

Pendeteksian Penyakit pada Daun Cabai dengan Menggunakan Metode *Deep Learning*

<http://dx.doi.org/10.28932/jutisi.v6i3.2857>

Rosalina [✉]#1, Ardi Wijaya^{#2}

[#]Program Studi Teknik Informatika, Universitas Presiden, Bekasi, Jawa Barat, Indonesia

¹rosalina@president.ac.id

²ardywijaya1997@gmail.com

Abstract — Chili is one of the most essential horticultural plants in Indonesia. In addition to the lack of supply of plants, the price of chili on the market has increased dramatically. The shortage is affected by unpredictable climate changes, which have to result in many chili plants suffering from crop failure. It was because the disease infects chili plants so that harvests are decreased. This work would incorporate Deep Learning for image processing in Disease Detection Systems. This disease detection method will be used to help users, in particular chili farmers, identify whether or not the leaves of their chili plants are contaminated with the disease. This system would take a picture of chili leaf using a Raspberry Pi camera and implement image processing on the chili leaf image to collect valuable information on the image to find out whether or not the chili leaf is contaminated with the disease. The purpose of this research is to make a desktop application for a disease detection system that has the ability to detect whether or not a chili leaf is infected by several diseases, display the condition of the chili leaves, display the type of disease that infects the chili leaves (if any), and provide a percentage probability of the system in detecting the image of the chili leaves correctly (whether it is healthy chili leaves or sick chili leaves). The system reaches 100 percent accuracy with good brightness and distance less than 1 meter, while the system reaches 68.8 percent accuracy with poor brightness and distance greater than or equal to 100 percent.

Keywords— chili leaf; deep learning; disease detection; raspberry pi

I. PENDAHULUAN

Komoditas Cabai bukan termasuk pangan pokok bagi masyarakat Indonesia, akan tetapi perannya sebagai bumbu pelengkap masakan, ditunjang harganya yang selalu fluktuatif, tak jarang cabai menyumbang inflasi bagi perekonomian nasional. Seiring dengan kelangkaan tanaman cabai di Indonesia mengakibatkan meningkatnya harga cabai secara drastis di pasaran. Kelangkaan tersebut disebabkan oleh perubahan cuaca yang tidak menentu, yang mengakibatkan banyak perkebunan cabai mengalami

kegagalan panen. Hal tersebut dikarenakan penyakit yang menyerang tanaman cabai sehingga hasil panen berkurang. Petani cabai banyak mengalami kerugian, serta konsumen cabai harus mengeluarkan biaya yang lebih besar daripada biasanya untuk membeli cabai. Cabai adalah salah satu tanaman hortikultura yang penting di Indonesia. Cabai diklasifikasikan sebagai tanaman buah dan sayuran yang memiliki protein untuk dikembangkan dan memiliki nilai ekonomi tinggi [1]. Seperti yang terlihat pada Gambar 1, produksi cabai selama 2012-2014 perbulannya cenderung menurun hingga titik terendahnya terjadi pada bulan November dan Desember. Pengurangan produksi cabai diikuti oleh kebutuhan konsumen yang tinggi. Pada musim hujan, produksi cabai berkurang, sementara permintaan konstan dan terus menerus setiap hari, bahkan meningkat pada musim-musim tertentu. Salah satu penyebab menurunnya produksi adalah penyakit tanaman cabai, penyakit utama pada tanaman cabai antara lain adalah busuk buah, bercak daun, layu fusarium, penyakit virus.

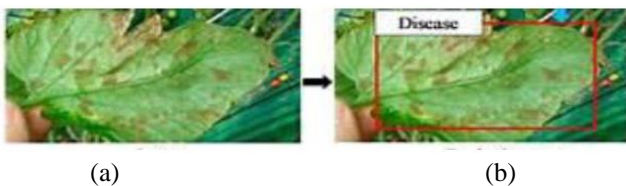


Gambar 1. Produksi Cabai di Indonesia (Sumber: BPS dan Direktorat Jenderal Hortikultura Kementan (diolah))

Adanya serangan penyakit yang menyerang tanaman cabai dapat menyebabkan gagal panen. Karena itu, kontrol yang tepat diperlukan untuk menghindarinya. Kontrol yang tepat bukan hanya ketika serangan sudah terjadi, tetapi yang paling penting adalah tindakan pencegahan. Keterlambatan

proses diagnosis disebabkan oleh kurangnya pengetahuan petani tentang jenis penyakit yang menyerang tanaman cabai [2]. Saat ini untuk mendeteksi penyakit pada tanaman cabai dilakukan secara manual oleh spesialis. Dalam praktiknya, diperlukan tim ahli yang tidak sedikit jumlahnya serta diperlukan pemantauan tanaman secara terus-menerus, sehingga memerlukan biayanya sangat tinggi bila dilakukan pada pertanian yang luas. Pada saat yang sama, petani di Indonesia umumnya tidak memiliki fasilitas yang memadai untuk dapat menghubungi para ahli, selain itu juga tenaga ahli yang terbatas menjadi kendala bagi petani untuk berkonsultasi. Berdasarkan uraian yang diuraikan di atas, diperlukan sistem pakar yang dapat membantu petani cabai untuk mendeteksi apakah tanaman cabai mereka terinfeksi oleh penyakit atau tidak.

Penelitian ini akan mengimplementasikan *Deep Learning* untuk pemrosesan gambar ke dalam sistem deteksi penyakit. Sistem deteksi penyakit ini akan digunakan untuk membantu pengguna terutama petani cabai untuk mendeteksi apakah daun tanaman cabai mereka terinfeksi oleh penyakit atau tidak. Sistem ini akan mengambil gambar daun cabai dengan menggunakan kamera Raspberry Pi dan melakukan pemrosesan gambar pada gambar daun cabai untuk mendapatkan informasi penting tentang gambar tersebut untuk mengetahui apakah daun cabai terinfeksi oleh penyakit atau tidak. Tujuan dari penelitian ini adalah membuat aplikasi desktop untuk sistem deteksi penyakit yang memiliki kemampuan untuk mendeteksi apakah daun cabai terinfeksi oleh beberapa penyakit atau tidak, menampilkan kondisi daun cabai, menampilkan jenis penyakit yang menginfeksi cabai daun (jika ada), dan berikan nilai persentase probabilitas sistem dalam mendeteksi gambar daun cabai dengan benar (apakah itu daun cabai sehat atau daun cabai sakit), seperti yang ditunjukkan pada Gambar 2, penelitian yang diusulkan ini dapat memperkirakan probabilitas penyakit pada daun cabai yang ditunjukkan sebagai kotak pembatas yang berisi area tanaman yang terinfeksi.



