

# Aplikasi Berbasis Android untuk Mendeteksi Kulit Kucing Berdasarkan Model CNN

Riyandi Aditya Fitrah  
Universitas Buana Perjuangan Karawang  
Karawang, Indonesia  
if19.riyandifitrah@mhs.ubpkarawang.ac.id

Anis Fitri Nur Masruriyah  
Universitas Buana Perjuangan Karawang  
Karawang, Indonesia  
anis.masruriyah@ubpkarawang.ac.id

Ayu Ratna Juwita  
Universitas Buana Perjuangan Karawang  
Karawang, Indonesia  
ayurj@ubpkarawang.ac.id

**Abstract** — Penyakit kulit kucing mendatangkan dampak negatif terhadap pemilik kucing dan hewan tersebut. Penyakit kulit kucing seperti *scabies* atau kudis, dan *ringworm* bersifat menular melalui sentuhan langsung yang terjangkit penyakit tersebut. Tungau telinga pada kucing bertempat di bawah rongga telinga kucing. Kucing yang terjangkit penyakit kulit dapat terkena kerusakan pada tubuhnya. Penelitian ini membantu menunjukkan deteksi penyakit kulit yang diderita oleh kucing melalui aplikasi *mobile* berbasis android. Aplikasi ini dibangun untuk mengetahui jenis penyakit kulit kucing pada kucing seperti kudis, *ringworm*, dan tungau. Pada penelitian ini, merancang suatu model deteksi objek menggunakan metode *deep learning* algoritma CNN dengan *tensorflow lite*. Arsitektur yang digunakan *MobileNetV2 FPN Lite* untuk memproses *training* model deteksi objek dengan *dataset* berjumlah besar, sehingga dapat diterapkan pada aplikasi *mobile* berbasis android. Berdasarkan penelitian, hasil perancangan model mendapatkan nilai mAP sebesar 42%, dan nilai mAR 23%. Hasil evaluasi sistem dan validasi pakar diperoleh dengan nilai sebesar 73%.

**Kata kunci** — algoritma CNN, *tensorflow lite*, *deep learning*

## I. PENDAHULUAN

Berdasarkan Nurajizah dan Saputra [1] kucing memiliki peran sebagai sahabat pemiliknya. Perilaku kucing sebagai hewan peliharaan manusia selalu berada di rumah dan terkadang juga di luar rumah. Kebiasaan tersebut dapat menumbuhkan suatu penyakit yang bisa dilihat secara kasat mata pada tubuh kucing [1]. Memelihara kucing tidak hanya melalui makanan, melainkan juga perlu diperhatikan kesehatannya [2]. Hal tersebut dapat menimbulkan kebingungan pemilik kucing dalam mengetahui penyakit yang diderita oleh kucingnya [1]. Kebingungan pemilik kucing disebabkan kurangnya pengetahuan pemilik kucing tentang penyakit hewan tersebut [2]. Hewan peliharaan yang menderita penyakit dapat berdampak negatif pada hewan peliharaannya dan juga pemiliknya [3]. Ketidaktahuan pemilik kucing terkait informasi jenis penyakit kucing yang dilihat secara kasat mata, dan sulitnya menemukan seorang pakar dalam keadaan mendesak, serta mahal biaya juga menjadi alasan pemilik hewan tidak membawa peliharaannya ke dokter hewan [1].

Pada penelitian Kirana [4] melakukan penelitian pengenalan pola aksara sunda menggunakan algoritma *Convolutional Neural Network*. CNN ini terbukti mampu mengklasifikasi dalam mengenali aksara sunda dengan akurasi 100.0% berdasarkan citra dari tulisan tangan responden yang diambil menggunakan pemindai mendapatkan akurasi 86.5%. Penelitian lainnya membuat sistem deteksi wajah keamanan pintu menggunakan metode *convolutional neural network* berbasis Arduino [5]. Pada penelitian ini, CNN digunakan untuk mendeteksi wajah seseorang menggunakan kamera atau *webcam* dengan perolehan nilai akurasi sebesar 76,6%. CNN berbasis arduino ini dapat meningkatkan keamanan pintu agar tidak terjadi pembobolan atau pencurian sistem yang diharapkan.

Penelitian oleh Maulana [6] tentang penerapan *convolutional neural network* pada timbangan pintar sayuran menggunakan Raspberry Pi. Pada penelitian ini, *Convolutional Neural Network* (CNN) digunakan untuk mendeteksi objek sayuran, dan menggunakan Raspberry Pi untuk melakukan pembacaan nilai berat sayuran. Hasil penelitian ini metode CNN memiliki akurasi hingga 90%. Pada penelitian Afifur [7] tentang klasifikasi sampah logam dan plastik berbasis Raspberry Pi dengan metode *convolutional neural network*. Pada penelitian ini, CNN teruji memiliki nilai akurasi 98%, dan melalui hasil percobaan rata-rata akurasi pengujian model 81,387%. CNN juga mampu mengatasi permasalahan pengelolaan sampah yang kurang peduli.

