

**KLASIFIKASI PENYAKIT KULIT DERMATITIS ATOPIK
DAN PSORIASIS MENGGUNAKAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK
DENGAN MODEL ARSITEKTUR RESNET-50**

LAPORAN TUGAS AKHIR

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



DISUSUN OLEH :

RAKHIMATULFITRIA MEKACAHYANI

NIM 32602000094

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

2024

FINAL PROJECT

**CLASSIFICATION OF ATOPIC DERMATITIS AND PSORIASIS SKIN
DISEASES USING CONVOLUTIONAL NEURAL NETWORK ALGORITHM
WITH RESNET-50 ARCHITECTURE MODEL.**

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



RAKHIMATULFITRIA MEKACAHYANI

NIM 32602000094

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

2024

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **“Klasifikasi Penyakit Kulit Dermatitis Atopik Dan Psoriasis Menggunakan Algoritma Convolutional Neural Network Dengan Model Arsitektur ResNet-50”** ini disusun oleh :

Nama : Rakhimatulfitria Mekacahyani

NIM : 3260200094

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :


Hari : Rabu

Tanggal : 28 Februari 2024

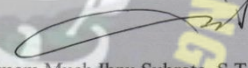
Mengesahkan,

Pembimbing I

Pembimbing II


Badie'ah, S.T., M.Kom

NIDN. 0619018701


Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D

NIDN. 0613037301

Mengetahui,

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




Ir. Sri Mulyono, M.Eng

NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “**Klasifikasi Penyakit Kulit Dermatitis Atopik Dan Psoriasis Menggunakan Algoritma *Convolutional Neural Network* Dengan Model Arsitektur ResNet-50**” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Kamis

Tanggal : 29 Februari 2024

TIM PENGUJI

Anggota I

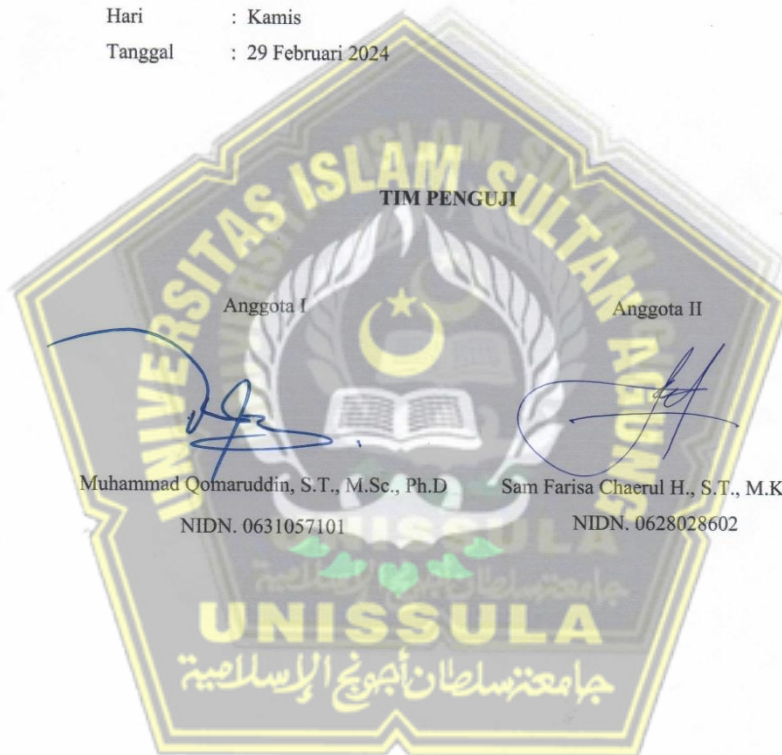
Anggota II

Muhammad Oomaruddin, S.T., M.Sc., Ph.D

Sam Farisa Chaerul H., S.T., M.Kom

NIDN. 0631057101

NIDN. 0628028602



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Rakhimatulfitria Mekacahyani

NIM : 32602000094

Judul Tugas Akhir : Klasifikasi Penyakit Kulit Dermatitis Atopik Dan Psoriasis
Menggunakan Algoritma *Convolutional Neural Network*
Dengan Model Arsitektur ResNet-50

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

Semarang, 29 Februari 2024

Yang Menyatakan,



Rakhimatulfitria Mekacahyani

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayahnya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul “Klasifikasi Penyakit Kulit Dermatitis Atopik Dan Psoriasis Menggunakan Algoritma *Convolutional Neural Network* Dengan Model Arsitektur ResNet-50” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

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

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H., yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana S.T., M.T., IPU., ASEAN.Eng.
3. Dosen Pembimbing I penulis Ibu Badie'ah, S.T., M.Kom yang telah membimbing dan memberikan banyak nasehat dan saran.
4. Dosen Pembimbing II penulis Bapak Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D yang telah membimbing dan memberikan banyak nasehat dan saran.
5. Orang tua dan keluarga penulis yang menjadi *support system* dan mengizinkan untuk menyelesaikan laporan ini.
6. Dan kepada semua pihak yang tidak dapat penulis sebutkan satu persatu atas doa, *support* serta bantuannya.

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

Semarang, 21 Desember 2023

Rakhimatulfitria Mekacahyani

DAFTAR ISI

COVER	i
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
ABSTRAK	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Pembatasan Masalah	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori	7
2.2.1 Dermatitis Atopik.....	7
2.2.2 Psoriasis	8
2.2.3 <i>Deep Learning</i>	9
2.2.4 <i>Convolutional Neural Network</i>	10
2.2.5 ResNet-50.....	12
2.2.6 <i>PyTorch</i>	14
BAB III METODE PENELITIAN	15
3.1 Metode Penelitian.....	15
3.1.1 Studi <i>Literature</i>	15
3.1.2 Pengumpulan Data (<i>Data Collecting</i>).....	15
3.1.3 Pemangkasan Data (<i>Cropping</i>)	16

3.1.4	Pra-Pemrosesan Data (<i>Preprocessing</i>).....	17
3.1.5	Perancangan Model.....	18
3.1.6	Pelatihan Model	20
3.1.7	Perbandingan Model	20
3.1.8	Evaluasi Model.....	22
3.1.9	<i>Deployment Model</i>	23
3.2	<i>Software</i> Yang Digunakan	24
3.3	Perancangan Sistem.....	26
BAB IV HASIL DAN ANALISIS PENELITIAN.....		28
4.1	Hasil dan Analisis	28
4.1.1	Pengujian Parameter yang Berbeda	28
4.1.2	Pengujian <i>Cropping</i> Dengan Kode Program.....	30
4.1.3	Pengujian Model MobileNetV2	31
4.1.4	Hasil Akurasi.....	32
4.1.5	Hasil <i>Confusion Matrix</i>	35
4.2	<i>Deployment</i>	37
4.2.1	<i>Link</i> Streamlit.....	37
4.2.2	<i>Dashboard</i> Streamlit	39
4.2.3	Hasil Prediksi	40
4.3	Pengujian Sistem	40
BAB V KESIMPULAN DAN SARAN.....		42
5.1	Kesimpulan.....	42
5.2	Saran.....	43

DAFTAR PUSTAKA

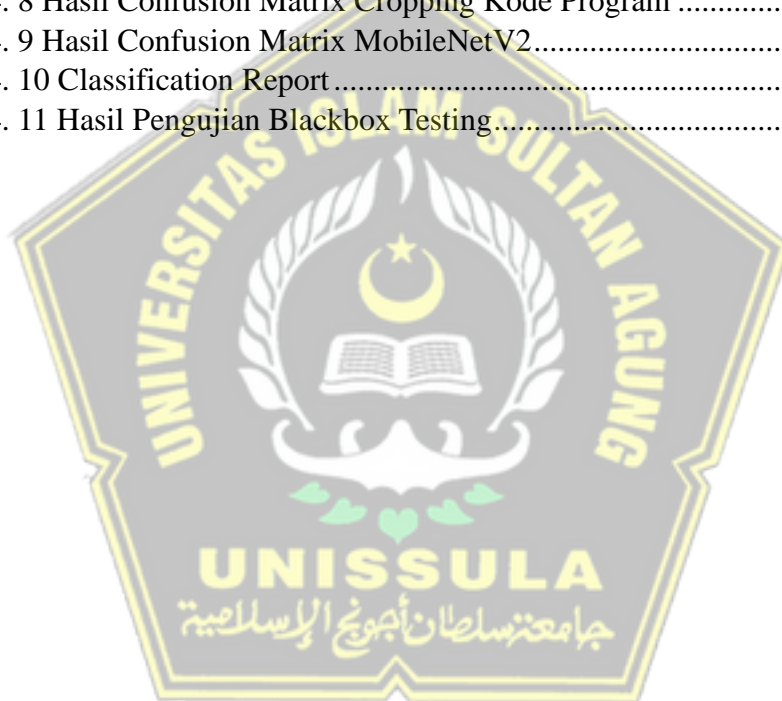
LAMPIRAN

DAFTAR GAMBAR

Gambar 2. 1 Kulit Terjangkit Penyakit Dermatitis Atopik.....	7
Gambar 2. 2 Kulit Terjangkit Penyakit Psoriasis	8
Gambar 2. 3 Lapisan Pada CNN	10
Gambar 2. 4 Model Arsitektur ResNet-50	12
Gambar 3. 1 Data Gambar Dermatitis Atopik.....	16
Gambar 3. 2 Data Gambar Psoriasis	16
Gambar 3. 3 Model Arsitektur ResNet-50 yang Dimodifikasi	19
Gambar 3. 4 Flowchart Streamlit Klasifikasi Penyakit Dermatitis Atopik dan Psoriasis	27
Gambar 4. 1 Pengujian Parameter 1	28
Gambar 4. 2 Pengujian Parameter 2.....	29
Gambar 4. 3 Pengujian Parameter 3.....	29
Gambar 4. 4 Pengujian Parameter 4.....	30
Gambar 4. 5 Pengujian Parameter 5.....	30
Gambar 4. 6 Pengujian Cropping dengan Kode Program.....	31
Gambar 4. 7 Pengujian Model MobileNetV2	32
Gambar 4. 8 Link Streamlit dan Kode Localtunnel	38
Gambar 4. 9 Masukkan Kode Localtunnel	38
Gambar 4. 10 Tampilan Awal Streamlit	39
Gambar 4. 11 Meng-upload gambar	39
Gambar 4. 12 Hasil Prediksi Dermatitis Atopik.....	40
Gambar 4. 13 Hasil Prediksi Psoriasis	40

DAFTAR TABEL

Tabel 3. 1 Pembagian Data Train, Test dan Valid	18
Tabel 3. 2 Konfigurasi Model	21
Tabel 4. 1 Plot Accuracy dan Loss	32
Tabel 4. 2 Model Terbaik Berdasarkan Hasil Akurasi	34
Tabel 4. 3 Hasil Confusion Matrix Skenario 1	35
Tabel 4. 4 Hasil Confusion Matrix Skenario 2	35
Tabel 4. 5 Hasil Confusion Matrix Skenario 3	35
Tabel 4. 6 Hasil Confusion Matrix Skenario 4	35
Tabel 4. 7 Hasil Confusion Matrix Skenario 5	35
Tabel 4. 8 Hasil Confusion Matrix Cropping Kode Program	36
Tabel 4. 9 Hasil Confusion Matrix MobileNetV2	36
Tabel 4. 10 Classification Report	36
Tabel 4. 11 Hasil Pengujian Blackbox Testing	41



ABSTRAK

Dermatitis atopik (eksim) dan psoriasis merupakan penyakit kulit yang umum terjadi dan memiliki kesamaan gejala yang ditandai dengan lesi epidermis yang memerah atau meradang secara tidak normal dan mengalami penebalan yang bervariasi dalam intensitas dan area permukaan kulit yang terkena. Akan tetapi diluar dari kesamaan tersebut, kedua penyakit merupakan jenis penyakit kulit yang berbeda dan memiliki banyak perbedaan diantaranya dari segi penyebab, lokasi gejala, tampilan, bentuk, mekanisme dan pengobatan. Dengan memanfaatkan teknologi AI yaitu menggunakan algoritma *convolutional neural network* model arsitektur ResNet-50 dapat memberikan hasil yang akurat dalam membedakan tipikal kulit yang mengidap psoriasis dan eksem. Data yang dilatih pada penelitian ini sebanyak 500 gambar dengan rincian 250 gambar dermatitis atopik dan 250 gambar psoriasis. Setelah melakukan proses pengujian terhadap berbagai konfigurasi didapatkan hasil akurasi terbaik sebesar 92,75% untuk data *train* dan 88,00% untuk data *test* dengan perbandingan data *train*, data *test* dan data *valid* sebesar 80%:10%:10%. Adapun hasil *confusion matrix* dari model terbaik menghasilkan nilai *accuracy* untuk data *valid* sebesar 88% dan nilai *precision*, *recall*, *f1 score* untuk dermatitis atopik sebesar 95%, 80%, 87% serta nilai *precision*, *recall*, *f1 score* untuk psoriasis sebesar 83%, 96%, 89%. Oleh karena itu, metode ResNet-50 pada skenario 1 tersebut lebih unggul daripada menggunakan skenario 2 – 5, *cropping* dengan kode program, dan metode MobileNetV2 dalam mengklasifikasikan penyakit kulit dermatitis atopik dan psoriasis.

