

Deteksi Penyakit Kulit dengan Pendekatan Machine Learning Berbasis Convolutional Neural Network (CNN)

Deliya Syafa^{1*}, Ardin Hersandini²

^{1,2}Program Studi Teknik Informatika, Fakultas Informatika dan Komputer, Universitas Binaniaga Indonesia

Email¹: syafadeliya@gmail.com

Email²: ahersandini@gmail.com

*Corresponding Author

ABSTRACT

Measles is a contagious disease that is often experienced, especially by children, so it requires early detection to prevent serious complications. This study aims to apply a Convolutional Neural Network (CNN)-based Machine Learning approach to detect measles quickly and accurately. The data used consists of 1,041 skin images (measles and non-measles) obtained from the research dataset. The data is processed through the stages of pre-processing, data augmentation, dataset division (training, validation, testing), model training using the ResNet34 architecture, and performance evaluation with a confusion matrix. The results show that the developed measles detection system has an accuracy of 95%, Precision of 90%, Recall of 100%, and F1-Score of 94.7%. The feasibility test shows that the system is very feasible with a feasibility level of 100% and a user satisfaction level of 100% based on the PSSUQ questionnaire. This study proves that the CNN algorithm with the ResNet34 architecture implemented on the website is able to provide accurate, fast, and effective detection in supporting the early detection of measles.

Keywords: CNN, Deep Learning, Measles Detection, Skin Image, Artificial Intelligence, Measles.

ABSTRAK

Penyakit campak merupakan salah satu penyakit menular yang sering dialami terutama oleh anak-anak sehingga membutuhkan deteksi dini untuk mencegah komplikasi yang serius. Penelitian ini bertujuan untuk menerapkan pendekatan Machine Learning berbasis Convolutional Neural Network (CNN) dalam mendeteksi penyakit campak secara cepat dan akurat. Data yang digunakan terdiri dari 1.041 citra kulit (campak dan bukan campak) yang diperoleh dari dataset penelitian. Data tersebut diproses melalui tahapan pre-processing, augmentasi data, pembagian dataset (training, validasi, testing), pelatihan model menggunakan arsitektur ResNet34, hingga evaluasi performa dengan confusion matrix. Hasil penelitian menunjukkan bahwa sistem deteksi campak yang dikembangkan memiliki akurasi 95%, Precision 90%, Recall 100%, dan F1-Score 94.7%. Uji kelayakan menunjukkan bahwa sistem sangat layak dengan tingkat kelayakan 100% serta tingkat kepuasan pengguna sebesar 100% berdasarkan kuesioner PSSUQ. Penelitian ini membuktikan bahwa algoritma CNN dengan arsitektur ResNet34 yang diimplementasikan pada website mampu memberikan deteksi yang akurat, cepat, dan efektif dalam mendukung deteksi dini penyakit campak.

Kata kunci: CNN, Deep Learning, Deteksi Campak, Citra Kulit, Artificial Intelligence, Campak.

A. PENDAHULUAN

Kulit merupakan bagian terluar dari tubuh manusia yang berfungsi untuk melapisi dan melindungi tubuh dari cedera, infeksi, dan gangguan lingkungan (Aritonang dkk., 2023). Kurangnya perhatian masyarakat terhadap kesehatan kulit, sehingga berisiko mengalami berbagai penyakit kulit yang dapat berdampak signifikan terhadap kualitas hidup, penyakit kulit juga dapat menyerang siapa saja dan di mana saja pada bagian seluruh tubuh manusia. Penyakit kulit yang sering terjadi ditandai dengan beberapa gejala seperti ruam, gatal, atau luka yang sulit sembuh karena disebabkan oleh infeksi, alergi, bakteri maupun virus.

Salah satu penyakit kulit yang disebabkan oleh virus dan perlu diwaspadai adalah Campak (measles) merupakan penyakit menular yang disebabkan oleh Virus Measles Morbivirus yang menyebar melalui percikan udara dari batuk atau bersin individu yang terinfeksi. Penyakit ini sangat mudah menular, terutama pada anak-anak, orang dewasa yang belum pernah menerima vaksin campak juga tetap berisiko tertular. Penyakit campak memiliki masa inkubasi sekitar 7–18 hari, rata-rata 10 hari setelah terpapar virus, di mana gejala awal mulai muncul secara bertahap seperti demam tinggi, batuk, pilek, mata merah, dan ruam khas.

Ruam ini berbentuk makulopapular (bercak merah yang sedikit menonjol) dan dapat bertahan selama 5–7 hari sebelum akhirnya memudar secara bertahap. Pada tahap akhir, ruam bisa berubah warna menjadi kecokelatan dan disertai dengan pengelupasan kulit ringan.

Campak merupakan penyakit yang epidemis di Indonesia dan telah di laporan setiap tahunnya. Pada tahun 2022 dan 2023, terjadi peningkatan signifikan jumlah kasus terkonfirmasi, dibandingkan dengan yang dilaporkan setiap tahun sejak 2018: terdapat 920 kasus yang dilaporkan pada tahun 2018, 639 pada tahun 2019, 310 pada tahun 2020, dan 132 pada tahun 2021. Berdasarkan data WHO Dari tanggal 1 Januari hingga 3 April 2023, tercatat sebanyak 2.161 kasus measles (848 terkonfirmasi secara laboratorium dan 1.313 dicurigai terinfeksi secara klinis) di 18 dari 38 provinsi di Indonesia, terutama di provinsi Jawa Barat (796 kasus), Papua Tengah (770 kasus), dan Banten (197 kasus).

