligned} &= \frac{0,99 + 0,99 + 0,97 + 1,00 + 0,95 + 0,91 + 1,00}{7} \quad (4.6) \\ &= \frac{6,81}{7} = 0,97 \end{aligned}$$

2. *Weighted Avg*

a) *Precision Weighted Avg*

$$\begin{aligned}
& \frac{1.664,19 + 1.564,08 + 1.526,65 + 1.608,00}{11.375} \quad (4.7) \\
& = \frac{1.506,04 + 1.541,54 + 1.673,00}{11.375} \\
& = \frac{11.083,50}{11.375} = 0,97
\end{aligned}$$

b) *Recall Weighted Avg*

$$\begin{aligned}
& \frac{1.681,00 + 1.59,00 + 1.590,93 + 1.608,00}{11375} \quad (4.8) \\
& = \frac{1.620,63 + 1.321,32 + 1.673,00}{11375} \\
& = \frac{11.090,88}{11.375} = 0,97
\end{aligned}$$

c) *F1-Score Weighted Avg*

$$\begin{aligned}
& \frac{1.664,19 + 1.580,04 + 1.558,79 + 1.608,00}{11.375} \quad (4.9) \\
& = \frac{1.555,15 + 1.431,43 + 1.673,00}{11.375} \\
& = \frac{11.070,60}{11.375} = 0,97
\end{aligned}$$

3. Accuracy

$$= \frac{\text{Total prediksi benar}}{\text{Total sampel}} = \frac{11071}{11375} = 97,38\% \quad (4.10)$$

b. *Confusion Matrix:*

Confusion matrix tersebut memiliki 7 kelas (0 sampai 6), dengan data sebagai berikut.

Tabel 4. 8 Confusion matrix

Class	TP	FP	FN	TN	Precision	Recall	F1-Score
0	1700	0	0	0	1.00	1.00	1.00
1	1600	5	5	0	0.99	0.99	0.99
2	1600	17	5	0	0.99	0.99	0.99

3	1600	0	0	0	1.00	1.00	1.00
4	1600	14	11	0	0.99	0.99	0.99
5	1300	123	120	0	0.91	0.91	0.91
6	1700	0	0	0	1.00	1.00	1.00

Perhitungan:

$$Precision = \frac{1700}{1700 + 0} = 1.00 \quad (4.11)$$

$$Recall = \frac{1700}{1700 + 0} = 1.00 \quad (4.12)$$

$$F1 - Score = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} = 1.00 \quad (4.13)$$

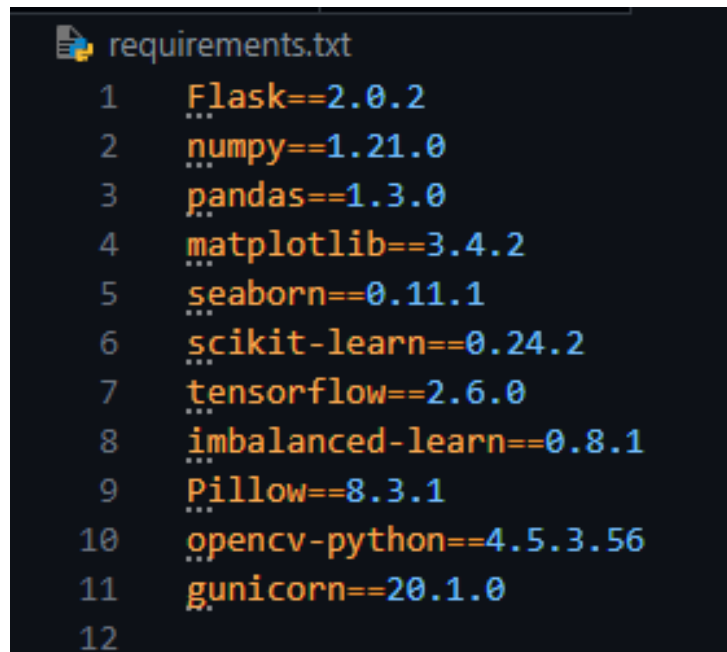
4) Implementasi Sistem

1. Pembuatan Flask

Dalam penelitian ini, Flask digunakan sebagai *framework* untuk pembuatan sistem. Meskipun Flask dikenal dalam pengembangan web, penelitian ini mengimplementasikannya ke dalam aplikasi Android. Dalam konteks ini, Flask berfungsi sebagai media untuk pembuatan API dengan metode POST. Sebelum memulai pembuatan aplikasi menggunakan Flask, penulis mempersiapkan berbagai kebutuhan (*requirements*) yang mendukung pembuatan API menggunakan Flask. Berikut adalah langkah-langkah yang biasanya dilakukan dalam proses ini.

- a. Menyiapkan Lingkungan Pengembangan: Instalasi Python: Pastikan Python sudah terinstal di sistem.
- b. Virtual Environment: Buat virtual environment untuk mengisolasi dependensi proyek.

- c. Menginstal Flask: Instal Flask dan pustaka lain yang diperlukan dengan menggunakan pip.
- d. Mendefinisikan Dependencies: Buat file requirements.txt yang berisi semua dependensi proyek yang diperlukan



```
requirements.txt
1  Flask==2.0.2
2  numpy==1.21.0
3  pandas==1.3.0
4  matplotlib==3.4.2
5  seaborn==0.11.1
6  scikit-learn==0.24.2
7  tensorflow==2.6.0
8  imbalanced-learn==0.8.1
9  Pillow==8.3.1
10 opencv-python==4.5.3.56
11 gunicorn==20.1.0
12
```

Gambar 4. 20 *Requiremenst.txt*

Pada Tabel 4.20 adalah *Requirements.txt* terdapat beberapa *requirements* yang digunakan untuk mengelola dependensi dalam proyek ini. Flask digunakan untuk membangun aplikasi web, sementara Gunicorn berfungsi sebagai server HTTP Python yang menangani permintaan HTTP dari aplikasi Flask. NumPy digunakan untuk komputasi numerik yang kuat dan efisien, sedangkan Pillow digunakan untuk pemrosesan citra. Pandas digunakan untuk manipulasi dan analisis data, sementara Matplotlib dan Seaborn digunakan untuk visualisasi data. Scikit-learn digunakan untuk alat-alat pembelajaran mesin, dan imbalanced-learn digunakan untuk mengatasi masalah data yang tidak seimbang. TensorFlow adalah requirement paling penting yang

digunakan untuk pembangunan model *machine learning*, dan *OpenCV* digunakan untuk pemrosesan citra dan visi komputer.

Tahap selanjutnya adalah penulisan kode untuk pembuatan aplikasi menggunakan Flask. Dalam proses ini, model berformat .tflit digunakan untuk pemrosesan citra dan prediksi label. Kode juga mencakup perubahan format label yang awalnya di *encode* sebagai integer saat pelatihan model CNN menjadi string. Misalnya, label 0 diubah menjadi 'Actinic keratoses'. Berikut kode flask prediksi jenis penyakit kanker kulit dapat dilihat pada kode dibawah ini.

```
from flask import Flask, request, jsonify
from PIL import Image
import numpy as np
import tensorflow as tf

app = Flask(__name__)

# Mapping for label indices to class names
label_mapping = {
    0: 'Actinic keratoses',
    1: 'Basal cell carcinoma',
    2: 'Benign keratosis-like lesions',
    3: 'Dermatofibroma',
    4: 'Melanocytic nevi',
    5: 'Vascular lesions',
    6: 'Melanoma'
}

def preprocess_image(image):
    # Convert image to grayscale
    image = image.convert('L')
    # Resize image to 28x28
    image = image.resize((28, 28))
    # Convert image to numpy array
    image_array = np.array(image)
    # Normalize image
    image_array = image_array / 255.0
```

```

# Convert image array to FLOAT32
image_array = image_array.astype(np.float32)
# Expand dimensions to match model input shape
image_array = np.expand_dims(image_array, axis=0)
# Add channel dimension (for grayscale)
image_array = np.expand_dims(image_array, axis=-
1)
# Repeat grayscale channel to match RGB channel
image_array = np.repeat(image_array, 3, axis=-1)
return image_array

@app.route('/predict', methods=['POST'])
def predict():
    # Check if an image was uploaded
    if 'image' not in request.files:
        return jsonify({'error': 'No image
provided'})

    # Read the image file
    image = request.files['image']

    try:
        # Open and preprocess the image
        img = Image.open(image)
        img_array = preprocess_image(img)

        # Load the TensorFlow Lite model
        interpreter =
tf.lite.Interpreter(model_path='D:/Flask/model3.tflit
e')
        interpreter.allocate_tensors()

        # Set the input tensor
        input_details =
interpreter.get_input_details()
        interpreter.set_tensor(input_details[0]['inde
x'], img_array)

        # Run inference
        interpreter.invoke()

        # Get the output tensor
        output_details =
interpreter.get_output_details()