Kata Kunci : Dermatitis Atopik, Psoriasis, ResNet-50

ABSTRACT

Atopic dermatitis (eczema) and psoriasis are common skin diseases and have similarity to symptoms characterized by infant or inflamed epidermis lesions and explicitly varying in the intensity and affected skin surface area. However, outside of the similarity, both diseases are different types of skin diseases and have many differences such as in terms of causes, location of symptoms, display, shape, mechanism and treatment. By utilizing AI technology that is using the Convolutional Neural Network algorithm model of the Resnet-50 architecture can provide accurate results in distinguishing typical skin that psoriasis and eczema. Data trained in this study were 500 images with details of 250 atopic dermatitis images and 250 psoriasis images. After doing the test process against various configurations obtained the best accuracy of 92.75% for Train data and 88.00% for data tests with comparative data train, data test and valid data of 80%: 10%: 10%. The result of the confusion matrix of the best model produces accuracy value for valid data of 88% and precision value, recall, F1 Score for atopic dermatitis by 95%, 80%, 87% and precision value, recall, F1 Score for psoriasis by 83%, 96%, 89%. Therefore, the resnet-50 method in the scenario 1 is superior to using 2-5 scenario, cropping with the program code, and the mobilenetv2 method in classifying atopic dermatitis skin disease and psoriasis.

Keywords : Atopic Dermatitis, Psoriasis, ResNet-50

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dermatitis atopik atau eksim adalah salah satu penyakit kulit kronis yang ditandai dengan gatal yang hebat, ruam merah, dan kulit kering yang dapat menyebabkan kerusakan pada kulit, tekstur kulit menjadi bersisik, infeksi sekunder, dan gangguan tidur. Kondisi ini biasanya terjadi pada bayi/anak – anak, namun tidak menutup kemungkinan orang dewasa juga dapat terjangkit penyakit ini. Termasuk jenis penyakit yang dapat terjadi dalam jangka panjang (kronis), dermatitis atopik dapat terlihat membaik kemudian kembali muncul bahkan dapat menjadi lebih buruk dari sebelumnya sehingga diperlukannya pengetahuan akan penyakit ini (Pratiwi dan Kamardi, 2019). Hal penting yang dapat dilakukan oleh penderita penyakit ini adalah pencegahannya, karena seringkali terjadi penderita tidak mengetahui diri mereka telah mengidap dermatitis atopik. Adapun faktor penyebab dermatitis atopik, diantaranya faktor genetik, iritasi terhadap zat tertentu, penggunaan produk kosmetik/pembersih, obat – obatan yang dioleskan di kulit, keringat, stres, infeksi, alergi dan perubahan hormon. Pada umumnya dermatitis atopik sendiri merupakan penyakit yang belum diketahui pasti penyebab terjangkitnya. Bahkan, penyakit ini memiliki prevalensi yang meningkat secara global dan menjadi masalah kesehatan yang semakin mendesak.

Psoriasis adalah penyakit autoimun dan inflamasi yang ditandai oleh pertumbuhan sel-sel kulit yang melakukan regenerasi lebih cepat dari biasanya sehingga mengakibatkan kulit menebal, kulit mengalami bercak kemerahan, dan kulit menjadi bersisik putih sampai terkelupas (Dewi, 2021). Menurut *National Institute of Health*, penyakit ini tidak memandang usia karena jumlah penderita psoriasis diseluruh dunia mencapai lebih dari 125 juta pasien. Penyakit psoriasis memiliki 8 jenis area dan intensitas yaitu plak, kuku, kulit kepala, inversi, gutata, pustular, entrodermik, artiris dan

vulgaris. Selain menyerang kulit, psoriasis juga menyerang persendian yang mana hal tersebut akan menjadi sangat berbahaya karena dapat menyebabkan kehilangan fungsinya bahkan kematian. Adapun kemunculan penyakit ini adalah berlangsung dalam jangka waktu lama dan bersifat *residif* (hilang-timbul) sehingga sampai saat ini belum dapat disembuhkan ataupun dicegah, namun yang bisa dilakukan hanyalah menghilangkan gejalanya (Izzati dan Waluya, 2018).

Dua penyakit ini termasuk dalam penyakit kulit yang umum terjadi dan memiliki kesamaan gejala yang ditandai dengan lesi epidermis yang memerah atau meradang secara tidak normal dan mengalami penebalan yang bervariasi dalam intensitas dan area permukaan kulit yang terkena (Palareti *dkk.*, 2016). Akan tetapi diluar dari kesamaan tersebut, kedua penyakit merupakan jenis penyakit kulit yang berbeda dan memiliki banyak perbedaan diantaranya dari segi penyebab, lokasi gejala, tampilan, bentuk, mekanisme dan pengobatan. Banyak dari orang awam yang merasa kesulitan dalam membedakannya, hal itu tentu dikarenakan kurangnya pengetahuan tentang perbedaan diantara kedua penyakit sehingga dengan adanya kecerdasan buatan diharapkan mampu membantu masyarakat dalam memberikan hasil yang akurat dalam membedakannya.

Kecerdasan Buatan (AI) merupakan salah satu teknologi yang dalam beberapa tahun terakhir mengalami kemajuan yang signifikan dan terus berkembang dan berinovasi sehingga memiliki peran penting dalam masa depan teknologi informasi dan masyarakat secara umum. Teknologi AI memanfaatkan berbagai teknik dan algoritma, termasuk pembelajaran mesin (*Machine Learning*), jaringan saraf tiruan (*Neural Networks*), pemrosesan bahasa alami (*Natural Language Processing*), dan berbagai pendekatan lainnya untuk menciptakan sistem yang dapat berpikir dan bertindak secara cerdas layaknya manusia. Tidak dapat dipungkiri bahwa teknologi AI saat ini sering digunakan dalam bidang kesehatan seperti diagnosis penyakit, pengembangan obat, manajemen data kesehatan, pemantauan pasien, pengobatan personalisasi, asisten kesehatan virtual, prediksi wabah

penyakit, konsultasi dokter secara online, robotik kesehatan, pengelolaan data genomik, dan lain - lain.

Dengan memanfaatkan teknologi AI diharapkan dapat memberikan hasil yang akurat dalam membedakan tipikal kulit yang mengidap psoriasis dan eksim. Oleh karena itu untuk menangani permasalahan yang ada, penelitian ini bertujuan membahas sebuah teknologi kecerdasan buatan (AI) yang dapat membedakan secara akurat tipikal kulit yang teridentifikasi mengidap penyakit kulit dermatitis atopik atau psoriasis. Dengan menggunakan model arsitektur ResNet-50 yang ada pada algoritma *Convolutional Neural Network*, output dari penelitian ini diharapkan dapat menjadi alternatif untuk memberi akurasi yang tinggi dalam mengklasifikasi penyakit kulit yang mengalami dermatitis atopik dan psoriasis.

1.2 Perumusan Masalah

Bagaimana algoritma *Convolutional Neural Network* menggunakan model arsitektur ResNet-50 melakukan klasifikasi antara gambar kulit yang mengidap penyakit dermatitis atopik dan psoriasis.

1.3 Pembatasan Masalah

Adapun batasan masalah dari penulisan proposal sebagai berikut :

- 1) Gambar kulit yang diambil yaitu bagian kulit yang mengalami kemerahan atau gejala semisalnya yang berkaitan dengan dermatitis atopik dan psoriasis.
- 2) Penyakit kulit yang dibahas pada penelitian ini adalah dermatitis atopik dan psoriasis, tidak penyakit kulit lainnya.
- 3) Algoritma yang digunakan adalah *Convolutional Neural Network* dengan model arsitektur ResNet-50.
- 4) Selain model arsitektur ResNet-50, pada penelitian ini juga menggunakan model arsitektur MobileNetV2 sebagai pembanding.

- 5) Output dari sistem ini adalah menampilkan hasil prediksi dari suatu gambar kulit yang diklarifikasikan mengidap penyakit dermatitis atopik atau psoriasis.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah mengklasifikasikan gambar kulit yang termasuk dermatitis atopik dan psoriasis menggunakan model arsitektur ResNet-50 berdasarkan sampel kulit yang diambil.

1.5 Manfaat

Manfaat yang didapatkan dari penelitian ini adalah memberikan hasil akurasi yang tinggi dalam mengklasifikasikan gambar kulit dermatitis atopik dan psoriasis berdasarkan model arsitektur ResNet-50 yang digunakan.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan pada pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : Pada bab 1 mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

BAB II : Pada bab II memuat penelitian – penelitian sebelumnya dan dasar teori untuk membantu dalam memahami teori yang berhubungan dengan algoritma *Convolutional Neural Network* dengan model arsitektur ResNet-50 sehingga dapat menyelesaikan penelitian ini.

BAB III : Pada bab III mengungkapkan proses tahapan – tahapan penelitian yang dimulai dari mendapatkan data hingga proses klasifikasi data yang ada.

BAB IV : Pada bab IV mengungkapkan hasil klasifikasi dari penelitian yang menggunakan *Convolutional Neural Network* dengan model arsitektur RestNet-50 beserta pengujian klasifikasinya.

BAB V : Pada bab V memaparkan kesimpulan dari keseluruhan proses penelitian dari awal hingga akhir.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penyakit atopik dermatitis (eksim) dan psoriasis adalah dua penyakit inflamasi kronis pada kulit yang umum terjadi di negara tropis seperti Indonesia. Kedua penyakit ini terkadang sulit dibedakan oleh orang awam dikarenakan memiliki kesamaan ciri, penampilan, gejala dan kurangnya pengetahuan akan kedua penyakit. Presentasi terjadinya penyakit eksem di seluruh dunia yaitu sekitar 20% bagi bayi dan 3% bagi orang dewasa. Adapun penyakit psoriasis juga umum terjadi pada anak – anak dibandingkan orang dewasa (Padilla *dkk.*, 2019). Dengan adanya beberapa aspek kesamaan pada kedua penyakit, klasifikasi penyakit eksem dan psoriasis menjadi solusi bagi orang awam sehingga dapat membedakan tipikal kulit yang teridentifikasi penyakit kulit eksem atau psoriasis.

Klasifikasi kedua penyakit dilakukan dengan mengimplementasikan AI menggunakan metode *Convolutional Neural Network* (CNN). CNN adalah jenis jaringan saraf di bidang pembelajaran mesin yang sangat berguna saat berhadapan dengan data visual dan memiliki potensi penerapan yang tinggi untuk diaplikasikan dibidang medis dalam menggunakan data citra (Schielein *dkk.*, 2023). Pada tahapan arsitektur CNN, hidden layer mencakup lapisan yang melakukan proses konvolusi sehingga ketika kernel konvolusi menyelusuri matriks masukan di lapisan tersebut maka operasi konvolusi menghasilkan feature map yang berkontribusi di masukan lapisan berikutnya (Tukar dan Bhavana, 2021). Untuk mendapatkan suatu hasil maka dibutuhkannya proses training dan testing dataset dari data citra yang dikumpulkan. Selanjutnya menghitung nilai loss dan akurasi dari hasil training dan testing menggunakan adam optimizer (Umam dan Handoko, 2021). Dalam melatih suatu data citra, CNN memiliki kinerja yang lebih baik, namun kurang dalam hal data pengujian sehingga diperlukannya tambahan model arsitektur dalam menguji suatu data citra.

Adapun dari penelitian sebelumnya telah membuktikan bahwa kemampuan deteksi penyakit kulit dengan menambahkan model arsitektur didalamnya menghasilkan akurasi yang lebih baik dibandingkan hanya menggunakan arsitektur CNN yang menampilkan akurasi ditingkat rata – rata (Sari *dkk.*, 2023). Model arsitektur yang teruji menghasilkan akurasi lebih tinggi dari arsitektur lainnya pada penelitian yang berjudul *Detection and diagnosis of skin disease using residual neural network (ResNet)* adalah ResNet-50 dengan akurasi 95% (Sharma *dkk.*, 2021). Dengan adanya hasil akurasi yang tinggi maka dapat mengklasifikasikan penyakit kulit dengan akurat sehingga membantu peneliti masa depan dalam meningkatkan diagnosis penyakit kulit.

