

APLIKASI MOBILE DETEKSI PENYAKIT DEMODEKOSIS PADA ANJING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK

Ardy Wijaya¹, Duman Care Khrisne², Lie Jasa³

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

Email: hardywijaya13@gmail.com¹, duman@unud.ac.id²,

liejasa@gmail.com³

ABSTRAK

Demodex adalah penyakit kulit pada anjing yang disebabkan oleh jenis tungau protozoa demodex follicularumvar canis atau demodex canis. Tungau ini terdapat pada akar bulu dan terkadang terdapat pada klenjar, hampir seluruh siklus hidup jenis tungau ini berada pada kulit, jika berlangsung dalam jangka waktu yang lama maka dapat mengakibatkan kematian pada anjing yang diinggapi oleh kutu tersebut. Penelitian ini bertujuan untuk mendeteksi penyakit demodekosis pada anjing dengan aplikasi *mobile* menggunakan metode *convolutional neural network* model arsitektur *squeezenet* dengan optimasi *radam*. Aplikasi ini bekerja dengan pengambilan gambar secara langsung melalui kamera *handphone* kemudian diproses dengan model yang sudah terlatih kemudian *output* dari aplikassi ini menampilkan nilai probabilitas anjing tersebut terindikasi demodex atau tidak. Berdasarkan hasil pengujian menggunakan metode *black box testing*, keseluruhan fungsionalitas aplikasi telah dinyatakan berhasil dijalankan sesuai dengan fungsi dan tujuannya masing-masing. Dari hasil pengujian data validasi menggunakan 60 data validasi, berdasarkan akurasi aplikasi deteksi demodekosis mampu mengenali data validasi dengan benar sebanyak 45 data dan 15 data belum tepat dikenali, sehingga diperoleh nilai akurasi sebesar 75% dan menghasilkan nilai *f1-score* 0,595.

Kata kunci : Demodekosis, Convulitional Neural Network, Optimasi RAdam

ABSTRACT

Demodex is a skin disease in dogs caused by the protozoan mite Demodex follicularumvar canis or Demodex canis. These mites are found in the hair roots and sometimes in the glands, almost the entire life cycle of this type of mite is on the skin, if it lasts for a long time it can cause death in dogs that are infested by these lice This study aims to detect demodicosis in dogs with a mobile application using a convolutional neural network method with a squeezenet architecture model with radam optimization. This application works by taking pictures directly through the cellphone camera then processing it with a trained model then the output of this application displays the probability value of the dog being indicated as demodex or not. Based on the results of testing using the black box testing method, the overall functionality of the application has been declared successful in accordance with its respective functions and objectives. From the results of validation data testing using 60 validation data, based on the accuracy demodecosis detection application is able to recognize validation data correctly as many as 45 data and 15 data has not been properly recognized, so that the accuracy value is 75% and the f1-score value is 0,595.

Keywords : Demodecosis, Convulitional Neural Network, RAdam Optimization.

1. PENDAHULUAN

Anjing pada umumnya merupakan salah satu hewan jinak yang dapat diajak bermain, tinggal dalam lingkungan manusia serta dapat diajak bersosialisai. Beberapa masyarakat tertarik terhadap anjing untuk hewan peliharaan. Anjing juga dapat dilatih

untuk beberapa keperluan yang bermanfaat seperti menuntun orang tua, sebagai penjaga rumah dan membantu polisi dalam penyelidikan berbagai. Sebagai hewan peliharaan anjing juga memiliki keistimewaan pada setiap pemilik anjing [1]. Pada anjing betina yang terjangkit

demodikosis, dapat langsung menularkan tungau demodex kepada anaknya segera setelah melahirkan. Gejala klinis yang muncul pada anjing berupa kerontokan pada rambut, pruritus (rasa gatal sehingga menyebabkan keinginan untuk menggaruk), dan lesi pada kulit. Gejala pruritus tidak selalu muncul pada anjing penderita, tapi hampir selalu dipastikan muncul pada kasus demodex yang disertai infeksi sekunder seperti pustul. Alopesia pada anjing dapat merugikan secara estetika, terlebih pada anjing dengan rambut yang panjang. Alopesia umumnya terjadi di daerah sekitar mata [2].

Beberapa pemilik anjing tidak memiliki pengetahuan tentang perawatan anjing dengan baik serta mengenali penyakit yang sering terjadi pada anjing, pada saat muncul gejala-gejala fisik yang berubah yang mengindikasikan anjing sakit. Sebagian pemilik anjing yang awam tentang penyakit anjing susah mengetahui penyakit yang diderita oleh anjing peliharaan mereka. Sedangkan jika ingin mengetahui penyakit yang diderita oleh anjing peliharaannya pemilik melakukan pengecekan ke klinik hewan, untuk beberapa orang yang tinggal jauh dari tempat klinik hewan akan merasa susah ataupun malas untuk membawa anjing peliharaannya untuk di periksa.

Penelitian [3], terjadi gejala klinis dari penyakit demodekosis adalah pada kulit terjadi alopecia, berkerak pada kulit, munculnya kemerahan disertai rasa gatal jika terinfeksi sekunder. Muncul demodex umumnya pada daerah kepala, kaki depan, hidung, ekor.