```

```

        output_data =
interpreter.get_tensor(output_details[0]['index'])

        # Get the predicted class label
        predicted_label_index =
np.argmax(output_data)
        predicted_label =
label_mapping[predicted_label_index]

        # Get confidence scores for each class
        confidence_scores = {label_mapping[i]:
float(output_data[0][i]) for i in
range(len(output_data[0]))}

        return jsonify({'predicted_class':
predicted_label, 'confidence_scores':
confidence_scores})

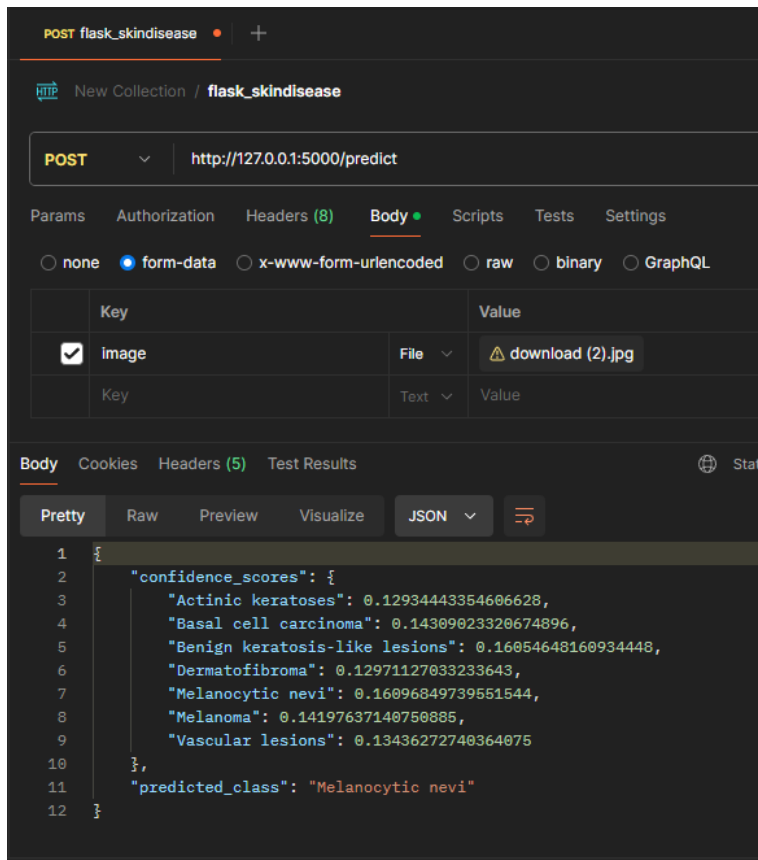
    except Exception as e:
        return jsonify({'error': str(e)})

if __name__ == '__main__':
    app.run(debug=True)

```

2. Pengujian API *Localhost* ke *Postman*

Pengujian dilakukan untuk memastikan apakah Flask berfungsi dengan baik. Penelitian ini menggunakan *Postman* sebagai alat untuk menguji API pada localhost. Pengujian membutuhkan *link localhost* yang telah diaktifkan melalui proyek Flask. Di *Postman*, diperlukan form-data untuk input file jenis gambar yang akan diprediksi. *Output* nya berupa JSON yang memberikan hasil prediksi gambar tersebut. Dapat dilihat pada Gambar 4.21.



Gambar 4. 21 Pengujian API Localhost di Postman

Gambar 4.21 adalah penjelasan dari hasil request Flask yang terlihat pada gambar.

a. *Endpoint*

- 1) URL: <http://127.0.0.1:5000/predict>
- 2) Method: POST

b. *Request*

- 1) Type: form-data
- 2) Key: image
- 3) Value: download (2).jpg (nama file gambar yang diupload)

c. *Response*

Response dari server memiliki format JSON yang berisi dua bagian utama: *confidence_scores* dan *predicted_class*.

1) *Confidence_scores* adalah *dictionary* yang berisi nilai probabilitas (*confidence score*) untuk masing-masing kelas dari model klasifikasi. Nilai-nilai ini menunjukkan seberapa yakin model bahwa gambar yang diupload termasuk dalam masing-masing kelas. Berikut adalah penjelasan masing-masing kelas dan nilainya.

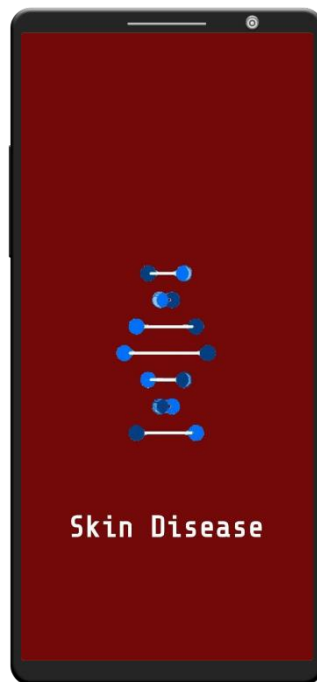
1. *Actinic keratoses*: 3.37024344458152 (sangat rendah, hampir 0%).
2. *Basal cell carcinoma*: 1.61904007328127 (sangat rendah, hampir 0%).
3. *Benign keratosis-like lesions*: 1.2040394318984596 (sangat rendah, hampir 0%).
4. *Dermatofibroma*: 0.01106695105815446 (sekitar 1.11%).
5. *Melanocytic nevi*: 0.9871448278420641 (sekitar 98.71%)
6. *Melanoma*: 1.9077714579661e-05 (sangat rendah, hampir 0%)
7. *Vascular lesions*: 0.001718022739634076 (sekitar 0.17%)

2) *Predicted class* adalah kelas yang diprediksi oleh model berdasarkan nilai probabilitas tertinggi. Dalam kasus ini, kelas yang diprediksi adalah *Melanocytic nevi* (Nevus melanositik) dengan *confidence score* tertinggi yaitu 0.9871448278420641 atau sekitar 98,71%.

5) Hasil Aplikasi

a. Halaman *Splash Screen*

Pada halaman ini terdapat informasi tentang aplikasi. Halaman ini menandakan bahwa aplikasi berhasil dijalankan. Halaman ini adalah halaman pertama yang akan diarahkan ke halaman berikutnya. Tampilan halaman *splash screen* bisa dilihat pada Gambar 4.22.