2.2 Dasar Teori

2.2.1 Dermatitis Atopik



Gambar 2. 1 Kulit Terjangkit Penyakit Dermatitis Atopik

Dermatitis Atopik atau eksim adalah salah satu penyakit kulit yang menyebabkan kulit menjadi kemerahan, gatal dan peradangan, biasanya terjadi pada kulit tangan, kaki, leher, wajah serta kulit kepala (Hutasoit *dkk.*, 2021). Peradangan yang terjadi pada penderita eksim dapat berlangsung dalam jangka waktu yang lama dan dapat terjadi berulang kali sehingga dibutuhkan pengetahuan dalam menanggulangnya. Penyakit ini biasanya

menjangkit bayi atau anak-anak dibawah 5 tahun, namun tidak menutup kemungkinan dapat terjadi pada remaja dan dewasa bahkan lansia. Kasus eksim juga bisa terjadi dikarenakan faktor keturunan/genetik dan lingkungan. Jika orang tersebut terkena eksim karena faktor genetik maka perlu penanganan secara serius dan rutin.

Selain faktor genetik, eksim juga bisa terjadi karena faktor alergi. Karena faktor alergi, seorang yang terkena eksim rentan terhadap iritan terhadap makanan, bahan kimia, sabun atau deterjen, gangguan emosi atau stress, pakaian yang bersifat iritatif, suhu ekstrem dan kelembapan. Adapun salah satu cara untuk mencegah kambuhnya eksim adalah menghindari faktor pencetusnya (Susanto, 2022). Pada sebagian besar orang, sensasi gatal yang dialami penderita dapat mengganggu kenyamanan orang tersebut sehingga rasa keinginan untuk menggaruknya sulit ditahan padahal hal tersebut dapat memperburuk kondisi kulit penderita. Oleh karena itu, penting bagi masyarakat untuk mengetahui kondisi kulit yang dialaminya.

2.2.2 Psoriasis



Gambar 2. 2 Kulit Terjangkit Penyakit Psoriasis

Psoriasis adalah salah satu penyakit kulit dengan peradangan kronis yang mengalami lesi kulit berupa plak kemerahan yang menebal berwarna putih keperakan serta berbatas tegas, kulit bersisik, mudah terkelupas dan terasa gatal. Kondisi ini biasa terjadi pada kulit bagian siku, lutut, kepala,

punggung, pusar, dan pinggang. Adapun rentang usia awal terjadinya psoriasis berdasarkan beberapa penelitian adalah sekitar 15-20 tahun, sedangkan dapat juga terjadi di usia kurang dari 10 tahun sekitar 10-15% kasus. Adapun untuk pengobatan psoriasis membutuhkan waktu yang relatif lama, hal itu dikarenakan pengobatan tidak dapat menyembuhkan tetapi dapat membantu dalam mengendalikan gejalanya sehingga membutuhkan waktu pendekatan secara individual (Rachman *dkk.*, 2022).

Penebalan pada kulit yang terjangkit psoriasis terjadi karena pergantian sel – sel kulit yang terlalu cepat sehingga terjadilah tumpukan sel kulit ekstra diatas kulit. Faktor berikut menjadi salah satu alasan yang menjadikan psoriasis dikatakan penyakit autoimun. Faktor lain yang menjadi penyebab psoriasis antara lain adalah faktor genetik, infeksi virus atau bakteri, stres, obesitas, merokok serta konsumsi alkohol. Selain itu, penyakit ini dapat menimbulkan komplikasi serius apabila tidak segera dikenali serta diobati gejalanya.

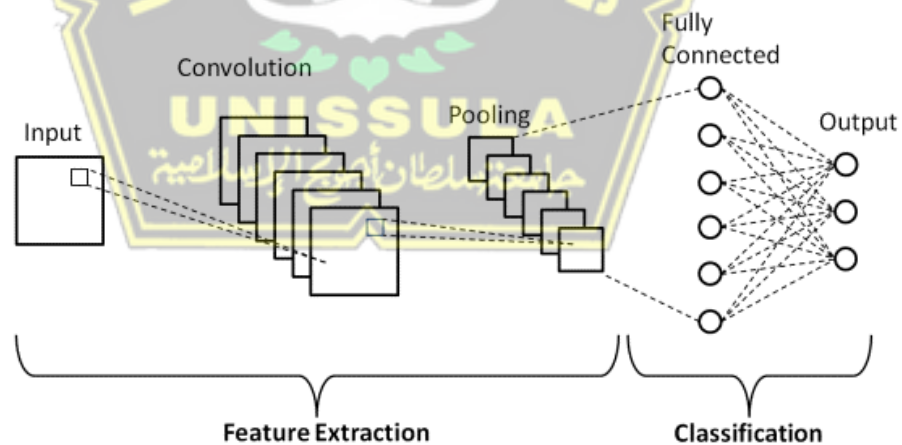
2.2.3 Deep Learning

Deep learning adalah metode dalam kecerdasan buatan (*Artificial Intelligence*) yang termasuk ke dalam sub-bidang *machine learning* dimana algoritma didalamnya terinspirasi dari struktur otak manusia yang strukturnya mengandalkan jaringan saraf buatan (*Artificial Neural Networks*) untuk memecahkan masalah (Arifin *dkk.*, 2021). Semakin populer dan menarik perhatian menjadikan teknologi *deep learning* diterapkan dalam berbagai produk atau project baik itu aplikasi besar maupun kecil. Jenis algoritma yang terdapat pada *deep learning* antara lain *convolutional neural network* (CNN), *recurrent neural network* (RNN), *long short term memory network* (LSTM), *self organizing maps* (SOM), dan lain sebagainya. Beberapa penerapan yang digunakan dalam produk sehari – hari yaitu meliputi pengenalan gambar, pengenalan suara, deteksi anomali, penglihatan komputer, pemrosesan bahasa alami (NLP) dan mesin rekomendasi.

Adapun cara kerja yang diaplikasikan oleh *deep learning* adalah menggunakan metode *neural network* yakni bekerja pada tingkatan atau layers dalam jumlah yang besar. Oleh karena itu, pada penelitian ini menggunakan jenis *neural network* CNN yang digunakan dalam mengekstraksi fitur secara langsung dari data gambar.

2.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu metode *deep learning* yang menerapkan algoritma jaringan saraf tiruan untuk mempelajari atau mengenali data dan mendeteksi fitur pada gambar yang berbeda (Intyanto, 2021). Cara kerja CNN yaitu memanfaatkan mekanisme pada konvolusi dengan mengaktifkan suatu filter yang memiliki ukuran tertentu ke dalam sebuah gambar sehingga komputer memperoleh informasi sampel baru dari hasil perkalian gambar tersebut. Berdasarkan cara kerja, CNN terdiri dari tiga lapisan yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer* (FC) yang mana setiap lapisan memiliki tugas masing-masing mulai dari mengidentifikasi hal yang terbilang mudah sampai kompleks.



Gambar 2. 3 Lapisan Pada CNN

Convolution layer adalah lapisan utama yang mendasari CNN untuk melakukan operasi konvolusi pada output layer sebelumnya. Tujuannya adalah untuk mengekstraksi fitur dari citra *input* sehingga dapat menspesifikasikan *kernel* konvolusi berdasarkan *input* pada CNN setelah

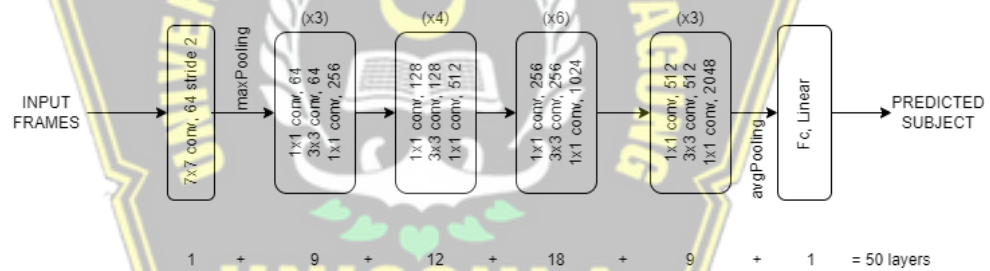
dilatih sebelumnya. Kernel adalah matriks bobot yang bergerak dari sudut kiri atas ke kanan bawah untuk menjalankan operasi konvolusi pada input gambar yang sedang dianalisis didalam jaringan CNN.

Lapisan selanjutnya adalah mereduksi ukuran/mengurangi dimensi sebuah data citra dari matriks hasil konvolusi yang merupakan tugas dari *pooling layer*. Adapun tujuan dari penggunaan *pooling layer* adalah untuk mempercepat proses komputasi, mengurangi overfitting dan mencegah model menjadi terlalu kompleks. *Pooling layer* memiliki dua model yang umum digunakan yaitu *max pooling* dan *average pooling*. Pada penelitian (Alwanda dkk., 2020) menerangkan bahwa *max pooling* berfungsi untuk mengembalikan nilai maksimum suatu gambar yang dijangkau oleh kernel, sedangkan *average layer* berfungsi untuk mengembalikan nilai rata – rata suatu gambar yang dijangkau oleh kernel.

Lapisan terakhir adalah *fully connected layer* (FC) yang merupakan sebuah neural network *multilayer perceptron* (MLP). Tujuannya adalah untuk melakukan transformasi pada dimensi data sehingga dapat mengklasifikasi data masukan secara linear. Masukan pada FC-layer adalah hasil dari kedua proses lapisan sebelumnya yang berupa *feature map* karena biasanya proses FC dilakukan setelah proses *convolution* dan *pooling layer*. Pada *fully connected layer* terdapat beberapa *hidden layer*, *activation function*, *output layer*, dan *loss function* yang berperan dalam mengklasifikasi data masukan. *Activation function* adalah fungsi yang memberikan non-linieritas ke jaringan untuk memungkinkan pembelajaran representasi yang lebih kompleks sehingga dapat melakukan pemotongan yang lebih baik dalam mengklasifikasi data dan memodelkan relasi non-linier dalam data tersebut, sedangkan *loss function* adalah fungsi yang mengukur sejauh mana prediksi model mendekati label sebenarnya.

2.2.5 ResNet-50

ResNet-50 adalah salah satu model arsitektur *residual network* (ResNet) yang biasa digunakan pada algoritma CNN dan termasuk ke dalam keluarga ResNet. Dikatakan ResNet-50 karena layer pada model arsitektur ini adalah sejumlah 50 lapisan yang terdiri dari berbagai blok penyusun yang disebut blok sisa dimana masing – masing dari blok tersebut berisi beberapa lapisan konvolusional. Penggunaan blok – blok residual pada ResNet-50 memungkinkan informasi melewati lapisan tanpa mengalami transformasi yang membantu mengatasi masalah gradien yang melemah sehingga dapat mempermudah aliran informasi. ResNet-50 dapat mencapai keseimbangan antara kompleksitas dan performa karena menggunakan blok – blok yang dikonfigurasi dengan baik sehingga lebih mudah untuk mengembangkan model deep learning yang dalam tanpa menghadapi tantangan pelatihan yang sulit.



Gambar 2. 4 Model Arsitektur ResNet-50

Secara keseluruhan, ResNet-50 terdiri dari 4 tahap proses konvolusi yang melibatkan blok – blok konvolusi lalu dilanjutkan lapisan *average pooling* serta diakhiri *fully connected layer* dimana layer ini sebagai lapisan prediksi. ResNet-50 memperkenalkan sebuah konsep baru yaitu *shortcut connections* sebagai elemen kunci dalam arsitektur ResNet-50 yang memiliki keterkaitan dengan *vanishing gradient problem* sehingga dapat meminimalkan hilangnya fitur – fitur penting ketika proses konvolusi (Nashrullah dkk., 2020). Adapun tujuan utama *shortcut connection* adalah untuk memastikan aliran gradien yang efisien selama pelatihan model. Karena ketika model menjadi lebih dalam maka gradien akan melemah

sehingga penambahan lapisan baru dapat menyebabkan penurunan kinerja model.

Shortcut Connection sendiri dirancang untuk mempresentasikan identitas (*identity mapping*) dari input. Jikalau blok tersebut menghasilkan representasi fitur yang diinginkan maka *shortcut connection* akan menyertakan identitas dari input aslinya. Sedangkan jika blok melakukan transformasi maka identitas tersebut dapat diubah sesuai kebutuhan. Oleh karena itu, manfaat dari *shortcut connection* adalah menghasilkan model yang dapat mempelajari perubahan tambahan dari identitas dan fokus pada pembelajaran representasi fitur yang sulit untuk dipelajari sehingga mengurangi risiko degradasi performa model akibat penambahan lapisan.

Pada lapisan awal (Conv1), input gambar dengan ukuran $224 \times 224 \times 3$ yang mana panjang dan lebarnya adalah 224 piksel dengan kedalaman adalah 3 yang mewakili tiga kanal warna RGB (*Red, Green, Blue*), membentuk ekstraksi fitur dengan kernel 7×7 (stride = 2) yang memiliki 64 filter sehingga lapisan ini diikuti normalisasi batch dan aktivasi ReLU. Tahapan selanjutnya yaitu melibatkan 4 blok utama atau disebut dengan ResNet Bloks yang mana blok 1 terdiri dari beberapa sub-blok yang berulang sehingga memiliki 3 sub-blok. Dilanjutkan pada blok 2 yang memiliki 4 sub-blok. Lalu blok 3 memiliki 6 sub-blok dan blok 4 memiliki 6 sub-blok yang mana blok ini membentuk bagian akhir dari struktur hierarki ResNet-50. Adapun setiap blok tersebut menggunakan *shortcut connection* didalamnya.