Gambar 2. (a) Input Gambar Daun Cabai , (b) Deteksi Penyakit Pada Daun Cabai

II. TINJAUAN PUSTAKA

Beberapa penelitian telah dilakukan untuk mendiagnosis penyakit pada tanaman dengan menggunakan data gambar daun. Penelitian-penelitian tersebut menggunakan data gambar daun karena penyakit tampak nyata pada daun.

Salah penelitian diterbitkan oleh Aduwo et. Al. [3], yang menyajikan penggunaan *computer vision* untuk mendiagnosis penyakit singkong. Mereka menggunakan gambar daun dari tanaman singkong yang diambil di laboratorium dengan pencahayaan dan latar belakang yang seragam. Algoritma dua kelas dikembangkan untuk mendeteksi apakah gambar daun berasal dari tanaman yang sakit atau sehat. Tiga set fitur diekstraksi dari gambar daun termasuk fitur yang berkaitan dengan rona dan intensitas gambar (HSV), dan fitur yang menangkap penekanan tombol pada gambar, termasuk fitur *Scale In-variant Feature Transformation* (SIFT) dan *Speeded-Up Robust Features* (SURF).

Metode klasifikasi yang digunakan ditingkatkan [4 - 6], di mana implementasi yang ditingkatkan dari skema klasifikasi berbasis prototipe. Beberapa penelitian lain memperluas masalah dua kelas menjadi masalah multi-kelas dengan berbagai penyakit dan tingkat penyakit yang berbeda-beda [7]. Beberapa fitur lain juga digunakan dalam ekstensi dari karya awal oleh Aduwo et.al.

Pendekatan berbasis gambar yang lebih baru untuk deteksi penyakit cabai didasarkan pada metode *deep learning*, misalnya [8] Dalam makalah ini, model jaringan saraf convolutional dikembangkan untuk melakukan deteksi dan diagnosis penyakit tanaman menggunakan gambar daun sederhana dari tanaman yang sehat dan berpenyakit, melalui metodologi *deep learning*. Beberapa model arsitektur dilatih, dengan kinerja terbaik mencapai tingkat keberhasilan 99,53% dalam mengidentifikasi kombinasi (tanaman, penyakit) yang sesuai (atau tanaman sehat). Tingkat keberhasilan yang sangat tinggi membuat model ini menjadi alat penasehat atau peringatan dini yang sangat berguna, dan sebuah pendekatan yang dapat dikembangkan lebih lanjut untuk mendukung sistem identifikasi penyakit tanaman terintegrasi untuk beroperasi dalam kondisi budidaya nyata. Penelitian lainnya yang menggunakan metode *deep learning* adalah [9-10] dimana penelitian ini menggunakan pendekatan gambar daun ditangkap oleh pengguna dan kemudian gambar diproses untuk menentukan status kesehatan daun tanaman. Deteksi penyakit daun cabai melalui gambar daun dan teknik pengolahan data sangat berguna dan sistem yang murah terutama untuk membantu petani dalam memantau area tanaman yang luas.

A. Raspberry Pi

Raspberry Pi adalah serangkaian komputer papan tunggal kecil yang dikembangkan di Inggris oleh Yayasan *Raspberry Pi* yang digunakan untuk mempromosikan pengajaran ilmu komputer dasar di sekolah dan di negara berkembang [11]. Tujuan dari dikembangkannya *Raspberry Pi* adalah untuk menciptakan perangkat berbiaya rendah untuk meningkatkan keterampilan pemrograman dan pemahaman perangkat keras di tingkat pra-universitas.

Sementara tujuan akhir mereka adalah untuk menumbuhkan pendidikan ilmu komputer dan mereka berharap bahwa komputer kecil dan terjangkau ini akan menjadi alat yang memungkinkan itu [12]. *Raspberry Pi* lebih lambat dari laptop atau desktop modern tetapi masih merupakan komputer Linux yang lengkap dan dapat memberikan semua kemampuan yang diharapkan yang menyiratkan, pada tingkat konsumsi daya rendah [13].

B. Computer Vision

Computer Vision adalah bidang ilmu komputer yang memungkinkan komputer untuk dapat melihat, mengidentifikasi, dan memproses gambar dengan cara yang sama seperti yang dilakukan oleh penglihatan manusia, dan kemudian memberikan *output* yang sesuai. Seperti menanamkan kecerdasan manusia kedalam komputer. Pada kenyataannya, agar komputer dapat mengenali gambar atau objek yang berbeda tersebut merupakan tugas yang sulit. Dalam hal ini, penglihatan komputer terkait erat dengan kecerdasan buatan, karena komputer harus menafsirkan apa yang dilihatnya, dan kemudian melakukan analisis yang sesuai atau bertindak sesuai.

Tujuan dari *computer vision* tidak hanya untuk melihat, tetapi juga untuk memproses dan memberikan hasil yang bermanfaat berdasarkan pengamatan. Misalnya, perkebunan cabai dapat dilengkapi dengan visi komputer yang akan dapat mengidentifikasi apakah daun cabai terinfeksi oleh penyakit atau tidak. Kemudian, perangkat pintar dapat menginformasikan kepada petani cabai jika ada daun cabai yang terinfeksi penyakit.

C. Image Processing

Image Processing merupakan metode untuk melakukan beberapa operasi pada gambar untuk mendapatkan gambar yang disempurnakan atau untuk mengekstrak informasi yang dibutuhkan. Proses sinyal di mana inputnya adalah gambar dan luarannya adalah gambar atau karakteristik / fitur yang terkait dengan gambar tersebut. Pemrosesan gambar mencakup tiga langkah; yang pertama adalah mengimpor gambar melalui alat akuisisi gambar, yang kedua adalah menganalisis dan memanipulasi gambar, dan yang terakhir adalah output di mana hasilnya dapat diubah gambar atau laporan yang didasarkan pada analisis gambar [14].

D. Deep Learning

Deep Learning merupakan bagian dari *Artificial Intelligence* dan *Machine Learning* di mana jaringan saraf tiruan dan algoritmanya terinspirasi oleh otak manusia dan belajar dari sejumlah besar data [15]. Demikian pula dengan bagaimana kita belajar dari pengalaman, algoritma *deep learning* akan melakukan tugasnya berulang-ulang. *Deep learning* menggunakan jaringan saraf tiruan berlapis-lapis untuk memberikan akurasi yang tinggi dalam melakukan

deteksi objek, pengenalan suara, terjemahan bahasa, dan lainnya. Selain itu, arsitekturnya sangat fleksibel karena dapat belajar langsung dari data mentah dan dapat meningkatkan akurasi prediktif ketika diberikan atau dimasukkan lebih banyak data [16].