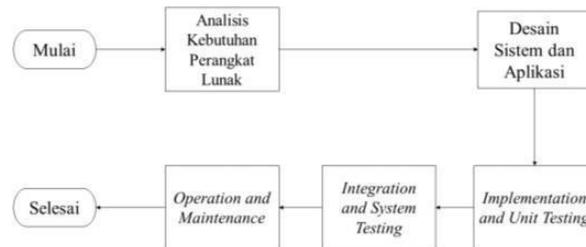
Penelitian yang dilakukan oleh Mustabinnur [8], tentang sistem kendali suhu dan pemantauan kelembapan udara berbasis android dengan sensor DHT11. Penelitian ini menggunakan aplikasi berbasis android dengan sensor DHT11 yang dirancang menggunakan arduino uno yang terbukti membantu kegiatan sehari-hari manusia di dalam ruangan tertutup. Penelitian oleh Juwita [9], tentang identifikasi citra batik dengan metode *convolutional neural network*. Pada penelitian ini, CNN digunakan untuk mengklasifikasikan citra batik dengan nilai tingkat akurasi 73,33%. CNN ini terbukti dapat menyelesaikan permasalahan dalam mengklasifikasikan citra batik berdasarkan bentuk dan motifnya. Penelitian yang dilakukan oleh Baihaqi [10] tentang penerapan algoritma CNN untuk deteksi jenis padi menggunakan Yolo V3. Pada penelitian ini, CNN mampu mendukung pekerjaan manusia pada sektor pertanian khususnya kegiatan-kegiatan produksi usaha tani. Hasil tingkat akurasi menggunakan *convolutional neural network* mencapai 100%.

Berdasarkan permasalahan terkait penyakit kucing dan beberapa uraian solusi penelitian yang telah dilakukan, penelitian ini membuat sistem untuk mendeteksi penyakit kulit yang diderita oleh kucing. Penelitian ini membangun aplikasi berbasis android yang belum dilakukan oleh penelitian terdahulu, dengan memanfaatkan framework *tensorflow* menggunakan metode *deep learning* dengan algoritma CNN arsitektur *MobileNet*. Pengguna sistem operasi android juga telah banyak digunakan karena bersifat *user friendly* atau mudah dijalankan [11]. Sistem ini memudahkan pemilik kucing agar dapat memahami, mengetahui, dan mengerti jenis penyakit pada kucing yang dipelihara berdasarkan *input* citra yang dilakukan pemilik kucing melalui aplikasi *mobile* berbasis android.

## II. METODE PENELITIAN

### A. Prosedur Penelitian

*System Development Life Cycle* (SDLC) merupakan metode pengembangan yang memelihara, dan menghasilkan sistem yang standar [12]. Prosedur penelitian menggunakan metode SDLLC yang dilakukan secara bertahap untuk memberikan hasil dari suatu permasalahan. Menurut Sommerville, mendefinisikan model air *waterfall* sebagai tahapan terpenting dalam pembangunan suatu sistem informasi [13]. Prosedur penelitian terdapat analisis kebutuhan perangkat lunak, desain sistem dan aplikasi, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*. Alur dari prosedur penelitian ditunjukkan pada Gambar 1.



Gambar 1 Prosedur Penelitian

#### 1. Analisis Kebutuhan Perangkat Lunak

Penelitian dimulai dari tahapan analisis kebutuhan perangkat lunak dengan melakukan identifikasi perangkat lunak dan kebutuhan sistem yang dibuat [14]. Tahapan ini juga memahami konsep sistem aplikasi berbasis android untuk mendeteksi penyakit kulit pada kucing melalui observasi dan juga wawancara terhadap seorang pakar dokter hewan Drh. Fyar F. H TA. Dokter pakar hewan tersebut memiliki pengetahuan tentang penyakit kulit pada kucing. Analisis kebutuhan diselesaikan dengan melakukan analisis citra penyakit kulit kucing untuk melakukan deteksi objek citra penyakit kulit kucing. Parameter yang digunakan *scabies* atau kudis, tungau, dan *ringworm*.

#### 2. Desain Sistem dan Aplikasi

Prosedur penelitian selanjutnya desain sistem dan aplikasi, tahap desain meliputi desain sistem pada aplikasi. Pada desain sistem terdapat fungsi desain dan operasi yang dijelaskan secara rinci, termasuk tata letak tampilan pada aplikasi berbasis android, dan aktivitas *user*.