Dari penelitian [4]-[8], didapatkan kesimpulan bahwa *convolutional neural network* dapat membantu melakukan proses klasifikasi dengan performa yang cukup baik, bahkan untuk mengenali objek-objek rumit seperti gambar dan suara.

Dari penelitian yang sudah dilakukan masih menggunakan cara manual sehingga untuk sebagian orang awam akan kesulitan untuk mengetahui penyakit demodekosis pada anjing peliharaannya. Sehingga pada

penelitian ini, akan dibuat sebuah aplikasi pengenalan gambar penyakit anjing yang terkena demodekosis menggunakan pendekatan *computer vision* dan *deep learning*. Pendekatan ini melakukan pengenalan terhadap citra anjing yang terjangkit demodekosis yang dibaca oleh kamera, dan aplikasi akan memberi informasi mengenai penyakit demodekosis atau bukan demodekosis yang dibaca oleh kamera. Diharapkan dengan dibuatnya aplikasi ini akan membantu sebagian orang yang awam untuk mengetahui penyakit demodekosis pada anjing.

2. KAJIAN PUSTAKA

2.1 Demodex

Demodex adalah penyakit kulit pada anjing yang disebabkan oleh jenis tungau protozoa demodex follicularumvar canis atau demodex canis. Tungau ini berukuran lebih dari 0,25mm untuk jantan sedangkan pada tungau betina memiliki ukuran 0,3mm. Tungau ini terdapat pada akar bulu dan terkadang terdapat pada klenjar lemak, siklus kehidupan pada tungau ini hampir selalu pada kulit [9].

Jika penyakit yang disebabkan oleh tungau demodex tidak cepat ditindak lanjuti maka akan menyebabkan infeksi lainnya atau sekunder. ini dapat mengakibatkan munculnya jamur atau bakteri ke dalam tubuh lewat luka tersebut sehingga menyebabkan luka sulit diobati. Luka yang diderita oleh anjing yang terkena demodex mengakibatkan anjing mengalami kegelisahan dan menurunnya nafsu makan [10].



Gambar 1 Tungau demodex [10]

Gambar 1 adalah gambar bagaimana hewan demodex yang berkembang biak pada wajah manusia ataupun kulit hewan

yang tidak bisa dilihat dengan mata telanjang ataupun kasat mata.

2.2 Demodekosis

Demodekosis adalah penyakit kulit yang disebabkan oleh tungau demodex sp. Tungau ini termasuk jenis tungau pembuat terowongan dalam kulit induk semangnya, tungau ini menyerang semua mamalia termasuk manusia. Tungau demodex hidup dalam folikel rambut dan kelenjar sebacea [11]. Spesies tungau demodex yang telah dilaporkan antara lain demodex D. canis (anjing), D. bovis (sapi), D. phylloides (babi), dan D. folliculorum pada manusia, D. equi (kuda), D. musculi (tikus), D. caviae (*guinea pig*). Tungau demodex sp merupakan flora normal pada kulit, peningkatan populasi tungau ini secara berlebihan mengakibatkan kerusakan jaringan kulit, seperti pada sapi [12].

Demodekosis dapat terjadi secara lokal dan general. Demodekosis lokal sering terjadi pada anjing muda dengan umur antara 3 sampai 6 bulan dengan lesi yang tidak terlalu parah, namun apabila tidak dilakukan penanganan atau perawatan maka dapat menjadi demodekosis yang bersifat general. Demodekosis general adalah kejadian penyakit kulit yang terjadi baik pada hewan muda maupun tua. Gambar 2 memperlihatkan kondisi fisik anjing yang terinfeksi oleh tungau demodex atau terinfeksi penyakit demodekosis.



Gambar 2 Kondisi fisik anjing demodex [12]

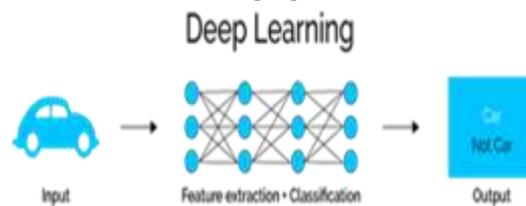
2.3 Computer Vision

Computer Vision atau visi komputer memiliki dua tujuan. Dari sudut pandang ilmu biologi, visi komputer bertujuan untuk menghasilkan model komputasi dari sistem penglihatan manusia. Dari sudut pandang teknik, visi komputer bertujuan untuk membangun sistem otonom yang dapat melakukan beberapa tugas yang dapat dilakukan oleh sistem penglihatan manusia (dan bahkan mengunggulinya dalam banyak kasus). Banyak tugas penglihatan terkait dengan ekstraksi informasi 3D dan temporal dari data 2D dengan waktu beragam seperti yang diperoleh oleh satu atau lebih kamera televisi, dan lebih umum lagi pemahaman tentang pemandangan dinamis semacam itu [13].