III. SISTEM ANALISIS DAN DESAIN

Ada empat langkah dalam fitur Sistem deteksi penyakit pada daun cabai agar dapat berjalan dengan tepat:

A. Sistem Inisialisasi

Sistem inisialisasi menggambarkan proses inisialisasi sistem yang dilakukan ketika pengguna menjalankan sistem untuk pertama kalinya. Inisialisasi sistem ini juga mencakup inisialisasi model dan pelatihan serta proses evaluasi model yang akan digunakan nanti ketika melakukan pemrosesan gambar. Pada awalnya, pengguna akan mengklik tombol "*Train Model*". Dan kemudian, model akan diinisialisasi. Selanjutnya, sistem akan mulai melatih model berdasarkan set data pelatihan dan mengevaluasi model berdasarkan set data pengujian. Setelah selesai, model kemudian akan disimpan pada sistem yang nantinya akan digunakan untuk pemrosesan gambar dan jalur model akan disimpan ke file teks.

B. Persiapan Sistem

Sistem deteksi penyakit pada daun cabai memiliki beberapa persiapan sebelum sistem mulai memproses gambar daun cabai. Sebelum sistem mulai memproses gambar daun cabai, pengguna harus mengklik tombol "Mulai Kamera" untuk mengaktifkan Kamera i. Setelah diaktifkan, tombol "Capture Image" akan diaktifkan dan akan ada kotak gambar yang akan menampilkan pratinjau kamera langsung dari *Raspberry Pi Camera*. Setelah itu, pengguna harus mengklik tombol "Capture Image" untuk mengambil gambar. Kemudian, gambar yang diambil akan ditampilkan pada kanvas di antarmuka pengguna dan disimpan di *Raspberry Pi*. Dan juga, jalur gambar yang diambil akan disimpan ke file teks. Nantinya, gambar ini akan diproses untuk mendapatkan beberapa informasi penting yang akan berguna untuk mengidentifikasi apakah daun cabai terinfeksi oleh penyakit atau tidak.

C. Sistem Pemrosesan

Dalam melakukan pemrosesan deteksi penyakit pada daun cabai ada beberapa proses yang harus dilakukan oleh pengguna, pertama, pengguna harus mengklik tombol "*Process Image*", kemudian, sistem akan membaca jalur gambar daun cabai dan memuat model. Selanjutnya, sistem akan mengubah gambar daun cabai menjadi gambar abu-abu dan mengubah ukurannya. Kemudian, sistem akan mencoba untuk mendapatkan nilai RGB dari setiap pixel pada gambar,

menormalkan nilai, dan membentuk kembali gambar menjadi 128x128.

Selanjutnya, sistem memprediksi gambar daun cabai dengan menggunakan model terlatih dan menunjukkan nilai probabilitasnya. Setelah itu, sistem akan memeriksa nilai probabilitas kedua kelas. Jika nilai probabilitas sehat lebih besar dari probabilitas bintik ungu, maka daun cabai sehat. Sebaliknya, jika nilai probabilitas sehat kurang dari nilai probabilitas titik ungu, maka daun cabai terinfeksi oleh beberapa penyakit dengan jenis penyakit titik ungu.

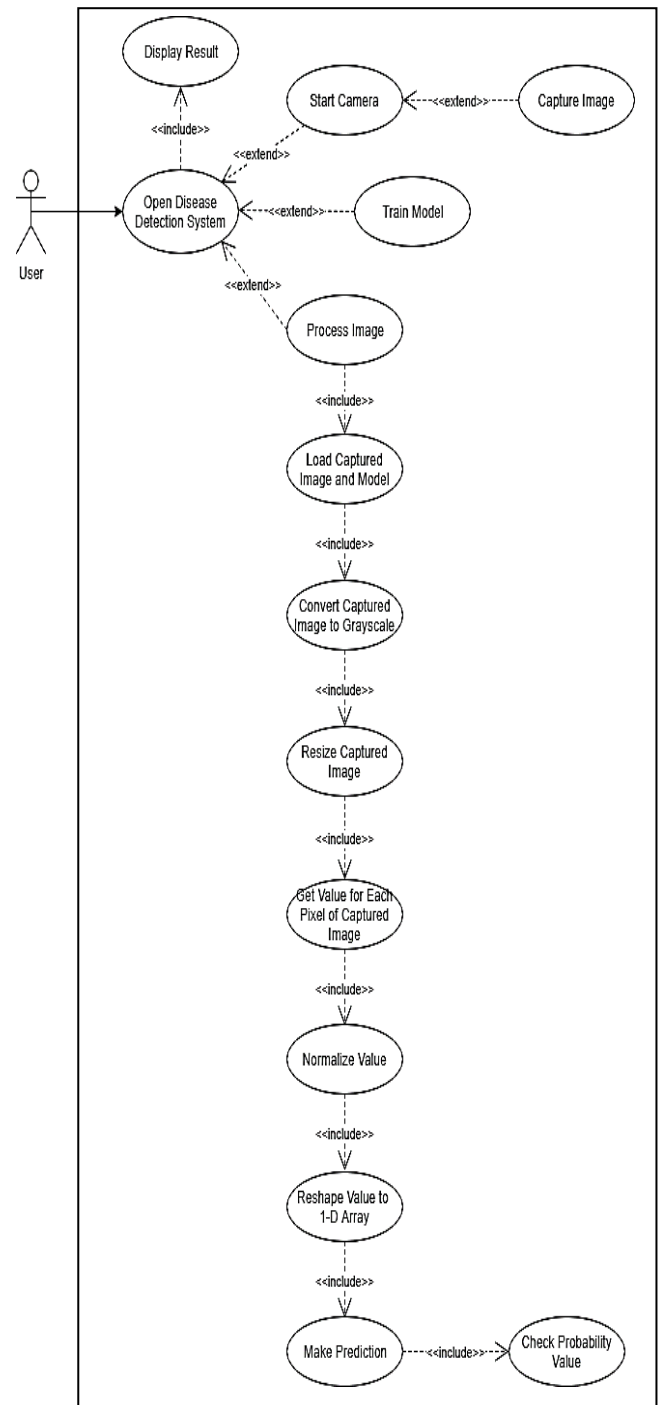
Akhirnya, sistem akan menampilkan hasil yang merupakan gambar yang diambil dengan beberapa deskripsi seperti kondisi daun cabai, jenis penyakit yang menginfeksi daun cabai (jika ada), dan nilai persentase probabilitas sistem dalam mendeteksi cabai gambar daun dengan benar (apakah itu daun cabai sehat atau daun cabai sakit) pada kanvas di antarmuka pengguna.

Untuk mendapatkan hasil terbaik dalam proses deteksi penyakit, berikut hal-hal yang harus dilakukan oleh pengguna:

- Pengguna harus menangkap gambar di lingkungan dengan pencahayaan yang cukup untuk mendapatkan gambar yang layak.
- Jarak antara kamera dan objek tidak boleh lebih dari 1 meter.

Gambar 3 menunjukkan bahwa ada satu aktor dan lima kasus penggunaan utama untuk sistem. Aktor adalah pengguna yang menggunakan sistem. Sementara, use case adalah tindakan yang dapat dilakukan sistem, mulai kamera, menangkap gambar, melatih model, memproses gambar, dan menampilkan hasilnya.

