RANCANG BANGUN APLIKASI IDENTIFIKASI PENYAKIT TANAMAN PEPAYA *CALIFORNIA* BERBASIS ANDROID MENGGUNAKAN METODE CNN MODEL ARSITEKTUR *SQUEEZENET*

Ferry Angga Irawan[1], Made Sudarma [2], Duman Care Khrisne[3]
Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana
Email: ferrysembilanb@gmail.com[1], imasudarma@gmail.com[2],
duman@unud.ac.id[3]

ABSTRAK

Permasalahan yang sering terjadi di Kebun Percobaan Pertanian Universitas Udayana terutama di bidang *Agrikultular* pada tanaman *Holtikultura* adalah mengenai penyakit tanaman, hal ini menyebabkan terjadinya penurunan hasil produksi, sehingga diperlukannya diagnosa penyakit pada tanaman lebih dini. Penelitian ini memfokuskan pada tanaman pepaya *California*. Penyakit pada tanaman ini sering muncul pada bagian daun dan buah. Dengan kemajuan teknologi di bidang image *processing* dapat membantu permasalahan yang terjadi pada bidang pertanian. Pada penelitian ini membangun sebuah aplikasi Android dengan metode CNN (*Convolutional Neural Network*) menggunakan arsitektur *SqueezeNet*. Pengklasifikasikan penyakit pada tanaman ini yaitu *Antraknosa,* dan *Ringspot Viruse*, serta mengklasifikasi pepaya sehat. Berdasarkan hasil validasi, aplikasi yang dibangun menggunakan metode CNN dan Arsitektur *SqueezeNet*, dapat mengenali penyakit *Antraknosa*, *Ringspot Viruse* serta Pepaya sehat melalui daun dengan *accuracy* 97% sedangkan melalui buah *accuracy* mencapai 70%.

Kata kunci : Penyakit Pepaya California, *Convolutional Neural Natwork*, *Arsitektur Squeezenet*, *Image Processing*

ABSTRACT

The problem that often occurs in the Agricultural Experimental Garden of Udayana University, especially in the field of agricultural crops in holticulture is about plant diseases, this causes a decrease in production results, so the need for early diagnosis of diseases of plants. The study focused on California papaya plants. Diseases of this plant often appear on the leaves and fruits. With advances in technology in the field of image processing can help problems that occur in the field of agriculture. In this study build an Android application with CNN (Convolutional Neural Network) method using SqueezeNet architecture. Classifying diseases in this plant are Anthracnose, and Ringspot Viruse, as well as classifying healthy papaya. Based on the validation results, the application built using CNN method and SqueezeNet Architecture, can recognize Anthracnose disease, Ringspot Viruse and Papaya healthy through leaves with accuracy of 97% while through fruit accuracy reaches 70%.

Keywords *: California Papaya Disease, Convolutional Neural Natwork, Squeezenet Architecture, Image Processing*

1. PENDAHULUAN

Pepaya adalah tanaman buah *family caricaceae* yang berasal dari Amerika Tengah dan Hindia Barat. Tanaman ini banyak ditanam di daerah tropis maupun subtropis [1]. Saat ini jenis pepaya yang mulai banyak dibudidayakan adalah jenis Pepaya *California*. Jenis Pepaya ini menjadi pepaya unggulan di Kebun Percobaan Pertanian Universitas Udayana karena memiliki nilai jual yang lebih tinggi dibandingkan dengan tanaman *hortikultura* yang lainnya. Saat ini tanaman Pepaya *California* di Kebun Percobaan Pertanian sudah mencapai 50 pohon, namun dalam budidaya Pepaya *California* terdapat kendala yang dapat menyebabkan produksi Pepaya *California* berkurang, yakni penyakit tanaman. Penyakit yang sering menyerang tanaman Pepaya *California* ada 6, dengan setiap penyakit mempunyai ciri dan cara yang berbeda dalam penanganannya. Dari 6 penyakit tersebut, ada 2 penyakit yang sering menyerang tanaman Pepaya California di Kebun Percobaan Pertanian Universitas Udayana, yaitu penyakit *Antraknosa* dan *Ringspot*

Viruse atau penyakit Bercak Cincin sehingga diperlukan pengetahuan khusus mengenai hama dan penyakit. Tidak semua petani yang ada di Kebun Percobaan Fakultas Pertanian Universitas Udayana memiliki pengetahuan tersebut. Hasil wawancara dengan petani di Kebun Percobaan Fakultas Pertanian Universitas Udayana bahwa, jika tanaman Pepaya California yang ada di Kebun Percobaan Pertanian terserang penyakit, para petani melakukan penanganan dari pengetahuan umum dan pengalaman saja, sehingga untuk penanganannya masih kurang tepat. Sedangkan untuk mendapatkan informasi serta solusi penanganan penyakit tersebut, para petani harus menunggu dosen dari Fakultas Pertanian yang ahli dalam penyakit dan hama tanaman untuk memberikan solusi.