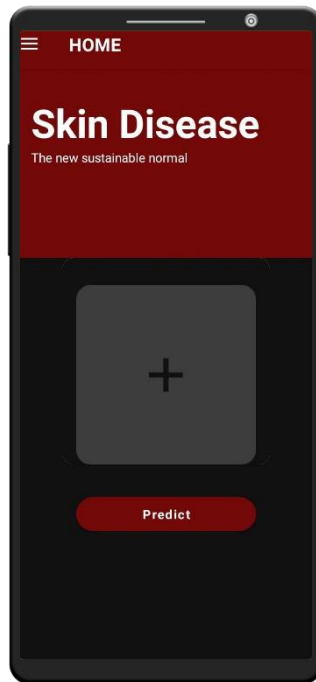


Gambar 4. 22 Halaman *Splash Screen*

b. Halaman Prediksi

Halaman prediksi adalah halaman utama yang paling penting dalam aplikasi ini. Pada halaman ini, pengguna dapat memprediksi jenis penyakit kanker kulit. Halaman ini menampilkan pratinjau gambar atau hasil gambar yang telah dimasukkan oleh pengguna, memiliki tombol galeri dan untuk *input* gambar yang akan diprediksi, menampilkan hasil

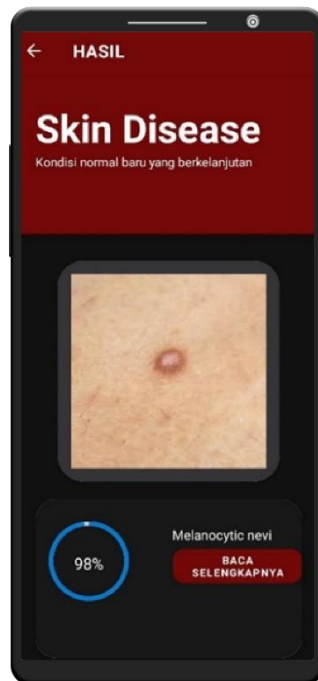
prediksi, dan memiliki tombol untuk melakukan prediksi gambar. Tampilan prediksi dapat dilihat pada Gambar 4.23.



Gambar 4. 23 Halaman Prediksi

c. Halman Hasil Prediksi

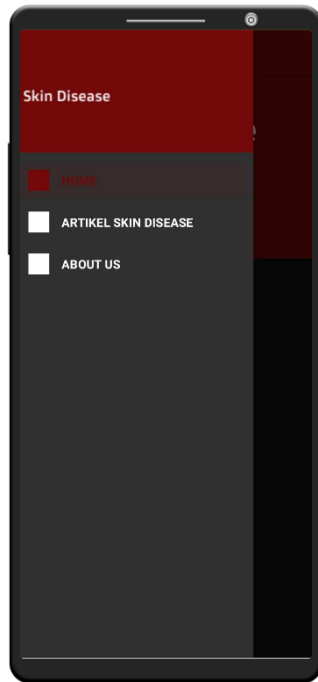
Halaman hasil prediksi adalah halaman setelah melakukan prediksi di halaman utama tadi. Halaman ini menampilkan hasil *input* gambar dan terdapat akurasi dari gambar tersebut. Halaman ini juga terdapat menu baca selengkapnya yang bertujuan untuk memudahkan pengguna mengetahui artikel dari gambar tersebut. Tampilan hasil prediksi dapat dilihat pada Gambar 4.24.



Gambar 4. 24 Halaman Hasil Prediksi

d. Halaman *Navigation*

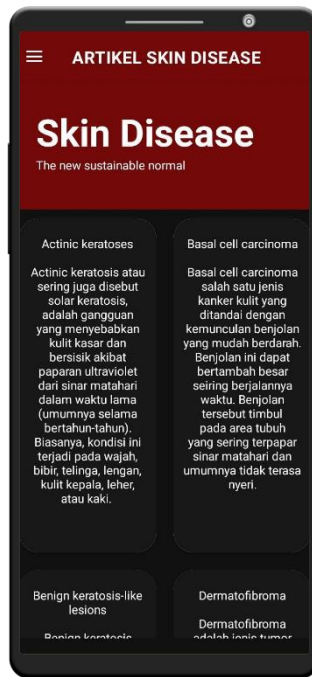
Halaman *navigation* adalah halaman yang mengarahkan pengguna untuk untuk memilih menu lain. Halaman ini terdapat tiga pilihan menu yaitu menu prediksi, menu artikel dan menu tentang. Tampilan *navigation* dapat dilihat pada Gambar 4.25.



Gambar 4. 25 Halaman *Navigation*

e. Halaman Artikel

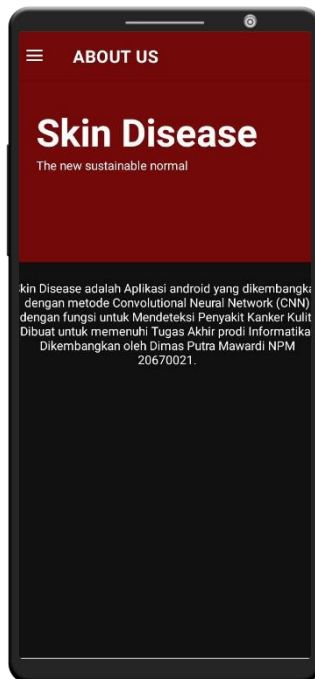
Halaman artikel berisi informasi tentang jenis penyakit kanker kulit. Pengguna dapat melihat informasi mengenai kategori tersebut. Artikel ini mencakup pengertian tentang jenis penyakit kanker kulit. Tampilan artikel dapat dilihat pada Gambar 4.26



Gambar 4. 26 Halaman Artikel

f. Halaman Tentang

Halaman tentang adalah halaman yang menyediakan informasi mengenai aplikasi ini dan pengembangnya. Pada halaman ini, pengguna dapat menemukan deskripsi umum tentang aplikasi, latar belakang pembuatannya. Tampilan halaman tentang ini mencakup teks informatif terkait. Tampilan tentang dapat dilihat pada Gambar 4.27



Gambar 4. 27 Halaman Tentang

4. Pengujian

Pengujian dilakukan untuk memastikan bahwa aplikasi berfungsi dengan baik dan memenuhi standar kualitas. Dalam penelitian ini, tiga jenis pengujian digunakan, yaitu *white box testing*, *black box testing*, dan *user acceptance testing*. Rincian pengujian yang dilakukan dalam penelitian ini adalah sebagai berikut.

a. *White Box Testing*

White box testing adalah teknik pengujian perangkat lunak yang melibatkan pemeriksaan kode sumber dari program untuk mendeteksi adanya kesalahan. Tujuan dari pengujian *white box* adalah memastikan setiap bagian dari kode berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Hasil pengujian *white box* dapat dilihat pada Tabel 4.9.

Tabel 4. 9 *White Box Testing*

Node	Code	Pengertian
1	<code>buttonPredict.setOnClickListener(new View.OnClickListener() { ... });</code>	Memastikan bahwa <i>OnClickListener</i> diatur untuk Button.
2	<code>@Override protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { super.onActivityResult(requestCode, resultCode, data); if (requestCode == 100) { imageView.setImageURI(data.getData()); Uri uri = data.getData(); try { img = MediaStore.Images.Media.getBitmap(this.getContentResolver(), uri); } catch (IOException e) { e.printStackTrace(); } }</code>	Mengatur tampilan pratinjau gambar. Memastikan gambar yang dipilih pengguna ditampilkan di <i>ImageView</i> dan diubah menjadi <i>Bitmap</i> .
3	<code>@Override public void onClick(View view) { img = Bitmap.createScaledBitmap(img, 224, 224, true); try { Shades model = Shades.newInstance(getApplicationContext());</code>	Memastikan model <i>Shades</i> diinisialisasi dengan benar..
4	<code>// Creates inputs for reference. TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 224, 224, 3}, DataType.FLOAT32); TensorImage tensorImage = new TensorImage(DataType.FLOAT32);</code>	Membuat <i>TensorBuffer</i> dengan ukuran tetap, memuat gambar ke dalam <i>TensorImage</i> , dan