Pada bagian proses gambar, ada beberapa tindakan / proses yang harus dilakukan untuk mendeteksi apakah daun cabai terinfeksi penyakit atau tidak. Pada awalnya, sistem harus memuat gambar dan model yang diambil. Selanjutnya, warna gambar yang diambil akan dikonversi dari RGB ke skala abu-abu. Kemudian, gambar yang diambil akan diubah ukurannya menjadi 128 x 128. Setelah itu, sistem akan mengambil nilai untuk setiap piksel pada gambar yang diambil dan menormalkan nilai itu. Kemudian, sistem akan membentuk kembali nilai itu dari array 2-D ke array 1-D. Akhirnya, sistem akan mencoba membuat prediksi pada gambar yang diambil. Proses ini akan menghasilkan nilai probabilitas sehat dan sakit. Oleh karena itu, sistem akan memeriksa nilai probabilitas antara sehat dan sakit. Setelah selesai, sistem akan menampilkan hasil yang merupakan gambar yang diambil dengan beberapa deskripsi seperti kondisi daun cabai, jenis penyakit yang menginfeksi daun cabai (jika ada), dan nilai probabilitas sistem dalam mendeteksi dengan benar gambar daun cabai (apakah itu daun cabai sehat atau daun cabai sakit).



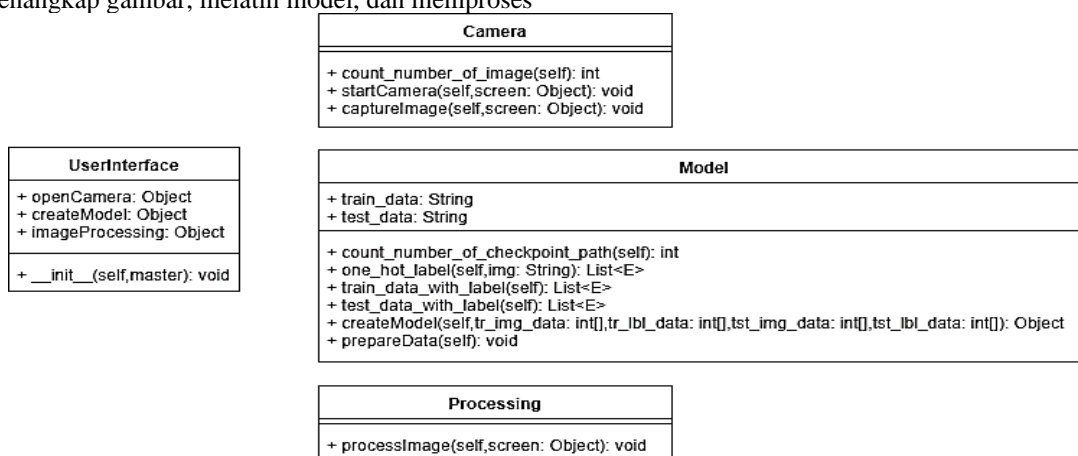
Gambar 3. Diagram use case dari sistem deteksi penyakit pada daun cabai

Desain antarmuka pengguna untuk sistem ini hanya akan berisi satu layar yang merupakan layar antarmuka utama. Detail layar akan dijelaskan lebih lanjut pada Gambar 4.



Gambar 4. Tampilan Antarmuka dari Sistem Deteksi Penyakit pada Daun Cabai

Seperti yang terlihat pada Gambar 4, terdapat empat tombol dan dua kanvas. Kedua kanvas adalah untuk menampilkan gambar input dan gambar output. Keempat tombol untuk memulai kamera, menangkap gambar, melatih model, dan memproses



Gambar 5. Tampilan Class Diagram dari Sistem Deteksi Penyakit pada Daun Cabai

Kelas UserInterface digunakan untuk menangani antarmuka pengguna sistem. Kelas ini hanya berisi satu metode yang digunakan untuk membuat antarmuka pengguna sistem. Antarmuka pengguna sistem terdiri dari dua kanvas dan empat tombol. Kanvas pertama untuk input dan kanvas kedua untuk output. Sedangkan empat tombol untuk memulai kamera, menangkap gambar, melatih model, dan memproses gambar. Setiap tombol memiliki aksi dan fungsinya sendiri.

Kelas Kamera adalah kelas yang digunakan untuk mengakses kamera. Kelas ini berisi tiga metode yang count_number_of_image, startCamera, dan captureImage. Metode pertama adalah count_number_of_image. Metode count_number_of_image digunakan untuk menghitung berapa banyak gambar dengan ekstensi .jpg di folder tertentu. Metode ini berguna saat menyimpan gambar ke Raspberry Pi karena digunakan untuk memberi nama input file gambar. Metode kedua adalah startCamera. Metode startCamera digunakan untuk mengaktifkan kamera Raspberry Pi, mengatur resolusi kamera, menampilkan pratinjau kamera langsung di layar, dan mengaktifkan

tombol "Ambil Gambar". Dan metode terakhir adalah captureImage. Metode captureImage digunakan untuk mengambil gambar, menyimpan gambar yang diambil ke Raspberry Pi, menampilkan gambar yang diambil di kanvas, dan menyimpan jalur gambar yang diambil pada file teks yang akan digunakan nanti saat memproses gambar.

Kelas Model adalah kelas yang digunakan untuk menginisialisasi dan melatih model. Kelas ini berisi enam metode yang count_number_of_checkpoint_path, one_hot_label, train_data_with_label, test_data_with_label, createModel, dan prepData. Metode pertama adalah count_number_of_checkpoint_path. Metode ini digunakan untuk menghitung berapa banyak model dengan ekstensi .ckpt di folder tertentu. Metode ini juga berguna ketika menyimpan model ke Raspberry Pi karena digunakan untuk memberi nama file model. Metode kedua adalah one_hot_label. Metode ini digunakan untuk menambahkan label ke semua gambar dalam dataset berdasarkan nama gambar. Metode ketiga adalah train_data_with_label. Metode ini digunakan untuk memuat semua gambar pelatihan, mengubah semua gambar pelatihan menjadi

gambar skala abu-abu, mengubah ukuran semua gambar pelatihan menjadi ukuran 128 * 128, mendapatkan nilai untuk setiap piksel pada semua gambar pelatihan, menambahkan yang disandikan satu-panas beri label pada semua gambar pelatihan, simpan semua gambar latihan dengan label encode satu-panas ke dalam daftar, dan putar semua gambar latihan untuk memastikan model mendapatkan gambar dari kedua kelas di setiap batchnya. Metode keempat adalah `test_data_with_label`. Metode ini sama dengan metode `train_data_with_label`. Satu-satunya perbedaan adalah gambar pengujian tidak perlu dikocok. Metode kelima adalah `createModel`. Model ini digunakan untuk menginisialisasi model, mendefinisikan struktur model (lapisan konvolusi, lapisan penyatuan maks, lapisan putus sekolah, lapisan ratakan, lapisan padat, dan lapisan yang terhubung penuh), melatih model pada dataset pelatihan, mengevaluasi model pada pengujian dataset, menyimpan model yang terlatih ke Raspberry Pi, dan menyimpan jalur model yang terlatih ke file teks yang akan digunakan nanti saat memproses gambar. Dan metode terakhir adalah `prepData`. Metode ini digunakan untuk menyiapkan data yang akan digunakan untuk pelatihan model. Data tersebut meliputi data pelatihan dan data pengujian.