#### 3. *Implementation and Unit Testing*

Selanjutnya tahapan *implementation and unit testing* model CNN yang telah dibangun oleh Fitrah [9] pada penelitian sebelumnya. Perolehan data didapat dengan data sekunder melalui sumber lain yang sudah ada diinternet. Data citra yang dapat dikumpulkan sebesar 43 citra penyakit kulit kucing. Akuisisi citra yang digunakan memiliki kualitas masing-masing 80% untuk data *train* dan 20% untuk data *test*. Pelabelan citra menggunakan *tools labeling* yang dijalankan menggunakan *command prompt*. Pembuatan *labelmap* disimpan pada format pbtx sebagai catatan biner untuk menyimpan informasi. Pada tensorflow memerlukan tensorflow *record* sebagai generalisasi data sebelum melakukan pelatihan model dengan mengatur konfigurasi *pipeline* sebagai kebutuhan pelatihan model deteksi objek. Penerapan model meliputi alur *input* citra yang dimasukkan, proses ekstraksi penyakit kulit kucing, klasifikasi CNN, dan deteksi penyakit kulit kucing.

#### 4. *Integration and System Testing*

Kemudian tahapan *integration and system testing* dengan melakukan uji coba pada sistem untuk memastikan fungsionalitas pada aplikasi.

#### 5. *Operation and Maintenance*

Tahap terakhir ini meliputi evaluasi seluruh program dengan memastikan data-data pada pada aplikasi berbasis android bekerja sesuai rencana dan fungsi sistem yang belum jalan sebagaimana mestinya. Maintenance untuk pemeliharaan pada sistem aplikasi berbasis android, dan juga pengembang perlu melakukan pemantauan jika perlu meningkatkan layanan sistem yang dikelola.

### B. Pengujian

Pengujian dilakukan untuk menguji model deteksi training model tensorflow, dan menguji sistem aplikasi mobile. Pengujian model deteksi objek untuk menguji hasil model training menggunakan tensorboard, dan pengujian sistem menggunakan *black box testing* teknik *equivalence partitioning* dan *white box testing* dengan teknik *cyclomatic*. Menentukan kompleksitas logika program menggunakan *cyclomatic* dari node grafik dengan rumus perhitungan persamaan 1.

$$V(G) = E - N + 2 \quad (1)$$

### C. Evaluasi Hasil

Evaluasi hasil penelitian menggunakan parameter yang dikenal *performance metric* terdiri dari TP, FP, FN, dan TN. *Confusion matrix* digunakan untuk mengevaluasi model pendeteksian objek yang terdiri dari *precision*, *recall*, dan *mean average precision* (mAP). Gambar 2 menunjukkan *confusion matrix*.

		Actually	
		Positif (1)	Negatif (0)
Predicted	Positif (1)	True Positive	False Positive
	Negatif (0)	False Negative	True Negative

Gambar 2 *Confusion Matrix*

*Precision* mendeskripsikan nilai presisi yang didapatkan dari pembagian jumlah total contoh positif, saat benar diklasifikasikan melalui model [15]. Rumus *precision* yang digunakan untuk kalkulasi nilai *precision* ditulis melalui persamaan 2.

$$Precision = \frac{TP (True Positive)}{TP (True Positive) + FP (False Positive)} \tag{2}$$

*Recall* mendeskripsikan kemampuan model deteksi objek. *Recall* mendefinisikan nilai rasio dari jumlah total contoh positif yang diklasifikasikan dengan benar [15]. Nilai *recall* yang baik menampilkan klasifikasi dikenali dengan benar [15]. Rumus *recall* rumus yang digunakan untuk kalkulasi nilai *recall* ditunjukkan pada persamaan 3.

$$Recall = \frac{TP (True Positive)}{TP (True Positive) + FN (False Negative)} \tag{3}$$

Nilai mAP dan mAR menunjukkan tingkat akurasi deteksi objek dan posisinya. Nilai parameter mAP menjadi peran utama untuk mengukur akurasi suatu model yang akan diaplikasikan, dan seberapa baik nya model deteksi objek mengatasi objek tanpa kesalahan. Persamaan menghitung mAP dapat ditulis melalui persamaan 4.

$$mAP@ a = \frac{1}{n} \sum_{i=1}^n AP_i \tag{4}$$

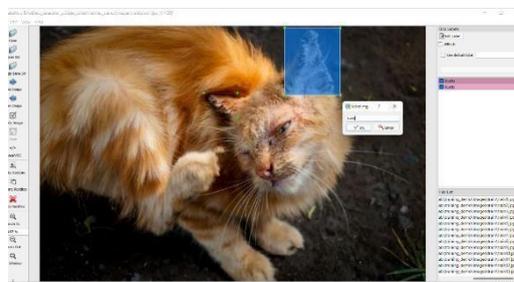
### III. HASIL DAN PEMBAHASAN

#### A. Akuisisi Citra

Total 43 citra *dataset* yang berhasil didapatkan dan dikelompokkan ke dalam folder *images*. Sebanyak 34 atau 80% dari jumlah *dataset* pada folder *train* penyakit kulit kucing yang terdiri dari kudis, *ringworm*, dan tungau. Sedangkan folder *test* sebanyak 9 atau 20% dari jumlah *dataset* penyakit kulit kucing terdiri dari kudis, *ringworm*, dan tungau.