Berdasarkan informasi diatas maka salah satu solusi adalah memberikan penanganan awal. Bentuk awal penanganan yang dapat dilakukan salah satunya adalah melalui diagnosa atau identifikasi penyakit yang ada pada Pepaya California dengan memanfaatkan teknologi yang ada saat ini.

Maka pada penelitian ini akan membangun sebuah aplikasi identifikasi penyakit tanaman Pepaya *California* berbasis Android menggunakan metode *convolutional neural network* arsitektur *SqueezeNet*. Diharapkan dengan adanya aplikasi ini, nantinya dapat menambah wawasan serta pembelajaran bagi masyarakat umum dalam pengidentifikasian serta penanganan penyakit tanaman Pepaya *California* secara tepat.

2. KAJIAN PUSTAKA

2.1 Tanaman Pepaya

Pepaya adalah tanaman buah *family caricaceae* yang berasal dari Amerika Tengah serta Hindia Barat. Tanaman ini banyak ditanam baik di wilayah tropis ataupun subtropis serta mudah berkembang melalui media tanah berhumus campur pasir, dan cukup sinar matahari [1].

2.2 Pepaya California

Tanaman Pepaya *California* merupakan salah satu jenis pepaya yang saat ini banyak ditanam para petani dikarena dari segi keuntungannya yang lumayan besar. Pepaya *California* ini memiliki bobot 0,8 – 1,5 kg/buah, berkulit hijau tebal serta mulus, berbentuk lonjong, buah matang berwarna kuning, rasanya manis, daging buah kenyal serta tebal [4].

2.3 Penyakit Tanaman Pepaya

A. *Antraknosa*

Penyakit *Antraknosa* diakibatkan oleh Jamur *Colletotrichum gloeosporioides (Penz) Sacc.* Penyakit pada buah yang masih muda ditandai dengan terdapatnya bintik kecil kebasah-basahan, bagian ini menghasilkan getah yang berbentuk bintik [5]. Penyakit ini pada buah muda tumbuh sangat lambat dan akan berkembang sangat cepat saat buah menjelang masak. Pada bagian daun, terjadi bercak kecil kebasah-basahan dan bentuknya tidak teratur, meluas berwarna coklat muda [6], seperti gambar 1 (a) dan (b).

(a) (b)

Gambar 1 (a) Penyakit *Antraknosa* Pada Buah (b) Penyakit *Antraknosa* Pada Daun

B. Bercak Cincin/ *Ringspot Viruse*

(a) (b)

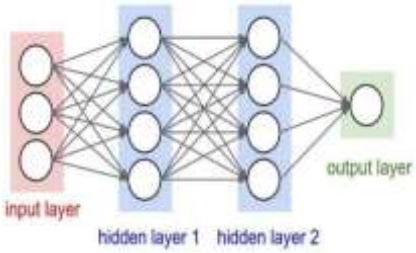
Gambar 2 (a) Penyakit Bercak Cincin Pada Buah (b) Penyakit Bercak Cincin Pada Daun

Penyakit bercak cincin diakibatkan oleh *Papaya Ringspot Virus (PRV),* yang menimbulkan daun berganti bentuk dan sempit serta membuat pertumbuhan pepaya terganggu dan buah yang dihasilkan akan menurun [7]. Gejala awal dari serangan virus ini membuat warna daun menjadi kekuningan dan transparansi tulang-tulang muda. Pada daun juga terdapat bercak kuning serta daun

terpelintir dengan wujud yang tidak teratur. Terdapat garis-garis hijau gelap dan bercak seperti cincin pada tangkai daun dan batang, sedangkan pada buah seperti bercak cincin atau mirip huruf C ini berwarna lebih gelap dari pada kulit buah Pepaya [5], seperti pada gambar 2.

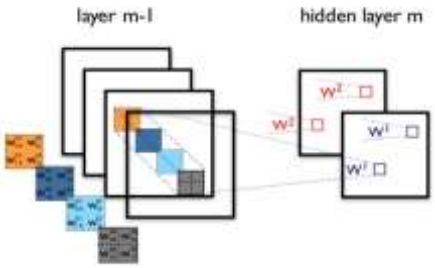
2.4 *Convolutional Neural Network*

Convolutional neural network ialah salah satu tipe *neural network* yang biasa digunakan pada data *image* atau gambar. CNN dapat digunakan untuk mengetahui serta mengidentifikasi objek pada suatu image [9]. Metode kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak semacam MLP yang tiap neuron hanya berukuran satu dimensi