Lapisan selanjutnya yaitu melalui tahapan GPA atau Grade Point Average untuk merata – ratakan representasi fitur pada setiap saluran diseluruh gambar yang menghasilkan vektor fitur yang bersifat global. Adapun lapisan terakhir dari arsitektur ResNet-50 adalah melalui tahapan *fully connected* yang diaktivasi dengan fungsi *linear* untuk menghasilkan distribusi probabilitas kelas sehingga model dapat memprediksi subjek dari input gambar sebelumnya. Fungsi *linear* juga dikenal sebagai lapisan dense yang mana berfungsi untuk melakukan transformasi *linear* sederhana pada

data input dengan menggunakan bobot dan bias tanpa memperkenalkan *non-linearitas*. Blok – blok yang ada pada ResNet-50 menggunakan arsitektur “BottleNeck” sehingga terdiri dari lapisan – lapisan konvolusi dengan ukuran 1×1 , 3×3 , dan 1×1 . Lapisan konvolusi ukuran 1×1 diawal digunakan untuk mengurangi dimensi, lapisan konvolusi 3×3 digunakan untuk mengekstrak fitur – fitur yang lebih kompleks, dan lapisan konvolusi 1×1 diakhir digunakan untuk memulihkan dimensi ruang sebelumnya.

2.2.6 PyTorch

PyTorch adalah sebuah library pada bahasa pemrograman yang biasa digunakan dalam *machine learning*, *deep learning* bahkan *natural language processing* (NLP) yang dikembangkan oleh tim *Facebook AI research* (sekarang berganti menjadi *Meta AI*) pada tahun 2016 (Qodryantha dkk., 2023). Dengan *library PyTorch*, pengguna dapat menjalankan dan menguji bagian dari kode secara *real-time* sesuai dengan sifatnya yang *open-source*. Walaupun dikembangkan menggunakan *c++*, namun *PyTorch* memiliki *frontend* API berbasis *python* yang sangat mempermudah pengguna dalam penggunaan *library* tersebut. *PyTorch* juga memiliki beberapa fitur yang memberikan banyak keuntungan, diantaranya grafik komputasi yang dinamis, dukungan *backend* yang berbeda, gaya yang imperatif, memiliki API yang intuitif dan mudah digunakan serta memiliki dukungan komunitas yang besar.

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

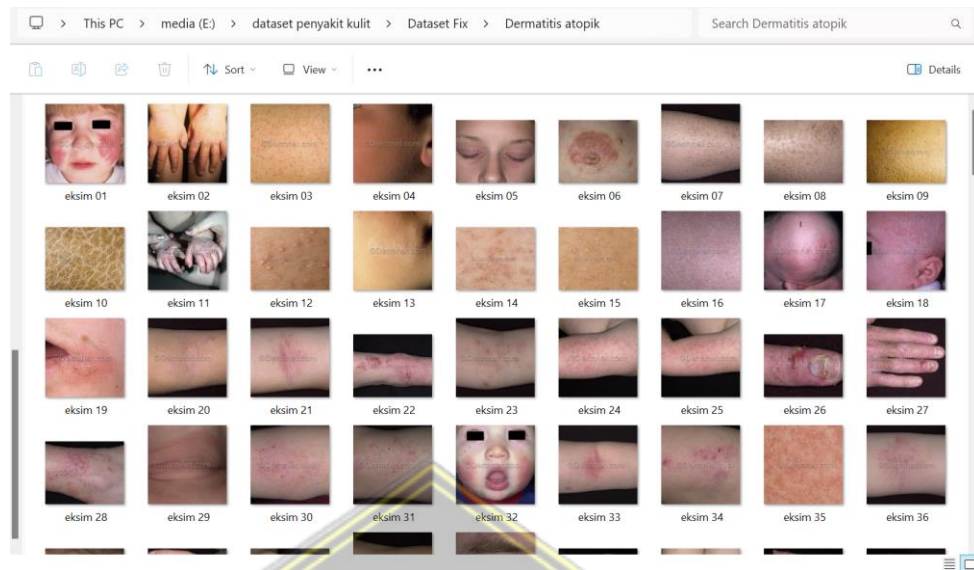
Penelitian ini menggunakan metode atau algoritma *Convolutional Neural Network* (CNN) dengan menambahkan model arsitektur ResNet-50 dalam prosesnya. Penambahan model arsitektur digunakan untuk membantu model dalam menghasilkan akurasi dan evaluasi model yang baik. Untuk dapat melakukan klasifikasi terhadap gambar kulit dermatitis atopik dan psoriasis, penelitian ini menggunakan metode dengan melakukan beberapa tahapan didalamnya. Berikut langkah – langkah yang dilakukan dalam penelitian ini :

3.1.1 Studi Literature

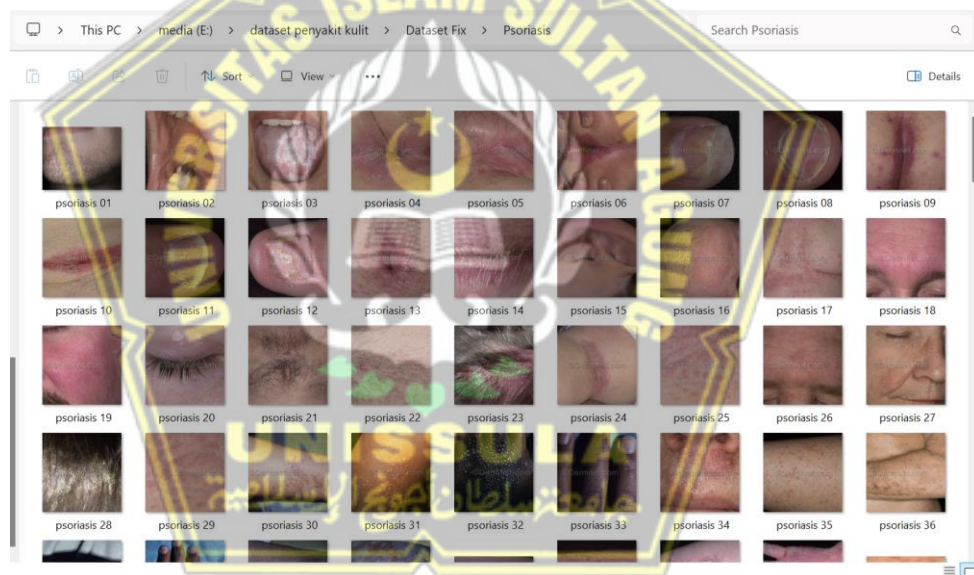
Teori yang dipelajari dalam melakukan penelitian ini adalah mengenai *image classification, deep learning, convolutional neural network*, dan ResNet-50 yang bersumber dari berbagai macam artikel, jurnal, dan situs *website* yang tersedia di google. Tujuan melakukan tinjauan adalah untuk mempelajari model arsitektur terbaik yang dapat menghasilkan hasil akurasi dan evaluasi model yang baik pada gambar kulit yang dilatih.

3.1.2 Pengumpulan Data (*Data Collecting*)

Data collecting atau biasa disebut dataset merupakan tahapan awal pada perancangan model dalam mengumpulkan data citra dari penyakit kulit dermatitis dan psoriasis. Dataset yang digunakan dalam penelitian ini diambil dari salah satu platform online yaitu *Kaggle* yang kemudian diseleksi lalu disimpan kedalam *google drive* sebagai berikut : <https://drive.google.com/drive/folders/1Xj17iKORrYrzjDJ370UFibWuG9t-xmAt?usp=sharing>. Adapun jumlah data citra pada dataset sebanyak 250 untuk penyakit dermatitis atopik dan 250 untuk penyakit psoriasis sehingga total data citra sebanyak 500 gambar. Pada gambar 3.1 – 3.2 merupakan folder gambar kulit dari kedua penyakit.



Gambar 3. 1 Data Gambar Dermatitis Atopik



Gambar 3. 2 Data Gambar Psoriasis

3.1.3 Pemangkasan Data (*Cropping*)

Pemangkasan atau *Cropping* data dilakukan untuk menyamaratakan ukuran gambar kulit agar tepat terpangkas di area kulit yang mengalami dermatitis atopik atau psoriasis. Ada banyak cara yang dapat dilakukan untuk memangkas data gambar yaitu contohnya dengan cara manual memangkas satu persatu dan menggunakan codingan. Pada penelitian ini cara yang digunakan yaitu dengan kedua cara tersebut, adapun yang pertama melakukan pemangkasan manual agar ketika proses konvolusi jenis filter

akan di-*training* secara mandiri oleh model untuk menemukan filter terbaik bagi model tersebut sehingga gejala yang timbul tidak terpangkas ketika dilakukan dengan cara pemangkasan menggunakan codingan. Ukuran gambar dalam pemangkasan yaitu dengan perbandingan 1 : 1 dan 5 : 4. Kedua ukuran tersebut dilakukan agar jika gejala yang timbul tepat di tengah gambar dan tidak meluas maka ukuran yang digunakan yaitu 1 : 1. Sedangkan jika sebaliknya maka ukuran yang digunakan yaitu 5 : 4. Hal ini dilakukan untuk memfokuskan area yang diinginkan. Selain itu, tujuan dilakukan *cropping* adalah untuk mencegah tidak sempurnanya tampilan gambar ketika di-*resize* sebesar 224×224 pada tahap *preprocessing*.

Cara kedua yaitu menggunakan teknik *crop* dengan kode program pada proses *preprocessing* data. Hal ini dilakukan untuk melihat perbandingan antara kedua teknik yang akan menghasilkan akurasi yang baik dan tidak menyebabkan adanya *overfitting* dan *underfitting*. Adapun teknik ini dilakukan dengan melakukan *center crop*(400) karena dataset memiliki ukuran gambar >400 sehingga nantinya hasil *crop* memuat ukuran gambar 1 : 1 untuk menghasilkan *resize* yang baik.

3.1.4 Pra-Pemrosesan Data (*Preprocessing*)

Tahapan *preprocessing* merupakan langkah utama pada proses pemodelan sebelum data gambar dianalisis untuk melihat karakteristik data tersebut sehingga komputer dapat membaca sebuah gambar menjadi *array* yang bermakna serta menghasilkan gambar dengan kualitas yang lebih baik untuk diproses lebih lanjut. Banyak tahapan *preprocessing* yang dapat dilakukan untuk meningkatkan kualitas gambar sehingga mudah dikenali untuk dilakukannya proses klasifikasi kedua penyakit. Dalam penelitian ini tahapan *preprocessing* yang dilakukan diantaranya *split* data, *resize*, augmentasi, dan *color jitter*. *Split* data dilakukan untuk membagi data menjadi tiga set yaitu data *train*, data *test*, dan data *valid* dengan ratio model 80% : 10% : 10% sehingga jumlah pembagian dapat dilihat pada tabel 3.1.

Tabel 3. 1 Pembagian Data *Train*, *Test* dan *Valid*

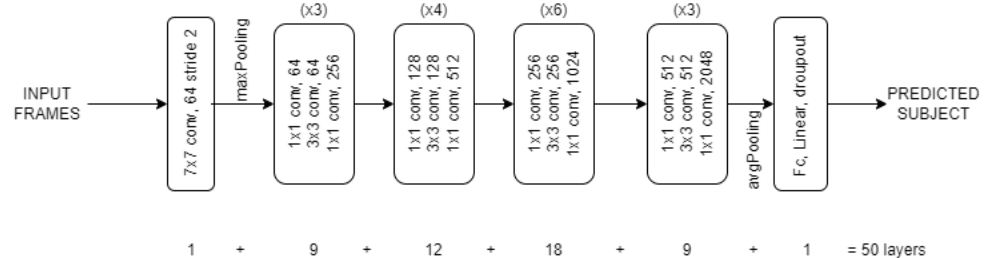
Nama Data	Jumlah (%)	Jumlah Angka	Dermatitis Atopik	Psoriasis
Data <i>Train</i>	80%	400	200	200
Data <i>Test</i>	10%	50	25	25
Data <i>Valid</i>	10%	50	25	25
Total	100%	500	250	250

Adapun proses *resizing* dilakukan agar terciptanya efisiensi dan efektifitas pada sistem input CNN sehingga menyamakan ukuran gambar serta mengetahui perbedaan tingkat akurasi pada gambar input yang berbeda. Ukuran *resize* untuk ResNet-50 yaitu $224 \times 224 \times 3$. Selanjutnya pada proses augmentasi, gambar akan dilakukan *horizontal flip* dan *vertical flip*. *Horizontal flip* dilakukan untuk membalikkan gambar secara horizontal dan *vertical flip* dilakukan untuk membalikkan gambar secara vertikal. Hal ini dilakukan untuk memodifikasi data gambar sehingga dapat terhindar dari adanya masalah *overfitting*. Selain itu juga melakukan proses rotasi sebesar $(-45, 45)$ untuk meningkatkan keberagaman pola yang dapat dikenali oleh model. Dan proses *color jitter* dilakukan agar model dapat mengenali gambar dalam kondisi gelap.