2.4 Deep Learning

Deep Learning yaitu sebuah pembelajaran lebih dalam pada bidang machine learning yang memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk melakukan representasi pada data dengan tingkat abstraksi. Metode menggunakan deep learning secara dramatis memperbaiki state of the art dalam pengenalan objek visual, pengenalan suara, deteksi objek dan banyak lainnya [14].

Fungsi dari deep learning yang utama adalah digunakan untuk mengekstraksi pola yaitu feature engineering yang berguna untuk mengekstraksi pola dari data untuk memudahkan model dalam membedakan kelas [14].



Gambar 3 Merupakan Struktur dari pemodelan jaringan pada Deep Learning [14]

Gambar 3 memperlihatkan struktur pemodelan jaringan pada *deep learning*

variabel dalam tabel kontingensi) [17]. Betuk *confusion matrix* dapat dilihat pada

		True condition			
		Condition positive	Condition negative	Prevalence = $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive Type I error	Positive predictive value (PPV) Precision = $\frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\sum \text{False positive}}{\sum \text{Predicted condition positive}}$
	Predicted condition negative	False negative Type II error	True negative	False omission rate (FOR) = $\frac{\sum \text{False negative}}{\sum \text{Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\sum \text{True negative}}{\sum \text{Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+} - 1}{\text{LR-} - 1}$
		False negative rate (FNR), Miss rate = $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	

Gambar 4 Confusion Matrix [17]

pada proses klasifikasi dilakukan dengan *input* langsung tanpa diekstraksi terlebih dahulu karena pada proses klasifikasi langsung dilakukan ekstraksi fitur.

2.5 Convolutional Neural Network

Convolutional Neural Network adalah salah satu dari metode dalam *machine learning* dari beberapa pengembangan yang dilakukan *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data 2D. CNN juga termasuk dalam jenis *deep learning* dimana dalamnya tingkat jaringan dan banyak diimplementasikan dalam sebuah data citra [15].

2.6 Confusion Matrix

Confusion matrix juga sering disebut *error matrix*. Pada dasarnya *confusion matrix* dapat memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang dapat menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Pada setiap baris matriks dapat mewakili *instance* kedalam kelas yang diprediksi [16]. Nama dari *confusion matrix* berasal dari fakta bahwa dalam keadaan kebingungan mudah untuk melihat dua kelas (biasanya salah mengartikannya sebagai yang lain). *Confusion Matrix* adalah sebuah tabel kontingensi yang khusus, dengan 2D atau dua dimensi ("aktual" dan "perkiraan"), dan rangkaian "kelas" yang identik di kedua dimensi (masing-masing kombinasi dimensi dan kelas adalah

gambar 4

2.7 Precision, Recall dan Akurasi

Precision dapat didefinisikan sebagai kemiripan dari hasil kategorisasi suatu dokumen. Kemudian nilai *precision* didapatkan menggunakan persamaan (1) dimana menyatakan perbandingan antara dokumen yang dinyatakan benar (A) dengan seluruh dokumen yang dikategorikan pada kategori tersebut (A+B) (1)[17].

$$Precision = \frac{A}{A + B} \quad (1)$$

Recall dapat didefinisikan sebagai perbandingan antara jumlah dokumen yang relevan yang ditemukan oleh sistem dengan jumlah dokumen pada kategori (A+C) [17]. Nilai *recall* didapatkan dengan menggunakan persamaan (2)[17].

$$Recall = \frac{A}{A + C} \quad (2)$$

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual [16]. Nilai akurasi didapatkan dengan menggunakan persamaan (3)[17].

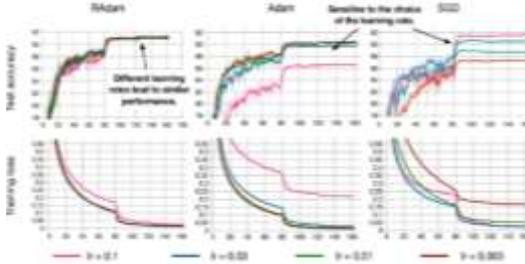
$$Akurasi = \frac{A + D}{A + B + C + D} \quad (3)$$

Dengan A (*True Positif*), B (*False Positif*), C (*False Negatif*), D (*True Negatif*).

2.8 Rectified Adam (RAdam)

Radam atau *adam* yang diperbaiki adalah variasi baru dari pengoptimal *adam* klasik yang menyediakan penyesuaian otomatis dan dinamis untuk tingkat pembelajaran adaptif berdasarkan studi terperinci mereka tentang efek varian dan momentum selama pelatihan [18]. Radam

memegang janji untuk segera meningkatkan setiap arsitektur AI dibandingkan dengan *adam* klasik seperti pada gambar 5.