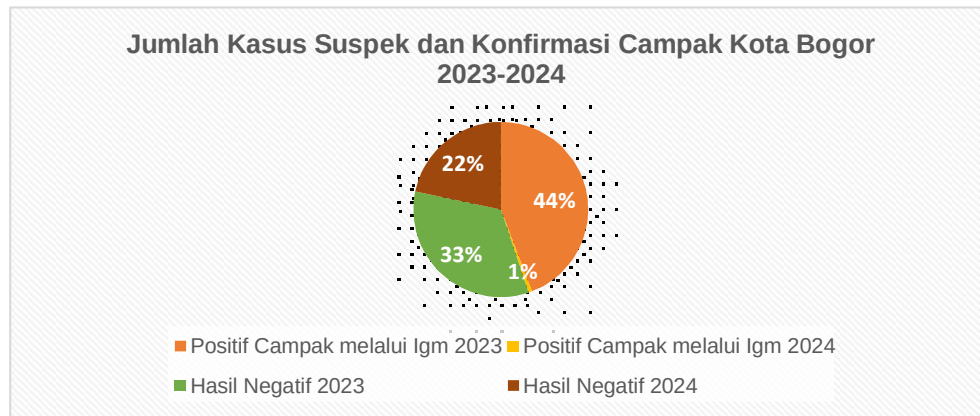
Meski banyak kasus campak dapat sembuh dengan perawatan medis yang tepat, apabila tidak ditangani dengan baik, penyakit ini dapat mengakibatkan komplikasi serius, terutama pada anak-anak dengan sistem imun lemah, ibu hamil, dan orang dewasa yang belum divaksinasi. Komplikasi tersebut seperti pneumonia, diare berat, atau radang otak, yang dapat berisiko jiwa dan berdampak signifikan pada kualitas hidup penderitanya. Penyebaran yang cepat dan potensi komplikasi ini membuat deteksi serta penanganan campak sangat penting. Seseorang yang pernah mengalami campak akan memiliki kekebalan terhadap penyakit tersebut sepanjang hidupnya.

Secara umum, pemeriksaan campak dilakukan dengan memperhatikan gejala klinis namun, untuk memastikan diagnosis, tes laboratorium seperti pemeriksaan serologi atau PCR (Polymerase Chain Reaction) sering dilakukan untuk mendeteksi virus campak dalam tubuh. Tes laboratorium ini memberikan hasil yang lebih tepat untuk memastikan infeksi campak, terutama ketika gejala awalnya menyerupai penyakit lain seperti rubella atau alergi kulit.

Meskipun pemeriksaan medis atau laboratorium seperti pemeriksaan serologi (IgM) dan RT-PCR memberikan hasil yang lebih tepat dan akurat namun, dalam pelaksanaannya membutuhkan waktu, tenaga medis terlatih, dan biaya yang tidak sedikit, terutama di wilayah dengan akses kesehatan yang terbatas. Sebagaimana dilaporkan oleh Pan American Health Organization (PAHO) (de Soárez dkk., 2024), dalam biaya untuk penanganan kasus campak di wilayah Amerika dapat mencapai USD 2.979 dengan total anggaran pengendalian wabah mencapai lebih dari USD 107 juta. Biaya ini mencakup dengan pelacakan kontak, vaksinasi ulang, dan biaya logistik lainnya, yang menunjukkan beban ekonomi yang besar apabila deteksi dini tidak dilakukan secara efektif.

Sementara itu, menurut laporan WHO tahun 2023 juga menyoroti keterbatasan dalam ketersediaan reagen laboratorium di beberapa negara berkembang, termasuk Indonesia. Keterbatasan ini menjadi salah satu faktor penghambat dalam pelaksanaan pemeriksaan laboratorium seperti serologi IgM dan RT-PCR, yang sejatinya diperlukan untuk memastikan diagnosis campak secara akurat.

Berdasarkan data dari Dinas Kesehatan Kota Bogor tahun 2023–2024 seperti pada Gambar 1 , tercatat pada tahun 2023 jumlah kasus suspek campak sebanyak 610 orang, dengan hasil pemeriksaan IgM positif pada 347 orang dan negatif pada 263 orang, yang kemungkinan disebabkan oleh penyakit lain. Sementara itu, pada tahun 2024, tercatat 175 kasus orang yang diduga campak (suspek) yang didiagnosis secara klinis. Namun, hasil konfirmasi laboratorium IgM menunjukkan hanya 5 kasus yang benar-benar positif, sedangkan 170 kasus dinyatakan negatif, yang kemungkinan juga disebabkan oleh penyakit lain yang memiliki gejala serupa. Hal ini menunjukkan bahwa pemeriksaan awal secara konvensional yang hanya melihat ruam pada kulit tidak cukup akurat, karena ruam campak bisa sangat mirip dengan ruam penyakit lain seperti cacar air, atau alergi. Oleh sebab itu, dibutuhkan cara deteksi awal yang lebih tepat agar pasien yang benar-benar terduga campak bisa segera dirujuk untuk pemeriksaan laboratorium.



Gambar 1. Jumlah Kasus Suspek dan Konfirmasi Campak Kota Bogor 2023-2024
 Sumber: Dinas Kesehatan Kota Bogor

Sebagai data pendukung Chart di atas yang menampilkan perbandingan kasus suspek campak tahun 2023 dan 2024, di bawah ini disajikan data distribusi kasus suspek campak per kecamatan di Kota Bogor pada tahun 2024. Tabel ini menampilkan jumlah kasus yang diperoleh melalui pemeriksaan konvensional serta hasil pemeriksaan laboratorium (IgM positif dan negatif) di setiap kecamatan. Informasi ini memberikan gambaran penyebaran kasus suspek dan hasil konfirmasi laboratorium campak di Kota Bogor selama tahun 2024.

Tabel 1. Data Suspek Pasien Campak Berdasarkan data Dinkes di Kota Bogor Tahun 2024

Kecamatan	Konvensional		Pemeriksaan Lab	
	Igm Positif	Igm Negatif	Igm Positif	Igm Negatif
Tanah Sareal	13	19	2	30
Bogor Barat	32	30	1	61
Bogor Selatan	10	17	1	26
Bogor Timur	7	14	-	21
Bogor Tengah	2	17	1	18
Bogor Utara	8	6	-	14

Meskipun tes laboratorium memiliki peranan penting dalam memastikan diagnosis, proses ini sering kali memakan waktu yang cukup lama. Rata-rata, waktu yang diperlukan dari kemunculan gejala hingga hasil laboratorium diterima adalah sekitar 10-14 hari. Oleh karena itu, teknologi berbasis citra digital yang bisa membantu menganalisis ruam kulit dapat menjadi alternatif yang efektif untuk mempercepat proses deteksi dan membantu diagnosa awal penyakit campak melalui pengamatan ruam pada kulit. Menurut Aritonang dkk. (2023), sistem berbasis pemrosesan citra digital merupakan solusi yang inovatif dan efektif untuk membantu tenaga kesehatan dalam mendiagnosis penyakit kulit.

[NO_PRINTED_FORM] [2] menyebutkan bahwa pemanfaatan teknologi memiliki potensi yang besar untuk memberikan manfaat bagi berbagai sektor, termasuk kesehatan. Hal ini juga memungkinkan praktisi di berbagai bidang untuk mengumpulkan, menyimpan, dan memproses data dengan lebih efisien, sehingga menghasilkan informasi berguna yang dapat digunakan oleh pengambil keputusan untuk merumuskan langkah-langkah yang lebih tepat dalam diagnosis dan pengobatan.