3.1.5 Perancangan Model

ResNet-50 merupakan model dengan konsep *shortcut connections* dan menjadikan input *layer* sebelumnya sebagai input terhadap output *layer* sebelumnya. Hal ini dilakukan untuk meminimalkan hilangnya fitur – fitur penting ketika menjalankan proses konvolusi. Pada tahap ini yaitu membuat perancangan model ResNet-50 yang digunakan pada penelitian yang sedang dilakukan. Tahapan pada ResNet-50 terdiri dari lima *stage* proses konvolusi yang dilanjutkan dengan proses *average pooling* serta diakhiri dengan proses *fully connected layer* yang tersajikan seperti pada gambar 2.4 diatas. Namun pada penelitian ini melakukan modifikasi dengan menambahkan *dropout* pada *layer fully connected* yang bertujuan untuk meminimalisasi terjadinya *overfitting*, dimana model menjadi terlalu spesifik terhadap data

pelatihan serta tidak dapat menggeneralisasi dengan baik ke data yang belum pernah dilihat sebelumnya. Gambar 3.3 menunjukkan model arsitektur ResNet-50 yang telah dimodifikasi.



Gambar 3. 3 Model Arsitektur ResNet-50 yang Dimodifikasi

Pada gambar 3.3 menunjukkan bahwa lapisan awal konvolusi dengan ukuran $7 \times 7 \times 64$ filter akan dimasukkan ke dalam blok utama ResNet-50 melalui lapisan max pooling terlebih dahulu. Lalu akan melewati lapisan konvolusi 1×1 , 3×3 , dan 1×1 dalam blok utama tersebut. Lapisan konvolusi 1×1 bertujuan untuk mengurangi dimensi kedalaman (jumlah saluran) dari fitur yang diperoleh dari lapisan 7×7 . Hasil lapisan konvolusi 1×1 dimasukkan ke dalam lapisan konvolusi 3×3 untuk mengekstraksi fitur-fitur yang lebih kompleks dan abstrak dari fitur yang sudah diurutkan sebelumnya sehingga jaringan dapat memperhatikan pola-pola yang lebih besar dan lebih kompleks dalam data gambar. Selanjutnya hasil dari lapisan tersebut akan dimasukkan ke dalam lapisan konvolusi 1×1 kedua untuk mengembalikan dimensi kedalaman fitur kembali ke dimensi yang diharapkan. Dengan demikian, melalui serangkaian lapisan konvolusi 1×1 , 3×3 , dan 1×1 pada blok utama ResNet-50 yang berjumlah 4 sub-blok dapat menghasilkan representasi fitur yang lebih dalam dan semakin abstrak dari fitur input aslinya.

Setelah melewati lapisan konvolusi tersebut maka kedalaman dimensi yang dihasilkan yaitu 2048 lalu akan dimasukkan ke dalam lapisan *global average pooling* untuk meratakan hasil dari setiap lapisan gambar menjadi satu vektor rata – rata. Vektor fitur tersebut digabungkan untuk klasifikasi akhir pada *layer fully connected*. Transformasi *linear pada layer fully connected* dilakukan untuk menghasilkan output yang dapat

diinterpretasikan sebagai probabilitas kelas. Lalu *dropout* dilakukan dengan mengabaikan secara acak sebagian dari neuron pada lapisan sebelumnya selama proses pelatihan. Nilai dropout yang dilakukan pada penelitian ini tergantung dari perbandingan model yang dapat dilihat pada tabel 3.2. *Dropout* merupakan modifikasi yang dilakukan pada penelitian ini sebagai teknik regularisasi yang digunakan selama proses pelatihan untuk mencegah *overfitting*.

3.1.6 Pelatihan Model

Pada tahap ini menggunakan model arsitektur ResNet-50 yang dapat digunakan untuk mengklasifikasi gambar penyakit. Ketika proses training dijalankan maka perlu mengatur beberapa konfigurasi parameter untuk mengklasifikasikan data citra yaitu dengan menentukan nilai *batch size*, *epoch*, *optimizer* dan *learning rate*. Nilai *batch size* yang digunakan pada penelitian ini untuk melatih model yaitu senilai 64 dengan *epoch* sebanyak 10 kali, tipe optimizer SGD dan learning rate sebesar 0,001. Selain itu, model yang dilatih juga dilakukan modifikasi dengan menambahkan *dropout* untuk meminimalisasi *overfitting* dan *underfitting* pada model. Adapun tipe GPU yang digunakan ketika mentraining data adalah T4 GPU yang merupakan Tesla T4 dengan memori GDDR6 16 GB dan 2.560 inti CUDA untuk menghadirkan peningkatan kinerja sehingga cocok untuk aplikasi pembelajaran mendalam. Untuk mendapatkan hasil akurasi yang tinggi dan minim *overfitting* dan *underfitting* maka dilakukannya beberapa perbandingan dengan mengganti parameter dan mencoba melatih model dengan model arsitektur lain yaitu MobileNetV2. Selanjutnya dataset yang telah di training menghasilkan sebuah model yang akan disimpan dengan format “.pt”.

3.1.7 Perbandingan Model

Demi mendapatkan hasil akurasi dan *confusion matrix* yang tinggi ada beberapa pengujian terhadap model yang dilakukan diantaranya, melakukan perubahan parameter, melakukan *cropping* menggunakan kode program dan melakukan perbandingan dengan model arsitektur MobileNetV2.

Keseluruhan model menggunakan perbandingan 80 : 10 : 10 untuk data *train*, *test* dan *valid* dengan jumlah 400 data *train*, 50 data *test* dan 50 data *valid* sehingga total 500 data gambar. Perbedaan parameter yang dilakukan yaitu pada proses *preprocessing* data dan modifikasi *dropout* pada model arsitektur ResNet-50. Adapun untuk *optimizer* keseluruhan model menggunakan SGD (*Stochastic Gradient Descent*) dengan *learning rate* = 0,001 dan *epoch* sebanyak 10 kali. Tabel 3.2 menunjukkan perbandingan model pada penelitian yang telah dilakukan.

Tabel 3. 2 Konfigurasi Model

No.	Model	Teknik Crop	Preprocessing Data	
1	ResNet-50	Manual	- <i>Rezise</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i> - <i>Color Jitter</i>	0,2
2	ResNet-50	Manual	- <i>Rezise</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i> - <i>Color Jitter</i> - <i>Rotation</i> (-45,45)	0,2
3	ResNet-50	Manual	- <i>Rezise</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i>	0,3
4	ResNet-50	Manual	- <i>Rezise</i> (224,224) - <i>Horizontal flip</i> - <i>Color Jitter</i> - <i>Rotation</i> (-45,45)	0,3
5	ResNet-50	Manual	- <i>Rezise</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i> - <i>Rotation</i> (-45,45)	<i>None</i>
6	ResNet-50	Kode	- <i>Center Crop</i> (400)	0,2

No.	Model	Teknik Crop	Preprocessing Data	
		Program	- <i>Resize</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i> - <i>Color Jitter</i>	
7	MobileNetV2	Manual	- <i>Resize</i> (224,224) - <i>Horizontal flip</i> - <i>Vertical flip</i> - <i>Normalize</i> - <i>Rotation</i> (-45,45)	0,2

3.1.8 Evaluasi Model

Evaluasi model dilakukan untuk melihat dan menemukan kombinasi model yang terbaik untuk mengklasifikasi kedua penyakit dengan hasil yang akurat. Untuk itu, hasil akurasi dan *Confusion matrix* digunakan dalam membandingkan model yang dilatih sebelumnya. Akurasi merupakan rasio prediksi yang benar terhadap keseluruhan data gambar yang dilatih. *Confusion matrix* merupakan alat pengukur performa model yang sering digunakan pada *machine learning* dalam permasalahan klasifikasi biner maupun *multiclass*. Kolom *Confusion matrix* menunjukkan hasil *class* prediksi, sedangkan baris *Confusion matrix* menunjukkan hasil *class* sebenarnya sehingga dapat digunakan dalam menghitung seluruh kemungkinan pada masalah klasifikasi. Ada tiga jenis *Confusion matrix* yaitu *precision*, *recall*, dan *f1 score*. Akurasi dan *Confusion matrix* dapat dilihat seperti pada persamaan rumus (1), (2), (3), dan (4).

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

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

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

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

A. Accuracy

Accuracy adalah tingkat kedekatan nilai prediksi dengan nilai aktual ketika model dilatih.

B. Confusion Matrix

Confusion matrix adalah tabel yang menyimpulkan seberapa sukses model dalam memprediksi data uji. Ada beberapa istilah yang mempresentasikan hasil dari model yang dilatih dalam *confusion matrix*:

1. *True Positive* (TP), hasil prediksi positif sesuai dengan nilai sebenarnya yaitu positif.
2. *False Positive* (FP), hasil prediksi positif tidak sesuai dengan nilai sebenarnya yaitu negatif.
3. *True Negative* (TN), hasil prediksi negatif sesuai dengan nilai sebenarnya yaitu negatif.
4. *False Negative* (FN), hasil prediksi negatif tidak sesuai dengan nilai sebenarnya yaitu positif.

Sedangkan tiga jenis *Confusion matrix* yang digunakan dalam menghitung prediksi antara lain :

1. *Precision* adalah rasio prediksi benar dengan jumlah hasil prediksi positif dan negatif.
2. *Recall* adalah rasio prediksi benar dengan jumlah nilai aktual positif atau negatif.
3. *F1 Score* adalah perbandingan rata – rata *precision* dan *recall* yang dibobotkan.

3.1.9 Deployment Model

Software yang digunakan untuk melakukan proses *deployment* pada penelitian ini adalah streamlit. Streamlit adalah sebuah python framework untuk membangun aplikasi web di bidang data science. Streamlit menyediakan berbagai fitur seperti *checkbox*, *text input*, *selectbox*, dan banyak lagi fitur yang dapat digunakan untuk mengontrol aplikasi web yang akan dibuat. Pada pemilihan bentuk *deployment* menggunakan *Local Deployment* dengan melakukan *install* streamlit : `!pip install -q`

streamlit' lalu menjalankan aplikasi : `'streamlit run <nama_file.py>'`. Penggunaan streamlit pada penelitian ini dikarenakan streamlit sendiri memiliki banyak manfaat diantaranya sederhana, mudah dipelajari, *rapid prototyping*, interaktif, *responsive*, integrasi dengan pandas, *deploy* dengan mudah, *open source*, dan mendukung untuk *Machine Learning* (ML). Tidak hanya itu, streamlit juga memudahkan dalam pengembangan yaitu deklaratif, terdapat *widget* dan elemen UI bawaan, *update* otomatis, peningkatan dengan *library* eksternal, dokumentasi yang baik, fleksibilitas dan kustomisasi, serta siklus pengembangan yang cepat. Oleh karena itu, pada penelitian ini menggunakan *local deployment* yang mampu memberikan kemudahan dalam pengembangan dan pengujian.

3.2 Software Yang Digunakan

Dalam pengembangan aplikasi, penelitian ini membutuhkan beberapa *software* yang digunakan selama proses *input* hingga *output* sehingga dapat berjalan sesuai dengan yang diharapkan. Berikut daftar *software* yang digunakan :

A. Python 3.11.3

Penggunaan python versi 3.11.3 sebagai bahasa pemrograman adalah karena memiliki *library* yang lengkap dan bersifat *open source*. Python merupakan bahasa pemrograman tingkat tinggi yang bersifat fleksibel dengan focus pada keterbacaan dan kesederhanaan sehingga mudah dalam penggunaannya.

B. Library Torch

Library Torch merupakan bagian dari PyTorch yang mana menjadi pustaka paling digemari oleh peneliti dan praktisi *Artificial Intelligence* (AI) di bidang industri dan akademisi khususnya dalam pengembangan model neural networks. PyTorch adalah pustaka tensor pada *deep learning* yang dioptimalkan berdasarkan Python dan Torch sehingga lebih digemari daripada *framework deep learning* lainnya

seperti Tensorflow dan Keras karena PyTorch menggunakan grafik komputasi dinamis dan *full Pythonic*.

C. *Library Torchvision*

Library Torchvision adalah bagian dari pustaka PyTorch yang menyediakan utilitas untuk beroperasi dengan *computer vision* seperti datasets, transformasi gambar, dan lainnya.

D. *Library Numpy*

Library Numpy adalah pustaka Python yang sangat populer untuk komputasi numerik yang mana menyediakan objek *array* yang efisien terhadap fungsi matematika dalam memanipulasi *array*.