	<pre> tensorImage.load(img); ByteBuffer byteBuffer = tensorImage.getBuffer(); // Toast.makeText(getApplicationContext(), "" + byteBuffer.toString() + " = " + inputFeature0.getBuffer().toString(), Toast.LENGTH_SHORT).show(); inputFeature0.loadBuffer(byteBuffer); </pre>	mendapatkan ByteBuffer.
5	<pre> // Runs model inference and gets result. Shades.Outputs outputs = model.process(inputFeature0); TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer() ; // Releases model resources if no longer used. </pre>	Memproses input dengan model dan mendapatkan output.
6	<pre> model.close(); float[] arr = outputFeature0.getFloatArray(); int max_ind = 0; String str = "\n" + arr[0]; </pre>	Menutup model.
7	<pre> for (int i=1; iarr[max_ind]) { max_ind = i; str += "\n" + arr[i]; } } Toast.makeText(getApplicationContext(), "" + Utils.hashMap.get(max_ind), Toast.LENGTH_SHORT).show(); </pre>	Menampilkan hasil prediksi menggunakan Toast.
8	<pre> } catch (IOException e) { // TODO Handle the exception } } </pre>	Menangani pengecualian yang mungkin terjadi selama pemrosesan gambar dan model.

1) Basis *Path* untuk Fungsi Prediksi

Basis *path* (jalur dasar) untuk pengujian *white box* dari fungsi prediksi yang diimplementasikan di MainActivity.java. Basis *path* testing adalah teknik yang digunakan untuk memastikan bahwa semua jalur eksekusi yang mungkin dalam kode telah diuji setidaknya sekali. Berikut adalah langkah-langkah dalam basis path testing untuk fungsi prediksi.

a) Mengidentifikasi Jalur Dasar

Pertama, kita akan mengidentifikasi semua jalur eksekusi dalam kode yang relevan dengan fungsi prediksi. Setiap node yang telah kita tetapkan sebelumnya akan menjadi bagian dari jalur eksekusi.

b) Menghitung Kompleksitas Siklomatik

Kompleksitas siklomatik adalah metrik yang digunakan untuk mengukur jumlah jalur linier independen dalam program. Rumus untuk menghitung kompleksitas siklomatik ($V(G)$) adalah:

$$V(G) = E - N + 2 \quad (4.14)$$

Dimana:

- E adalah jumlah *edge* (garis alur dalam *flow graph*)
- N adalah jumlah *node* (titik keputusan dalam *flow graph*)

2) *Flow Graph* untuk Fungsi Prediksi

Berikut adalah representasi *flow graph* dari kode fungsi prediksi.

Nodes: 8

Edges: 8

Menggunakan rumus kompleksitas siklomatik:

$$V(G) = E - N + 2 = 8 - 8 + 2 = 2 \quad (4.15)$$

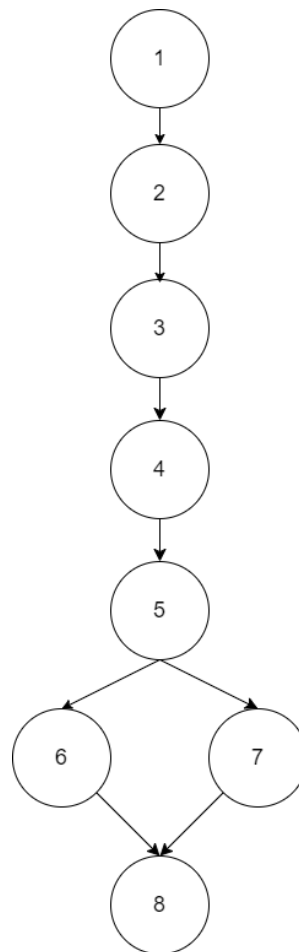
Ini menunjukkan bahwa ada 2 *path* jalur independen dalam fungsi prediksi.

Jalur Independen

Path 1 : 1, 2, 3, 4, 5, 6, 8

Path 2 : 1, 2, 3, 4, 5, 7, 8

Hasil pengujian *white box* yang mendapatkan hasil 2 *independent path* dengan rentang nilai 1 sampai 10, yang berarti tidak kompleks dan mudah untuk perbaikan pada *code* program sehingga dikatakan baik. Pengujian *white box* memastikan bahwa setiap path diuji dengan baik untuk menjaga kualitas dan stabilitas aplikasi.



Gambar 4. 28 *Flowgraph Basis Path*

b. *Black Box Testing*

Black box testing digunakan untuk memastikan aplikasi bekerja sesuai dengan harapan pengguna. Pengujian ini dilakukan oleh tiga dosen informatika. Pengujian *black box* dibagi menjadi tiga bagian perencanaan pengujian, hasil pengujian, dan kesimpulan pengujian. Pengujian ini bertujuan untuk mengevaluasi fungsionalitas aplikasi dari perspektif pengguna.

1) *Form Black Box Testing*

Form black box testing aplikasi dapat dilihat pada Tabel 4.10.

Tabel 4. 10 *Black Box Testing*

Nama Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
<i>Splash Screen</i>	Pengguna menunggu transisi ke halaman berikutnya.	Pengguna melihat <i>splash screen</i> .	Aplikasi akan langsung berpindah ke halaman berikutnya.		
Halaman menu Prediksi	Pengguna menekan tombol upload galeri.	Pengguna dapat masuk ke galeri.	Aplikasi akan menampilkan galeri.		
	Pengguna memilih gambar di galeri.	Pengguna dapat memilih gambar yang akan diprediksi dan	Aplikasi akan menampilkan hasil gambar yang telah dipilih pengguna.		

		menekan tombol prediksi.			
Halaman menu <i>Navigation</i>	Pengguna menekan tombol menu <i>navigation</i> .	Pengguna dapat memilih salah satu menu dan masuk ke halaman yang dituju.	Aplikasi akan menampilkan halaman yang dituju.		
Halaman menu Arikel	Pengguna menekan menu artikel pada halaman menu <i>navigation</i> dan melakukan <i>scrolling text</i> .	Pengguna dapat melihat daftar artikel	Aplikasi akan menampilkan daftar artikel		
Halaman menu Tentang	Pengguna menekan menu tentang pada halaman	Pengguna dapat melihat isi halaman menu tentang aplikasi.	Aplikasi akan menampilkan halaman menu tentang aplikasi.		

	menu <i>navigation.</i>				
--	----------------------------	--	--	--	--

2) Hasil Perhitungan *Black Box Testing*

Hasil perhitungan *black box testing* aplikasi dapat dilihat pada Tabel 4.11.

Tabel 4. 11 Hasil *Black Box Testing*

Nama Pengujian	<i>Test Case</i>	<i>Hasil yang Diharapkan</i>	Hasil yang Didapatkan	Keterangan					
				Diterima			Ditolak		
				1	2	3	1	2	3
<i>Splash Screen</i>	Pengguna menunggu transisi ke halaman berikutnya .	Pengguna melihat <i>splash screen.</i>	Aplikasi akan langsung berpindah ke halaman berikutnya.	√	√	√			
Halaman menu Prediksi	Pengguna menekan tombol upload galeri.	Pengguna dapat masuk ke galeri.	Aplikasi akan menampilkan galeri.	√	√	√			
	Pengguna memilih gambar di galeri.	Pengguna dapat memilih gambar yang akan diprediksi dan menekan	Aplikasi akan menampilkan hasil gambar yang telah dipilih pengguna.	√	√	√			

		tombol prediksi.							
Halaman menu <i>Navigation</i>	Pengguna menekan tombol menu <i>navigation</i> .	Pengguna dapat memilih salah satu menu dan masuk ke halaman yang dituju.	Aplikasi akan menampilkan halaman yang dituju.	√	√	√			
Halaman menu Arikel	Pengguna menekan menu artikel pada halaman menu <i>navigation</i> dan melakukan <i>scrolling text</i> .	Pengguna dapat melihat daftar artikel	Aplikasi akan menampilkan daftar artikel	√	√	√			
Halaman menu Tentang	Pengguna menekan menu tentang pada halaman menu	Pengguna dapat melihat isi halaman menu tentang aplikasi.	Aplikasi akan menampilkan halaman menu tentang aplikasi.	√	√	√			

	<i>navigation</i>								
--	-------------------	--	--	--	--	--	--	--	--

3) Kesimpulan Hasil *Black Box Testing*

Berdasarkan black box testing dari 6 pengujian pada aplikasi yang didapat dari 3 responden, berikut ini hasil *black box testing*.