#### B. Preprocessing Citra

Pelabelan ini dilakukan bersama dengan pakar agar model deteksi dapat belajar sesuai dengan ciri penyakit kulit kucing yang terjangkau. Proses pelabelan citra menggunakan *labelimg* yang dipanggil menggunakan CMD. Proses pelabelan citra dapat dilakukan dengan memberikan *bounding box* terhadap objek yang dideteksi. Proses pelabelan dan penamaan citra dapat ditunjukkan pada Gambar 3.



Gambar 3 Pelabelan Citra

#### C. Membuat Labelmap

File labelmap berisi 3 item dengan id 1 sampai dengan 3 dan name yang berisi nama objek “kudis”, “ringworm”, dan “tungau”.

File labelmap format pbtX digunakan saat pelatihan model deteksi objek untuk mengidentifikasi kelas label yang digunakan.

D. TF Record

Dataset pada folder train, test dan juga file labelmap yang sudah disiapkan, kemudian diubah ke dalam format record. Hasil dari program tersebut berupa berkas yang dinamakan test.record yang dapat ditunjukkan pada Tabel 1.

Tabel 1 Tensorflow Record

11101111 10111111 10111101 00001010 11101111 10111111 10111101 00100010 11101111 10111111 10111101 01111101 01101111 00111001 11101111
10111111 10111101 11101111 10111111 10111101 11101111 10111111 10111101 00110100 01111100 11101111 10111111 10111101 11101111 10111111
10111101 11001011 10101101 11101111 10111111 10111101 11101111 10111111 10111101 01010000 11101111 10111111 10111101 00010110 11101111
10111111 10111101 00010001 00100010 01001001 00010010 11101111 10111111 10111101 11101111 10111111 10111101 01110101 01101101 11001010
10000100 11101111 10111111 10111101 10000100 11101111 10111111 10111101 11101111 10111111...

Pada Tabel 1 memiliki nilai komputasi komputer dengan catatan biner yang memiliki angka 1 dan 0, tensorflow record menyimpan data tersebut sebagai menyimpan data saat melakukan proses pelatihan dengan menggunakan nilai 0 dan 1. Representasi nilai 0 memberikan sinyal “tidak ada” pada komputer, sementara nilai 1 memberikan sinyal “ada” pada komputer.

E. Konfigurasi Pipeline

Penelitian ini menggunakan model arsitektur MobileNetV2 FPN Lite 320x320, berkas pipeline berada pada berkas model arsitektur yang digunakan. File pipeline.config memiliki struktur yang harus dikonfigurasi sebelum masuk ke dalam penelitian.

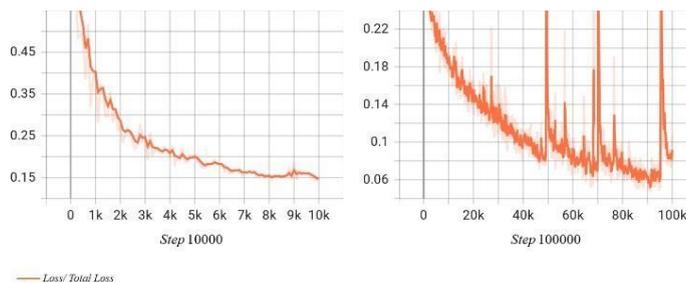
F. Training Model

Proses pelatihan berhenti ketika jumlah steps yang terdapat di dalam berkas pipeline.config. Selama proses pelatihan berjalan, dapat dipantau dari log yang ditampilkan google colab. Normalnya, nilai total loss semakin mengecil dengan bertambahnya step yang dilewati. Hasil dari percobaan pelatihan dengan jumlah step yang berbeda dapat dilihat pada Tabel 2.

Tabel 2 Tabel Pelatihan Model

Total Step	Hasil Training		
	Waktu Pelatihan	mAP	mAR
10000	16 menit	0.420080	0.230000
100000	4 jam 52 menit	0.205778	0.216667

Berdasarkan Tabel 2 telah dilakukan pengujian model deteksi objek terhadap total step yang berbeda. Normalnya, bertambah step nilai mAP dan mAR semakin meningkat, namun berdasarkan tabel tersebut terjadi penurunan. Pengujian model ini menunjukkan model pendeteksian objek mengalami overfitting. Hal ini terjadi karena dataset yang diperoleh terbatas dan jumlah yang sedikit. Model dengan step 10000 memiliki nilai mAP dan nilai mAR yang tinggi. Sedangkan model step 100000 memiliki mAP dan nilai mAR yang rendah. Sehingga model step 10000 menunjukkan performa yang bagus untuk mendeteksi karena memiliki nilai mAP dan nilai mAR. Perolehan nilai total loss model deteksi objek berdasarkan jumlah step dapat ditunjukkan pada Gambar 4.