Salah satu tantangan utama dalam pemeriksaan campak adalah proses yang membutuhkan waktu relatif lama, terutama ketika tanda awal seperti ruam merah sulit dikenali atau mirip dengan kondisi kulit lainnya. Tanda-tanda campak, seperti ruam merah pada permukaan kulit, sering kali memiliki kemiripan dengan kondisi penyakit lain, seperti alergi, cacar air atau penyakit lainnya yang juga menimbulkan ruam. Hal ini dapat membuat petugas kesehatan kesulitan membedakan antara campak dan penyakit kulit lainnya, yang berpotensi menyebabkan kesalahan diagnosis atau keterlambatan dalam perawatan yang tepat.

Mengamati beragam kesulitan dalam mendiagnosis campak, diperlukan suatu pendekatan yang lebih efisien untuk membantu mengenali penyakit ini. Rencana dari penelitian ini untuk menghasilkan aplikasi berbasis website dengan menerapkan algoritma Convolutional Neural Network untuk mendeteksi penyakit campak

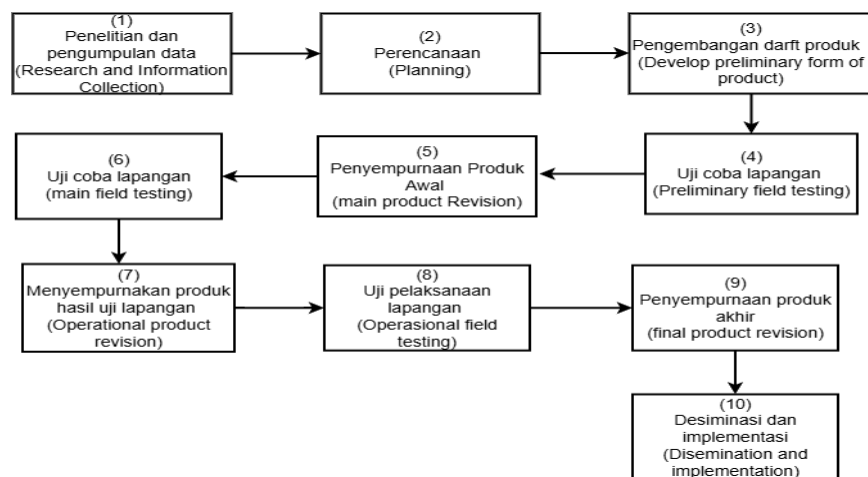
dengan cepat, akurat, dan objektif untuk membantu diagnosis awal. Teknologi ini diharapkan dapat memberikan solusi terhadap tantangan dalam proses diagnosis serta menjadi alat bantu yang efektif bagi tenaga medis dan dapat dilakukan dengan lebih efektif dan efisien, sehingga meningkatkan kualitas layanan kesehatan.

B. METODE

Algoritma Convolutional Neural Network adalah salah satu metode deep learning yang memiliki kemampuan untuk mengenali pola pada citra dengan cara mengekstraksi fitur penting melalui lapisan konvolusi dan pooling, sehingga mampu mendeteksi tepi, tekstur, dan bentuk secara otomatis tanpa perlu proses ekstraksi manual. Oleh karena itu algoritma ini sangat cocok untuk dimanfaatkan dalam menganalisis citra ruam kulit pada penderita campak.

1. Tahapan Penelitian

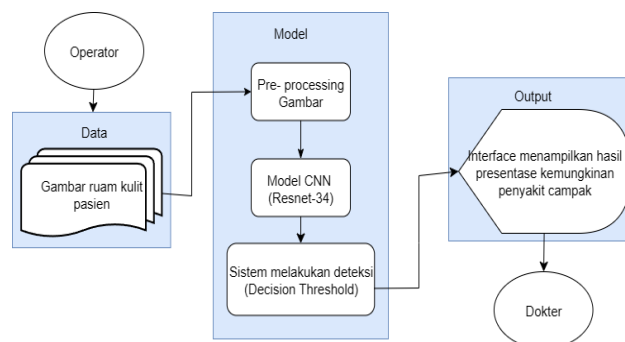
Metode yang digunakan pada penelitian dan pengembangan ini, menggunakan metode Research and Development yang bertujuan untuk menciptakan atau memperbaiki produk, jasa, atau teknik melalui langkah-langkah penelitian ilmiah dan evaluasi. Berikut langkah-langkah yang dikenal sebagai siklus R&D yang dikembangkan oleh Borg & Gall, [3, hlm. 764] yang akan diterapkan dalam penelitian ini:



Gambar 2. Langkah-langkah metode R&D menurut Borg And Ball
 Sumber : [3, hlm. 764]

2. Model Konseptual

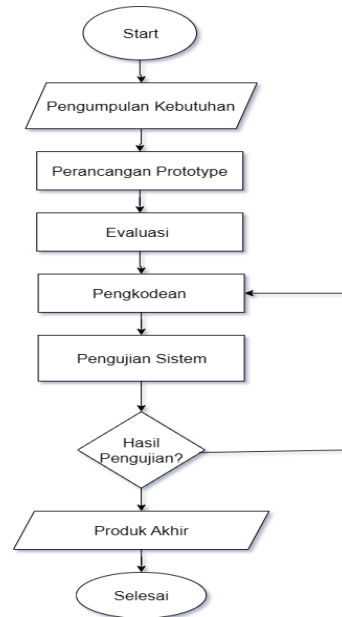
Model yang diusulkan pada penelitian ini adalah AI Medical Skin Disease System, yang terdiri dari empat komponen utama: operator, model CNN, interface, dan dokter. Pada model ini Operator bertugas untuk menginputkan data pasien beserta citra ruam kulit pasien, kemudian model CNN memproses citra melalui tahapan pre-processing, feature extraction, dan classification. Hasil deteksi ditampilkan melalui interface sebagai media interaksi pengguna dengan sistem. Output akhir digunakan oleh dokter sebagai bahan pertimbangan dalam pengambilan keputusan medis.