E. *Library Pandas*

Library Pandas berfungsi dalam menyediakan struktur data dan fungsi manipulasi data yang nyaman khususnya DataFrame yang memungkinkan analisis data tabular.

F. *Library Matplotlib.pyplot*

Matplotlib merupakan pustaka yang bekerja untuk visualisasi data pada Python. Adapun '*matplotlib.pyplot*' adalah modul khusus dalam Matplotlib yang menyediakan fungsi untuk membuat suatu plot grafik.

G. *Library CV2*

OpenCV (*Open Source Computer Vision Library*) merupakan pustaka yang menyediakan alat untuk pengolahan gambar dan *computer vision*. Adapun '*cv2*' adalah modul pada OpenCV yang umum digunakan dalam pengembangan *software*.

H. *Library OS*

Operating System (OS) atau modul `os` merupakan pustaka yang menyediakan antarmuka untuk berinteraksi dengan sistem operasi seperti operasi pada sistem file, manipulasi direktori dan lainnya.

I. *Library Pickle*

Library Pickle merupakan pustaka pada Python yang digunakan untuk serialisasi dan deserialisasi objek Python. Pada penelitian ini

menggunakan deserialisasi dalam menyimpan dan mengambil objek Python. Deserialisasi adalah proses mengambil representasi *string* yang sudah disimpan atau dikirim dan mengembalikannya menjadi objek Python.

J. *Library Scikit-Learn*

Library scikit-learn atau *sklearn* merupakan pustaka yang digunakan untuk *machine learning* yang ditulis dalam bahasa pemrograman Python dan sangat populer. Dalam penelitian ini, *library scikit-learn* digunakan sebagai evaluasi performa model dengan cara mengukur model menggunakan *confusion matrix*, *accuracy*, *precision*, *recall*, dan *F1 score* dari model yang telah dilatih.

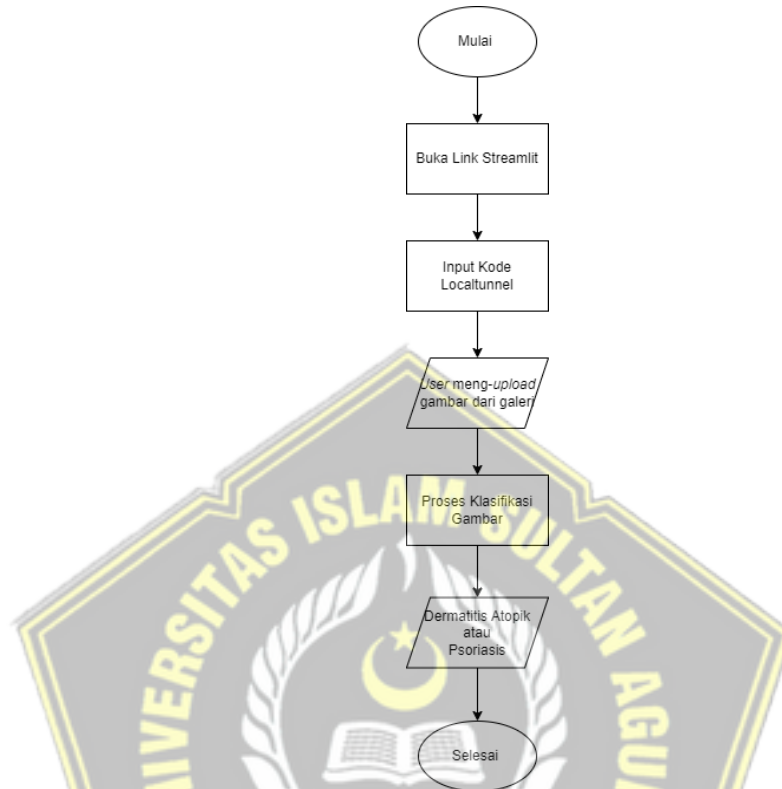
K. *Framework Streamlit*

Streamlit merupakan *framework* pada Python yang digunakan untuk membangun aplikasi web dibidang *data science* dan *machine learning* bersifat *open source*. Dalam penelitian ini, *framework streamlit* digunakan untuk men-*deploy* aplikasi yang telah dibuat menggunakan bahasa pemrograman Python.

3.3 Perancangan Sistem

Tahap ini merupakan rancangan untuk menetapkan alur kerja pengembangan *streamlit* yang akan dibuat dalam *flowchart*. Tahap pertama dimulai dengan *user* mengakses *link* *streamlit* lalu memasukkan *Endpoint IP*. *Endpoint IP* adalah kode publik IP dari kreator *localtunnel* untuk dapat mengakses aplikasi web yang telah dibuat. Setelah memasukkan kode, *user* dapat mengklik *submit* maka tampilan *dashboard* muncul di antarmuka *user*. Pada *dashboard* terdapat *fitur* untuk meng-*upload* gambar dari penyakit dermatitis atopik atau psoriasis yang ingin diprediksi sehingga nantinya akan menampilkan hasil prediksi apakah gambar tersebut terklasifikasi ke dalam *class* dermatitis atopik atau *class* psoriasis. Jika hasil prediksi telah muncul maka tulisan prediksi berwarna hijau pada sisi kiri dan sisi kanan akan menampilkan label prediksi, penjelasan tentang

penyakit dan faktor penyebab dari gambar tersebut. *Flowchart* dari streamlit dapat dilihat pada gambar 3.3.



Gambar 3. 4 *Flowchart* Streamlit Klasifikasi Penyakit Dermatitis Atopik dan Psoriasis

Gambar 3.3 merupakan *flowchart* streamlit yang menerangkan alur kerja aplikasi web dalam mengklasifikasikan gambar kulit yang teridentifikasi penyakit dermatitis atopik dan psoriasis. Adapun tahapan alur kerja streamlit tersebut dijelaskan sebagai berikut :

- a. Pertama *user* mengakses *link* streamlit dari kreator aplikasi web.
- b. Lalu *user* memasukkan kode localtunnel untuk dapat menampilkan *dashboard* dari aplikasi web tersebut.
- c. Pada fitur *upload* gambar, *user* memasukkan gambar kulit yang ingin diklasifikasikan.
- d. Setelah gambar kulit terproses maka hasil prediksi akan mengklasifikasikan apakah gambar tersebut terkategori *class* dermatitis atopik atau psoriasis yang mana tulisan prediksi berwarna hijau pada sisi kiri dan sisi kanan akan menampilkan label prediksi, penjelasan penyakit dan faktor penyebabnya.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil dan Analisis

Berdasarkan tabel 3.2 di bab III dilakukannya perbandingan model dengan hasil sebagai berikut :

4.1.1 Pengujian Parameter yang Berbeda

Skenario 1 pada gambar 4.1 menggunakan *resize* (224, 224), *horizontal flip*, *vertical flip* dan *color jitter* untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Selain itu, modifikasi model yang dilakukan yaitu menambahkan *dropout* dengan $p = 0,2$.

```
# Performing the Image Transformation and Data Augmentation on the
transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.ToTensor(),
    transforms.ColorJitter(brightness=0, contrast=0.2, saturation=0, hue=0),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),
])
```

Gambar 4. 1 Pengujian Parameter 1

Skenario 2 pada gambar 4.2 menggunakan *resize* (224, 224), *horizontal flip*, *vertical flip*, rotasi(-45, 45), dan *color jitter* untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Modifikasi yang dilakukan pada proses ini adalah menambahkan *droupout* dengan $p = 0,2$.

```

# Performing the Image Transformation and Data Augmentation on the

transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.RandomRotation(degrees=(-45, 45)),
    transforms.ToTensor(),
    transforms.ColorJitter(brightness=0, contrast=0.2, saturation=0, hue=0),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),

])

```

Gambar 4. 2 Pengujian Parameter 2

Skenario 3 pada gambar 4.3 menggunakan *resize* (224, 224), *horizontal flip* dan *vertical flip* untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Modifikasi yang dilakukan pada proses ini adalah menambahkan *droupout* dengan $p = 0,3$.

```

# Performing the Image Transformation and Data Augmentation on the

transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.ToTensor(),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),

])

```

Gambar 4. 3 Pengujian Parameter 3

Skenario 4 pada gambar 4.4 menggunakan *resize* (224, 224), *horizontal flip*, rotasi (-45, 45) dan *color jitter* untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Modifikasi yang dilakukan pada proses ini adalah menambahkan *droupout* dengan $p = 0,3$.


```

# Performing the Image Transformation and Data Augmentation on the

transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(degrees=(-45, 45)),
    transforms.ToTensor(),
    transforms.ColorJitter(brightness=0, contrast=0.2, saturation=0, hue=0),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),

])

```

Gambar 4. 4 Pengujian Parameter 4

Skenario 5 pada gambar 4.5 menggunakan *resize* (224, 224), *horizontal flip*, *vertical flip*, dan rotasi (-45, 45) untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Adapun pengujian ini tidak menambahkan *droupout* didalamnya.

```

# Performing the Image Transformation and Data Augmentation on the

transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.RandomRotation(degrees=(-45, 45)),
    transforms.ToTensor(),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),

])

```

Gambar 4. 5 Pengujian Parameter 5

4.1.2 Pengujian *Cropping* Dengan Kode Program

Tidak hanya menguji model dengan *cropping* manual, pada penelitian ini juga melakukan *cropping* dengan kode program ketika tahap *preprocessing* data. Adapun pengujian yang dilakukan yaitu menggunakan *center crop* (400), hal itu dikarenakan ukuran gambar berada di angka 400 ke atas sehingga keseluruhan gambar dapat di *crop* oleh program tanpa terkecuali dan memuat ukuran gambar 1 : 1. Dengan demikian proses

preprocessing yang dilakukan selain *center crop* (400) pada pengujian ini adalah menggunakan *resize* (224, 224), *horizontal flip*, *vertical flip*, dan *color jitter* untuk data *train* dan *resize* (224, 224) untuk data *test* pada *preprocessing* data. Selain itu, modifikasi model yang dilakukan yaitu menambahkan *dropout* dengan $p = 0,2$.

```
# Performing the Image Transformation and Data Augmentation on the
transform_train = transforms.Compose([
    transforms.CenterCrop(400),
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.ToTensor(),
    transforms.ColorJitter(brightness=0, contrast=0.2, saturation=0, hue=0),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),
])
```

Gambar 4. 6 Pengujian *Cropping* dengan Kode Program

4.1.3 Pengujian Model MobileNetV2

Pada penelitian ini juga melatih dan menguji model dengan MobileNetV2 yang merupakan model arsitektur lainnya pada CNN selain dari ResNet-50. Hal ini dilakukan sebagai pembandingan untuk mendapatkan hasil akurasi yang tinggi dengan minim *overfitting* dan *underfitting* pada data gambar yang dilatih. Adapun tahap *preprocessing* yang dilakukan yaitu dengan *resize* (224, 224), *horizontal flip*, *vertical flip*, rotasi (-45, 45) dan normalisasi untuk data *train* dan *resize* (224, 224) untuk data *test*. *Split* data yang diterapkan yaitu dengan perbandingan 80 : 10 : 10 untuk data *train*, *test*, dan *valid*. Selain itu, modifikasi yang dilakukan juga sama dengan model ResNet-50 yaitu menambahkan *droupout* dengan $p = 0,2$.

```

# Performing the Image Transformation and Data Augmentation on the
transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
    transforms.RandomRotation(degrees=(-45, 45)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])

# Augmentation on test images not needed
transform_tests = torchvision.transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),
])

```

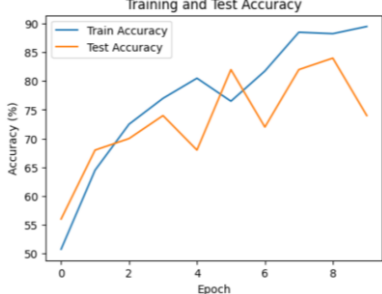
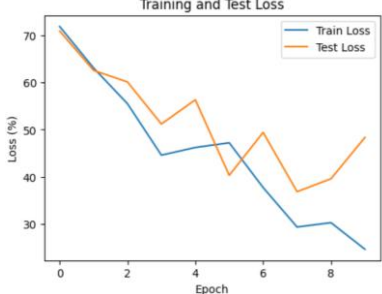
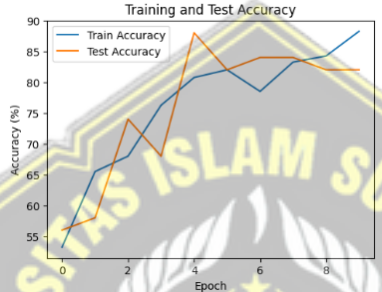
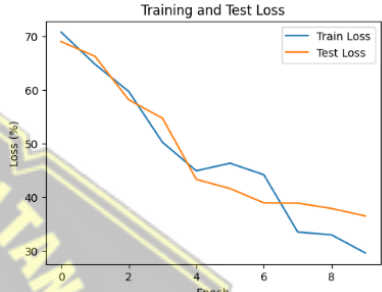
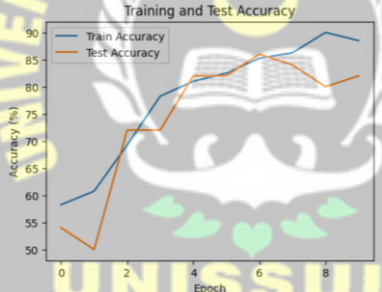
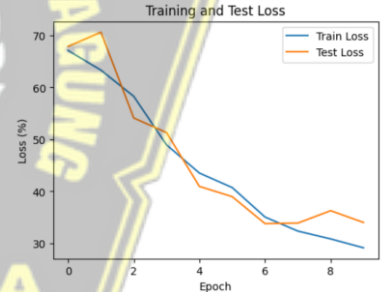
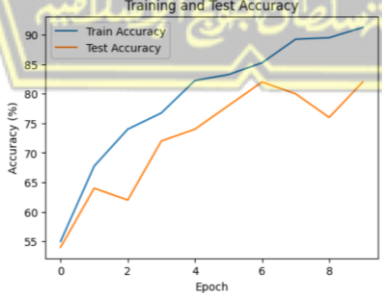
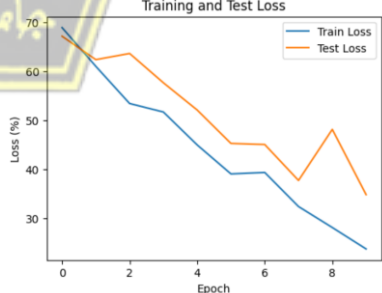
Gambar 4. 7 Pengujian Model MobileNetV2

4.1.4 Hasil Akurasi

Setelah melakukan beberapa pengujian diatas didapatkan hasil akurasi yang digambarkan pada tabel 4.1.