- Pengujian pertama

$$\text{Tercapai : } \frac{6}{6} \times 100\% = 100\% \quad (4.16)$$

$$\text{Gagal : } \frac{0}{6} \times 100\% = 0\% \quad (4.17)$$

- Pengujian kedua

$$\text{Tercapai : } \frac{6}{6} \times 100\% = 100\% \quad (4.18)$$

$$\text{Gagal : } \frac{0}{6} \times 100\% = 0\% \quad (4.19)$$

- Pengujian ketiga

$$\text{Tercapai : } \frac{6}{6} \times 100\% = 100\% \quad (4.20)$$

$$\text{Gagal : } \frac{0}{6} \times 100\% = 0\% \quad (4.21)$$

$$\begin{aligned} \text{Jumlah presentase rata-rata} &= \frac{300\%}{3} = 100\% \quad (4.22) \\ \text{tercapai} & \end{aligned}$$

$$\begin{aligned} \text{Jumlah presentase rata-rata} &= \frac{0\%}{3} = 0\% \quad (4.23) \\ \text{gagal} & \end{aligned}$$

Berdasarkan hasil perhitungan tersebut, dari enam pengujian yang dilakukan oleh 3 responden, semua pengujian black box berhasil mencapai tingkat keberhasilan 100%, sementara yang gagal mencapai tingkat kegagalan 0%. Dari hasil ini, dapat

disimpulkan bahwa aplikasi berfungsi sesuai dengan fungsionalitas yang diharapkan.

- *User Acceptance Testing (UAT)*

User acceptance testing dilakukan untuk memverifikasi bahwa aplikasi klasifikasi jenis penyakit kanker kulit memenuhi persyaratan dan ekspektasi pengguna. Proses pengujian ini memberikan kesempatan bagi pengguna untuk memberikan umpan balik dan memastikan bahwa aplikasi siap untuk digunakan dalam lingkungan produksi atau operasional. Pengujian *user acceptance* ini melibatkan 5 responden yang diminta untuk mengisi kuesioner dengan skala likert dari 1 hingga 5. Hasil kelayakan aplikasi dapat ditemukan dalam Tabel 4.12.

Tabel 4. 12 *User Acceptance Testing (UAT)*

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi klasifikasi jenis penyakit kanker kulit dapat bermanfaat bagi pengguna?					
2.	Apakah aplikasi klasifikasi jenis penyakit kanker kulit memberikan informasi tentang jenis penyakit					

	penyakit kanker kulit dan cara mengatasinya?					
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit mudah dioperasikan?					
4.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai yang diharapkan?					
5.	Apakah klasifikakasi jenis penyakit kanker kulit berisi informasi yang dibutuhkan?					
6.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai keperluan Anda?					
Aspek <i>User Interface</i> (UI)						
7.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang mudah dipahami?					

8.	Apakah aplikasi klasifikasi jenis penyakit kanker kulit memiliki tampilan yang menarik?					
9.	Apakah aplikasi klasifikasi jenis penyakit kanker kulit memiliki tema warna yang enak dilihat?					
10.	Apakah aplikasi klasifikasi jenis penyakit kanker kulit perlu dikembangkan lagi?					

Keterangan :

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Berikut merupakan hasil kuesioner *user acceptance testing* yang telah dibagikan kepada 3 responden. Hasil *user acceptance testing* dapat dilihat pada Tabel 4.13.

Tabel 4. 13 Hasil *User Acceptance Testing* (UAT)

Pernyataan	Hasil Pengujian		
	Responden 1	Responden 2	Responden 3
1	5	4	3
2	3	3	2

3	4	4	4
4	5	5	4
5	5	4	4
6	4	5	4
7	5	4	5
8	4	4	4
9	4	5	4
10	5	5	5
Jumlah Skor	44	43	39
Presentase	88%	86%	78%
Total	252%		

Dari hasil persentase yang diberikan oleh 3 responden untuk setiap pertanyaan yang mencakup aspek kegunaan, kemudahan penggunaan, dan antarmuka pengguna (UI), kami kemudian mencari nilai rata-ratanya. Tujuannya adalah untuk mengetahui tingkat penerimaan aplikasi yang dikembangkan oleh responden. Nilai rata-rata ini dapat dihitung menggunakan persamaan berikut.

$$\text{Presentase rata-rata} = \frac{\text{Jumlah tota presentase}}{\text{Jumlah responden}} \quad (4.24)$$

$$\text{Presentase rata-rata} = \frac{252\%}{3} = 84\% \quad (4.25)$$

Dengan daftar kategori presentase sebagai berikut.

0% - 20% = Sangat Kurang

21% - 40% = Kurang

41% - 60% = Cukup Baik

61% - 80% = Baik

81% - 100% = Sangat Baik

Dari perhitungan tersebut, diperoleh presentase rata-rata dari ketiga aspek sebesar 84%. Dengan demikian, dapat disimpulkan bahwa pengujian UAT pada aplikasi ini mendapat kategori "Sangat Baik".

B. Pembahasan

Dari hasil penelitian yang menggunakan metode *waterfall*, aplikasi deteksi jenis kanker kulit dibuat melalui 4 tahapan, yaitu analisis kebutuhan, desain, implementasi, dan pengujian.

1. Analisis Kebutuhan

Pada tahap pertama, yaitu tahap analisis kebutuhan, dilakukan pengumpulan, pemahaman, dan penjelasan secara rinci tentang kebutuhan yang diperlukan untuk pengembangan aplikasi. Beberapa jenis analisis kebutuhan termasuk analisis kebutuhan data, analisis kebutuhan sistem atau aplikasi, serta analisis kebutuhan fungsional. Dalam konteks pengembangan aplikasi deteksi jenis penyakit kanker kulit, langkah-langkah ini bertujuan untuk mengidentifikasi dan merinci kebutuhan yang spesifik untuk aplikasi tersebut.

2. Desain

Setelah menyelesaikan tahap pengumpulan kebutuhan aplikasi, langkah selanjutnya adalah tahap desain. Dalam tahap ini, penggunaan model perancangan *unified modeling language* (UML) seperti *use case*, *activity*, *sequence*, dan *class diagram* menjadi penting. Melalui diagram-diagram tersebut, penulis dapat mengidentifikasi fungsi-fungsi aplikasi serta alur kerja yang terjadi di dalamnya. *Use Case Diagram* membantu dalam memahami interaksi pengguna dengan sistem, *activity diagram* menjelaskan langkah-langkah proses secara rinci, *sequence diagram* menggambarkan interaksi antara objek dalam waktu tertentu, dan *class diagram* menunjukkan struktur sistem dalam hal kelas-kelas dan hubungan antar kelas tersebut. Selain UML, penulis juga merancang *wireframe* sederhana untuk desain antarmuka pengguna, memastikan

tata letak dasar elemen-elemen UI sesuai dengan kebutuhan fungsional dan mudah digunakan oleh pengguna.