Gambar 3. Komponen AI Medical Skin Disease System

3. Model Prosedural

Salah satu model pengembangan yang diterapkan dalam penelitian ini adalah model prototipe, yaitu rancangan awal produk untuk memahami kebutuhan pengguna. Prototipe memudahkan proses penelitian dan pengembangan secara efektif melalui versi kecil sistem yang dapat diperluas. Dalam penelitian ini, metode prototipe dibagi menjadi beberapa tahap proses pengembangan. Tahapan-tahapan tersebut meliputi :



Gambar 4. Tahapan Metode Prototyping

4. Uji Coba Produk

Uji coba produk dilakukan untuk mengumpulkan data yang akan digunakan dalam perhitungan untuk mengukur tingkat akurasi produk, serta sebagai dasar untuk menentukan kelayakan produk yang telah dihasilkan. pada penelitian ini menggunakan 2 instrumen, yaitu pengujian dilakukan untuk ahli dan pengujian pengguna. Alat ukur yang digunakan untuk ahli yaitu menggunakan Blackbox Testing dan untuk pengguna menggunakan PSSUQ. Rumus yang digunakan adalah sebagai berikut;

$$\text{Presentase kelayakan (\%)} = \frac{\text{Skor yang di observasikan}}{\text{Skor yang di harapkan}} \times 100\%$$

Penilaian kelayakan untuk aspek yang diteliti terbagi menjadi lima kategori yaitu: sangat layak, layak, cukup layak, tidak layak, dan sangat tidak layak. Skala ini menggunakan hasil dalam bentuk presentase, dengan nilai maksimum 100% dan minimum 0%. Berikut adalah rentang pembagian untuk setiap kategori kelayakan yaitu sebagai berikut:

Tabel 2. Kategori Kelayakan

Presentase Pencapaian	Kategori Kelayakan
< 21%	Sangat Belum Layak
21% - 40%	Belum Layak
41% - 60%	Cukup Layak
61% - 80%	Layak
81%- 100%	Sangat Layak

Sumber : (Aritonang 2009)

5. Uji Hasil

a. Uji efektivitas

Hasil prototype pada aspek efektivitas akan diuji menggunakan metode Field Observation yaitu dengan cara mengobservasi pengguna secara langsung bagaimana mereka menggunakan aplikasi tersebut, tingkat efektivitas akan diukur menggunakan tingkat keberhasilan pada pengguna. Berikut rumus persamaan untuk menghitung efektivitas :

$$Efektifitas (\%) = \frac{Skor\ yang\ didapatkan}{Skor\ Maksimal} \times 100\%$$

Selanjutnya digunakan untuk menghitung nilai kepuasan yaitu menghitung skor yang dapatkan dengan nilai maksimal. Pada skala Likert 1 sampai 5 yang dibuat dengan skala sangat setuju hingga sangat tidak setuju dibuat untuk membantu pengguna dalam menilai efektivitas kerja sistem.

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Cukup

4 = Setuju

5 = Sangat Setuju

Acuan dari tingkat efektivitas bersumber dari Litbang Depdagri (1991), dengan kategori di bawah:

Tabel 3. Ukuran Standar Efektivitas Acuan Litbang Depdagri

Rasio Efektivitas	Tingkat Capai
Di bawah 40%	Sangat Tidak Efektif
40 -59,99%	Tidak Efektif
60 -79,99%	Cukup Efektif
Diatas 80%	Sangat Efektif

b. Confusion Matrix

Confusion matrix pada penelitian ini digunakan untuk mengevaluasi performa model klasifikasi terhadap data yang tersedia (Sharma, 2020). Matriks ini terdiri dari empat komponen utama: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).

Nilai-nilai tersebut digunakan untuk menghitung beberapa metrik performa model, yaitu:

$$Akurasi (\%) = \frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100\%$$

Akurasi menggambarkan seberapa akurat model machine learning dalam mengidentifikasi objek dengan benar

$$Precision (\%) = \frac{TP}{(TP + FP)} \times 100\%$$

Precision menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model.

$$Recall (\%) = \frac{TP}{(TP + FN)} \times 100 \%$$

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

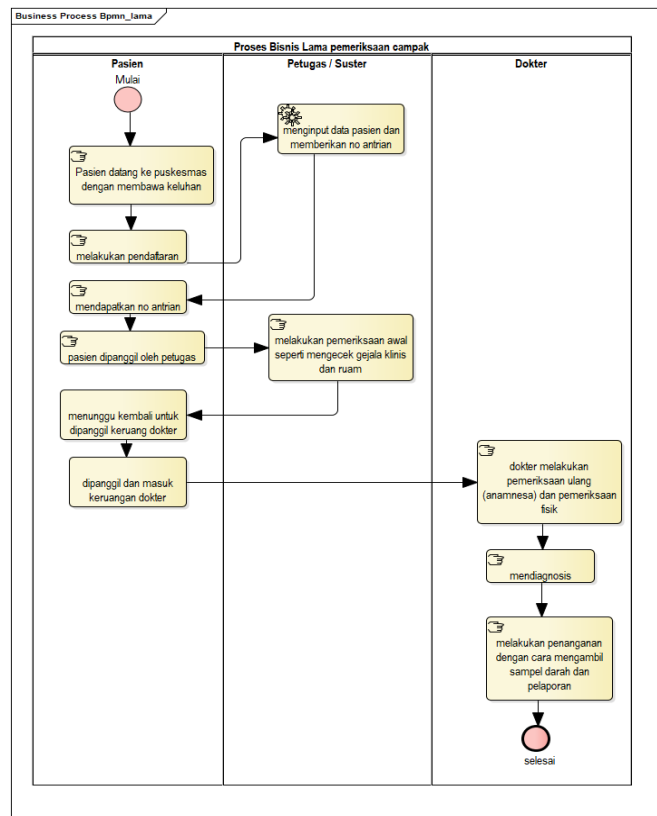
$$F1\ Score (\%) = \frac{(2 * Recall * Precision)}{(Recall + Precision)} \times 100 \%$$