Pemrosesan Kelas adalah kelas yang digunakan untuk memproses gambar, memprediksi gambar yang diambil berdasarkan model yang dilatih, dan menampilkan hasilnya. Kelas ini hanya berisi satu metode yaitu `processImage`. Metode ini digunakan untuk memproses gambar, memprediksi gambar yang diambil berdasarkan model yang dilatih, dan menampilkan hasilnya pada kanvas. Hasilnya terdiri dari gambar yang diambil dengan beberapa teks / deskripsi seperti kondisi daun cabai, jenis penyakit yang menginfeksi daun cabai (jika ada), dan nilai persentase probabilitas sistem dalam mendeteksi gambar yang diambil dengan benar (apakah itu adalah daun cabai sehat atau daun cabai sakit).

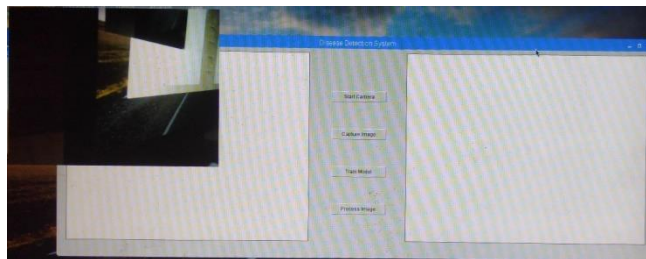
IV. IMPLEMENTASI

Gambar 6 menunjukkan bagaimana membuat kamera Raspberry Pi menjadi mudah diakses dan aktif. Pada awalnya, sistem akan memanggil modul PiCamera dan mengatur resolusi kamera ke 1024x768. Selanjutnya, pratinjau kamera langsung ditampilkan di layar dengan ukuran 500x500. Dan kemudian, sistem akan mengaktifkan tombol "Capture Image".

```
Def startCamaera(self, screen) :  
    self.camera=PiCamera()  
    self.camera.resolution=(1024,768)  
    self.camera.start_preview(fullscreen=False,  
    window=(50,50,500,500))  
    screen.captureImageButton["state"]=NORMAL
```

Gambar 6. Fungsi untuk Memulai Kamera

Gambar 7 menunjukkan pratinjau langsung kamera dari kamera Raspberry Pi. Pratinjau kamera langsung ini akan ditampilkan di layar ketika pengguna mengklik tombol "Starting Camera". Dan juga, ketika pratinjau kamera langsung ditampilkan di layar, tombol "Capture Image" akan diaktifkan.



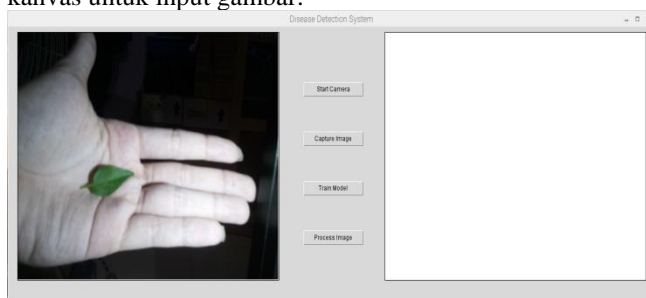
Gambar 7. Tampilan Pratinjau Kamera

Gambar 8 menunjukkan cara menampilkan gambar pada kanvas untuk input gambar. Pada awalnya, sistem akan memuat gambar yang diambil. Selanjutnya, sistem akan mengubah ukuran gambar yang diambil menjadi 575 x 450 dan menampilkannya di kanvas untuk gambar input.

```
self.image = Image.open(self.path)  
self.image = self.image.resize((575,450),Image.ANTIALIAS)  
self.photo = ImageTk.PhotoImage(self.image)  
screen.imageCanvas.create_image(0,0,image=self.photo,anchor=NW)  
screen.imageCanvas.image=self.photo
```

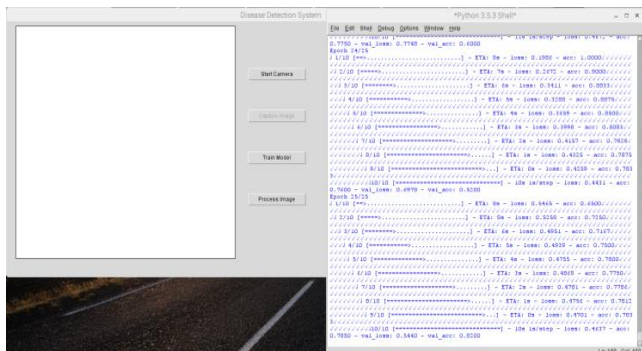
Gambar 8. Fungsi *Displaying Image*

Gambar 9 menunjukkan gambar yang diambil ditampilkan pada kanvas untuk input gambar. Ketika pengguna mengklik tombol "Capture Image", pertama gambar yang diambil akan diubah ukurannya menjadi 575 x 450 piksel. Setelah itu, gambar yang diambil ditampilkan di kanvas untuk input gambar.



Gambar 9. Tampilan *Displaying Image*

Gambar 10 menunjukkan proses pelatihan model. Ketika pengguna mengklik tombol "Train Model", sistem akan secara otomatis melatih model tersebut. Proses pelatihan akan ditampilkan pada kulit python bersama dengan waktu estimasi, akurasi, kehilangan, kehilangan validasi, dan akurasi validasi.