Gambar 5 Radam berkonvergensi dengan cepat dan mencapai akurasi lebih besar [18]

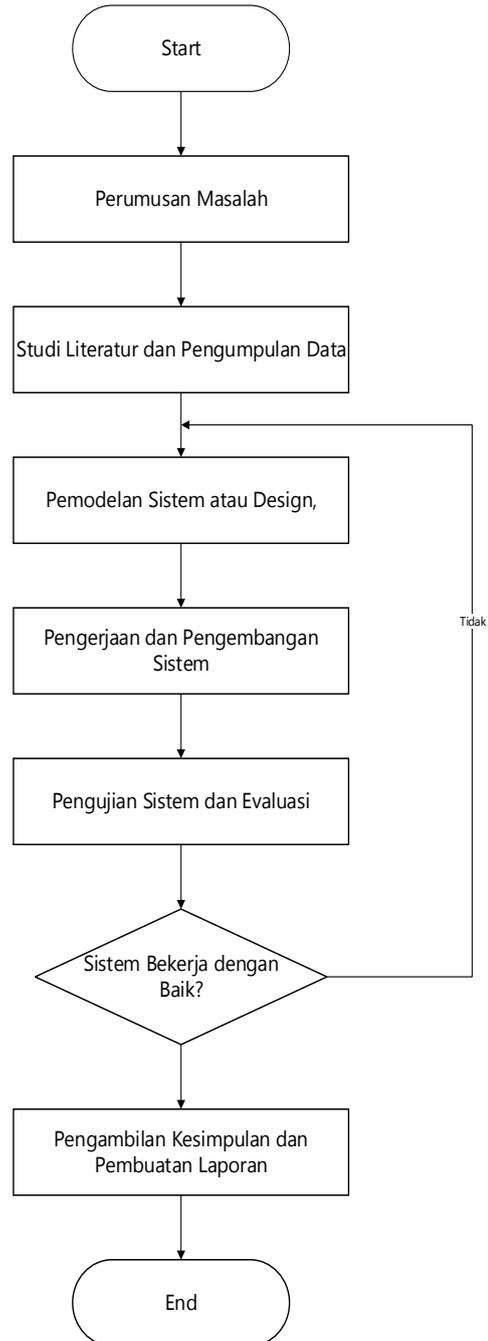
2.9 SqueezeNet

SqueezeNet adalah sebuah nama dari jaringan saraf tiruan dalam untuk *computer vision* yang dirilis pada tahun 2016. *SqueezeNet* yang dikembangkan oleh para peneliti *DeepScale*, University of California, Berkeley, dan Stanford University. Dalam merancang *SqueezeNet*, tujuan penulis adalah membuat jaringan saraf yang lebih kecil dengan parameter yang lebih sedikit yang dapat lebih mudah masuk ke dalam memori komputer dan dapat lebih mudah ditransmisikan melalui jaringan komputer [18].

3 METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan dari penelitian pembuatan aplikasi deteksi penyakit demodekosis pada anjing menggunakan metode *convolutional neural network* dilakukan dengan langkah-langkah seperti gambar 6.



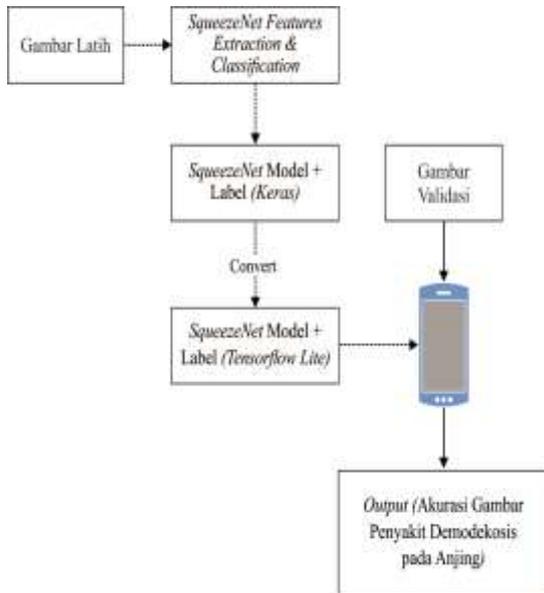
Gambar 6 Flowchart Tahapan Penelitian

Gambar 6 penjelasan alur dari tahapan-tahapan penelitian.

3.2 Gambaran Umum Sistem

Gambaran umum dari aplikasi pengenalan gambar penyakit demodekosis pada anjing terdiri dari 2 proses yaitu proses *latih/training* menggunakan laptop dan proses *uji/testing* menggunakan laptop

dan diujikan pada *handphone* dengan sistem operasi *android* seperti gambar 7.