C. HASIL DAN PEMBAHASAN

1. Hasil

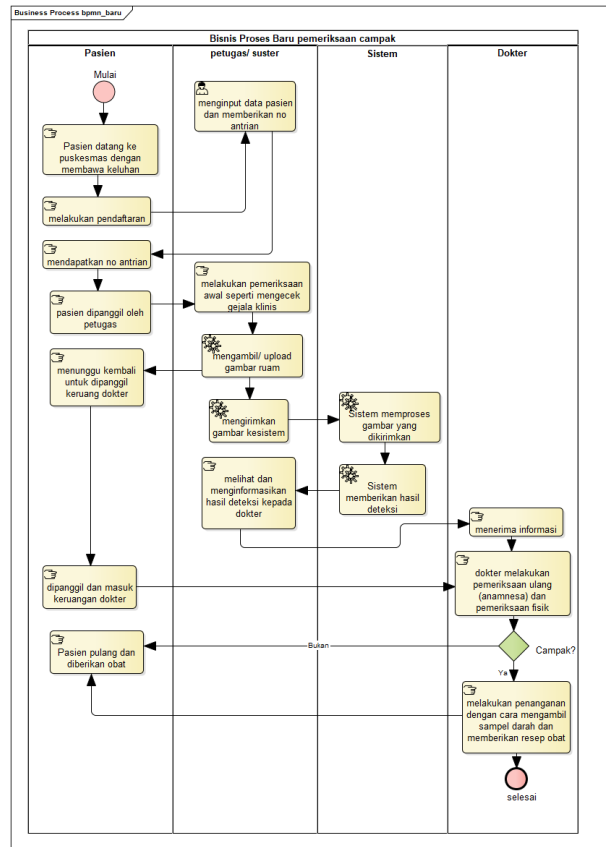
a. Hasil Analisis Proses

Berdasarkan pada proses analisis kebutuhan diperlukan agar dapat menangani permasalahan terkait deteksi penyakit kulit yaitu menggunakan algoritma CNN. Terdapat beberapa tahap yang dapat dilakukan untuk menangani masalah tersebut yaitu membuat proses bisnis baru. Berikut dijelaskan proses bisnis lama yang berlangsung oleh pihak puskesmas:



Gambar 5. Proses Bisnis Lama Pemeriksaan Campak

Berdasarkan Gambar 5, sistem pemeriksaan lama masih bersifat konvensional dan bergantung pada pengamatan langsung, sehingga rentan terhadap kesalahan diagnosis, terutama saat jumlah pasien tinggi. Untuk mengatasi hal tersebut, dirancang sistem baru berbasis Convolutional Neural Network (CNN). Proses dimulai dari pendaftaran pasien di Puskesmas Pasir Mulya, dilanjutkan dengan pemeriksaan awal oleh petugas yang mengunggah citra ruam kulit pasien. Citra tersebut diproses oleh sistem untuk menghasilkan diagnosis awal, yang kemudian diverifikasi oleh dokter guna menentukan diagnosis akhir apakah pasien terindikasi campak atau tidak.



Gambar 6. Proses Bisnis Baru Pemeriksaan Campak

b. Hasil Analisis Metode

Berdasarkan hasil analisis, diketahui bahwa sistem deteksi ruam campak yang berjalan saat ini belum menerapkan komputerasi, khususnya metode Convolutional Neural Network (CNN). Metode CNN dipilih karena kemampuannya dalam mengenali pola visual secara otomatis melalui proses ekstraksi fitur langsung dari gambar, sehingga sesuai untuk tugas klasifikasi gambar seperti deteksi gejala ruam kulit.

1. Dataset

Dataset yang digunakan terdiri dari dua kelas, yaitu campak dan bukan campak, dengan total 260 citra (114 citra campak dan 146 citra bukan campak). Untuk memberikan gambaran, pada Gambar 7 dan 8 ditampilkan contoh citra dari masing-masing kelas.



Gambar 7. Gambar Dataset Bukan Campak



Gambar 8. Gambar Dataset Campak

2. Pre-processing Data

Tahapan pre-processing dilakukan untuk mempersiapkan citra sebelum dilatih pada model, meliputi:

- Resize : Setiap citra diubah ukurannya menjadi 384×384 piksel agar memiliki dimensi seragam sesuai kebutuhan arsitektur CNN.
- Normalisasi : Nilai piksel dikonversi ke rentang 0–1 menggunakan rumus:

$$x^1 = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Selain itu digunakan juga normalisasi standar ImageNet dengan rumus :

$\mu = [0.485, 0.456, 0.406]$ (mean tiap kanal R, G, B),

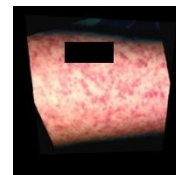
$\sigma = [0.229, 0.224, 0.225]$ (standar deviasi tiap kanal R, G,B).

- Augmentasi Data : Tahap ini berfungsi untuk memperbanyak variasi citra agar model tidak mudah overfitting dilakukan augmentasi seperti rotasi acak ($\leq 20^\circ$), flipping horizontal, zoom, color jitter, random perspective, grayscale, gaussian blur, dan random erasing.

Berikut adalah hasil gambar yang telah di augmentasi:



Gambar 9, Hasil Augmented Bukan Campak

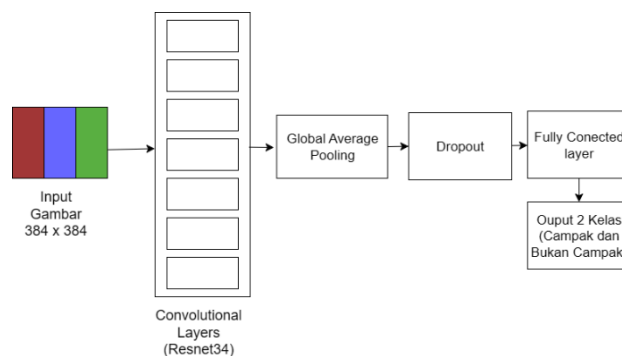


Gambar 10. Hasil Augmented Campak

Dataset kemudian dibagi menjadi tiga bagian menjadi Data latih: 70%, Data validasi: 15%, Data uji: 15%.

3. Arsitektur Model

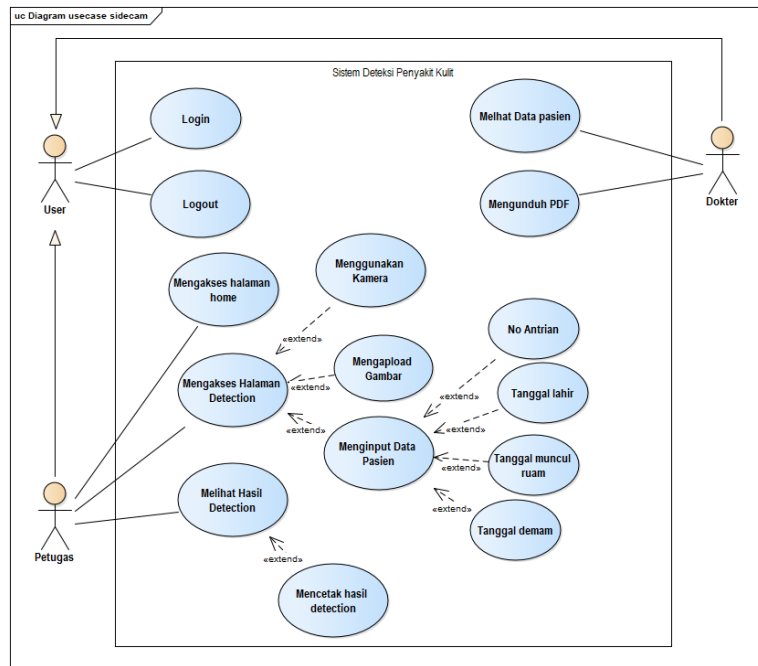
Model yang digunakan adalah ResNet34 dengan tambahan Dropout layer ($p=0.5$) pada bagian fully connected layer guna mengurangi overfitting. ResNet34 dipilih karena struktur residual block-nya mempermudah aliran gradien selama pelatihan, sehingga efektif meskipun dataset terbatas. Berikut adalah alur kerja Metode CNN dalam penelitian ini adalah sebagai berikut :