Gambar 10. Tampilan Train Model

Fungsi model pelatihan sistem deteksi penyakit pada daun cabai ditunjukkan pada Gambar 11 hingga Gambar 15. Fungsi ini digunakan untuk memuat semua pelatihan dan menguji gambar dari dataset, pre-process semua gambar itu, memberikan label untuk setiap gambar, menyimpan semua gambar dengan label pada daftar, menginisialisasi model, menentukan struktur model (menambahkan beberapa lapisan), kompilasi model, dan latih model.

```
def one_hot_label(self, img):
    self.label = img.split('.')[0]
    self.ohl = []
    if self.label == 'bercak':
        self.ohl = np.array([1,0])
    elif self.label == 'sehat':
        self.ohl = np.array([0,1])
    return self.ohl
```

Gambar 11. Tampilan Label gambar daun

Gambar 11 menunjukkan cara memberikan label daun untuk semua gambar untuk pengujian dan pelatihan. Fungsi ini digunakan untuk menambahkan label ke semua gambar berdasarkan nama gambar tersebut. Gambar 12 menunjukkan bagaimana mempersiapkan data untuk pelatihan. Pada awalnya, sistem akan memuat semua gambar pelatihan. Selanjutnya, sistem akan memproses ulang gambar pelatihan dengan mengubah gambar menjadi skala abu-abu, mengubah ukuran gambar menjadi 128x128, mengubah gambar menjadi array piksel, dan menambahkan label ke gambar. Dan kemudian, sistem akan mencocok data pelatihan. Terakhir, data pelatihan akan disimpan ke dalam daftar.

```
def train_data_with_label(self):
    self.train_images = []
    for i in os.listdir(train_data):
        for j in os.listdir(os.path.join(train_data,i)):
            self.filepath = os.path.join(train_data,i)
            self.imageFile = cv2.imread(os.path.join(self.filepath,j), cv2.IMREAD_GRAYSCALE)
            self.imageFile = cv2.resize(self.imageFile, (128,128))
            self.train_images.append((np.array(self.imageFile),self.one_hot_label(j)))
    np.random.shuffle(self.train_images)
    return self.train_images
```

Gambar 12. Mempersiapkan Fungsi Data Pelatihan

Gambar 13 menunjukkan bagaimana mempersiapkan data untuk pengujian. Pada awalnya, sistem akan memuat semua gambar pengujian. Selanjutnya, sistem akan memproses ulang gambar pengujian dengan mengubah gambar menjadi skala abu-abu, mengubah ukuran gambar menjadi 128x128, mengubah gambar menjadi array piksel, dan menambahkan label ke dalamnya. Terakhir, data pengujian akan disimpan ke daftar.

```
def test_data_with_label(self):
    self.test_images = []
    for i in os.listdir(test_data):
        for j in os.listdir(os.path.join(test_data,i)):
            self.filepath = os.path.join(test_data,i)
            self.imageFile = cv2.imread(os.path.join(self.filepath,j), cv2.IMREAD_GRAYSCALE)
            self.imageFile = cv2.resize(self.imageFile, (128,128))
            self.test_images.append((np.array(self.imageFile),self.one_hot_label(j)))
    return self.test_images
```

Gambar 13. Mempersiapkan Fungsi Data Pengujian

Gambar 14 menunjukkan cara membuat model dan melatih model secara otomatis di Raspberry Pi. Pada awalnya, sistem akan menginisialisasi model. Selanjutnya, sistem akan menentukan lapisan pertama, lapisan kedua, dan lapisan yang terhubung penuh. Lapisan pertama dan lapisan kedua terdiri dari lapisan 2D Konvolusional dan lapisan 2D Max-Pooling. Sedangkan lapisan yang sepenuhnya terhubung terdiri dari Dropout Layer, Flatten Layer, dan Dense Layer. Kemudian, sistem akan menginisialisasi fungsi kerugian yaitu binary crossentropy, optimizer yaitu Adam Optimizer dengan tingkat pembelajaran = 0,001, dan nilai metrik yang akurat. Terakhir, sistem akan menyusun model, melatih model berdasarkan kumpulan data pelatihan, dan mengevaluasi atau memvalidasi model berdasarkan pada kumpulan data pengujian.

```
def createModel(self, tr_img_data, tr_lbl_data, tst_img_data, tst_lbl_data):  
    self.model = tf.keras.Sequential()  
  
    self.model.add(tf.keras.layers.Conv2D(filters=16, kernel_size=3, strides=1,  
                                           padding='same', activation='relu', input_shape=(128,128,1)))  
    self.model.add(tf.keras.layers.MaxPooling2D(pool_size=2,padding='same'))  
  
    self.model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, strides=1,  
                                           padding='same', activation='relu'))  
    self.model.add(tf.keras.layers.MaxPooling2D(pool_size=2,padding='same'))  
  
    self.model.add(tf.keras.layers.Dropout(0.25))  
    self.model.add(tf.keras.layers.Flatten())  
    self.model.add(tf.keras.layers.Dense(256, activation='relu'))  
    self.model.add(tf.keras.layers.Dropout(rate=0.5))  
    self.model.add(tf.keras.layers.Dense(2, activation='sigmoid'))  
  
    self.opt = tf.keras.optimizers.Adam(lr=0.001)  
    self.model.compile(loss='binary_crossentropy',  
                      optimizer=self.opt,  
                      metrics=['accuracy'])  
  
    self.model.fit(tr_img_data, tr_lbl_data, batch_size=50, epochs=100,  
                  validation_data = (tst_img_data, tst_lbl_data))  
  
    return self.model
```

Gambar 14. Fungsi Pembuatan Model Pengujian

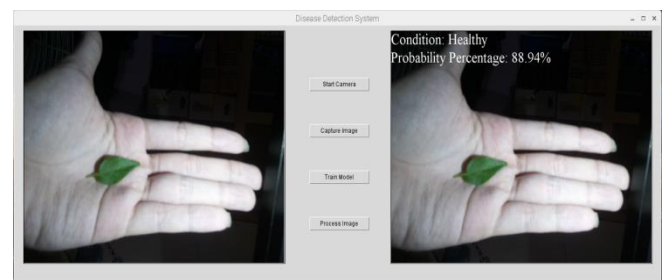
Gambar 15 menunjukkan bagaimana mempersiapkan data pelatihan dan menguji data. Pada awalnya, sistem akan mempersiapkan data untuk pelatihan dan pengujian. Selanjutnya, data pelatihan dan data pengujian akan dinormalisasi, dibentuk kembali menjadi array 1-D dengan ukuran 128 x 128 piksel, dan disimpan dalam variabel yang berbeda. Setelah itu, label data pelatihan dan label data pengujian juga akan disimpan dalam variabel yang berbeda. Setelah itu, sistem akan membuat model dengan menggunakan data itu.

```
def prepareData(self):  
    self.training_data = self.train_data_with_label()  
    self.testing_data = self.test_data_with_label()  
  
    self.tr_img_data = (np.array([i[0] for i in self.training_data],  
                                 dtype="float")/255.0).reshape(-1,128,128,1)  
    self.tr_lbl_data = np.array([i[1] for i in self.training_data])  
  
    self.tst_img_data = (np.array([i[0] for i in self.testing_data],  
                                 dtype="float")/255.0).reshape(-1,128,128,1)  
    self.tst_lbl_data = np.array([i[1] for i in self.testing_data])  
  
    self.model = self.createModel(self.tr_img_data,self.tr_lbl_data,  
                                  self.tst_img_data,self.tst_lbl_data)
```