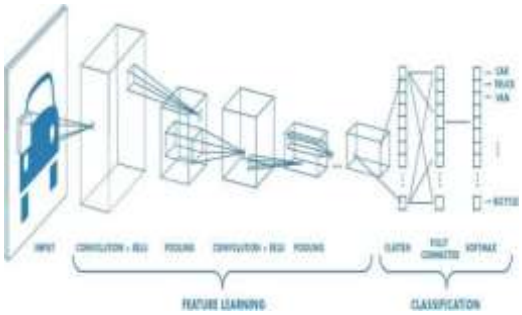
Gambar 3 Arsitektur MLP Sederhana

Gambar 3 menampilkan arsitektur MLP yang mempunyai layer (kotak merah serta biru) dengan tiap-tiap layer berisikan neuron (lingkaran putih). MLP menerima input data satu ukuran serta menyebarkan data tersebut pada jaringan hingga menghasilkan output. Pada CNN operasi linear memakai operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, tetapi berupa empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar 4. Karena sifat proses konvolusi, maka CNN hanya bisa digunakan pada data yang mempunyai struktur dua dimensi seperti citra dan suara.



Gambar 4 Proses Konvolusi pada CNN

CNN juga memiliki sebuah susunan *neuron* 3D (lebar, tinggi, dan kedalaman), untuk ukuran pada layer terdapat pada lebar dan tinggi, sedangkan untuk kedalaman mengacu pada jumlah layer. Jumlah layer pada CNN terdapat 2 layer yaitu: *feature learning* dan *classification* seperti pada gambar 5.



Gambar 5 Layer Pada *Convolutional Neural Network*

Layer pada *Convolutional Neural Network* adalah sebagai berikut :

1. Layer *feature learning*/ekstraksi fitur gambar, tersusun atas beberapa layer dan setiap layer tersusun atas neuron yang terkoneksi pada daerah lokal layer sebelumnya. Layer jenis pertama adalah layer konvolusi dan layer kedua adalah layer *pooling*, yang letaknya berselang-seling.
2. Layer *classification*/ klasifikasi, tersusun atas beberapa *layer* dengan setiap layernya tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan layer lain. Hasil keluaran dari layer ini adalah berupa skoring kelas untuk klasifikasi.

2.8 *Squeezenet*

SqueezeNet ialah arsitektur jaringan saraf tiruan yang menggunakan CNN. *SqueezeNet* sanggup menggapai akurasi *AlexNet*(pemenang *ImageNet classification task* 2012) dengan parameter 50 kali lebih sedikit serta waktu pelatihan 2 kali lebih cepat [10].

SqueezeNet banyak mengubah susunan konvolusi 3x3 dengan 1x1 serta filter yang lebih sedikit untuk mengecilkan ukuran *activation map* (*squeeze*). Metode reduksi dimensi ini pertama kali dikenalkan dalam model *Network In Network* (NIN) [11].

2.9 Keras

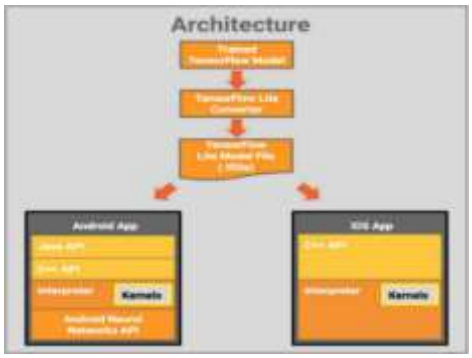
Keras merupakan *interface library* yang digunakan untuk menyederhanakan implementasi algoritma-algoritma *deep Learning* di atas *Tensorflow* [12]. Pembuatan model jaringan syaraf menggunakan *keras* tidak memerlukan penulisan kode untuk mengekspresikan perhitungan matematisnya. Hal ini dikarenakan *keras* sudah menyediakan beberapa model dasar untuk CNN dan dioptimasi untuk mempermudah penelitian tentang *deep learning*. Proses perhitungan atau komputasi menggunakan *keras* akan berjalan baik dengan menggunakan CPU maupun GPU [13].

2.10 Tensorflow

Tensorflow adalah koleksi software open source untuk komputasi numerik yang menggunakan grafik aliran data. Node pada grafik menunjukkan operasi matematika, sedangkan tepi-tepi grafik menunjukkan susunan data multidimensi (tensor) yang dikomunikasikan antartepi grafik. Terdapat 4 komponen *computional graph* terkait *Tensorflow* yaitu: *operation, tensors, variabels,* dan *sessions*. Kelebihan *Tensorflow* antara lain, cepat, *fleksibel* dan siap produksi [14].