Tabel 4. 1 Plot Accuracy dan Loss

Percobaan	Plot	
	Accuracy	Loss
Skenario 1	<p>Training and Test Accuracy</p>	<p>Training and Test Loss</p>
Skenario 2	<p>Training and Test Accuracy</p>	<p>Training and Test Loss</p>

Percobaan	Plot	
	Accuracy	Loss
Skenario 3	 <p>Training and Test Accuracy</p> <p>Y-axis: Accuracy (%) (50 to 90) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Accuracy (blue), Test Accuracy (orange)</p>	 <p>Training and Test Loss</p> <p>Y-axis: Loss (%) (30 to 70) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Loss (blue), Test Loss (orange)</p>
Skenario 4	 <p>Training and Test Accuracy</p> <p>Y-axis: Accuracy (%) (55 to 90) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Accuracy (blue), Test Accuracy (orange)</p>	 <p>Training and Test Loss</p> <p>Y-axis: Loss (%) (30 to 70) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Loss (blue), Test Loss (orange)</p>
Skenario 5	 <p>Training and Test Accuracy</p> <p>Y-axis: Accuracy (%) (50 to 90) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Accuracy (blue), Test Accuracy (orange)</p>	 <p>Training and Test Loss</p> <p>Y-axis: Loss (%) (30 to 70) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Loss (blue), Test Loss (orange)</p>
Cropping Kode Program	 <p>Training and Test Accuracy</p> <p>Y-axis: Accuracy (%) (55 to 90) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Accuracy (blue), Test Accuracy (orange)</p>	 <p>Training and Test Loss</p> <p>Y-axis: Loss (%) (30 to 70) X-axis: Epoch (0 to 8)</p> <p>Legend: Train Loss (blue), Test Loss (orange)</p>



Tabel 4.1 merupakan grafik dari plot *accuracy* dan *loss* data *train* dan *test* pada setiap percobaan yang dilakukan. Penilaian untuk grafik tersebut yaitu semakin naik garis plot *accuracy* sampai mendekati angka 100 dan semakin turun garis plot *loss* mendekati angka 0, serta jarak diantara grafik *train* dan *test* saling berdekatan maka model tersebut semakin baik. Penjelasan grafik pada tabel 4.1 diterangkan pada tabel 4.2.

Tabel 4. 2 Model Terbaik Berdasarkan Hasil Akurasi

Percobaan	Epochs	Train Acc	Test Acc	Train Loss	Test Loss
Skenario 1	10/10	92,75%	88,00%	22,48%	34,45%
Skenario 2	10/10	85,75%	82,00%	32,21%	37,75%
Skenario 3	10/10	89,50%	74,00%	24,73%	48,35%
Skenario 4	10/10	88,25%	82,00%	29,53%	36,43%
Skenario 5	9/10	90,00%	82,00%	30,85%	34,02%
Cropping Kode Program	10/10	91,25%	82,00%	23,84%	34,89%
MobileNetV2	10/10	87,00%	76,00%	27,45%	52,28%

Dari tabel 4.2 menunjukkan bahwa skenario 1 merupakan model dengan hasil akurasi yang tinggi yaitu 92,75% untuk *train accuracy* dan 88,00% untuk *test accuracy*. Pada skenario 1 model mengalami *overfitting* namun terkategori minim karena berjarak sekitar 4,75% untuk plot *accuracy* dan 11,97% untuk plot *loss*. Namun, jika dilihat dari terminim *overfitting* maka skenario 2 yaitu berjarak sekitar 3,75% untuk plot *accuracy* dan

5,54% untuk plot *loss*. Oleh karena itu, selain melihat hasil akurasi penelitian ini juga melakukan *confusion matrix* untuk menentukan model terbaik dalam mengklasifikasikan dermatitis atopik dan psoriasis.

4.1.5 Hasil *Confusion Matrix*

Perhitungan *confusion matrix* digunakan untuk mencari nilai *accuracy*, *precision*, *recall*, dan *f1 score* yang mana dilakukan ke semua percobaan untuk menemukan konfigurasi terbaik dalam mengklasifikasikan dermatitis atopik (eksim) dan psoriasis.

Tabel 4. 3 Hasil *Confusion Matrix* Skenario 1

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	20 (TP)	5 (FP)
<i>Predicted Psoriasis</i>	1 (FN)	24 (TN)

Tabel 4. 4 Hasil *Confusion Matrix* Skenario 2

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	18 (TP)	7 (FP)
<i>Predicted Psoriasis</i>	2 (FN)	23 (TN)

Tabel 4. 5 Hasil *Confusion Matrix* Skenario 3

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	13 (TP)	12 (FP)
<i>Predicted Psoriasis</i>	1 (FN)	24 (TN)

Tabel 4. 6 Hasil *Confusion Matrix* Skenario 4

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	19 (TP)	6 (FP)
<i>Predicted Psoriasis</i>	3 (FN)	22 (TN)

Tabel 4. 7 Hasil *Confusion Matrix* Skenario 5

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	20 (TP)	5 (FP)
<i>Predicted Psoriasis</i>	4 (FN)	21 (TN)

Tabel 4. 8 Hasil *Confusion Matrix Cropping* Kode Program

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	23 (TP)	2 (FP)
<i>Predicted Psoriasis</i>	7 (FN)	18 (TN)

Tabel 4. 9 Hasil *Confusion Matrix MobileNetV2*

	<i>Actually Positive</i>	<i>Actually Negative</i>
<i>Predicted Eksim</i>	23 (TP)	2 (FP)
<i>Predicted Psoriasis</i>	10 (FN)	15 (TN)

Berdasarkan hasil *confusion matrix* pada tabel 4.3 – 4.9 menunjukkan bahwa hasil skenario 1 dapat mengklasifikasikan penyakit eksim dengan benar sebanyak 20 dari 25 gambar dan penyakit psoriasis 24 dari 25 gambar. Lalu skenario 2 mengklasifikasikan kedua penyakit dengan benar lebih sedikit dibandingkan skenario 1. Skenario 3 mengklasifikasikan penyakit eksim dengan benar paling sedikit dari skenario lainnya namun pada penyakit psoriasis sama dengan skenario 1. Skenario 4 mengklasifikasikan kedua penyakit dengan benar lebih sedikit dari skenario 1. Skenario 5 mengklasifikasikan penyakit eksim dengan benar sebanyak skenario 1 namun pada penyakit psoriasis lebih sedikit dari skenario – skenario sebelumnya. Dan untuk skenario menggunakan teknik *crop* kode program dan model MobileNetV2 mengklasifikasikan kedua penyakit dengan benar hampir sama. Perhitungan tersebut dapat mengetahui nilai *accuracy*, *precision*, *recall*, dan *f1 score* dari setiap konfigurasi yang mana dapat dilihat pada tabel 4.10.

Tabel 4. 10 *Classification Report*

Konfigurasi	Penyakit	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
Skenario 1	Eksim	0,88	0,95	0,80	0,87
	Psoriasis		0,83	0,96	0,89
Skenario 2	Eksim	0,82	0,90	0,72	0,80
	Psoriasis		0,77	0,92	0,84
Skenario 3	Eksim	0,74	0,93	0,52	0,67

Konfigurasi	Penyakit	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
	Psoriasis		0,67	0,96	0,79
Skenario 4	Eksim	0,82	0,86	0,76	0,81
	Psoriasis		0,79	0,88	0,83
Skenario 5	Eksim	0,82	0,83	0,80	0,82
	Psoriasis		0,81	0,84	0,82
Cropping Kode Program	Eksim	0,82	0,77	0,92	0,84
	Psoriasis		0,90	0,72	0,80
MobileNetV2	Eksim	0,76	0,70	0,92	0,79
	Psoriasis		0,88	0,60	0,71

Berdasarkan tabel 4.10 menunjukkan bahwa performa terbaik dari setiap konfigurasi adalah model skenario 1 dengan *accuracy* = 0,88 dan nilai *precision*, *recall* serta *f1 score* berada diantara angka 0,80 – 0,96. Selain dari hasil *confusion matrix* yang unggul, skenario 1 juga memiliki hasil akurasi yang tinggi dan minim *overfitting* dengan nilai 92,75% untuk *train accuracy* dan 88,00% untuk *test accuracy* sehingga menjadikan model skenario 1 sebagai model terbaik dalam mengklasifikasikan dermatitis atopik dan psoriasis.

4.2 Deployment

Pemilihan untuk menggunakan streamlit sebagai *platform* untuk *deployment* adalah dikarenakan kemudahan penggunaan, kecepatan pengembangan dan kebutuhan spesifik penelitian.

4.2.1 Link Streamlit

Pertama yang dilakukan untuk dapat mengakses aplikasi web yang telah dibuat adalah dengan mengakses *link* streamlit yang didapatkan dari hasil *run* ketika pengembang menjalankan seluruh proses *deployment*. Berikut *link* streamlit dan kode *localtunnel* yang dapat dilihat pada gambar 4.8.


```

✓ [1] !pip install -q streamlit
17s
-----8.4/8.4 MB 18.9 MB/s eta 0:00:00
-----190.6/190.6 kB 15.7 MB/s eta 0:00:00
-----4.8/4.8 MB 39.9 MB/s eta 0:00:00
-----82.1/82.1 kB 5.7 MB/s eta 0:00:00
-----62.7/62.7 kB 2.9 MB/s eta 0:00:00

✓ [2] !wget -q -O - ipv4.icanhazip.com
0s
34.73.191.33

! streamlit run app.py & npx localtunnel --port 8501
... [.....] / rollbackFailedOptional: verb npm-session 29c445d640eff40
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.28.0.12:8501
External URL: http://34.73.191.33:8501

npx: installed 22 in 2.941s
your url is: https://shy-loops-grow.loca.lt

```

Gambar 4. 8 *Link* Streamlit dan Kode Localtunnel

Pada gambar 4.8, kode “34.73.191.33” merupakan kode localtunnel dan <https://shy-loops-grow.loca.lt> merupakan *link* streamlit. Setelah *user* berhasil mengakses *link* berikut maka tahapan yang dilakukan yaitu memasukkan kode localtunnel pada gambar 4.9.

shy-loops-grow.loca.lt

Friendly Reminder

This website is served via a localtunnel. This is just a reminder to always check the website address you're giving personal, financial, or login details to is actually the real/official website.

Phishing pages often look similar to pages of known banks, social networks, email portals or other trusted institutions in order to acquire personal information such as usernames, passwords or credit card details.

Please proceed with caution.

To access the website, please confirm the tunnel creator's public IP below.

If you don't know what it is, please ask whoever you got this link from.

This password-like gate is now sadly required since too many phishing portals are being hosted via localtunnel and I'm getting bombarded with abuse notices.