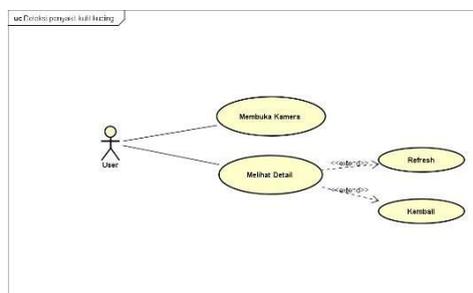


Gambar 4 Grafik Total Loss Berdasarkan Step

Berdasarkan Gambar 4 menunjukkan grafik total loss berdasarkan jumlah step 10000, dan 100000. Garis oranye menunjukkan nilai total loss, jumlah total loss ketika step 10000 bernilai 1.31. Sedangkan ketika step 100000 bernilai 2.68.

G. Desain Sistem dan Aplikasi

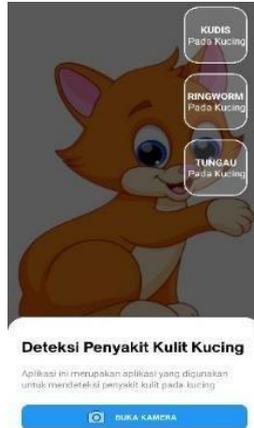
1. Use Case Diagram



Gambar 5 Use Case Diagram

Pada Gambar 5 use case diagram menunjukkan sebuah aktivitas dalam sistem yang di mana terdapat aktor yang berperan sebagai user. Aktivitas user yang dilakukan terdapat membuka kamera, dan melihat detail. Ketika user melakukan aktivitas detail, maka di dalam aktivitas detail terdapat fungsi refresh, dan kembali.

2. Hasil Desain



Gambar 6 Hasil Desain Tampilan Menu

Berdasarkan Gambar 6 hasil desain tampilan menu utama terdapat tombol buka kamera yang berfungsi untuk menampilkan kamera *mobile*.



Gambar 7 Hasil Desain Tampilan Kamera

Gambar 7 hasil desain tampilan kamera menampilkan kamera pada sehingga *user* dapat mendeteksi penyakit kulit kucing. Terdapat dua tombol detail dan cek yang di mana pada tombol detail untuk menampilkan halaman detail. Tombol cek untuk melihat informasi nilai atau angka model deteksi saat melakukan proses ekstraksi yang dimunculkan pada cek sumber.



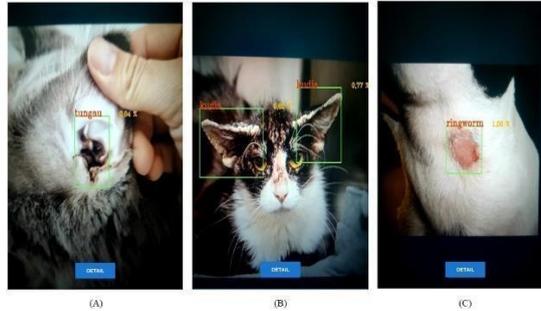
Gambar 8 Hasil Desain Tampilan Detail

Berdasarkan Gambar 8 hasil desain tampilan detail menampilkan informasi informasi terkait deteksi objek. Informasi tersebut seperti nilai *inference*, *input size*, nama model yang digunakan, dan informasi perangkat.

H. Implementasi Model

Model deteksi objek yang sudah dikonversi diterapkan ke dalam *mobile* dengan memberikan parameter nama folder, nama label, dan mengisi *input size*. File model yang digunakan dengan *step* 10000 karena memiliki nilai mAP tinggi dan mAR yang cukup tinggi. Nama label berupa file txt yang disamakan dengan kelas objek yang dideteksi. Objek penyakit kulit kucing yang terdeteksi

ditunjukkan melalui pelabelan bentuk kotak warna hijau. Kotak berwarna hijau menandakan lokasi dari objek yang terdeteksi. Sementara pada atas kotak terdapat teks yang menunjukkan kelas label yang dideteksi. Model deteksi objek mengembalikan nilai probabilitasnya berupa nilai persentase. Nilai persentase digunakan untuk menentukan *threshold* objek penyakit kulit kucing ditampilkan atau tidak. Model hasil penelitian sudah dapat digunakan pada aplikasi deteksi penyakit kulit kucing kucing seperti pada Gambar 9.