3. Implementasi

Tahap berikutnya adalah tahap implementasi, Proyek ini menggunakan metode klasifikasi citra dengan jaringan saraf konvolusi (CNN) untuk mengklasifikasikan tujuh jenis lesi kulit menggunakan dataset HAM10000 dari Kaggle. Dalam proyek ini, data yang terdiri dari 10.015 gambar lesi kulit dengan metadata seperti jenis kelamin, usia, dan lokasi lesi di tubuh diolah melalui beberapa tahap *preprocessing*, termasuk *resizing*, normalisasi, dan *one-hot encoding*. Model CNN yang dirancang terdiri dari beberapa lapisan konvolusi dan pooling, dengan lapisan *dropout* untuk mengurangi *overfitting*, dan dilatih menggunakan *optimizer* Adam serta *loss function categorical cross-entropy*.

Pelatihan model dilakukan selama 200 epoch, di mana setiap epoch merupakan satu siklus penuh melalui seluruh dataset pelatihan. Epoch yang lebih banyak memungkinkan model untuk belajar pola yang lebih kompleks, meskipun harus dipantau untuk menghindari *overfitting*, di mana model menjadi terlalu cocok dengan data pelatihan dan kurang mampu menggeneralisasi data baru. Setelah pelatihan, model dievaluasi dengan menggunakan metrik seperti confusion matrix dan classification report, yang menunjukkan performa akurasi yang baik. Akhirnya, model yang sudah dilatih dikonversi ke format TensorFlow Lite untuk penggunaan pada perangkat dengan sumber daya terbatas, menjadikannya alat yang efektif untuk membantu dalam diagnosis awal lesi kulit.

4. Pengujian

Tahap terakhir adalah pengujian untuk memastikan kualitas aplikasi yang berfungsi dengan baik. Penulis melakukan 3 jenis pengujian *Black Box*, *White Box*, dan *User Acceptance Testing (UAT)*. Pengujian *white box* yang mendapatkan hasil 2 *independent path* yang berarti tidak

kompleks dan mudah untuk perbaikan pada *code* program sehingga dikatakan baik. Dalam pengujian *Black Box*, berhasil mencapai presentase 100% keberhasilan, sedangkan kegagalan mendapat presentase 0% dari 3 responden dan 6 pengujian. Pengujian *user acceptance test* (UAT) juga berhasil dengan presentase 84% dari 3 responden dengan 10 pertanyaan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis, beberapa kesimpulan dapat ditarik sebagai berikut.

1. Dari hasil proses penelitian dataset penyakit kanker kulit menggunakan algoritma CNN dengan jumlah dataset 10015 citra. Dataset ini dibagi menjadi 2 yaitu *training* sebanyak 7.511 (75%) dan *test* sebanyak 2.504 (25%). Model ini menggunakan teknik *transfer learning*, mendapatkan *accuracy* sebesar sebesar 97,38%, *validation accuracy* sebesar 97,39% dari 200 *epoch*.
2. Penelitian ini mencapai tingkat *accuracy* sebesar 97,38% dengan rata-rata nilai *precision*, *recall* dan *f1_score* 97%. Penelitian ini melibatkan 10015 gambar dermoskopi kulit dengan pembagian ke dalam 7 kelas.
3. Penulis melakukan 3 jenis pengujian untuk memastikan kualitas aplikasi *white box*, *black box*, dan *user acceptance testing* (UAT). Pengujian *white box* yang mendapatkan hasil 2 *independent path* yang berarti tidak kompleks dan mudah untuk perbaikan pada *code* program sehingga dikatakan baik. Hasil pengujian *black box* menunjukkan keberhasilan 100%, sedangkan kegagalan mencapai 0% dari 3 responden dalam 6 pengujian. Pengujian *user acceptance test* (UAT) berhasil mencapai 84%.

B. Saran

Dari hasil penelitian yang telah dilakukan, penulis memberikan beberapa saran yang dapat menjadi pertimbangan untuk penelitian mendatang, antara lain:

1. Lakukan penelitian lebih lanjut untuk mengatasi keterbatasan yang ada, seperti memperbaiki kesalahan klasifikasi dan meningkatkan robustitas model terhadap gambar dengan kualitas rendah.

2. Eksplorasi berbagai arsitektur CNN dan teknik-teknik optimasi seperti ensemble learning, fine-tuning, dan penggunaan hyperparameter tuning untuk meningkatkan performa model.

DAFTAR PUSTAKA

- [1] Ricky Yohannes and Muhammad Ezar Al Rivan, “Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM,” *Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM*, vol. 2, no. 2, pp. 133–144, Apr. 2022.
- [2] M. Faruk and N. Nafi’iyah, “Telematika Klasifikasi Kanker Kulit Berdasarkan Fitur Tekstur, Fitur Warna Citra Menggunakan SVM dan KNN,” vol. 13, no. 2, pp. 100–109, 2020, doi: 10.35671/telematika.v13i2.987.
- [3] A. Lawi and A. Muh Amil Siddik, “Prosiding Seminar Nasional Teknik Elektro dan Informatika (SNTEI) 2022-Teknik Informatika,” 2022.
- [4] F. Nuraeni, Y. H. Agustin, and E. N. Yusup, “Aplikasi Pakar Untuk Diagnosa Penyakit Kulit Menggunakan Metode Forward Chaining Di Al Arif Skin Care Kabupaten Ciamis,” pp. 6–7, 2016.
- [5] H. P. Kekal and D. U. E. Saputri, “Optimization of Melanoma Skin Cancer Detection with the Convolutional Neural Network,” *Journal Medical Informatics Technology*, pp. 53–58, Jun. 2023, doi: 10.37034/medinftech.v1i2.10.
- [6] I. W. Prastika, E. Zuliarso, J. T. Lomba, J. No, and S. 50241, “Deteksi Penyakit Kulit Wajah Menggunakan Tensorflow Dengan Metode Convolutional Neural Network,” *Jurnal Manajemen informatika & Sistem Informasi*, vol. 4, no. 2, 2021, [Online]. Available: <http://e-journal.stmiklombok.ac.id/index.php/misi>
- [7] Luqman Hakim, Z. Sari, and H. Handhajani, “Klasifikasi Citra Pigmen Kanker Kulit Menggunakan Convolutional Neural Network,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 379–385, Apr. 2021, doi: 10.29207/resti.v5i2.3001.
- [8] A. Salam, F. Yanto, S. Agustian, and S. Ramadhani, “Kajian Ilmiah Informatika dan Komputer Perbandingan Klasifikasi Citra CT-Scan Kanker Paru-Paru Menggunakan Contrast Stretching Pada CNN dengan EfficientNet-B0,” *Media Online*, vol. 4, no. 3, pp. 1341–1351, 2023, doi: 10.30865/klik.v4i3.1448.
- [9] A. Pumsirirat and L. Yan, “Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine,” 2018. [Online]. Available: www.ijacsa.thesai.org
- [10] M. Ezar, A. Rivan, D. Alwyn, and G. Riyadi, “Perbandingan Arsitektur LeNet dan AlexNet Pada Metode Convolutional Neural Network Untuk Pengenalan American Sign Language,” 2021. [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jkt/>