2.11 Tensorflow Lite

Tensorflow Lite adalah *library machine learning* yang dirancang khusus untuk perangkat *mobile*. Hal ini memungkinkan mesin untuk belajar di perangkat dengan *latensi* rendah dan ukuran *binary* yang kecil [15]. *Tensorflow Lite* juga mendukung *platform Android* (Java/C++ API), iOS (C++ API), dan Linux (Python/Java/C++ API). Pada gambar 6 merupakan gambaran arsitektur *Tensorflow Lite.*



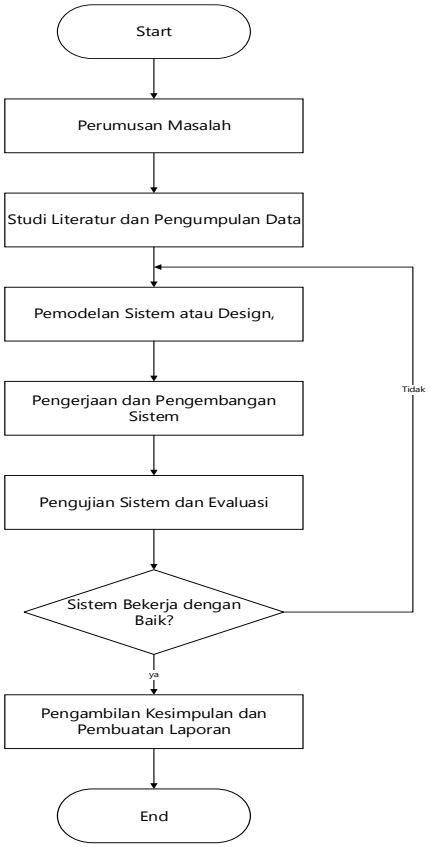
Gambar 6 Arsitektur *Tensorflow Lite*

3 METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian sistem identifikasi penyakit tanaman Pepaya *California* berbasis *Android* dilakukan dengan langkah-langkah seperti gambar 7 dan dijelaskan sebagai berikut :

1. Tahapan pertama yang dilakukan dalam penelitian ini adalah melakukan Perumusan masalah yaitu mengidentifikasi secara spesifik mengenai permasalahan atau topik penelitian. Serta bagaimana tingkat akurasi dari aplikasi identifikasi penyakit tanaman Pepaya *California* menggunakan metode *Convolutional Neural Network* model arsitektur *SqueezeNet* berbasis Android.
2. Tahapan kedua yaitu studi literatur dan pengumpulan data, studi literatur didapatkan dari jurnal, artikel, buku digital yang topiknya terkait dengan perumusan masalah. Serta dipelajari mengenai penyakit Pepaya *California*, serta metode CNN (*Convolutional Neural Network)*, jaringan syaraf tiruan dan arsitektur *SqueezeNet* sebagai referensi penelitian. Dan untuk pengumpulan data dari penyakit tanaman Pepaya berupa data gambar berwarna yang diambil dengan menggunakan kamera xiomi 5A yang bertempat di Kebun Percobaan Pertanian Universitas Udayana dan di Desa Waliang, Abang Karangasem. Gambar diambil dengan kondisi terkena cahaya matahari langsung. Terkumpul sebanyak 4860 gambar Pepaya *California* yang terkena penyakit dan pepaya sehat yang diidentifikasi melalui daun dan buah yaitu : *Antraknosa* 1620 citra (810 citra daun dan 810 citra buah), *Ringspot Viruse* atau penyakit cincin 1620 (810 citra daun dan 810 citra buah). Serta Pepaya Sehat 1620 (810 citra daun dan 810 citra buah).