Gambar 9 (A) Deteksi Penyakit Tungau (B) Deteksi Penyakit Kudis (C) Deteksi Penyakit Ringworm

Pada Gambar (9A) menunjukkan hasil deteksi objek penyakit kulit kucing tungau, Gambar (9B) mendeteksi objek penyakit kulit kucing kudis, dan Gambar (9C) mendeteksi penyakit kulit kucing *ringworm*. Nilai tertinggi akurasi saat mendeteksi memperoleh 1.00% ketika mendeteksi *ringworm*, sedangkan kudis 0.83%, dan tungau 0.84%. Pada halaman tampilan detail menunjukkan hasil nilai yang ditunjukkan pada saat melakukan deteksi penyakit kulit kucing melalui tampilan kamera. Nilai akurasi persentase ditampilkan dengan cara menekan tombol *refresh* saat menangkap objek. *Inference* berupa nilai saat proses eksekusi model tensorflow lite untuk membuat prediksi berdasarkan data masukan. Nilai *input size* menunjukkan angka *height* dan *width* yang digunakan. Tampilah halaman hasil detail deteksi penyakit kulit kucing dapat ditunjukkan pada Gambar 10.



Gambar 10 Hasil Detail

### I. White Box Testing

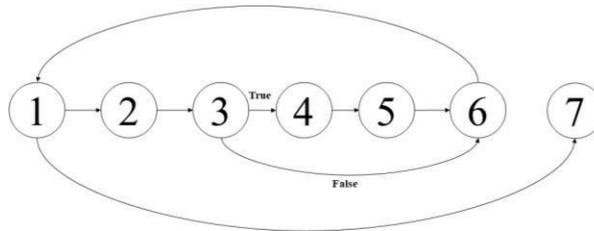
Pengujian *white box* menguji salah satu kode program yang berfungsi mendeteksi objek penyakit kulit kucing digunakan untuk menerima masukan kamera yang berfungsi untuk mendeteksi objek. Ketika mendeteksi, tampilan pada kamera terdapat kotak-kotak berwarna hijau dan tulisan teks diatas sebagai nama kelas objek. Kode program dapat dilihat pada Tabel 3.

Tabel 3 Kode Program Fungsi Deteksi Objek Penyakit Kucing

No	Kode Program
1	for (int i=0;i<10;i++){
2	class_value=(float) Array.get(Array.get(Object_class,0),i); score_value= (float) Array.get(Array.get(score,0),i);  xscore_value= String.format("%.2f",score_value);
3	if (score_value>0.5){
4	Object box1=Array.get(Array.get(value,0),i); float top = (float) Array.get(box1,0)*height; float left= (float) Array.get(box1,1)*width; float bottom =(float) Array.get(box1,2)*height; float right =(float) Array.get(box1,3)*width;

5	<code>Imgproc.rectangle(rotated_mat_image,new Point(left,top),new Point(right,bottom),new Scalar(100,255,100),2);</code> <code>Imgproc.putText(rotated_mat_image,labelList.get((int)class_value) , new Point(left,top),3,1,new Scalar(250,60,0),2);</code> <code>Imgproc.putText(rotated_mat_image," "+xscore_value+" %", new Point(right,top),3,0.7,new Scalar(250,200,0),2,Imgproc.LINE_4);</code>
6	<code>} end kondisi if</code>
7	<code>} end kondisi for</code>

1. Flow Graph



Gambar 11 Flow Graph

2. Complexity

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 7 + 2 \\
 &= 3
 \end{aligned}$$

Sehingga jumlah path sebanyak 3 :

- Jalur 1 : 1-7
- Jalur 2 : 1-2-3-4-5-6-1-7
- Jalur 3 : 1-2-3-6-1-7

3. Path

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 2 + 1 \\
 &= 3
 \end{aligned}$$

### J. Blackbox Testing

Pengujian dilakukan oleh dokter hewan Drh. Fyar F. H TA yang memiliki pengetahuan tentang penyakit kulit pada kucing. Pengujian *blackbox testing* berdasarkan menu aktivitas yang terdapat pada aplikasi dilakukan dengan menguji beberapa tombol dan mendeteksi objek. Pada tampilan utama terdapat tombol buka kamera, tampilan kamera terdapat tombol detail, tampilan detail terdapat tombol *refresh*, dan tombol kembali. Tabel 4 menunjukkan pengujian *blackbox testing*.