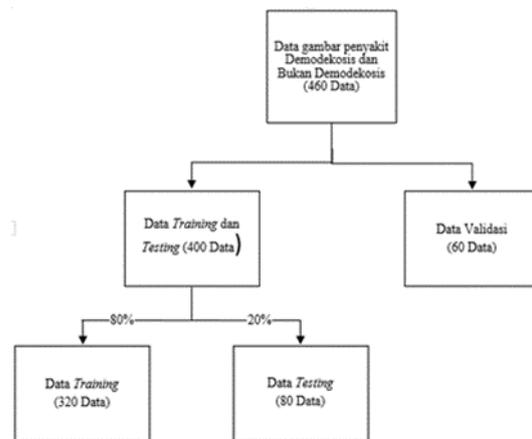
Gambar 7 Gambaran Umum Sistem

Gambar 7 memperlihatkan proses latih, gambar berwarna dengan *format* (.jpg) yang memiliki ukuran 227x227 sebagai *inputan*. Kemudian dibawa ke tahap ekstraksi fitur yang mengubah gambar menjadi ukuran yang lebih kecil melalui *layer-layer deep learning* seperti gambar 7 yang merupakan arsitektur model *squeezenet*. Selanjutnya *input* diberi label yang kemudian dilatih melalui beberapa *epoch*, sehingga menghasilkan sebuah *squeezenet* model dalam bentuk *keras*. Agar model ini dapat digunakan di perangkat *mobile*, model terlebih dahulu diubah menjadi *squeezenet* model dalam bentuk *tensorflow*, kemudian model ini diubah lagi menjadi *squeezenet* model dalam bentuk *tensorflow lite* untuk dapat digunakan di perangkat *mobile*. Pada proses uji dilakukan menggunakan ekstraksi fitur yang sama, namun setelah fitur didapatkan tidak lagi ditambahkan label. Fitur yang didapatkan dari gambar latih dicocokkan atau dicari kedekatannya dengan fitur yang sudah ada didalam *squeezenet* model, selanjutnya sistem akan memberikan sebuah *output* berupa akurasi penyakit anjing demodekosis dan bukan demodekosis kemudian pada sistem dapat menampilkan informasi tentang penyakit demodekosis pada anjing melalui *web* yang tersedia pada sistem.

4 HASIL DAN PEMBAHASAN

4.1 Struktur Data Aplikasi

Data dari penelitian ini dibagi menjadi dua kelompok data yaitu data *training* dan data validasi. Data *training* merupakan data yang digunakan untuk melatih model dalam mengenali penyakit demodekosis pada anjing. Data validasi merupakan data yang digunakan untuk menguji aplikasi dalam memprediksi data baru yang sebelumnya belum pernah dilihat atau data tersebut tidak digunakan saat proses *training*. Berikut data yang digunakan dalam penelitian ini dapat dilihat pada gambar 8.



Gambar 8. Struktur Data Aplikasi

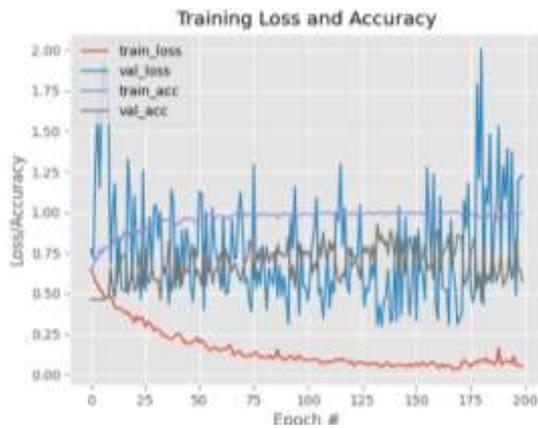
Data gambar penyakit demodekosis pada anjing yang digunakan pada penelitian ini terdiri dari 2 kelompok yaitu gambar penyakit demodekosis dan gambar anjing bukan demodekosis, dengan total data sebanyak 460 data. Pada proses *training* dan *testing*, data gambar anjing yang digunakan sebanyak 400 data. Untuk proses validasi telah disiapkan 60 data gambar anjing yang digunakan, proses *training* dan *testing* dilakukan melalui *desktop*, sedangkan proses validasi dilakukan melalui aplikasi *mobile*. Tabel 1 merupakan *dataset* berupa gambar penyakit anjing yang digunakan pada penelitian ini.

Tabel 1 *Dataset* Gambar Penyakit Anjing

No.	Nama	Jumlah Data	
		<i>Training</i>	Validasi
1	Demodex	200	30
2	BukanDemodex	200	30

4.2 Training dan Testing Model

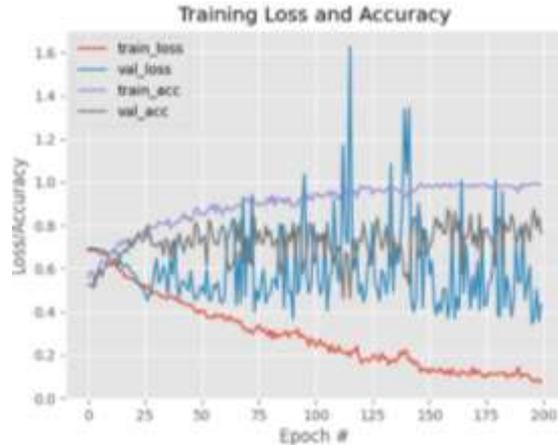
Model yang digunakan pada tahapan *training* dan *testing* menggunakan model dengan arsitektur *squeezenet* dengan optimasi yang berbeda seperti pada gambar