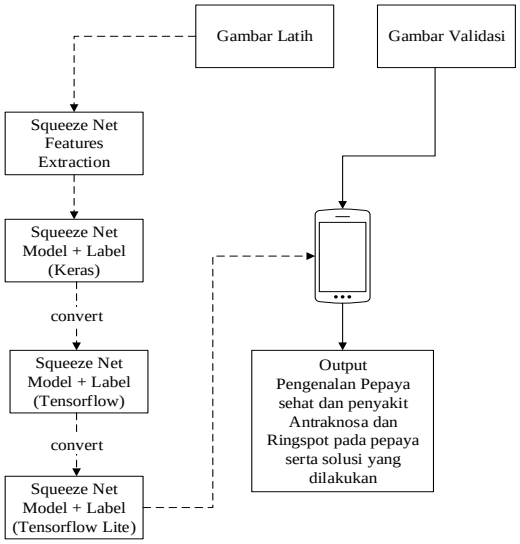
Gambar 7 *Flowchart* Tahapan Penelitian

3. Tahapan ketiga yaitu pemodelan sistem atau *design*. Tahap ini dilakukan penentuan metode yang akan digunakan dalam pembangunan sistem identfikasi penyakit tanaman Pepaya *California*, pada penelitian ini digunakannnya metode adalah *Convolutional Neural Network* (CNN) menggunakan model arsitektur *SqueezeNet*.
4. Tahapan keempat yaitu pengerjaan dan pengembangan sistem. Dalam pengerjaan sistem ini digunakannya *software Android Studio* dengan bahasa pemrograman java dan untuk proses *training* model menggunakan bahasa pemrograman *Python* dengan *library Tensorflow* serta *backend keras*, dan untuk pengembangan sistem digunakannya metode *Linear Sequential Model*, yang memiliki proses sistematis dengan pendekatan sekuensial yang dimulai dari analisis, perancangan, implementasi atau pengkodean dan pengujian.
5. Tahapan kelima adalah proses pengujian dan evaluasi sistem. Pada tahap ini dilakukan sebuah pengujian ke sistem dengan menggunakan *blackbox testing* yang digunakan untuk melihat aplikasi tersebut apakah sudah berjalan dengan baik atau tidak, setelah melakukan pengujian blackbox *testing* pada aplikasi, selanjutnya melakukan pengujian tingkat akurasi aplikasi dalam mengenali penyakit *Antraknosa*, *Ringspot Viruse* dan Pepaya sehat, serta melakukan proses evaluasi terhadap hasil yang didapatkan. Evaluasi ini dilakukan agar mendapatkan data akurasi yang dapat mengenali Pepaya sehat, penyakit *Antraknosa* dan *Ringspot Viruse*.
6. Tahapan keenam adalah penarikan kesimpulan dan pembuatan laporan. Pada tahap ini kesimpulan diambil dari hasil pengujian serta evaluasi sistem yang telah dilakukan.

3.2 Gambaran Umum Sistem

Gambaran umum aplikasi identifikasi penyakit tanaman Pepaya *California* terdiri dari 2 proses yaitu proses latih/*training* dan proses *validasi*, dapat dilihat pada gambar 8.

Pada proses latih, gambar berwarna dengan format (.jpg) yang memiliki ukuran 227×227 sebagai inputan. Kemudian akan dibawa ke tahap ekstraksi fitur yang mengubah gambar menjadi ukuran yang lebih kecil melalui layer-layer *deep learning*. Selanjutnya input diberi label yang kemudian dilatih melalui beberapa epoch, sehingga menghasilkan sebuah SqueezeNet model dalam bentuk keras.



Gambar 8. Gambaran Umum Aplikasi

Agar model ini dapat digunakan di perangkat *mobile*, model harus 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 *validasi* dilakukan dengan menggunakan aplikasi langsung. Pertama gambar *validasi* diambil menggunakan kamera pada perangkat *mobile*, kemudian citra input diproses melalui *SqueezeNet* model yang sudah dilatih untuk di ekstraksi fitur dan diklasifikasikan. Fitur yang didapatkan dari gambar *validasi* akan dicocokkan atau dicari kedekatannya dengan fitur yang sudah ada didalam *SqueezeNet* model, selanjutnya sistem akan memberikan sebuah output berupa identifikasi Pepaya sehat dan nama penyakit tanaman Pepaya yaitu *Antraknosa* dan *Ringspot Viruse* dengan nilai probabilitasnya.

Untuk mengetahui performa dari aplikasi dalam mengenali data validasi penyakit tanaman Pepaya *California* yaitu dengan melihat akurasi dalam mengenali gambar validasi dari penyakit tanaman Pepaya *California*. Data validasi yang digunakan berjumlah 60 data, dengan masing-masing kelas berjumlah 10 data. pengujian dilakukan berdasarkan *Accuracy*. Akurasi didapatkan dari perbandingan antara jumlah data benar dengan jumlah keseluruhan data di kalikan dengan 100%, maka nilai akurasi dapat diketahui dalam bentuk persentase seperti pada persamaan

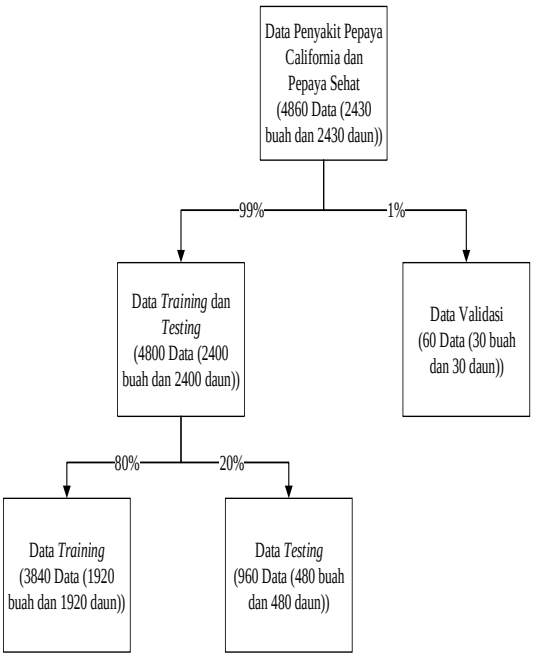
$$Accuracy = \frac{\text{Jumlah data yang dikenali dengan benar}}{\text{Jumlah keseluruhan data}} \times 100\% \quad (1)$$