Tabel 4 Tabel Pengujian *Blackbox Testing*

No	Deskripsi Pengujian	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1.	Menekan tombol kamera	Sistem menampilkan kamera dan mendeteksi objek penyakit kulit kucing	Sesuai	Valid
2.	Menekan tombol detail	Sistem menampilkan halaman detail	Sesuai	Valid
3.	Menekan tombol <i>refresh</i>	Sistem menampilkan informasi detail nilai akurasi, <i>inference</i> , <i>input size</i> , dan informasi perangkat	Sesuai	Valid
4.	Menekan tombol kembali	Sistem menampilkan halaman kamera	Sesuai	Valid

### K. Evaluasi Hasil

Model deteksi objek metode *deep learning* algoritma CNN membutuhkan *dataset* yang sudah terdapat pelabelan berupa *bounding box* dan kelas objeknya. Berdasarkan *dataset* yang dikumpulkan masih terbatas sehingga perlu diperbanyak untuk meningkat performa model yang lebih baik. Jumlah *dataset* yang dikumpulkan sebanyak 43 citra penyakit kulit kucing, namun data tersebut masih kurang. Hasil model deteksi objek metode *deep learning* algoritma CNN dengan arsitektur *MobileNetV2 FPN Lite 320x320* berhasil diterapkan. Hal tersebut ditunjukkan pada hasil mAP tinggi dengan *step* 10000 bernilai 42% dan berhenti meningkat ketika *step* 100000 bernilai 20%. Proses implementasi model deteksi objek pada aplikasi *mobile* dapat berjalan dengan baik. Melalui uji coba menggunakan aplikasi *mobile*, proses mendeteksi objek penyakit kulit kucing membutuhkan *inference* yang lebih sedikit.

Evaluasi sistem dilakukan bersama dengan pakar dokter hewan Drh. Fyar F. H TA yang memiliki pengetahuan tentang penyakit kulit pada kucing. Melalui penelitian ini, evaluasi hasil sistem menggunakan sebuah *input* citra kucing yang terjangkit penyakit kulit seperti kudis, *ringworm*, dan tungau. Saat sistem melakukan deteksi terhadap *input* citra, maka hasil sistem memberikan *output* hasil citra dan memberikan keterangan pada sistem (lampiran). Pada penelitian ini, proses validasi hasil dilakukan dengan mengetahui hasil penilaian melalui validasi pakar. Terdapat 2 data Setuju (S) dan Tidak Setuju (TS) melalui validasi pakar saat menyesuaikan hasil dari *output* dan keterangan pada hasil evaluasi sistem. Tabel 4 menunjukkan validasi hasil pakar.

Tabel 5 Tabel Validasi Pakar

No	Nama Penyakit	Hasil Pakar	Keterangan
1	Kudis	Mendeteksi penyakit kudis	Sesuai, mendeteksi penyakit kudis
2	Kudis	Mendeteksi penyakit kudis	Sesuai, mendeteksi penyakit kudis
3	Kudis	Mendeteksi penyakit <i>ringworm</i>	Tidak Sesuai, seharusnya penyakit kudis
4	Kudis	Mendeteksi penyakit kudis	Sesuai, mendeteksi penyakit kudis
5	Kudis	Mendeteksi penyakit kudis	Sesuai, mendeteksi penyakit kudis
6	<i>Ringworm</i>	Mendeteksi penyakit <i>ringworm</i>	Sesuai, mendeteksi penyakit <i>ringworm</i>
7	<i>Ringworm</i>	Mendeteksi penyakit <i>ringworm</i>	Sesuai, mendeteksi penyakit <i>ringworm</i>
8	<i>Ringworm</i>	Mendeteksi penyakit <i>ringworm</i>	Sesuai, mendeteksi penyakit <i>ringworm</i>
9	<i>Ringworm</i>	Mendeteksi penyakit <i>ringworm</i>	Sesuai, mendeteksi penyakit <i>ringworm</i>
10	<i>Ringworm</i>	Mendeteksi penyakit <i>ringworm</i>	Sesuai, mendeteksi penyakit <i>ringworm</i>

11	Tungau	Mendeteksi penyakit tungau	Sesuai, mendeteksi penyakit tungau
12	Tungau	Mendeteksi penyakit <i>ringworm</i>	Tidak Sesuai, seharusnya penyakit tungau
13	Tungau	Mendeteksi penyakit tungau	Sesuai, mendeteksi penyakit tungau
14	Tungau	Mendeteksi penyakit <i>ringworm</i>	Tidak Sesuai, seharusnya penyakit tungau
15	Tungau	Mendeteksi penyakit <i>ringworm</i>	Tidak Sesuai, seharusnya penyakit tungau

$$\text{Validasi Pakar} = \frac{S}{S + \bar{S}} \times 100\%$$

$$\text{Validasi Pakar} = \frac{11}{11 + 4} \times 100\%$$

$$\text{Validasi Pakar} = \frac{11}{15} \times 100\%$$

$$\text{Validasi Pakar} = 73\%$$

Berdasarkan perhitungan yang dilakukan maka hasil validasi pakar pada penelitian ini sebesar 73%.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, model CNN yang mendeteksi objek penyakit kulit kucing memerlukan dataset yang sudah diberi pelabelan citra berupa bounding box dengan nama kelas objeknya. Pembangunan aplikasi berbasis android dan penerapan model CNN tensorflow lite pada aplikasi mobile berbasis android berhasil diterapkan untuk mendeteksi penyakit kulit kucing. Hasil evaluasi model CNN menggunakan tensorboard menunjukkan nilai mAP dari model sebesar 42% dan mAR sebesar 23% saat step 10000. Hasil evaluasi sistem dan validasi pakar berdasarkan perhitungan 2 data Setuju (S) dan Tidak Setuju (TS) sebesar 73%.