Gambar 9. Grafik Performa *Training* dan *Testing* Optimasi Adam

Berdasarkan grafik gambar 9, nilai akurasi saat *training* (*train_acc*) mengalami peningkatan seiring dengan bertambahnya nilai *epoch*. Nilai *train_acc* mengalami peningkatan secara signifikan sampai pada *epoch* ke-50. Kemudian mengalami peningkatan tetapi tidak terlalu signifikan hingga *epoch* ke-200. Pada beberapa *epoch* juga terjadi kondisi *overfitting*. Kondisi *overfitting* ditunjukkan pada hasil *train_acc* yang didapatkan lebih baik dari nilai *val_acc*, artinya performa model lebih baik dalam memproses data *training* dan gagal melakukan prediksi terhadap data *testing*. Kondisi *overfitting* dapat terjadi karena beberapa faktor, yaitu jumlah *hidden layer* yang digunakan tidak optimal, jumlah *dataset* yang sedikit, dan juga dapat disebabkan karena adanya *noise* saat proses *training*. Kondisi *overfitting* dapat diatasi dengan menerapkan *layer dropout* pada arsitektur model dan melakukan *data augmentation* untuk menambah variasi *dataset*.

Kedua menggunakan optimasi *radam*, data hasil proses *training* dan *testing* dengan 200 *epoch* terlampir pada lampiran 2. Berdasarkan pada lampiran 2, waktu yang dibutuhkan untuk menyelesaikan *training* dan *testing* menggunakan optimasi *radam* dengan 200 *epoch* adalah 1,45Jam dan ukuran model yang dihasilkan dalam setiap *epoch* adalah sama yaitu 2,247 KB



diperoleh dari proses *training* dan *testing* model pada gambar 10.

Gambar 10. Grafik Performa *Training* dan *Testing* Optimasi RAdam

Berdasarkan grafik gambar 10, nilai akurasi saat *training* (*train_acc*) mengalami peningkatan seiring dengan bertambahnya nilai *epoch*. Nilai *train_acc* mengalami peningkatan secara signifikan sampai pada *epoch* ke-140. Kemudian mengalami peningkatan tetapi tidak terlalu signifikan hingga *epoch* ke-200. Pada beberapa *epoch* juga terjadi kondisi *overfitting*. Kondisi *overfitting* ditunjukkan pada hasil *train_acc* yang didapatkan lebih baik dari nilai *val_acc*, artinya performa model lebih baik dalam memproses data *training* dan gagal melakukan prediksi terhadap data *testing*. Kondisi *overfitting* dapat terjadi karena beberapa faktor, yaitu jumlah *hidden layer* yang digunakan tidak optimal, jumlah *dataset* yang sedikit, dan juga dapat disebabkan karena adanya *noise* saat proses *training*. Kondisi *overfitting* dapat diatasi dengan menerapkan *layer dropout* pada arsitektur model dan melakukan *data augmentation* untuk menambah variasi *dataset*.

Berdasarkan pengamatan proses *training* model dengan optimasi *adam* dan *radam* dari *epoch* 1 sampai *epoch* 200, model dengan menggunakan optimasi *radam* merupakan model dengan performa terbaik dikarenakan nilai akurasi saat *testing* (*val_acc*) sudah terlihat lebih baik dibandingkan dengan optimasi *adam* pada *epoch* ke-1 hingga *epoch* ke-60 dimana *radam* menunjukkan *testing* (*val_acc*) yang berangsur membaik dibandingkan menggunakan optimasi *adam*. Berdasarkan nilai yang digambarkan pada grafik tiap-tiap optimasi yang digunakan maka *fit* model

yang digunakan yaitu dengan optimasi *radam*. Selain terlihat pada grafik optimasi *radam* lebih unggul dibandingkan *adam* berdasarkan penelitian yang menjelaskan bahwa *radam* merupakan perbaikan dari optimasi *adam* meningkatkan generalisasi dengan laju pembelajaran dari model yang ada dengan menerapkan pembelajaran awal yang rendah dan dinamis untuk tingkat pembelajaran adaptif berdasarkan studi tentang efek varian dan momentum selama pelatihan [18]. Sehingga model dengan optimasi *radam* selanjutnya akan digunakan untuk membangun aplikasi deteksi penyakit demodekosis pada anjing berbasis *android*.

Tabel 2 Hasil Train setelah 200 Epoch

Setelah 200 Epoch	Train Acc	Val Acc	Train Loss	Val Loss
Adam	0,9938	0,5750	0,0530	1,2300
R-Adam	0,9906	0,7625	0,0724	0,4340

Tabel 2 menampilkan hasil *train* setelah *epoch* ke-200, menunjukkan tingkat *Train_Acc* lebih unggul optimasi *adam* sedangkan pada *Val_Acc* lebih unggul optimasi *radam*.

4.3 Performa Aplikasi

Aplikasi dibangun menggunakan bahasa pemrograman *kotlin* dan *library tensorflow*. Model *squeezenet* dengan format *keras* harus diubah ke format *tensorflow lite* menggunakan *converter* agar dapat digunakan pada aplikasi *mobile*. Hasil dari *converter* ini berupa *floating point* model dengan format *flatbuffer* (.tflite). *Floating point* model memiliki akurasi yang bagus namun ukuran model yang dihasilkan lumayan besar. *tensorflow lite* menyediakan optimasi model menggunakan *quantized* model untuk mengoptimalkan performa dan mengurangi ukuran *binary* model namun sedikit mengurangi akurasi dengan presisi yang kecil. Penelitian ini menggunakan *floating point* model karena model *squeezenet* sudah menghasilkan ukuran model yang kecil setelah melalui proses *training* yaitu 2.247 KB. Kemudian setelah *convert* menggunakan *tensorflow lite converter*

ukuran model menjadi 2.123 KB. Hasil model (.tflite) dan label (.txt) kemudian diinputkan ke dalam *assets* pada *software android studio*. Cara kerja aplikasi deteksi demodex seperti pada gambar 11.