Gambar 11. Arsitektur Model CNN

c. Hasil Analisis Kebutuhan Sistem

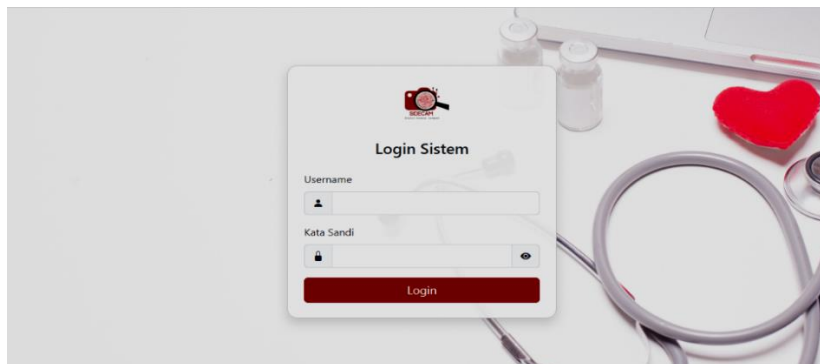
Sistem yang dikembangkan digambarkan melalui use case diagram yang menunjukkan interaksi antara pengguna dan sistem deteksi penyakit kulit berbasis website. Diagram ini bertujuan memastikan bahwa sistem sesuai dengan kebutuhan dan tujuan yang diharapkan.



Gambar 12. Use Case Diagram

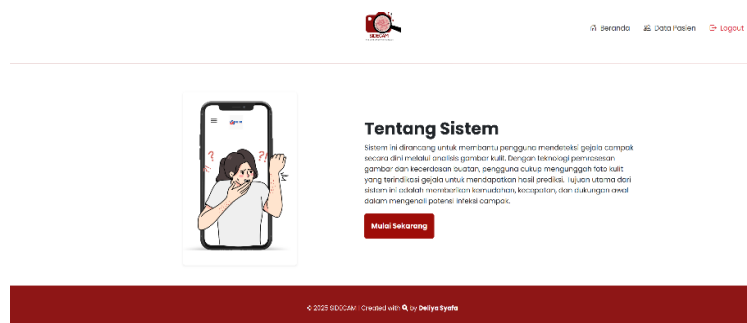
d. Prototipe Program

1. Halaman Login



Gambar 13. Halaman Login

2. Halaman Home



Gambar 14. Halaman Home

3. Halaman Deteksi

Gambar 15. Halaman Deteksi

4. Halaman Hasil Deteksi

Gambar 16. Halaman Hasil Deteksi

5. Halaman Lihat Data Pasien

Data Pasien & Hasil Diagnosa									
No	No Antrian	Tgl Lahir	Tgl Demam	Tgl Ruam	Kategori	Akurasi	Saran	Gambar	Heatmap
1	669h	2025-08-16	2025-08-09	2025-08-26	Bukan Campak	65.7%	Hasil deteksi menunjukkan ruam bukan disebabkan oleh campak. Perhatikan perkembangan gejala dan tetap jaga kesehatan tubuh.		
2	j990	2025-08-08	2025-08-28	2025-08-28	Campak	97.5%	Deteksi menunjukkan adanya indikasi campak. Disarankan segera konsultasi ke dokter untuk pemeriksaan lebih lanjut.		
3	h8900	2025-08-14	2025-08-30	2025-08-29	Bukan Campak	92.6%	Hasil deteksi menunjukkan ruam bukan disebabkan oleh campak. Perhatikan perkembangan gejala dan tetap jaga kesehatan tubuh.		
4	N700	2025-08-15	2025-08-29	2025-07-30	Bukan Campak	95.6%	Hasil deteksi menunjukkan ruam bukan disebabkan oleh campak. Perhatikan perkembangan gejala dan tetap jaga kesehatan tubuh.		
5	h990	2025-08-08	2025-08-28	2025-08-28	Bukan Campak	98.9%	Hasil deteksi menunjukkan ruam bukan disebabkan oleh campak. Perhatikan perkembangan gejala dan tetap jaga kesehatan tubuh.		
6	tas207	2025-08-16	2025-08-22	2025-08-18	Campak	72%	Deteksi menunjukkan adanya indikasi campak. Disarankan segera konsultasi ke dokter untuk pemeriksaan lebih lanjut.		

Gambar 17. Halaman Lihat Data Pasien

e. Pengkodean

Pengkodean pada penelitian ini menggunakan bahasa pemrograman Python dengan menerapkan algoritma Convolutional Neural Network (CNN). Proses pengkodean mencakup tahap pre-processing data, pelatihan model menggunakan arsitektur ResNet34 hingga mendapatkan hasil model yang akan diterapkan pada website berbasis laravel. Berikut adalah beberapa source code yang digunakan:

1. Preprocessing data

Pada tahap pertama yaitu melakukan pre-processing data, dimana tahap ini meliputi resize, augmentasi, dan split data. Gambar diubah ukurannya menjadi 384×384 piksel, kemudian diperbanyak melalui teknik augmentasi seperti rotasi dan flip. Setelah itu, dataset dibagi menjadi tiga bagian yaitu data latih, data validasi, dan data uji agar model CNN dapat dilatih, divalidasi, serta diuji secara optimal. Berikut adalah kode yang digunakan dalam proses ini:

(1) Input data bagian pre-processing

```
# Path dataset input dan output
input_root = '/content/drive/MyDrive/Sidecam/data/dataset'
output_root = '/content/drive/MyDrive/Sidecam/data/data_augmented_fix8'

# Rasio pembagian data
split_ratio = {'train': 0.7, 'val': 0.15, 'test': 0.15}

# Target minimal data latih per kelas setelah augmentasi
target_train = 400
```