1

4 HASIL DAN PEMBAHASAN

4.1 Struktur Data Aplikasi

Data citra yang digunakan pada penelitian ini sebanyak 4860 data gambar dengan format (.jpg), yang terdiri 2430 data dari 3 kelas yaitu penyakit *Antraknosa*, *Ringspot Viruse*, pepaya sehat yang di identifikasi melalui daun dan 2430 data dari 3 kelas penyakit *Antraknosa*, *Ringspot Viruse*, pepaya sehat yang di identifikasi melalui buah. Data *training* dan *testing* digunakan sebanyak 4800 data (2400 buah dan 2400 daun), untuk proses *validasi* digunakan 60 data (30 buah dan 30 daun). Gambar 9 merupakan bagan yang menggambarkan struktur dari *dataset* yang digunakan. Dan untuk penjelasan data secara terperinci dapat dilihat pada tabel 1



Gambar 9. Struktur Data Aplikasi

Tabel 1 Data Penyakit *Antraknosa*, *Ringspot Viruse* dan Pepaya Sehat

No.	Nama	Jumlah Data	
		Training	*Validasi*
1	*Antraknosa* Daun	800	10
2	*Ringspot Viruse* Daun	800	10
3	Pepaya Sehat Daun	800	10
4	*Antraknosa* Buah	800	10
5	*Ringspot Viruse* Buah	800	10
6	Pepaya Sehat Buah	800	10

4.2 Training dan Testing Model

Tahap *training* dan *testing* pada daun dan buah dilakukan secara terpisah agar aplikasi dapat belajar mengenali penyakit *Antraknosa*, *Ringspot Viruse*, dan pepaya sehat melalui daun dan buah. Proses ini dilakukan untuk memperoleh sebuah model

yang nantinya akan digunakan untuk klasifikasi. Proses *training* adalah sebuah proses yang modelnya akan dilatih untuk mengenali penyakit *Antraknosa*, *Ringspot Viruse* pada Pepaya *California* dan Pepaya sehat berdasarkan daun dan buah. Pada proses ini juga *training* model akan dibagi menjadi dua proses yaitu proses *training* dan proses *testing*. Analisa yang dapat dilihat pada proses ini adalah nilai akurasi (ACC) dan *loss model (Loss).* Pada gambar 10 dan gambar 11 dapat dilihat proses

training dan *testing* menggunakan model train.py. Aplikasi akan melakukan *training* dengan data yang disimpan pada folder *dataset*. pada desktop.

Gambar 10 Proses *Training* Buah Pepaya *California*

Gambar 11 Proses *Training* Daun Pepaya *California*

Dua parameter utama pada penelitian ini untuk proses *training*, adalah yaitu *epoch* dan *learning rate*. Pada saat proses *training* dengan model arsitektur *SqueezeNet*, nilai akurasi terbaik diperoleh dengan mengganti nilai *learning rate* 4e-2 menjadi 1e-3. Sehingga nilai *learning rate* yang digunakan pada penelitian ini yaitu 1e-3, yang secara linier diturunkan pada setiap *epoch* [10].

Fit model ditentukan berdasarkan nilai akurasi dan *loss* yang terjadi saat proses *training* dan *testing*. Waktu yang dibutuhkan untuk menyelesaikan *training* dengan 200 *epoch* pada buah adalah 25.49 Jam dan untuk daun selama 25.60 jam. Sehingga waktu total yang dibutuhkan untuk *training* daun dan buah Pepaya California adalah kurang lebih 51.09 jam dengan ukuran model yang dihasilkan dalam setiap *epoch* adalah sama yaitu 6.39 mb.

Berdasarkan model yang terbentuk melalui proses *training* dan *testing*, maka digunakannya model *epoch* 189 untuk identifikasi daun, dan untuk model buah digunakannya *epoch* 121. Perbandingan data hasil *loss* dan *accuracy* dapat dilihat pada tabel 2 dan tabel 3.

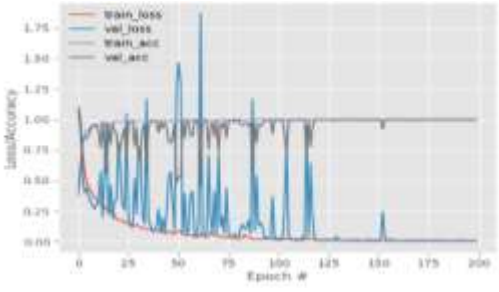
Tabel 2 Hasil Akurasi Data *Training* Dan *Testing* Model 189 Daun

Data	Jumlah Data	Nilai *Loss*	Nilai *Accuracy*
Training	80	0,0066	99.95 %
Testing	20	0.0098	99.58 %