#### PENGAKUAN

Naskah Ilmiah ini adalah sebagian dari penelitian Tugas Akhir milik Riyandi Aditya Fitrah, dengan judul “Deteksi Penyakit Kulit Kucing Menggunakan Algoritma CNN Dengan Tensorflow Lite Berbasis Android” yang dibimbing oleh Anis Fitri Nur Masruriyah dan Ayu Ratna Juwita.

#### DAFTAR PUSTAKA

- [1] S. Nurajizah and M. Saputra, “SISTEM PAKAR BERBASIS ANDROID UNTUK DIAGNOSA PENYAKIT,” vol. 14, no. 1, pp. 7–14, 2018.
- [2] sudirman Sudirman, A. Dasan, and A. Fortuna, “Deteksi penyakit kulit pada kucing dengan metode forward chaining berbasis android,” 2022.
- [3] H. Patria, A. Anton, and P. Astuti, “Sistem Pakar Menggunakan Metode Certainty Factor Untuk Mendiagnosa Penyakit Kulit Pada Hewan Kucing,” *Simpatik J. Sist. Inf. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2021, [Online]. Available: <http://103.75.24.116/index.php/simpatik/article/view/70>
- [4] A. Kirana, H. Hikmayanti, and J. Indra, “Pengenalan Pola Aksara Sunda dengan Metode Convolutional Neural Network,” *Sci. Student J. Information*,

*Technol. Sci.*, vol. 1, no. 2, pp. 95–100, 2020.

- [5] Kiki Wahyuddin, D. Wahiddin, and D. S. Kusumaningrum, “Sistem Deteksi Wajah Keamanan Pintu Menggunakan Metode Convolutional Neural Network ( CNN ) Berbasis Arduino,” vol. IV, pp. 15–23, 2023.
- [6] F. Maulana, J. Indra, S. A. P. Lestari, and Universitas, “Penerapan Convolutional Neural Network pada Timbangan Pintar Sayuran Menggunakan Raspberry Pi,” vol. II, pp. 1–9, 2021.
- [7] A. A. Rahman, U. B. Perjuangan, A. Fauzi, U. B. Perjuangan, J. Indra, and U. B. Perjuangan, “Klasifikasi Sampah Logam Dan Plastik Berbasis Raspberry Pi Dengan Metode Convolution Neural Network,” vol. IV, pp. 1–6, 2023.
- [8] T. R. Mustabinnur, Sutan Faisal, “Sistem Kendali Suhu Dan Pemantauan Kelembaban Udara Berbasis Android Dengan Sensor Dht11,” *Sci. Student J. Information, Technol. Sci.*, vol. 1, no. 2, pp. 108–115, 2020.
- [9] R. A. Fitrah, A. F. N. Masruriyah, and A. R. Juwita, “Deteksi Penyakit Kulit Kucing Menggunakan Algoritma CNN Dengan Tensorflow Lite Berbasis Android,” Karawang, Indonesia, 2023.
- [10] K. Baihaqi and Y. Cahyana, “Application of Convolution Neural Network Algorithm for Rice Type Detection Using Yolo v3,” *Systematics*, vol. 3, no. 2, pp. 272–280, 2021.
- [11] A. Ishak and N. Pakaya, “Sistem Informasi Wedding Organizer Berbasis Android,” *Jambura J. Informatics*, vol. 3, no. 2, pp. 97–108, 2021, doi: 10.37905/jji.v3i2.11746.
- [12] A. Y. Permana and P. Romadlon, “PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE,” *Биохимия*, vol. 84, no. 10, pp. 1511–1518, 2019, doi: 10.1134/s0320972519100129.
- [13] T. How, “Metode SDLC,” *kumparan*, 2021. <https://kumparan.com/how-to-teknologi/metode-sdlc-definisi-manfaat-dan-jenis-modelnya-1zqTZD1AzKH>
- [14] N. Nahrudin, Y. Cahyana, and K. A. Baihaki, “Deteksi Bentuk Candi Jiwa dan Candi Blandongan,” vol. IV, pp. 24–30, 2023.
- [15] H. Harani, N. Prianto, and M. Cahyo Hasanah, “Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python,” *J. Tek. Inform.*, vol. 11, no. 3, pp. 47–53, 2019.