Gambar 18. Input Data Preprocessing

(2) Proses Pengolahan dan Augmentasi

```
=== 4. Transformasi Augmentasi (Tensor-based, tidak error) ===
augmentation_transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(degrees=20),
    transforms.ColorJitter(brightness=0.4, contrast=0.4, saturation=0.4, hue=0.03),
    transforms.RandomAffine(degrees=0, translate=(0.05, 0.05), scale=(0.85, 1.15),
    shear=10),
    transforms.RandomPerspective(distortion_scale=0.3, p=0.7),
    transforms.RandomGrayscale(p=0.1),
    transforms.GaussianBlur(kernel_size=3, sigma=(0.1, 2.0)),
    transforms.RandomErasing(p=0.3, scale=(0.02, 0.2), ratio=(0.3, 3.3))
])
```

```
# Resize dasar
basic_transform = transforms.Resize((384, 384))

# Fungsi simpan dan augmentasi
def save_transformed_image(img, path, augment_times=0):
    img_resized = basic_transform(img)
    img_resized.save(path)
    for i in range(augment_times):
        aug_img = transforms.ToPILImage()(augmentation_transform(transforms.ToTensor()(img_resized)))
        aug_img.save(path.replace('.jpg', f'_aug{i}.jpg'))

# Loop per kelas dan pembagian train/val/test
for class_name in os.listdir(input_root):
    ...
    # Hitung jumlah augmentasi agar mencapai target_train
    augment_times = math.ceil((target_train - current_train) / current_train)
    save_transformed_image(img, save_path, augment_times)
```

Gambar 19. Proses Pengolahan dan Augmentasi

(3) Output Preprocessing

```
# Menampilkan jumlah gambar akhir per split dan per kelas
for split in ['train', 'val', 'test']:
    print(f"\n📁 {split.upper()}:")
    for class_name in os.listdir(os.path.join(output_root, split)):
        count = len(os.listdir(os.path.join(output_root, split, class_name)))
        print(f"    - {class_name}: {count} gambar")
```

Gambar 20. Output Preprocessing

2. Pelatihan Model CNN

Pada tahap pelatihan model ini menggunakan algoritma Convolutional Neural Network dengan arsitektur Resnet34 yang dipilih karena kemampuannya dalam mengekstraksi fitur visual secara mendalam. dan akurasi tinggi pada klasifikasi citra yang dikombinasikan dengan confusion matrix untuk keperluan evaluasi model:

(1) Input (Data & Persiapan)

```
# Device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

```
# Transformasi data (resize, to tensor, normalisasi)
transform = transforms.Compose([
    transforms.Resize((384, 384)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406],
                          [0.229, 0.224, 0.225])
])

# Dataset dan DataLoader
data_dir = '/content/drive/MyDrive/Sidecam/data/data_augmented_fix8'
train_dataset = datasets.ImageFolder(os.path.join(data_dir, 'train'),
transform=transform)
val_dataset = datasets.ImageFolder(os.path.join(data_dir, 'val'),
transform=transform)
test_dataset = datasets.ImageFolder(os.path.join(data_dir, 'test'),
transform=transform)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=16, shuffle=False)
test_loader = DataLoader(test_dataset, batch_size=16, shuffle=False)
```

Gambar 21. Persiapan Data

(2) Proses Arsitektur Model & Training

```

# Arsitektur ResNet34 + Dropout
class ResNetWithDropout(nn.Module):
    def __init__(self, base_model, num_classes):
        super().__init__()
        self.features = nn.Sequential(*list(base_model.children())[:-1])
        self.dropout = nn.Dropout(0.5)
        self.classifier = nn.Linear(base_model.fc.in_features, num_classes)
    def forward(self, x):
        x = self.features(x)
        x = torch.flatten(x, 1)
        x = self.dropout(x)
        return self.classifier(x)

# Inisialisasi model, loss, optimizer, scheduler
base_model = models.resnet34(weights=models.ResNet34_Weights.DEFAULT)
model = ResNetWithDropout(base_model,
num_classes=len(train_dataset.classes)).to(device)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode='min', factor=0.5,
patience=2)

# Training loop (inti)
for epoch in range(epochs):
    model.train()
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
# Validasi dan simpan model terbaik tiap epoch
```

Gambar 22. Proses (Arsitektur Model & Training)

(3) Output – Menyimpan Model & Evaluasi

```

# Simpan model terbaik
output_dir = '/content/drive/MyDrive/Sidecam/Model'
os.makedirs(output_dir, exist_ok=True)
torch.save(model.state_dict(), os.path.join(output_dir, 'model_resnet34_ep20.pth'))

# Evaluasi pada test set
model.eval()
all_preds, all_labels = [], []
with torch.no_grad():
    for images, labels in test_loader:
        outputs = model(images.to(device))
        _, preds = torch.max(outputs, 1)
        all_preds.extend(preds.cpu().numpy())
        all_labels.extend(labels.numpy())