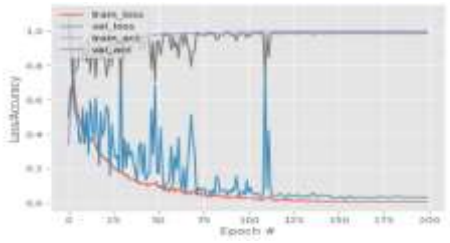
Tabel 3 Hasil Akurasi Data *Training* Dan *Testing* Model 121 Buah

Data	Jumlah Data	Nilai *Loss*	Nilai *Accuracy*
Training	80	0.0168	99.95 %
Testing	20	0.0285	99.17 %

Kinerja *loss* dan akurasi yang didapatkan, ditunjukkan pada gambar 12 dan 13 yaitu grafik performa *training* dan *testing* pada daun dan buah. Berdasarkan grafik pada gambar 12 dan 13, model sudah mampu berlatih dalam mengenali data, sehingga tidak mengalami *overfiting* ataupun *underfiting.*



Gambar 12 Grafik Performa *Training* dan *Testing* Daun Pepaya *California*



Gambar 13 Grafik Performa *Training* dan *Testing* Buah Pepaya *California*

Data tersebut didukung dengan nilai *accuracy* yang tinggi, untuk nilai *accuracy* dari proses *training* daun mencapai 99.95 % dan nilai *accuracy* pada proses *testing* sebesar 99.58 %. Hal ini menunjukkan bahwa model memiliki kecocokan dengan *dataset* yang ada dan mampu melakukan klasifikasi dengan baik. Sedangkan nilai *accuracy* untuk proses *training* buah mencapai 99.95 % dan nilai *accuracy* pada proses *testing* sebesar 99.17 % hal ini menunjukkan bahwa model yang didapatkan mampu mengklasifikasi secara baik.

Grafik loss yang mendekati nilai 0 atau rendah dan grafik accuracy yang semakin meningkat menunjukkan performa yang baik. *Loss* adalah sebuah fungsi yang digunakan untuk menggambarkan kemungkinan kerugian yang akan dihasilkan oleh model yang didapatkan. Sedangkan *accuracy* merupakan persentase dari data validasi yang diklasifikasikan ke kelas yang benar. Berdasarkan tabel 2 dan tabel 3, nilai *loss* pada data *training* daun adalah 0.0066. dan pada data *testing* daun adalah 0.0098. Sedangkan nilai loss pada *training* buah sebesar 0.0168 dan pada data *testing* sebesar 0.0285. Nilai loss pada data *training* lebih rendah dibandingkan dengan data *testing*. Sehingga model ini baik untuk digunakan dalam identifikasi penyakit tanaman pepaya california.

Berhentinya proses pelatihan atau *training* ditentukan dari jumlah *epoch* yang digunakan. Dalam penelitian ini digunakan sebanyak 200 *epoch*. Selanjutnya model yang dipilih akan digunakan untuk membangun aplikasi android identifikasi penyakit tanaman pepaya california berdasarkan daun dan buah.

4.3 Performa Aplikasi

Aplikasi dibangun menggunakan bahasa pemrograman *java* dan *library Tensorflow*. Model *SqueezeNet* dengan format *Keras* (h5) harus diubah ke format *Tensorflow Lite* menggunakan *converter* agar dapat digunakan pada aplikasi *mobile*. Untuk melihat performa aplikasi serta *accuracy* pada aplikasi android maka dilakukan proses *validasi* melalui *smartphone* dengan *dataset* yang baru secara langsung.

A. *Validasi* dan *Accuracy*

Tahap *validasi* ini dilakukan secara langsung di kebun mulai jam 09.00 pagi – 14.00 siang, karena menyesuaikan pencahayaan pada pengambilan dataset. Tahap *validasi* menggunakan 60 sampel data *validasi* dari 3 kelas daun (2 penyakit pepaya *california*, dan 1 pepaya sehat) dan 3 kelas buah (2 penyakit pepaya *california*, dan 1 pepaya sehat). *Validasi* dilakukan berdasarkan *Accuracy* yang didapatkan.

Hasil pengujian aplikasi dengan 60 data *validasi* berdasarkan *Accuracy* diperoleh jumlah data daun yang diklasifikasi dengan benar yaitu 29 data dan yang belum diklasifikasi dengan tepat yaitu 1 data. Sedangkan jumlah data buah yang diklasifikasi dengan benar adalah 21 dan yang belum diklasifikasi dengan tepat yaitu 9 data. Klasifikasi yang salah dikarenakan dataset hampir menyerupai dengan dataset yang lain sehingga mengakibatkan salah dalam memprediksi.