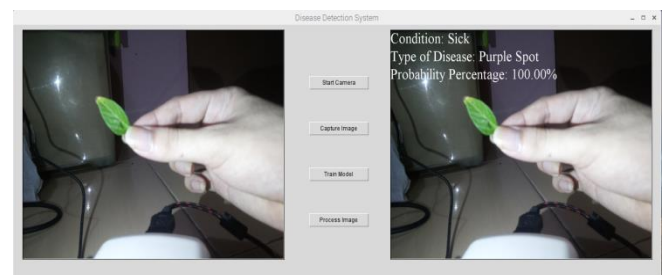
Gambar 15. Fungsi Persiapan Data Pelatihan

Gambar 16 dan Gambar 17 menunjukkan hasil pemrosesan gambar. Ketika pengguna mengklik tombol "Proses Gambar", sistem akan mulai memproses gambar dengan memprediksi gambar yang diambil apakah itu daun cabai sehat atau daun cabai sakit berdasarkan model yang dilatih. Ketika proses dilakukan, gambar hasil dengan deskripsi tentang kondisi daun cabai, jenis penyakit yang menginfeksi daun cabai (jika ada), dan probabilitas

persentase nilai sistem dalam mendeteksi dengan benar gambar yang ditangkap (apakah itu adalah daun cabai sehat atau daun cabai sakit) akan ditampilkan di kanvas untuk gambar keluaran.



Gambar 16. Menampilkan Hasil (Daun Cabai Sehat)



Gambar 17. Menampilkan Hasil (Daun Cabai Sakit)

A. Dataset

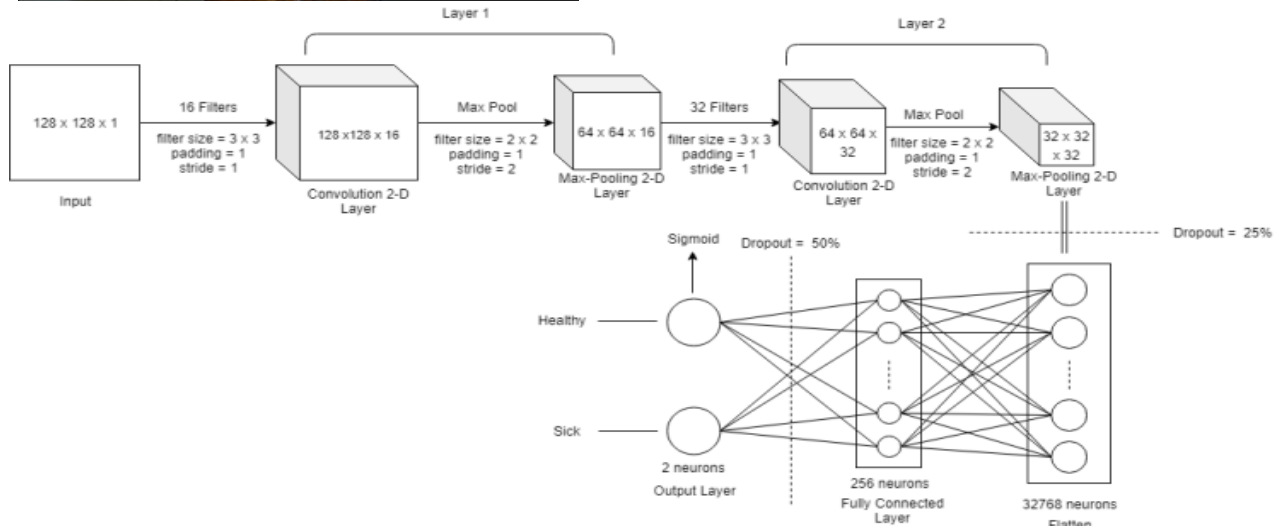
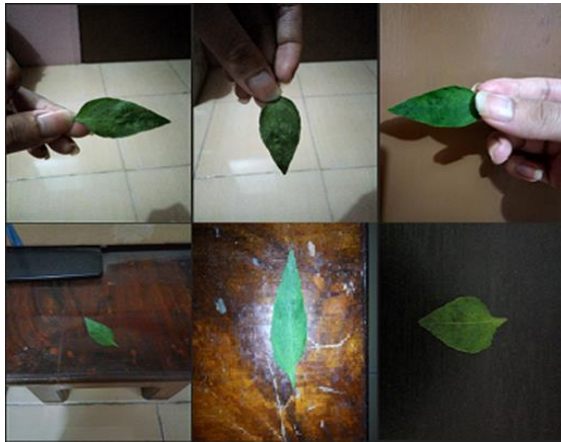
Untuk mengembangkan sistem ini, diperlukan dataset. Kumpulan data untuk sistem ini terdiri dari 1000 gambar yang berasal dari gambar koleksi pribadi, yang terdiri dari 500 gambar untuk daun cabai sehat dan 500 gambar untuk daun cabai sakit. Dari 500 gambar untuk daun cabai sehat dan daun cabai sakit, 100 gambar dari daun cabai sehat dan daun cabai sakit akan digunakan untuk menguji data,

sedangkan sisanya akan digunakan untuk data pelatihan. Kumpulan data untuk sistem ini ditunjukkan pada Gambar 18 dan Gambar 19. Kumpulan data ini akan digunakan untuk data pelatihan dan data pengujian saat melatih dan mengevaluasi model.

Gambar 18. Dataset Daun Cabai Sehat



Gambar 19. Dataset Daun Cabai Sakit



Gambar 20. Model Jaringan Saraf Sistem

Gambar 20 menunjukkan model jaringan saraf yang diimplementasikan pada sistem deteksi penyakit pada daun cabai. Pada awalnya, gambar input dengan bentuk $128 \times 128 \times 1$ diumpangkan ke lapisan pertama dan lapisan kedua yang terdiri dari lapisan Konvolusi 2-D dan lapisan Max-Pooling 2-D. Pada lapisan Konvolusi 2-D pertama, gambar input akan disaring dengan menggunakan ukuran filter 3×3 , jumlah filter = 16, padding = 1, dan langkah = 1. Bentuk output dari lapisan ini adalah $128 \times 128 \times 16$. Selanjutnya, output ini akan dimasukkan ke layer 2-D Max-Pooling untuk mengurangi ukuran dimensi, perhitungan, dan overfitting dengan ukuran filter 2×2 , padding = 1, dan langkah = 2. Bentuk output dari lapisan ini akan menjadi $64 \times 64 \times 16$. Kemudian, output ini akan diumpangkan ke lapisan 2-D Konvolusi kedua dengan ukuran filter 3×3 , jumlah filter = 32, bantalan = 1, dan langkah = 1. Dan

bentuk output dari layer ini adalah $64 \times 64 \times 32$. Setelah itu, output ini akan diumpangkan kembali ke layer 2-D Max-Pooling untuk mengurangi ukuran dimensi, perhitungan, dan overfitting lagi dengan ukuran filter 2×2 , padding = 1, dan langkah = 2. Bentuk output dari lapisan ini akan menjadi $32 \times 32 \times 32$. Selanjutnya, 25% dari output ini akan dikeluarkan dan output ini akan rata dari m array 2-D ke array 1-D. Proses ini akan menghasilkan 32.768 neuron. Kemudian, dua lapisan yang terhubung sepenuhnya ditambahkan pada akhirnya. Lapisan sepenuhnya terhubung pertama berisi 256 neuron, sedangkan lapisan terhubung sepenuhnya kedua hanya berisi 2 neuron yang mewakili kemungkinan gambar menjadi daun cabai sehat atau daun cabai sakit.