Gambar 11. Cara Kerja Aplikasi

Hasil aplikasi *mobile* deteksi demodekosis pada anjing menggunakan metode *convolutional neural network* seperti pada gambar 12, 13, dan 14.



Gambar 12 Tampilan awal aplikasi

Gambar 12 merupakan tampilan awal aplikasi pengenalan gambar penyakit demodekosis pada anjing. Pengguna dapat

mengambil gambar anjing yang ingin diklasifikasi dengan menekan tombol SELECT PHOTO (langsung dari galeri *handphone*) dan START CAMERA (mengggunakan kamera pada *handphone*).



Gambar 13 Aplikasi menampilkan hasil klasifikasi

Gambar 13 merupakan tampilan aplikasi saat menampilkan hasil klasifikasi. Pengguna menekan tombol SELECT PHOTO ketika ingin mengambil gambar dari galeri *handphone* dan START CAMERA ketika ingin mengambil gambar menggunakan kamera pada *handphone*. Aplikasi akan melakukan klasifikasi menggunakan model terlatih, kemudian aplikasi menampilkan hasil klasifikasi yaitu nama penyakit dan nilai confidence. Confidence adalah tingkat probabilitas sebuah gambar diklasifikasi kedalam kelas demodex atau bukan demodex jika probabilitasnya melebihi atau sama dengan 0.5 maka kelas tersebut dimunculkan pada aplikasi sebagai hasil

klasifikasi dengan confidence adalah nilai probabilitas dikalikan dengan 100%.



Gambar 14 Aplikasi menampilkan web

Gambar 14 merupakan tampilan aplikasi *web*. Pengguna mengklik *hyperlink* pada aplikasi, kemudian aplikasi akan membuka *browser* dan menampilkan *web* informasi tentang penyakit demodekosis lebih spesifik.

4.4 Validasi dan Akurasi

Proses validasi bertujuan untuk menguji aplikasi dalam memprediksi data baru yang sebelumnya belum pernah dilihat atau data diluar dari data *training* menggunakan model terlatih dengan optimasi *adam* dan optimasi *radam*. Proses validasi menggunakan 60 sampel data validasi dari 2 kelas. Keberhasilan validasi dihitung menggunakan nilai akurasi. Hasil pengujian aplikasi menggunakan model terlatih dengan optimasi *adam* digunakan 60 data validasi berdasarkan akurasi diperoleh jumlah data yang diklasifikasi dengan benar yaitu 35 data dan yang belum diklasifikasi dengan tepat yaitu 25 data, kemudian hasil pengujian dari aplikasi menggunakan model terlatih dengan optimasi *radam* digunakan 60 data validasi berdasarkan akurasi diperoleh jumlah data

yang diklasifikasi dengan benar yaitu 45 data dan yang belum diklasifikasi dengan tepat yaitu 15 data. Berdasarkan akurasi yang diperoleh jika dihitung dengan persamaan (1)[17] maka didapatkan nilai akurasi aplikasi dalam mengenali data validasi penyakit demodekosis pada anjing menggunakan model terlatih dengan optimasi yang berbeda, *adam* dan *radam* sebagai berikut

A. *Adam*

$$\frac{35}{60} \times 100\% = 58,3\%$$

B. *RAadam*

$$\frac{45}{60} \times 100\% = 75\%$$

Berdasarkan perhitungan di atas, aplikasi telah berhasil mengenali data dengan memperoleh *adam* akurasi sebesar 58% dan *radam* akurasi sebesar 75%.

Setelah mendapatkan akurasi dari model yang dihasilkan kita perlu memperhatikan apakah akurasi kita cukup baik merepresentasikan kemampuan model atau tidak. Untuk itu kita perlu mencari nilai *f-score*, Tabel 4 memperlihatkan nilai *f-score* dari model yang telah dilatih. Sebelum mendapatkan nilai *f-score* perlu kita lihat Tabel 3 yaitu *confusion matrix* dari proses validasi model yang telah dilatih. Skor F1 yaitu nilai rata-rata harmonis dari *precision* dan daya *recall*, nilai F1 dalam mencapai nilai terbaiknya pada 1 dan nilai terburuk pada 0 [19].

Tabel 3 Confusion Matrix

Kelas	precision	recall	f1-score	f1-Score
Demodex	0.48	0.73	0.58	0.595
Bukan Demodex	0.51	0.76	0.61	

Tabel 4 Nilai F Score

		PREDIKSI	
		DEMODEX	BUKAN DEMODEX
AKTUAL	DEMODEX	22	8
	BUKAN DEMODEX	23	7

5 KESIMPULAN

Berdasarkan pada penelitian yang dilakukan, telah berhasil dibangun sebuah aplikasi *mobile* dengan menggunakan metode *convolutional neural network* yang dapat mengenali gambar penyakit demodekosis pada anjing dan yang bukan demodekosis.