Endpoint IP:

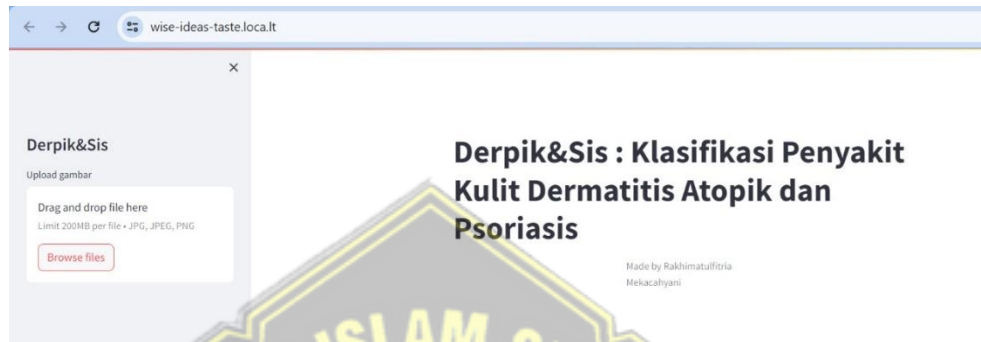
[Click to Submit](#)

Gambar 4. 9 Masukkan Kode Localtunnel

Pada gambar 4.9, *endpoint* IP merupakan kolom untuk *user* memasukkan kode localtunnel.

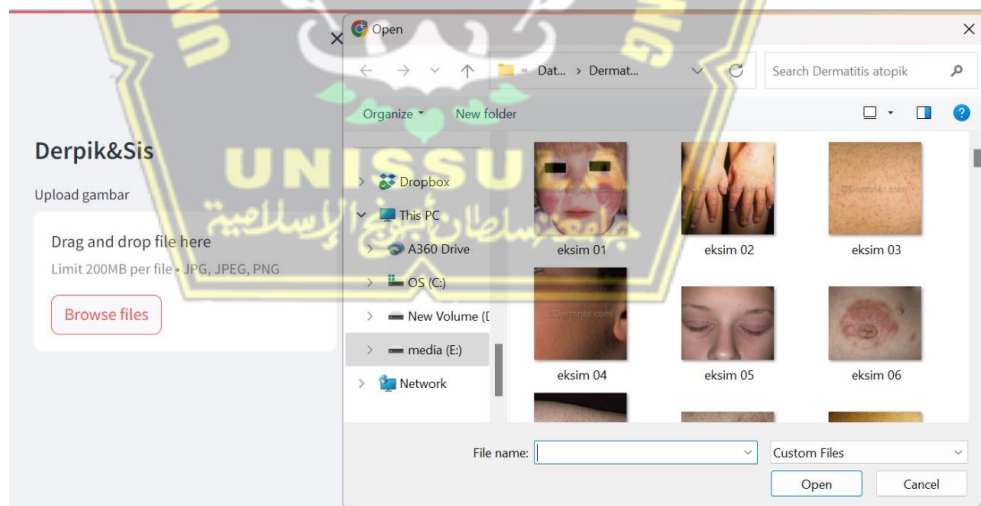
4.2.2 Dashboard Streamlit

Setelah *user* berhasil mengakses memasukkan kode localtunnel dengan benar maka dapat mengklik *submit*. Gambar 4.10 merupakan tampilan *dashboard* atau awal ketika *user* men-*submit* kode localtunnel yang telah dimasukkan.



Gambar 4. 10 Tampilan Awal Streamlit

Untuk dapat mengunggah gambar yang akan diprediksi maka *user* dapat mengklik tombol “Browse Files” pada gambar 4.10 sehingga akan muncul *bottom* yang menampilkan pilihan gambar yang ingin diprediksi seperti gambar 4.11.



Gambar 4. 11 Meng-upload gambar

Pada gambar 4.11, *user* dapat memilih gambar yang akan diprediksi.

4.2.3 Hasil Prediksi

Setelah memilih gambar, maka proses prediksi gambar akan dilakukan. Gambar yang dipilih akan terunggah pada sisi kiri beserta dengan nama file gambar tersebut lalu dibawah gambar akan menampilkan hasil prediksi apakah gambar tersebut terklasifikasikan *class* dermatitis atopik atau psoriasis yang mana tulisannya berwarna hijau. Adapun pada sisi kanan akan menampilkan gambar yang diunggah beserta label prediksi, penjelasan hasil prediksi dan faktor penyebabnya.



Gambar 4. 12 Hasil Prediksi Dermatitis Atopik



Gambar 4. 13 Hasil Prediksi Psoriasis

4.3 Pengujian Sistem

Metode yang digunakan pada tahap ini adalah *blackbox testing*. *Blackbox testing* adalah pengujian fungsionalitas sebuah aplikasi yang sedang dikembangkan pada *software* tanpa perlu mengetahui internal kode atau programnya sehingga pengembang cukup meninjau *input* dan

output-nya. Pada penelitian ini jenis *blackbox testing* yang digunakan yaitu *functional testing*. *Functional testing* adalah proses pengujian yang berfokus pada fitur-fitur dan fungsi spesifik, vital, dan utama dari aplikasi. Tabel 4.11 menunjukkan hasil *functional testing* pada aplikasi yang dikembangkan.

Tabel 4. 11 Hasil Pengujian *Blackbox Testing*

Input	Kasus Pengujian	Output	Kesimpulan
Kode localtunnel	Memasukkan kode localtunnel dari kreator	Sesuai	Normal
Upload gambar	Melakukan <i>upload</i> gambar kulit	Sesuai	Normal
Testing gambar kulit dermatitis atopik	Melakukan testing gambar kulit dermatitis atopik	Sesuai	Normal
Testing gambar kulit psoriasis	Melakukan testing gambar kulit psoriasis	Sesuai	Normal
Testing hasil klasifikasi	Mengecek hasil klasifikasi gambar	Sesuai	Normal
Penjelasan dan gejala penyakit	Menampilkan penjelasan dan gejala penyakit kulit yang terklasifikasi	Sesuai	Normal

Dari tabel 4.11 diterangkan bahwa pengembangan aplikasi dapat menjalankan semua fungsinya dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat disimpulkan bahwa menggunakan algoritma *convolutional neural network* dengan model arsitektur ResNet-50 dapat memuat hasil akurasi yang tinggi dalam mengklasifikasikan penyakit kulit dermatitis atopik dan psoriasis. Hal itu telah teruji dengan melakukan perbandingan terhadap beberapa percobaan seperti mengganti parameter dengan dataset yang telah di-*crop* manual sebelum dilatih, menggunakan dataset yang di-*crop* melalui kode program pada tahap *preprocessing*, dan melatih data dengan model arsitektur yang lain yaitu MobileNetV2. Perbandingan dari percobaan tersebut memiliki 7 konfigurasi dengan rincian 5 skenario dengan *crop* manual, 1 *crop* dengan kode program, dan 1 dengan model MobileNetV2.

Adapun hasil pengujian terbaik yaitu model pada skenario 1 dengan hasil akurasi 92,75% untuk *train accuracy* dan 88,00% untuk *test accuracy*. Selain itu, hasil *confusion matrix* menampilkan nilai *accuracy* untuk data *valid* sebesar 88% dan nilai *precision*, *recall*, *f1 score* untuk dermatitis atopik sebesar 95%, 80%, 87% serta nilai *precision*, *recall*, *f1 score* untuk psoriasis sebesar 83%, 96%, 89%. Dengan demikian, hasil model ini dapat dengan baik dalam mengklasifikasikan gambar antara dermatitis atopik dan psoriasis. Hal itu dapat dilakukan pada aplikasi web *streamlit*, yang mana *user* dapat mengunggah gambar dermatitis atopik atau psoriasis lalu gambar akan diproses dan diprediksi sehingga sistem akan menampilkan hasil prediksi apakah gambar yang diunggah terkategori penyakit kulit dermatitis atopik atau psoriasis serta memberikan penjelasan dan faktor penyebab penyakit tersebut.

5.2 Saran

Berdasarkan penelitian yang sudah ada, penulis menyarankan untuk penelitian yang akan datang adalah :

1. Dari hasil penelitian ini, sistem memiliki akurasi pengujian sebesar 92,75% sehingga diharapkan penelitian selanjutnya dapat memodifikasi penelitian yang telah dilakukan serta melatih data menggunakan model arsitektur yang bisa menghasilkan akurasi yang lebih baik.
2. Penelitian selanjutnya dapat menambahkan jenis penyakit kulit lainnya serta meningkatkan jumlah dan variasi data untuk meminimalisasi terjadinya overfitting.
3. Selain fitur upload gambar, diharapkan penelitian selanjutnya dapat menambahkan fitur secara *real time* dengan kamera untuk tambahan fitur dikala *user* ingin langsung mengecek kondisi kulit mereka.



DAFTAR PUSTAKA

- Alwanda, M.R. *dkk.* (2020) “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle,” *Jurnal Algoritme*, 1(1), hal. 45–56. Tersedia pada: <https://doi.org/10.35957/algoritme.v1i1.434>.
- Arifin, I. *dkk.* (2021) “Penerapan Computer Vision Menggunakan Metode Deep Learning pada Perspektif Generasi Ulul Albab,” *Jurnal Teknologi Terpadu*, 7(2), hal. 98–107. Tersedia pada: <https://doi.org/10.54914/jtt.v7i2.436>.
- Dewi, F.D.K. (2021) “Terapi pada Psoriasis,” *Jurnal Medika Hutama*, 2(02 Januari), hal. 631–641.
- Hutasoit, R.Y.P. *dkk.* (2021) “Implementasi Metode Forward Chaining untuk Identifikasi Penyakit Kulit dan Alternatif Penanganannya,” *INOVTEK Polbeng-Seri Informatika*, 6(1), hal. 90–104.
- Intyanto, G.W. (2021) “Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network),” *Jurnal Arus Elektro Indonesia*, 7(3), hal. 80. Tersedia pada: <https://doi.org/10.19184/jaei.v7i3.28141>.
- Izzati, A. dan Waluya, O.T. (2018) “Gambaran penerimaan diri pada penderita psoriasis,” *Jurnal Psikologi Esa Unggul*, 10(02), hal. 126366.
- Nashrullah, F. *dkk.* (2020) “COMPLETE Journal of Computer, Electronic, and Telecommunication Investigasi Parameter Epoch Pada Arsitektur ResNet-50 Untuk Klasifikasi Pornografi,” *Journal of Computer, Electronic, and Telecommunication*, 1(1).
- Padilla, D. *dkk.* (2019) “Differentiating Atopic Dermatitis and Psoriasis Chronic Plaque using Convolutional Neural Network MobileNet Architecture,” *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2019*, hal. 0–5. Tersedia pada: <https://doi.org/10.1109/HNICEM48295.2019.9073482>.

- Palareti, G. *dkk.* (2016) "Comparison between different D-Dimer cutoff values to assess the individual risk of recurrent venous thromboembolism: Analysis of results obtained in the DULCIS study," *International Journal of Laboratory Hematology*, 38(1), hal. 42–49. Tersedia pada: <https://doi.org/10.1111/ijlh.12426>.
- Pratiwi, H.I. dan Kamardi, R. (2019) "Pengembangan Sistem Web Sebagai Diagnosa Dini Penyakit Alergi Kulit Dermatitis Atopik Dengan Metode Forward Chaining," *Widyakala J*, 6(2), hal. 167.
- Qodryantha, A. *dkk.* (2023) "Pembangunan Sistem Pengawasan Cerdas Dengan Visualisasi 3D Development Of Smart Surveillance System With 3D Visualization," 9(1), hal. 465–480.
- Rachman, P.O. *dkk.* (2022) "Peran Balneoterapi pada Psoriasis," *MEDICINUS*, 35(2), hal. 62–70.
- Sari, D.M.R. *dkk.* (2023) "Dermatitis Atopic and Psoriasis Skin Disease Classification by using Convolutional Neural Network," *Computer Engineering and Applications Journal*, 12(1), hal. 1–10. Tersedia pada: <https://doi.org/10.18495/comengapp.v12i1.419>.
- Schielein, M.C. *dkk.* (2023) "Outlier detection in dermatology: Performance of different convolutional neural networks for binary classification of inflammatory skin diseases," *Journal of the European Academy of Dermatology and Venereology*, 37(5), hal. 1071–1079. Tersedia pada: <https://doi.org/10.1111/jdv.18853>.
- Sharma, M. *dkk.* (2021) "Detection and Diagnosis of Skin Diseases Using Residual Neural Networks (RESNET)," *International Journal of Image and Graphics*, 21(5). Tersedia pada: <https://doi.org/10.1142/S0219467821400027>.
- Susanto, P.M. (2022) "Tatalaksana Dermatitis Atopik Pada Anak," *Jurnal Medika Hutama*, 3(02 Januari), hal. 2248–2260.
- Tukar, T. dan Bhavana, P. (2021) "Deteksi dan Klasifikasi Penyakit Kulit," hal. 1096–1101.

Umam, C. dan Handoko, L.B. (2021) “Convolutional Neural Network (CNN) Untuk Identifikasi Karakter Hiragana,” in *Prosiding Seminar Nasional Lppm Ump*, hal. 527–533.



LAMPIRAN