V. SISTEM TESTING

UCAPAN TERIMA KASIH

Agar sistem berfungsi dengan baik, ada tiga batasan dalam proses pengujian. Batasan pertama berkaitan dengan cahaya. Untuk mendapatkan gambar yang layak dan jelas, the Kamera Raspberry Pi harus ditempatkan di lingkungan dengan pencahayaan yang cukup. Karena kondisi ini akan membuat sistem untuk dapat memprediksi gambar yang diambil secara akurat. Keterbatasan kedua adalah jarak. Itu jarak antara kamera Raspberry Pi dan daun cabai tidak boleh lebih dari 1 meter untuk membuat sistem untuk dapat memprediksi gambar yang diambil secara akurat. Karena jika jaraknya melebihi 1 meter, ukuran daun cabai menyala gambar akan menjadi lebih kecil dan sistem tidak akan mampu memprediksi gambar yang diambil secara akurat. Yang terakhir Keterbatasan berkaitan dengan objek. Objek yang digunakan untuk pengujian harus daun cabai. Karena sistem hanya bisa mendeteksi objek daun cabai untuk mendapatkan hasil akurat. Jika objeknya bukan daun cabai, sistemnya tidak boleh bisa mendapatkan hasilnya secara akurat. Hasil pengujian terlihat pada Tabel I.

TABEL I.
TABEL HASIL PENGUJIAN

Skenario Pengujian	Hasil Pengujian
Fungsi Utama (Penerangan yang Cukup dan Jarak < 1 meter)	100%
Fungsi Utama (Penerangan yang Cukup dan Jarak >= 1 meter)	72%
Fungsi Utama (Penerangan yang kurang dan Jarak < 1 meter)	85,2%
Fungsi Utama (Penerangan yang kurang dan Jarak >= 1 meter)	68,8%

VI. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *Deep Learning* untuk melakukan pemrosesan gambar daun cabai untuk mengetahui bahwa daun cabai tersebut sehat atau terdeteksi penyakit, dengan menggunakan gambar daun cabai yang berasal dari gambar koleksi pribadi. Telah dilakukan pengujian dan berdasarkan hasil pengujian tersebut diketahui bahwa 100% Fungsi Utama (Penerangan yang cukup dan jarak < 1 meter) sistem dapat dengan baik mengenali daun cabai dan dapat mendeteksi apakah daun cabai tersebut terinfeksi penyakit atau tidak, namun dengan fungsi Utama dimana penerangan yang kurang dan jarak >= 1 meter, akurasi sistem hanya sebesar 68,8%.

Penelitian ini terlaksana atas dana yang diberikan oleh Ristekdikti dalam Program Penelitian Dosen Pemula berdasarkan Surat Keputusan Nomor B/87/E3/RA.00/2020 dan perjanjian kontrak nomor 080/SP2H/AMD/LT/DRPM/2020, 010/SP2H/AMD/LT-MONO/LL4/2020, 090/LRPM-PDP/VII/PresUniv/2020

DAFTAR PUSTAKA

- Ralalahu, M. S., Hehanusa, M. L., and Oszaer, L.L., "Respon Tanaman Cabai Besar (*Capsicum annum L*) Terhadap Pemberian Pupuk Organik Hormon Tanaman Unggul", *Jurnal Ilmu Budaya Tanaman*, vol. 2, no.2, Oktober 2013.
- Muslim, "Sistem Pakar Diagnosa Hama Dan Penyakit Cabai Berbasis Teorema Bayes", *Jutisi : Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, Vol. 4, No. 3, Desember 2015, pp. 797 –876, 2015.
- J. R. Aduwo, E. Mwebaze, and J. A. Quinn, "Automated vision-based diagnosis of cassava mosaic disease", *Industrial Conference on Data Mining - Workshops*, 2010, pp. 114–122.
- E. Mwebaze and M. Biehl, "Prototype-based classification for image analysis and its application to crop disease diagnosis", *Advances in Self-Organizing Maps and Learning Vector Quantization - Proceedings of the 11th International Workshop WSOB*, 2016, pp. 329–339.
- E. Mwebaze, P. Schneider, F.-M. Schleif, J. Aduwo, J. Quinn, S. Haase, T. Villmann, and M. Biehl, "Divergence-based classification in learning vector quantization", *Neurocomputing*, vol. 74, no. 9, pp. 1429 – 1435, 2011.
- E. Mwebaze, M. Biehl, G. Bearda, and D. Zuehlke, "Combining dissimilarity measures for prototype based classification", *European Symposium on Artificial Neural Networks (ESANN)*, Vol. 23, pp. 31–36, 2015.
- G. Owumugisha and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images", *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 158–163.
- Konstantinos P. Ferentinos, "Deep learning models for plant disease detection and diagnosis", *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- M. Karuna, B. S Varsha, Sneha. R. M, G. K. Meghana, "Early Detection of Chili Plant Leaf Diseases using Machine Learning", *IJESC*, Vol.9, No.5, pp. 22328- 22335, 2019.
- Brahimi M., Arsenovic M., Laraba S., Sladojevic S., Boukhalfa K., Moussaoui A, *Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation*, Human-Computer Interaction Series. Springer, Cham. pp. 93-117, 2018.
- Bush, S. (2011) Dongle computer lets kids discover programming on a TV. [Online]. Tersedia: <https://www.electronicweekly.com/market-sectors/embedded-systems/dongle-computer-lets-kids-discover-programming-on-a-2011-05>
- Verry, T. (2018) What is the Raspberry Pi? [Online]. Tersedia: <https://www.extremetech.com/computing/124317-what-is-raspberry-pi-2>
- Richardson, M., & Wallace, S. *Getting Started with Raspberry Pi*, Maker Media, pp. 8-9, 2016.
- Anbarjafari. (2014) Digital Image Processing. University of Tartu. [Online]. Tersedia: <https://sisu.ut.ee/imageprocessing/book/1>
- Ertam, F., & Aydin, G. "Data Classification with Deep Learning using Tensorflow", *2nd International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 755-758.
- LeCun, Y., Bengio, Y., & Hinton, G, "Deep Learning", *Nature*, vol. 521, pp. 436, 2015