Dari hasil pengujian data validasi menggunakan 60 data gambar dengan optimasi berbeda, berdasarkan *adam* akurasi aplikasi deteksi demodekosis mampu mengenali gambar dengan benar sebanyak 35 data gambar dan 25 data gambar belum tepat dikenali, sehingga diperoleh nilai akurasi sebesar 58,3%, berdasarkan *radam* akurasi aplikasi deteksi demodekosis mampu mengenali gambar dengan benar sebanyak 45 data gambar dan 15 data gambar belum tepat dikenali, sehingga diperoleh nilai akurasi sebesar 75%.

Hal ini menunjukkan bahwa model sudah cukup baik dalam mendeteksi yaitu menggunakan model dengan optimasi *radam* dan dari hasil pengujian data validasi menggunakan *confusion matrix* menghasilkan nilai *f1-score* 0,595. Model yang dibuat kurang baik, diperlukan lebih banyak *dataset* untuk memperbaiki nilai *f1-score*.

6. Daftar Pustaka

- [1] Ningrum, N. C., Anra, H., & Nasution, H, 2016. Sistem Pakar Diagnosa Penyakit Menular Pada Anjing Menggunakan Metode Dempster Shafer, vol 1 no.1, pp. 95–99.

- [2] Gartner, M. C. , 2015. Pet personality: A review. *Personality and Individual Differences*, 75, 102–113.
- [3] Henfrey, J. I, 1990. Inheritable defects in dogs series of articles from reprinted as a booklet.
- [4] D. C. Khrisne and I. M. A. Suyadnya, 2018. "Indonesian herbs and spices recognition using smaller VGGNet-like network," pp. 221–224.
- [5] D. C. Khrisne and T. Hendrawati, 2020. "Indonesian Alphabet Speech Recognition for Early Literacy using Convolutional Neural Network Approach," *J. Electr. Electron. Informatics*, vol. 4, no. 1, pp. 34–37.
- [6] I. M. Wismadi, D. C. Khrisne, and I. M. A. Suyadnya, 2019. "Detecting the Ripeness of Harvest-Ready Dragon Fruit using Smaller VGGNet-Like Network," vol. 3, no. 2, p. 35.
- [7] K. H. Indrani, D. C. Khrisne, and I. M. A. Suyadnya, 2020. "Android Based Application for Rhizome Medicinal Plant Recognition Using SqueezeNet," *J. Electr. Electron. Informatics*, vol. 4, no. 1, pp. 10–14.
- [8] Wicaksono A. Y, Suciati N, Fatichah C, Uchimura K, and Koutaki G, 2017. Modified Convolutional Neural Network Architecture for Batik Motif Image Classification, *IPTEK, Journal Of Cience*, vol. 2, no. 1, pp. 26–30.
- [9] Sardjana, I. K. W, 2012. Pengobatan Demodekosis Pada Anjing Di Rumah Sakit Hewan Pendidikan Fakultas Kedokteran Hewan Universitas Airlangga, 9–14.
- [10] Budiartawan, I. K. A., & Batan, I. W, 2018. Infeksi Demodex canis pada Anjing Persilangan Pomeranian dengan Anjing Lokal. Infeksi Demodex Canis Pada Anjing Persilangan Pomeranian Dengan Anjing Lokal, 562–575.
- [11] Slingenbergh, J., Mohammed, A. N., & Bida, S. A, 1980. Studies on bovine demodecosis in northern Nigeria. Specification and host parasite relationships. *Tijdschrift Voor Diergeneeskunde*.
- [12] Huang, A. R., Mallet, L., Rochefort, C. M., Eguale, T., Buckeridge, D. L., & Tamblyn, R, 2012. "Medication-related falls in the elderly: Causative factors and preventive strategies. *Drugs and Aging*", 359–376.
- [13] Huang, T. S, 1996. *Computer Vision : Evolution and Promise*. 19th CERN School of Computing, 21–25.
- [14] Zein, A, 2019. *Jurnal Penelitian dan Pengkajian Sains dan Teknologi Pendeteksian Penyakit Malaria Menggunakan Medical Images Analisis Dengan Deep Learning Python*, 28–29.
- [15] Ridwan Prasetya, Parlindungan Siahaan, and Romi Wiryadinata, 2018. *Banten Batik Classification with Backpropagation Neural Network*.
- [16] Stehman, S. V, 1997. *Selecting and Interpreting Measures of Thematic Classification Accuracy. Remote Sensing Environment*, 77–89.
- [17] Powers, D. M. W, 2011. *Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation*. *International Journal of Machine Learning Technologies*, 37–63.
- [18] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J, 2019. *On the Variance of the Adaptive Learning Rate and Beyond*. 1–13.
- [19] Beitzel, S. M, 2006. *On Understanding and Classifying Web Queries*. Illinois Institute of Technology.