# Classification report dan confusion matrix
print(classification_report(all_labels, all_preds,
target_names=train_dataset.classes))
cm = confusion_matrix(all_labels, all_preds)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=train_dataset.classes,
            yticklabels=train_dataset.classes)
plt.show()

```

Gambar 23. Menyimpan Model

2. Pembahasan

a. Uji Hasil

Pada bagian pembahasan ini mencakup pengujian prototype sistem deteksi penyakit kulit campak berbasis Convolutional Neural Network (CNN). Pada pengujian ini terdiri dari dua tahap, yaitu uji coba oleh ahli (Blackbox Testing) dan uji coba oleh pengguna (Pertanyaan PSSUQ), dengan tujuan untuk menilai fungsionalitas, kelayakan, serta efektivitas sistem yang dikembangkan.

Pada pengujian hasil ini menggunakan rumus Field Observation yang ditunjukkan sebagai berikut:

$$\text{Efektifitas (\%)} = \frac{\text{Skor yang didapatkan}}{\text{Skor Maksimal}} \times 100\%$$

Berdasarkan hasil observasi, total skor yang diperoleh dari kuesioner Post-Study System Usability Questionnaire (PSSUQ) adalah 240. Nilai maksimum dihitung menggunakan rumus berikut:

Nilai maksimum = 5 (Skor maksimum per pertanyaan) x 16 (Jumlah pertanyaan) x 3 (jumlah responden)

Uji pengguna melibatkan tiga responden (dokter, suster, dan petugas laboratorium). Setiap responden memberikan penilaian terhadap 16 butir pertanyaan yang mencakup kepuasan secara keseluruhan (overall), kegunaan sistem (system usability), kualitas informasi (information quality), dan kualitas antarmuka (interface quality).

Tabel 4 Tabel Uji Pengguna PSSUQ

No	Responden	Pertanyaan															
		P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15	P 16
1	R1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	R2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	R3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Jumlah :												240					

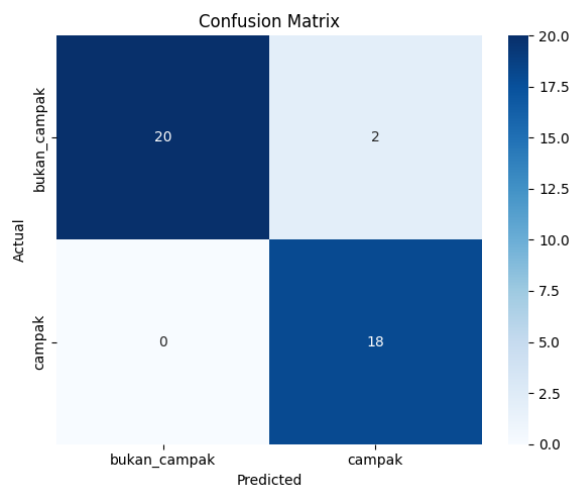
Sehingga hasil pengujian tersebut, diperoleh :

$$Efektifitas (\%) = \frac{240}{240} \times 100\% = 100\%$$

Dengan demikian, hasil perhitungan menunjukkan bahwa efektivitas dari prototipe aplikasi ini mencapai 100%, yang berarti sistem ini termasuk kategori “Sangat Efektif”.

b. Confusion Matrix

Selain uji fungsionalitas dan efektivitas sistem, adapun pengujian untuk menilai kemampuan model dalam mengklasifikasikan citra ruam campak dari algoritma Convolutinal Neural Network (CNN) yang dilakukan menggunakan Confusion Matrix. yang berfungsi untuk mengevaluasi performa model klasifikasi dengan membandingkan hasil prediksi dan label aktual. Melalui Confusion Matrix dapat diperoleh nilai akurasi, presisi, recall, dan F1-score yang menjadi indikator penting dalam mengukur sejauh mana sistem deteksi yang dikembangkan mampu mengenali citra campak secara tepat.



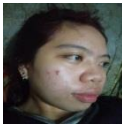
Gambar 24. Hasil Pengukuran Confusion Matrix

Tabel 5 Hasil Pengukuran Confusion Matrix

Accuracy	Precision	Recall	F1 Score
95%	90%	100%	94.7%

Tabel 6 Tabel Pengujian aktual dan test campak

No	Gambar penyakit Kulit	Aktual	Prediksi
1		Campak	Campak
...			
13		Campak	Campak

No	Gambar penyakit Kulit	Aktual	Prediksi
			
...			
25		Bukan campak	Campak
...			
31		Bukan campak	Campak
...			
40		Bukan Campak	Bukan campak

D. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan mengenai Deteksi Penyakit Kulit Dengan Pendekatan Machine Learning, maka dapat disimpulkan:

1. Penelitian ini berhasil mengembangkan sistem deteksi ruam campak berbasis Convolutional Neural Network (CNN) yang diimplementasikan dalam aplikasi web. Sistem yang dibangun dapat membantu proses pemeriksaan awal penyakit campak dalam mengenali ruam campak melalui gambar citra kulit;
2. Penerapan algoritma Convolutional Neural Network (CNN) pada sistem deteksi ruam campak melalui citra kulit membantu proses pemeriksaan awal sehingga dapat meningkatkan efektivitas dan konsistensi dalam penanganan pasien.
3. Sistem prototipe untuk deteksi penyakit kulit terutama ruam campak telah dibuat dengan baik dan sudah cukup mampu dalam melakukan deteksi penyakit kulit terutama mendeteksi ruam campak, dalam sistem ini menggunakan library flask python dan framework laravel;
4. Sistem mampu melakukan deteksi ruam pada campak dengan tingkat akurasi 95%, precision 90%, recall 100% dan F1-score 94.7%. Hasil ini menunjukkan bahwa tujuan dari penelitian untuk mendeteksi ruam pada campak melalui gambar secara akurat dan efektif sebagai pemeriksaan awal berhasil dicapai

Berdasarkan beberapa kesimpulan yang telah dijelaskan di atas, saran yang dapat disampaikan untuk pengembangan dan penyempurnaan penelitian selanjutnya antara lain sebagai berikut:

1. Untuk penelitian selanjutnya, disarankan menggunakan jumlah dataset yang lebih banyak dan bervariasi agar model dapat mengenali pola ruam campak secara lebih akurat dan generalisasi hasilnya meningkat.
2. Untuk pengembangan selanjutnya disarankan menggunakan arsitektur atau metode lain yang lebih sesuai untuk dataset terbatas, misalnya Generative Adversarial Network (GAN). Dengan pendekatan

ini, data yang terbatas dapat diperbanyak melalui proses augmentasi atau sintesis citra baru, sehingga model dapat belajar lebih baik tanpa harus memiliki jumlah data yang sangat besar.

3. Untuk pengembangan selanjutnya disarankan menggunakan penyakit kulit lain yang memiliki kemiripan gejala, seperti cacar air dan skabies (scabies), sehingga sistem yang dikembangkan menjadi lebih serbaguna.

E. DAFTAR PUSTAKA

- [1] R. M. Aritonang, M. Y. Florestiyanto, dan B. Yuwono, “Human Skin Disease Detection using Convolutional Neural Network Method with Hyperparameter Tuning to Determine the Best Parameter Combination,” *Jurnal Informatika dan Teknologi Informasi*, vol. 20, no. 2, hlm. 215–225, 2023, doi: 10.31515/telematika.v20i2.9161.
- [2] Y. Widya, A. Rustam, C. Chazar, dan M. A. Ramdhani, “Aplikasi Diagnosa Penyakit Kulit Menggunakan dengan Menggunakan Metode Convolutional Neural Networks,” *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, vol. 15, no. 2, hlm. 208–224, Nov 2023.
- [3] Sugiyono, *Metode Penelitian Pendidikan (Kuantitatif, Kualitatif, kombinasi, R&D dan Penelitian Pendidikan)*, Edisi ke 3. Bandung: ALFABETA, 2019.
- [4] Arikunto, Suharsimi. *Prosedur Penelitian: Suatu Pendekatan Praktik*. Rineka Cipta, 2009. Pressman, Roger, and Bruce Maxim. *Software Engineering: A Practitioner’s Approach*. McGraw-Hill, 2019.