- [11] R. Adi Saputra and U. Halu Oleo, “Klasifikasi Kanker Kulit Berdasarkan Data Citra Benign Dan Malignant Menggunakan Convolutional Neural Network,” 2024. [Online]. Available: www.kaggle.com/fanconic/skin-
- [12] A. Ahmad, “Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning,” 2017. [Online]. Available: www.teknoindonesia.com
- [13] S. R. Fakultas, T. Informasi, U. Islam, K. Muhammad, and A. Al Banjari, “Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm,” 2020.
- [14] D. A. Nurlitasari, R. Magdalena, and R. Y. N. Fu’adah, “Analisis Performansi Sistem Klasifikasi Kanker Kulit Menggunakan Convolutional Neural Network,” *Journal Of Electrical And System Control Engineering*, vol. 5, no. 2, pp. 91–99, Feb. 2022, doi: 10.31289/jesce.v5i2.5691.
- [15] R. F. Muharram *et al.*, “Implementasi artificial intelligence untuk deteksi masker secara realtime dengan tensorflow dan ssdmobilenet Berbasis python,” *Jurnal Widya*, vol. 3, no. 2, pp. 281–290, 2022, [Online]. Available: <https://jurnal.amikwidyaloka.ac.id/index.php/awl>
- [16] F. Royana, P. Yuniar Maulida, R. Nurul Hasanah, and S. Setia Rahayu, “Aplikasi Mobile Deteksi Dini Kanker Kulit Berdasarkan Image Processing | 100,” 2021. [Online]. Available: <http://journal.pwmjateng.com/index.php/jle>
- [17] A. Performa *et al.*, “Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin,” 2018. [Online]. Available: <https://www.researchgate.net/publication/329525878>
- [18] A. D. Prakarsa, K. Candra Brata, and H. Fitriyah, “Pembangunan Aplikasi Pengenalan Kanker Melanoma berbasis Android,” 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [19] Haviluddin, “Memahami Penggunaan UML (Unified Modelling Language),” *Jurnal Informatika Mulawarman*, vol. 06, no. 1, pp. 1–15, 2011.
- [20] M. Nazir, S. Fajariani Putri, and D. Malik, “Perancangan Aplikasi E-VOTING Menggunakan Diagram UML (Unified Modelling Language),” 2022.
- [21] A. Feby Prasetya and U. Lestari Dewi Putri, “Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language),” 2022.
- [22] M. Syarif and W. Nugraha, “Pemodelan Diagram Uml Sistem Pembayaran Tunai Pada Transaksi E-Commerce,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 4, no. 1, 2020.

- [23] W. Aliman, "Perancangan Perangkat Lunak untuk Menggambar Diagram Berbasis Android," *Syntax Literate ; Jurnal Ilmiah Indonesia*, vol. 6, no. 6, p. 3091, Jun. 2021, doi: 10.36418/syntax-literate.v6i6.1404.
- [24] M. E. Khan and F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," 2012. [Online]. Available: www.ijacsa.thesai.org
- [25] R. Roslidar, M. R. Syahputra, R. Muharar, and F. Arnia, "Adaptasi Model CNN Terlatih pada Aplikasi Bergerak untuk Klasifikasi Citra Ternal Payudara," *Jurnal Rekayasa ElektriKa*, vol. 18, no. 3, Sep. 2022, doi: 10.17529/jre.v18i3.8754.
- [26] S. Gordon *et al.*, "Best Practice Recommendations: User Acceptance Testing for Systems Designed to Collect Clinical Outcome Assessment Data Electronically," *Ther Innov Regul Sci*, vol. 56, no. 3, pp. 442–453, May 2022, doi: 10.1007/s43441-021-00363-z.
- [27] L. Setiyani, "Implementasi Cybersecurity pada Operasional Organisasi," 2021.

LAMPIRAN

Lampiran 1 Lembar Bimbingan Dosen Pembimbing 1



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrisng@gmail.com, Homepage : www.upgrisng.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Dimas Putra Mawardi
NPM : 20670021
Program Studi : Informatika
Judul Skripsi : Klasifikasi Jenis Penyakit Kanker Keras dengan Algoritma Convolution Neural Network

Dosen Pembimbing I : Mega Novita Ph.D
Dosen Pembimbing II : Hugroho Dwi Saputra S.kom., M.kom

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1	07/03/24	Judul	
2	14/03/24	Bab I	
3	21/03/24	Bab I	
4	02/05/24	Bab II	
5	13/05/24	Bab III	
6	30/05/24	Bab IV	
7	3/6/24	Bab IV Revisi	
8	3/6/24	Bab V ACC SKRIPSI	

Dosen Pembimbing I,

Mega Novita, S.Si., M.Si., M.Nat.Sc.,
Ph.D

NIDN. 0615118801

Mahasiswa,

Dimas Putra Mawardi

NPM. 20670021

Lampiran 2 Lembar Bimbingan Dosen Pembimbing 2



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrisng@gmail.com, Homepage : www.upgrisng.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Dimas Putra Mawardi
 NPM : 20670021
 Program Studi : Informatika
 Judul Skripsi : Klasifikasi Jenis Penyakit Kanker Kulit dengan Algoritma Convolution Neural Network
 Dosen Pembimbing I : Mega Novita Ph.D
 Dosen Pembimbing II : Mugroho Dwi Saputra S.kom., M.kom

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1	07/03/24	Judul	
2	14/03/24	Bab I	
3	21/03/24	Bab I	
4	02/05/24	Bab II	
5	13/05/24	Bab III	
6	30/05/24	Bab IV	
7	3/6/24	Bab IV Revisi	
8	3/6/24	Bab V ACC SKRIPSI	

Dosen Pembimbing I,

Mega Novita, S.Si., M.Si., M.Nat.Sc.,
Ph.D

NIDN. 0615118801

Mahasiswa,

Dimas Putra Mawardi

NPM. 20670021

Lampiran 3 Kuesioner Pengujian Black Box

Kuesioner Pengujian *Black Box* Pada Aplikasi Klasifikasi Jenis Penyakit Kanker kulit

Nama Penguji : Bambang Agus H. S.kom, M.kom

Tanggal Pengujian : 27 Mei 2024

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
<i>Splash Screen</i>	Pengguna menunggu transisi ke halaman berikutnya.	Pengguna melihat <i>splash screen</i> .	Aplikasi akan langsung berpindah ke halaman berikutnya.	U	
Halaman menu Prediksi	Pengguna menekan tombol upload galeri.	Pengguna dapat masuk ke galeri.	Aplikasi akan menampilkan galeri.	✓	
	Pengguna memilih gambar di galeri.	Pengguna dapat memilih gambar yang akan diprediksi dan menekan tombol prediksi.	Aplikasi akan menampilkan hasil gambar yang telah dipilih pengguna.	✓	
Halaman menu <i>Navigation</i>	Pengguna menekan tombol menu <i>navigation</i> .	Pengguna dapat memilih salah satu menu dan masuk ke halaman yang dituju.	Aplikasi akan menampilkan halaman yang dituju.	✓	

Halaman menu Arikel	Pengguna menekan menu artikel pada halaman menu <i>navigation</i> dan melakukan <i>scrolling text</i> .	Pengguna dapat melihat daftar artikel	Aplikasi akan menampilkan daftar artikel	✓	
Halaman menu Tentang	Pengguna menekan menu tentang pada halaman menu <i>navigation</i> .	Pengguna dapat melihat isi halaman menu tentang aplikasi.	Aplikasi akan menampilkan halaman menu tentang aplikasi.	✓	

Saran dari penguji :

Pada Artikel diberi pembeda antara antar kriteria pada Penyakit kulit yang terkena.

Semarang, 27 Mei 2024.