Jika dihitung dengan persamaan 1 didapatkan nilai akurasi aplikasi dalam mengenali penyakit *Antraknosa*, *Ringspot Viruse* serta pepaya sehat pada tanaman pepaya *california* sebagai berikut.

a. *Accuracy* Daun

$$\frac{29}{30} \times 100\% = 97\%$$

b. *Accuracy* Buah

$$\frac{21}{30} \times 100\% = 70\%$$

Berdasarkan perhitungan diatas, aplikasi telah berhasil mengenali data *validasi* sebesar dengan memperoleh

Accuracy daun sebesar 97% dan *Accuracy* buah sebesar 70%.

5 KESIMPULAN

Berdasarkan Penelitian yang dilakukan, penelitian ini telah berhasil membangun aplikasi berbasis android yang dapat mengenali penyakit *Antraknosa*, *Ringspot Viruse* serta Pepaya sehat. Dari hasil pengujian data *validasi* 30 data daun, dan 30 data buah, diperoleh Akurasi dari model *SqueezeNet* yang cukup baik, dengan tingkat akurasi daun mencapai 97%, sedangkan 70% didapatkan untuk akurasi buah. Hal ini menunjukan bahwa model sudah cukup baik dalam mendeteksi penyakit tanaman pepaya (*Antraknosa*, *Ringspot Viruse*, serta Pepaya Sehat) melalui daun dan buah dari data baru yang belum terlihat.

Adapun beberapa saran dalam pengembangan aplikasi ini adalah, untuk mendapatkan akurasi yang lebih baik sebaiknya memfokuskan ke citra penyakit tersebut agar mengurangi redudansi. Selanjutnya diharapkan agar menambahkan jumlah kelas penyakit pepaya *California* yang belum ada.

6. Daftar Pustaka

[1] A. Soedarya, Agribisnis Pepaya, Bandung: Pustaka Grafika, 2009.

[2] D. R. Priindaryanti, “Pengenalan Pola Citra Penyakit Tanaman Padi Pada Daun Menggunakan Gabor Wavelet dan Algoritma K-Means,” pp. 4-48, 2017.

[3] A. Hidayatuloh, “IDENTIFIKASI PENYAKIT TANAMAN TOMAT MELALUI DAUN DENGAN METODE CONVOLUTIONAL NEURAL NETWORK ARSITEKTUR SQUEEZENET,” 2018.

[4] RAR, “Cara menanam biji pepaya california,” Ayo Berkebun, 7 2016. [Online]. Available: https://ayoo-berkebun.blogspot.com/2016/07/cara-menanam-biji-pepaya-california.html. [Diakses 2019 Oktober 26].

[5] A. S. Indriyani, Pengelolaan Kebun Pepaya Sehat, Solok: Balai Penelitian Tanaman Buah Tropika, 2008, p. 23.

[6] M. Prasojo, “Pengendalian Penyakit Patek (Antraknosa) Pada Pepaya,” 2017. [Online]. Available: https://unsurtani.com/2017/01/pengendalian-penyakit-patek-Antraknosa-pada-Pepaya. [Diakses 2019 November 29].

[7] K. Triyanto, “Cara Mengatasi Hama dan Penyakit Tanaman Pepaya,” 2016. [Online]. Available: https://kabartani.com/cara-mengatasi-hama-dan-penyakit-pada-tanaman-pepaya.html. [Diakses 19 Juni 2019].

[8] W. S. Suartika, “Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101,” *Teknik ITS,* vol. V, no. 5, p. A65, 2016.

[9] S. Sena, “Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN),” 2017. [Online]. Available: https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94. [Diakses 4 Desember 2019].

[10] F. N. e. a. Iandola, “SqueezeNet: AlexNetlevel accuracy with 50x fewer parameters and,” 2016. [Online].

[11] Y. A. S. S. A. Arrizal Amin, “Klon Perilaku Menggunakan Jaringan Saraf Tiruan Konvolusional Dalam Game SuperTuxKart,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , vol. 3, no. 1, pp. 866-875, 2019.

[12] P. Pintar, “Implementasi Deep Learning Sederhana Menggunakan Keras,” 2018. [Online]. Available: https://medium.com/@planet.pintar/implementasi-deep-learning-sederhana-menggunakan-keras-3f5726f007e7. [Diakses 29 Desember 2019].

[13] T. Shafira, “IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORKS UNTUK KLASIFIKASI CITRA TOMAT MENGGUNAKAN KERAS,” p. 34, 2018.

[14] A. R. &. T. Ramdani, “Tensorflow yang Dikembangkan oleh Google Brain Team,” 2019. [Online]. Available: https://mmsi.binus.ac.id/2019/11/26/tensorflow-yang-dikembangkan-oleh-google-brain-team/. [Diakses 24 Desember 2019].

[15] A. Andrea, “Tensor Flow Lite Android,” 2016. [Online]. Available: https://adiandrea.id/articles/2018-05/tensor-flow-lite-android. [Diakses 24 Desember 2019].