NIDN 0601088201

Kuesioner Pengujian *Black Box* Pada Aplikasi Klasifikasi Jenis Penyakit Kanker kulit

Nama Penguji : *Ranadhan Renaldy, S.kom, m.kom*

Tanggal Pengujian : *27 Mei 2024*

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
<i>Splash Screen</i>	Pengguna menunggu transisi ke halaman berikutnya.	Pengguna melihat <i>splash screen</i> .	Aplikasi akan langsung berpindah ke halaman berikutnya.	✓	
Halaman menu Prediksi	Pengguna menekan tombol upload galeri.	Pengguna dapat masuk ke galeri.	Aplikasi akan menampilkan galeri.	✓	
	Pengguna memilih gambar di galeri.	Pengguna dapat memilih gambar yang akan diprediksi dan menekan tombol prediksi.	Aplikasi akan menampilkan hasil gambar yang telah dipilih pengguna.	✓	
Halaman menu <i>Navigation</i>	Pengguna menekan tombol menu <i>navigation</i> .	Pengguna dapat memilih salah satu menu dan masuk ke halaman yang dituju.	Aplikasi akan menampilkan halaman yang dituju.	✓	

Halaman menu Arikel	Pengguna menekan menu artikel pada halaman menu <i>navigation</i> dan melakukan <i>scrolling text</i> .	Pengguna dapat melihat daftar artikel	Aplikasi akan menampilkan daftar artikel	✓	
Halaman menu Tentang	Pengguna menekan menu tentang pada halaman menu <i>navigation</i> .	Pengguna dapat melihat isi halaman menu tentang aplikasi.	Aplikasi akan menampilkan halaman menu tentang aplikasi.	✓	

Saran dari penguji :

Bisa ditambahkan menu untuk cara penggunaan agar memudahkan
 pengguna baru

Semarang, 27 Mei 2024



Ramadhan Rivaldy, S.Kom, M.Kom

NIDN NPP. 299901639

Kuesioner Pengujian *Black Box* Pada Aplikasi Klasifikasi Jenis Penyakit Kanker kulit

Nama Penguji : Nur Latifah Dwi M.S., M.Kom.

Tanggal Pengujian : 27 Mei 2024.


Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
<i>Splash Screen</i>	Pengguna menunggu transisi ke halaman berikutnya.	Pengguna melihat <i>splash screen</i> .	Aplikasi akan langsung berpindah ke halaman berikutnya.	✓	
Halaman menu Prediksi	Pengguna menekan tombol upload galeri.	Pengguna dapat masuk ke galeri.	Aplikasi akan menampilkan galeri.	✓	
	Pengguna memilih gambar di galeri.	Pengguna dapat memilih gambar yang akan diprediksi dan menekan tombol prediksi.	Aplikasi akan menampilkan hasil gambar yang telah dipilih pengguna.	✓	
Halaman menu <i>Navigation</i>	Pengguna menekan tombol menu <i>navigation</i> .	Pengguna dapat memilih salah satu menu dan masuk ke halaman yang dituju.	Aplikasi akan menampilkan halaman yang dituju.	✓	

Halaman menu Arikel	Pengguna menekan menu artikel pada halaman menu <i>navigation</i> dan melakukan <i>scrolling text</i> .	Pengguna dapat melihat daftar artikel	Aplikasi akan menampilkan daftar artikel	✓	
Halaman menu Tentang	Pengguna menekan menu tentang pada halaman menu <i>navigation</i> .	Pengguna dapat melihat isi halaman menu tentang aplikasi.	Aplikasi akan menampilkan halaman menu tentang aplikasi.	✓	

Saran dari penguji :

Pada artikel sebaiknya penulisan antara keterangan dan judul serta format tersebut.

Semarang, 27 Mei 2024 .


 Himpunan Mahasiswa Ilmu Komunikasi
 NIDN 0623009001

Lampiran 4 Pengujian UAT

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Klasifikakasi Jenis Penyakit Kanker kulit

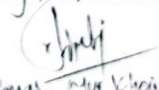
Nama Penguji : Ahmad Muckhois
Tanggal Pengujian : 27 Mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit dapat bermanfaat bagi pengguna?				✓	
2.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memberikan informasi tentang jenis penyakit penyakit kanker kulit dan cara mengatasinya?			✓		
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit mudah dioperasikan?				✓	
4.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai yang diharapkan?					✓
5.	Apakah klasifikakasi jenis penyakit kanker kulit berisi informasi yang dibutuhkan?				✓	
6.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai keperluan Anda?					✓

Aspek User Interface (UI)						
7.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang mudah dipahami?				✓	
8.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tema warna yang enak dilihat?					✓
10.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Semarang, 29 Mei 2024

 Almas Nur Hafid

**Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi
Klasifikakasi Jenis Penyakit Kanker kulit**

Nama Penguji : Tri Setyadudi
 Tanggal Pengujian : 27 Mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit dapat bermanfaat bagi pengguna?			✓		
2.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memberikan informasi tentang jenis penyakit penyakit kanker kulit dan cara mengatasinya?		✓			
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit mudah dioperasikan?				✓	
4.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai yang diharapkan?				✓	
5.	Apakah klasifikakasi jenis penyakit kanker kulit berisi informasi yang dibutuhkan?				✓	
6.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai keperluan Anda?				✓	

Aspek User Interface (UI)						
7.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Surabaya, 27 Mei 2024

[Signature]
Tri Setyobudi

**Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi
Klasifikakasi Jenis Penyakit Kanker kulit**

Nama Penguji : Ananta Agil Nusanjara
 Tanggal Pengujian : 27 Mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit dapat bermanfaat bagi pengguna?					✓
2.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memberikan informasi tentang jenis penyakit penyakit kanker kulit dan cara mengatasinya?			✓		
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit mudah dioperasikan?				✓	
4.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai yang diharapkan?					✓
5.	Apakah klasifikakasi jenis penyakit kanker kulit berisi informasi yang dibutuhkan?					✓
6.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit sesuai keperluan Anda?				✓	

Aspek <i>User Interface</i> (UI)						
7.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi klasifikakasi jenis penyakit kanker kulit perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Semarang, 27 Mei 2024


Ananta Agil Nusantara

Lampiran 5 Lembar Revisian Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Dimas Putra Mawardi
 N P M : 20670021
 Judul : Klasifikasi Jenis Penyakit Kanker Kulit Dengan Algoritma Convolutional Neural Network (CNN) Berbasis Mobile Apps

No	Uraian Revisi	Keterangan
1	Judul ditambahkan "Deteksi Awal Klasifikasi".	
2	Konsistensi penulisan karakter "berikut ini" "di bawah ini". Langsung sebut Tabel/Gambar.	
3	white box atau low black box.	
4	Kalimat untuk menjelaskan Gambar "SANGAT ANEH".	
5	Kesimpulan tidak perlu membandingkan dengan penelitian Lukman et al.	
6	Buat perbandingan hasil dengan Lukman et al dalam bentuk tabel supaya mudah terbaca.	

all
5 Juli 2024

Pengesahan Penguji I






Mega Novita, Ph.D
 NIP/NPP. 158801493


*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Dimas Putra Mawardi
 N P M : 20670021
 Judul : Klasifikasi Jenis Penyakit Kanker Kulit Dengan Algoritma Convolutional Neural Network (CNN) Berbasis Mobile Apps

No	Uraian Revisi	Keterangan
1	tambahkan pembaharuan variabel yg digunakan untuk algoritma CNN cost sperti epochs harus 100!	
2	balok juga harus black ^{white} box kenapa 2 part (black/white)!	
3	Daftar pustaka perbaiki!	 5/2009 10/07
4	Tata tulis!	.

Pengesahan Penguji II


 Nugroho Dwi Saputro, M.Kom
 NIP/NPP. 178801537

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Dimas Putra Mawardi
 N P M : 20670021
 Judul : Klasifikasi Jenis Penyakit Kanker Kulit Dengan Algoritma Convolutional Neural Network (CNN) Berbasis Mobile Apps

No	Uraian Revisi	Keterangan
1	Identifikasi Masalah? Solusi? menjustifikasi ini paling sempurna	all 2/7-2024
2	Punyan Masalah belum menentukan masalah ya sebenarnya.	
3	Use Case ajaib aplikasi tiba? digunakan.	
4	Simbol? activity/kegiatan kurang sesuai	
5	Tahap? Mulai data yang diperoleh sampai dengan didapat per hari belum dihubungkan di pembahasan.	
6	Scrap? tampilan/berikan Border top/tab	

Pengesahan Penguji III

Bambang Agus H. S. Kom, M. Kom
 NIP/NPP. 148201433

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi