

**IDENTIFIKASI PENYAKIT *DOWNY MILDEW* DAN *WATERMELON*
MOSAIC VIRUS PADA TANAMAN SEMANGKA DENGAN
MENGUNAKAN ALGORITMA *SSD MOBILENET V2*
BERBASIS *MOBILE***

SKRIPSI

FAKHRI RIZHA ANANDA

171402043



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

IDENTIFIKASI PENYAKIT *DOWNY MILDEW* DAN *WATERMELON*
MOSAIC VIRUS PADA TANAMAN SEMANGKA DENGAN
MENGUNAKAN ALGORITMA *SSD MOBILENET V2*
BERBASIS *MOBILE*

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

FAKHRI RIZHA ANANDA

171402043



PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

PERSETUJUAN

Judul : IDENTIFIKASI PENYAKIT *DOWNY*
MILDEW DAN *WATERMELON MOSAIC*
VIRUS PADA TANAMAN SEMANGKA
DENGAN MENGGUNAKAN ALGORITMA
SSD MOBILENET V2 BERBASIS *MOBILE*

Kategori : SKRIPSI

Nama : FAKHRI RIZHA ANANDA

Nomor Induk Mahasiswa : 171402043

Program Studi : SARJANA (S-1) TEKNOLOGI INFORMASI

Fakultas : FAKULTAS ILMU KOMPUTER DAN
TEKNOLOGI INFORMASI

Medan, 10 Juli 2024

Komisi Pembimbing:

Pembimbing 2,

Ivan Jaya S.Si., M.Kom.
NIP. 198407072015041001

Pembimbing 1,

Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc.
NIP. 198603032010121004

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,

Dedy Ardiyaningrum, M.Kom.
NIP. 97908311009121002



PERNYATAAN

IDENTIFIKASI PENYAKIT *DOWNY MILDEW* DAN *WATERMELON MOSAIC VIRUS* PADA TANAMAN SEMANGKA DENGAN MENGGUNAKAN ALGORITMA *SSD MOBILENET V2* BERBASIS *MOBILE*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Juli 2024

Fakhri Rizha Ananda
171402043

UCAPAN TERIMAKASIH

Puji dan syukur penulis ucapkan kepada Allah SWT, Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer, pada Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Dalam proses untuk menyelesaikan skripsi ini, penulisan telah banyak mendapatkan berbagai bimbingan, dukungan, maupun bantuan dan disertai doa dari berbagai pihak. Adapun dengan hal ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Diri penulis, yang telah berjuang dan tetap teguh dan semangat, baik untuk memotivasi diri sendiri dan tidak menyerah dalam pengerjaan skripsi ini sehingga akhirnya dapat dilalui.
2. Orangtua Tercinta, Bapak Didi Irawan dan Ibu Kurniati, yang telah memberikan banyak kasih sayang, serta doa dan dukungan sejak usia dini hingga beranjak ke pendidikan tinggi, begitu juga adik penulis yaitu Irsyad Dwi Kurniawan.
3. Keluarga Besar penulis yang telah memberikan doa serta dukungan kepada penulis.
4. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologin Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi S.T., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara
6. Bapak Ivan Jaya S.Si, M.Kom. selaku Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
7. Bapak Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc. selaku dosen pembimbing pertama penulis yang telah banyak membantu dan membimbing penulis dalam penelitian dan penulisan skripsi ini.

8. Bapak Ivan Jaya S.Si, M.Kom. selaku dosen pembimbing kedua penulis yang telah banyak membantu dan membimbing penulis dalam penelitian dan penulisan skripsi ini.
9. Bapak Dedy Arisandi ST., M.Kom selaku dosen pembimbing pertama penulis yang telah memberikan masukan dan saran dalam penyempurnaan penulisan skripsi ini.
10. Ibu Fanindia Purnamasari S.TI., M.IT selaku dosen pembimbing kedua penulis yang telah memberikan masukan dan saran dalam penyempurnaan penulisan skripsi ini.
11. Seluruh staff dan civitas academica dalam lingkungan Fasilkom-TI USU yang membantu penulis selama masa perkuliahan.
12. Teman-teman dekat penulis yaitu Ammar, Rizal, dan Alvinza yang telah kebersamai penulis sejak SMP hingga sekarang.
13. Teman-teman FRABS yaitu Fahmi, Faisal, Fakur, Bagus, Jon, Jalo, Syofie, Ratih, Nisa, dan Sisi yang banyak memberikan dukungan kepada penulis dalam penyelesaian skripsi ini.
14. Teman-teman semasa perkuliahan penulis yakni Jackie, Dinul, Muharriz, Ibnu, Rafif, Rizki Fatihah, Baskoro, Ali, dan Baihaqi yang telah membantu dan mendukung serta memberikan semangat dalam masa perkuliahan.
15. Teman-teman di Pemerintahan Mahasiswa periode 2019/2020 yang telah banyak membantu dalam penyelesaian skripsi penulis.
16. Teman-teman di HIMATIF-USU yang telah banyak membantu dalam penyelesaian skripsi penulis.
17. Teman-teman di Information Technology Laboratory Group (ITLG) USU.
18. Keluarga Teknologi Informasi Angkatan 2017 Fasilkom-TI USU yang telah kebersamai penulis semasa kuliah.
19. Teman-teman SMA Negeri 13 Medan yang telah kebersamai masa-masa SMA penulis.
20. Teman-teman SMP Negeri 2 Medan yang telah menemani penulis di masa SMP.

21. Teman-teman SD Al-Azhar Medan yang memberikan doa dan dukungan di masa Sekolah Dasar penulis.
22. Kepada abang, kakak, adik, serta rekan lainnya yang ikut serta dalam mendukung penulis untuk menyelesaikan penulisan skripsi ini.

Penulis sadar bahwa masih banyak terdapat kekurangan dalam penulisan dan perancangan skripsi ini, oleh karenanya penulis berharap dapat diberikan kritik ataupun saran yang konstruktif dalam menyempurnakan skripsi ini.

Medan, Juni 2024

Penulis

**IDENTIFIKASI PENYAKIT *DOWNY MILDEW* DAN *WATERMELON*
MOSAIC VIRUS PADA TANAMAN SEMANGKA DENGAN
MENGUNAKAN ALGORITMA *SSD MOBILENET V2*
BERBASIS *MOBILE***

ABSTRAK

Semangka (*Citrullus lanatus*) adalah salah satu jenis komoditas buah dengan produksi tinggi di Indonesia. Dalam proses pembudidayaan buah semangka, ada beberapa penyakit yang lazim menyerang tanaman semangka, diantaranya adalah *Downy Mildew* dan *Watermelon Mosaic Virus*. Penyakit ini memiliki gejala awal yang mirip, yakni terdapat bercak putih di sekujur daun, meskipun disebabkan oleh dua hal yang berbeda, yaitu Jamur dan Virus. Penyakit ini dapat dilihat dari bentuk dan warna pada daun semangka. Gejala awal pada kedua penyakit tersebut yakni bercak putih di sekujur daun. Pengujian dilakukan dengan menggunakan citra daun semangka yang bersumber dari set data *public Kaggle*. Hasilnya menunjukkan bahwa *SSD MobileNet V2* menghasilkan tingkat akurasi yang tinggi, dimana rata-rata akurasi keseluruhan mencapai 91.6%. Metode dalam aplikasi ini memiliki potensi untuk diterapkan dalam industri pertanian semangka guna mendeteksi sejak dini penyakit *Downy Mildew* dan *Watermelon Mosaic Virus*.

Kata Kunci: Identifikasi penyakit semangka, *Downy Mildew*, *Watermelon Mosaic Virus*, *SSD MobileNet V2*, *Charleston Grey*

**IDENTIFICATION OF DOWNY MILDEW AND WATERMELON MOSAIC
VIRUS DISEASE ON WATERMELON PLANT WITH MOBILE-BASED
SSD MOBILENET V2 ALGORITHM**

ABSTRACT

Watermelon (Citrullus lanatus) is a type of fruit commodity with high production in Indonesia. In the process of cultivating watermelon, there are several diseases that commonly attack watermelon plants, including Downy Mildew and Watermelon Mosaic Virus. This disease has similar initial symptoms, namely white spots all over the leaves, although it is caused by two different things, namely fungi and viruses. This disease can be seen from the shape and color of the watermelon leaves. The initial symptoms of both diseases are white spots all over the leaves. Testing was carried out using images of watermelon leaves sourced from the public Kaggle data set. The results show that the MobileNet V2 SSD produces a high level of accuracy, where the overall accuracy average reaches 91.6%. The method in this application has the potential to be applied in the watermelon farming industry to detect early Downy Mildew and Watermelon Mosaic Virus diseases.

Keywords: *Watermelon diseases identification, downy mildew, watermelon mosaic virus, SSD MobileNet V2, Charleston Gray*

DAFTAR ISI

PERSETUJUAN	i
PERNYATAAN	ii
UCAPAN TERIMAKASIH	iii
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	8
2.1 Semangka	8
2.2 Jenis Penyakit Semangka	8
2.2.1 Semangka	8
2.2.2 <i>Downy Mildew</i>	9
2.2.3 <i>Watermelon Mosaic Virus</i>	10
2.3 Pengolahan Citra	10
2.3.1 <i>Google Colab</i>	11
2.3.2 <i>Tensorflow</i>	12
2.3.3 Deteksi Objek	12
2.3.4 <i>MobileNet</i>	13
2.3.5 <i>SSD (Single Shot Multibox Detector)</i>	14
2.3.6 <i>SSD-MobileNet</i>	14
2.4 Citra	14
2.4.1 <i>RGB Image</i>	15
2.4.2 <i>Grayscale Image</i>	15
2.4.3 <i>Binary Image</i>	15

2.5	Penelitian Terdahulu	16
BAB 3	ANALISIS DAN PERANCANGAN SISTEM	21
3.1	Data yang Digunakan	21
3.2	Arsitektur Umum	22
3.2.1	<i>Image Acquisition</i>	23
3.2.2	<i>Image Preprocessing</i>	24
3.2.2.1	<i>Augmentation</i>	24
3.2.2.2	<i>Resizing</i>	24
3.2.2.3	<i>Labeling</i>	25
3.2.2.4	<i>Convert TFRecord</i>	25
3.2.3	<i>Image Identification</i>	26
3.2.3.1	<i>Extract Feature Maps</i>	26
3.2.3.2	<i>Convolution Filters</i>	28
3.2.4	<i>Learned Model</i>	28
3.2.5	<i>TFLite Model</i>	29
3.2.6	<i>Proses Training</i>	30
3.2.7	<i>Proses Testing</i>	32
3.2.8	<i>Output</i>	32
3.3	<i>Flowchart Sistem</i>	32
3.4	Perancangan Antarmuka	33
BAB 4	IMPLEMENTASI DAN PENGUJIAN	33
4.1	Implementasi Sistem	33
4.1.1	Perangkat Keras dan Perangkat Lunak	33
4.1.2	Implementasi Data	33
4.1.3	Implementasi Antarmuka	34
4.2	Prosedur Operasional	35
4.3	Pengujian Sistem	36
BAB 5	KESIMPULAN DAN SARAN	48
5.1	Kesimpulan	48
5.2	Saran	48
DAFTAR PUSTAKA		50

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	18
Tabel 3. 1 Jumlah data pelatihan dan data pengujian.....	22
Tabel 4. 1 Hasil Pengujian	36
Tabel 4. 2 <i>Confusion Matrix</i>	46
Tabel 4. 3 Nilai TP, FP dan FN dari Tanaman Semangka.....	47
Tabel 4. 4 Nilai <i>Precision</i> , <i>Recall</i> dan <i>F1-Score</i>	47

DAFTAR GAMBAR

Gambar 2. 1 Semangka Charleston Gray (<i>id.chinavegetableseeds.com</i>).....	9
Gambar 2. 2 Daun Yang Terserang Downy Mildew	9
Gambar 2. 3 Daun Yang Terserang Mosaic Virus	10
Gambar 2. 4 Posisi Kordinat Citra (<i>www.agussetiawanpcd.wordpress.com</i>).....	11
Gambar 2. 5 Arsitektur MobileNet (<i>www.researchgate.com</i>)	13
Gambar 3. 1 Data Daun Semangka (<i>www.depositphotos.com</i>).....	21
Gambar 3. 2 Arsitektur Umum	23
Gambar 3. 3 Foto Asli (Kiri), 4 Objek Prediksi Tiap Sel atau Lokasi (Kanan)....	28
Gambar 3. 4 <i>Flowchart System</i>	33
Gambar 3. 5 Rancangan Antarmuka Aplikasi.....	34
Gambar 4. 1 Data Citra Daun Semangka	34
Gambar 4. 2 Tampilan Halaman Identifikasi	35
Gambar 4. 3 Prosedur Operasional	35

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah salah satu dari beberapa negara yang berlokasi di garis lintang khatulistiwa. Hal inilah yang membuat Indonesia memiliki iklim tropis. Wilayah yang memiliki iklim tropis pada umumnya mendapatkan sinar matahari sepanjang waktu. Semangka (*Citrullus lanatus*), merupakan salah satu buah dari tanaman yang tumbuh dengan baik di daerah tropis maupun subtropis, dan juga memiliki masa panen yang cukup singkat, yakni hanya 6 bulan (Puspitasari et al., 2014).

Buah semangka merupakan satu dari sekian banyak buah yang lazim dikonsumsi masyarakat Indonesia dikarenakan cita rasa yang manis, disertai dengan banyaknya kandungan air. Semangka umumnya dikonsumsi dengan dimakan segar ataupun diolah menjadi jus. Semangka adalah buah yang tidak mengandung lemak karena kadar gulanya terbatas, tetapi kaya akan air. Air dan kalium dalam semangka penuh dengan antioksidan dan vitamin C. Buah ini juga mengandung sitrullin dan karotenoid yang memiliki likopen sebagai antioksidan untuk melawan radikal bebas. (Daniel, 2016).

Menurut data yang dikumpulkan oleh Badan Pusat Statistik (BPS), jumlah semangka yang di produksi di Indonesia sejumlah 414.242 ton pada tahun 2021. Jumlah ini cenderung menurun 26,07% jika dilihat dari produksi pada setahun sebelumnya yakni sebesar 560.317. Jumlah produksi semangka juga mengalami fluktuasi dalam 10 tahun belakang, dengan produksi tertinggi sebesar 653.995 ton pada 2014 dan terendah sebanyak 414.242 pada 2021.

Dari segi budidaya, menangani semangka ternyata tidaklah mudah. Hampir semua petani yang menanam buah musiman ini selalu menghadapi tantangan, seperti penyakit dan juga serangan hama. Kurangnya pengetahuan petani mengenai berbagai jenis penyakit semangka juga turut menjadi salah satu masalah utama di

lapangan, yang seringkali mengakibatkan diagnosa yang kurang tepat dan penanganan yang terlambat terhadap tanaman yang terinfeksi (Handayani, 2017).

Penyakit pada tanaman semangka dapat berasal dari banyak sumber, yaitu jamur, bakteri, serangga, nematoda, maupun virus. Secara umum, penyakit pada tanaman semangka dapat teridentifikasi lebih awal pada bagian daun, meskipun beberapa dari penyakit tersebut dapat menunjukkan gejala pada bagian tanaman lainnya, baik pada tangkai maupun buah.

Penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* (WMV) merupakan salah satu penyakit yang cukup sering menjangkit tanaman semangka. Kedua penyakit ini juga memiliki karakteristik atau gejala yang hampir sama pada tahap awal, yang ditandai dengan bercak-bercak putih pada sekujur daun.

Mariana (2019) dengan judul “Sistem Pakar Diagnosis Penyakit Tanaman Semangka Menggunakan Metode *Certainty Factor*” mendapati bahwa dari 11 responden yang terdiri dari 10 petani dan 1 orang pakar, yang menghasilkan kesimpulan bahwa 97% petani setuju dan 3% kurang setuju, sedangkan pakar menghasilkan 100% persetujuan. Penelitian ini menggunakan 10 penyakit dengan total 26 gejala sebagai sumber data.

Handayani et al. (2017) dengan judul “Sistem Pakar Diagnosa Penyakit Tanaman Semangka Menggunakan Metode *Dempster Shafer* Berbasis Web” dengan menganalisis 9 penyakit pada tanaman semangka dengan total 35 gejala mendapati bahwa uji sistem pakar menunjukkan bahwa dari total 21 data uji terdapat persentase 86%. Kemudian dari hasil kuesioner, didapati bahwa 75% *user* setuju aplikasi dibuat sesuai kebutuhan, 72.73% setuju aplikasi dapat mendeteksi penyakit dalam waktu singkat, 87.1% setuju bahwa gejala pada basis data sudah sesuai, 66.7% sangat setuju bahwa aplikasi mempermudah *user* dalam mengidentifikasi penyakit, 87.1% setuju aplikasi menghasilkan diagnosa yang akurat.

Kemudian penelitian yang dilakukan Arora et al. (2024) membuat sebuah sistem untuk mengevaluasi tingkat keseriusan WMV (*Watermelon Mosaic Virus*) pada semangka dengan menggunakan algoritma *Hybrid RNN* dan *Random Forest*. Penelitian dilakukan dengan menggabungkan kedua model tersebut dan

menghasilkan 99.8% hasil yang baik dari total lebih dari 16.000 gambar dengan tingkat WMV yang berbeda-beda.

Selanjutnya, penelitian yang dilakukan Banerjee et al. (2023) menggunakan algoritma CNN dan *SVM-based model* dalam klasifikasi penyakit pada semangka. Ada total 8 penyakit semangka yang akan diprediksi oleh model ini, yaitu *Bacterial Fruit Blotch*, *Anthrachnose*, *Gummy Rope Blight*, dan lainnya. Kedelapan penyakit tersebut menggunakan daun semangka sebagai objek utama dalam pembuatan modelnya. Hasil yang didapati bahwa model ini memiliki *micro-average precision* sebesar 76% dan *weighting-average precision* sebesar 70.26%.

Penelitian yang dilakukan Chan et al. (2022) membuat sebuah sistem untuk deteksi wajah dengan menggunakan *SSD MobileNet V2* dan *Geometrical Algorithm*. Algoritma ini terbukti unggul dengan tingkat akurasi 91.5% dibandingkan *OpenCV*, namun masih kurang akurat dibandingkan algoritma unggul lainnya untuk deteksi wajah seperti MTCNN dan Dlib. Meskipun begitu, *SSD MobileNet V2* terbukti untuk membantu mengurangi tingkat *over-detection* dan *misdetetection*.

Selanjutnya, penelitian yang dilakukan Alfarizi et al. (2023) membuat sistem deteksi *realtime* untuk mengenali kucing dalam sebuah sistem pengusir kucing dengan menggunakan algoritma *SSD MobileNet V2* dengan tingkat keberhasilan sebesar 82%. Meskipun begitu, faktor-faktor alamiah seperti jarak, kecerahan, dan resolusi kamera dapat mempengaruhi keakuratan model tersebut.

Oleh karena itu, penulis melakukan sebuah penelitian dengan judul: “IDENTIFIKASI PENYAKIT *DOWNY MILDEW* DAN *WATERMELON MOSAIC VIRUS* PADA TANAMAN SEMANGKA DENGAN MENGGUNAKAN ALGORITMA *SSD MOBILENET V2* BERBASIS *MOBILE*”.

1.2 Rumusan Masalah

Sebagai salah satu komoditas buah yang lazim dikonsumsi masyarakat di Indonesia, semangka memiliki beberapa permasalahan dalam proses budidayanya, salah satunya adalah penyakit. Beberapa penyakit pada tanaman ini juga kerap

mempengaruhi kualitas dan kuantitas produksi semangka di Indonesia. Salah satu dari sekian penyakit yang kerap kali menyerang tanaman semangka adalah *Downy Mildew* dan *Watermelon Mosaic Virus*. Kedua penyakit ini pada tahap awal memiliki karakteristik dan gejala yang mirip, yakni terdapat bercak putih di sekujur daun, meskipun disebabkan oleh hal yang berbeda. Akibatnya, beberapa petani salah melakukan tindakan pencegahan yang akhirnya bisa mengurangi kualitas dan jumlah hasil panen. Oleh karena itu, agar petani lebih mudah dalam mengenali perbedaan penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada tanaman semangka, diperlukan suatu sistem yang dapat mengidentifikasi dan mengenali penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada tanaman semangka dengan akurat, agar dapat membantu petani semangka dalam meningkatkan kualitas dan kuantitas produksi semangka di Indonesia.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini antara lain:

1. Varian semangka yang digunakan adalah semangka jenis hibrida yakni *Charleston Gray*.
2. Penyakit yang digunakan pada penelitian ini adalah *Downy Mildew* dan *Watermelon Mosaic Virus*.

1.4 Tujuan Penelitian

Tujuan pada penelitian ini berguna untuk mengidentifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada tanaman semangka dengan menggunakan algoritma *SSD MobileNet V2*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu:

1. Memberikan informasi kepada masyarakat terutama petani semangka tentang bagaimana mengidentifikasi penyakit pada tanaman semangka.

2. Mengetahui hasil identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dengan menggunakan algoritma *SSD MobileNet V2*.
3. Dapat menjadi referensi pada penelitian bidang *Image Processing*, khususnya dalam penerapan.

1.6 Metodologi Penelitian

Tahapan-tahapan yang akan dilakukan dalam penelitian ini untuk mencapai tujuan penelitian meliputi:

1. Studi Pustaka dan Literatur

Studi literatur ini dilakukan pada tahap awal penelitian dengan mengumpulkan informasi dari penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* serta metode dan teknologi yang telah digunakan sebelumnya untuk mengidentifikasi penyakit ini, kemudian mengkaji algoritma *SSD MobileNet V2* dan implementasinya dalam deteksi objek pada aplikasi *mobile* yang diperoleh dengan mengumpulkan bahan dari jurnal, skripsi, artikel, buku, panduan dan sumber referensi lainnya.

2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data citra daun semangka yang terinfeksi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dari berbagai sumber, termasuk foto lapangan maupun *database* publik, kemudian mengklasifikan dan memberikan label pada data citra berdasarkan jenis penyakit yang teridentifikasi.

3. Pengembangan Model

Setelah dilakukan pengumpulan data, tahap selanjutnya adalah pengembangan model. Model dilatih dengan menggunakan *dataset* yang telah dikumpulkan dan diberi label, kemudian dilakukan validasi dan pengujian model untuk memastikan tingkat akurasi yang diinginkan.

4. Pengembangan Aplikasi

Tahapan selanjutnya setelah model berhasil dikembangkan dan dilatih adalah mengembangkan aplikasi *mobile* dan mengintegrasikannya dengan model yang telah dikembangkan sebelumnya ke dalam aplikasi *mobile*.

5. Uji Sistem

Pengujian sistem dilakukan dengan menggunakan aplikasi *mobile* yang telah dikembangkan untuk mendeteksi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada tanaman semangka.

6. Evaluasi dan Analisa

Selanjutnya, pada tahap ini akan dilakukan proses penilaian tingkat akurasi, efisiensi, dan kecepatan deteksi aplikasi *mobile* serta menganalisa faktor-faktor alamiah seperti jarak, kecerahan, dan resolusi kamera terhadap keakuratan deteksi penyakit.

7. Dokumentasi

Ini adalah tahap terakhir dari seluruh proses penelitian. Pada tahap ini, laporan disusun dan kesimpulan diambil berdasarkan penelitian yang telah dilaksanakan.

1.7 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bagian utama sebagai berikut:

BAB 1: PENDAHULUAN

Pada bab pendahuluan, tentang hal-hal yang dibahas yaitu latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2: LANDASAN TEORI

Bab ini berisikan teori-teori atau istilah-istilah yang digunakan untuk memahami permasalahan yang dibahas dalam penelitian ini. Seperti pengertian *Computer Vision*, *Image Processing*, *SSD-MobileNet* dan teori lainnya yang terkait dalam penelitian ini.

BAB 3: ANALISIS DAN PERANCANGAN

Pada bab analisis dan perancangan, akan dijabarkan mengenai arsitektur umum penelitian, bagaimana proses *pre-processing* yang dilakukan, tahap *training* dan *testing* data dan metode *SSD-MobileNet*.

BAB 4: IMPLEMENTASI DAN PENGUJIAN

Di bab ini, akan dijelaskan penerapan dan pembahasan dari desain aplikasi yang telah dibuat pada bab 3. Hasil pengujian dan implementasi aplikasi juga akan dijelaskan di sini.

BAB 5: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan ringkasan dari desain yang dibahas pada bab 3 serta hasil penelitian yang dijelaskan pada bab 4. Selain itu, bab ini juga menyertakan saran-saran untuk pengembangan penelitian lebih lanjut.

BAB 2

LANDASAN TEORI

2.1 Semangka

Semangka, yang termasuk dalam keluarga *Cucurbitaceae* (labu-labuan), memiliki tekstur renyah dan berair serta banyak manfaat bagi tubuh. Beberapa manfaat semangka antara lain melindungi jantung, memperlancar pengeluaran *urine*, dan menjaga kesehatan kulit. Selain sebagai penghilang dahaga, semangka juga berfungsi sebagai antioksidan yang baik. (Tahir et al., 2016).

Dengan banyaknya manfaat yang dapat diperoleh dari buah berbentuk bulat ini semangka tidak heran banyak orang yang membudidayakan semangka sebagai komoditas bahan dagang yang sering diperjual belikan di dalam masyarakat umum, namun penanganan semangka ternyata tidaklah mudah, setiap petani yang menanam buah musiman ini selalu menghadapi kendala, terutama serangan penyakit, semangka membutuhkan cukup banyak perhatian khusus untuk dibudidaya dan rentan terkena penyakit.

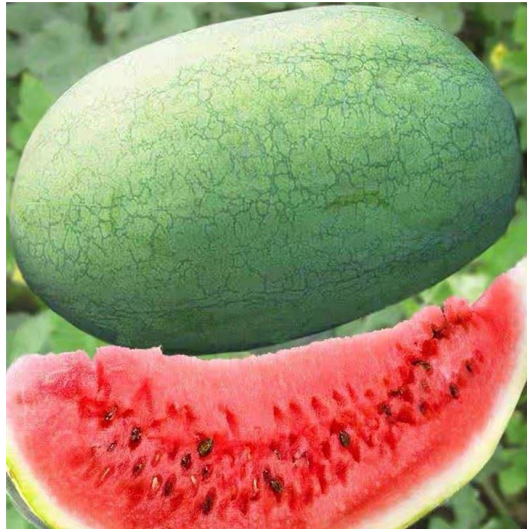
Beberapa faktor dapat menyebabkan terhambatnya produksi semangka. Salah satu faktor tersebut adalah keasaman tanah (pH) tanah (5.6-8), pengalaman dan kemampuan petani dalam mengolah lahan untuk ditanami semangka seperti pemilihan benih, jarak tanam (Silva Junior et al., 2023), pupuk, dan penggunaan pestisida serta polinasi, hama dan penyakit seperti layu fusarium, antraknosa, dan juga virus (Damicone et al., 2007), serta suplai air dan lahan yang digunakan. (Tunde, 2019).

2.2 Jenis Penyakit Semangka

2.2.1 Semangka

Daun semangka sehat merupakan jenis daun yang sehat tanpa teridentifikasi memiliki penyakit dan bercak terkena parasit. Semangka yang digunakan pada penelitian ini adalah semangka jenis hibrida dengan nama *Charleston Gray*.

Selanjutnya, pada penelitian ini penyakit yang akan dideteksi adalah *Downy Mildew* dan *Watermelon Mosaic Virus*.



Gambar 2. 1 Semangka *Charleston Gray* (*id.chinavegetableseeds.com*)

2.2.2 *Downy Mildew*

Downy Mildew merupakan penyakit pada daun semangka yang diakibatkan jamur cendawan *pseudoperonospora cubensis*. Munculnya bintik atau bercak pada bagian atas daun merupakan salah satu gejala awal sebuah tanaman semangka terkena penyakit ini, bintik atau bercak dapat berwarna biru pucat, keunguan, kekuningan, maupun tergantung jenis tanaman yang diserang.



Gambar 2. 2 Daun Yang Terserang *Downy Mildew*

2.2.3 *Watermelon Mosaic Virus*

Watermelon Mosaic Virus merupakan penyakit pada daun semangka yang diakibatkan oleh serangan virus mosaik semangka. Virus ini dapat menyebar melalui berbagai cara, termasuk vektor (terutama kutu daun) atau kontak fisik dengan seseorang atau alat, namun tidak melalui benih, karena virus ini tidak menyebar lewat benih. Kutu daun mendapatkan virus ini saat mengisap getah dan dapat menyebarkannya secara terus menerus selama beberapa jam setelah terinfeksi.

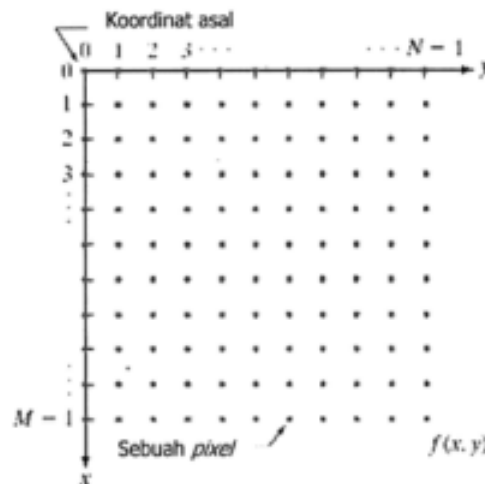


Gambar 2. 3 Daun Yang Terserang Mosaic Virus

2.3 Pengolahan Citra

Pengolahan citra (*Image Processing*) merupakan ilmu yang melakukan olahan citra berdasarkan input maupun output. Output dari pengolahan citra bisa berupa gambar atau sekumpulan karakteristik dan parameter yang berkaitan dengan gambar tersebut. Secara umum, istilah pengolahan citra digital didefinisikan sebagai pemrosesan gambar dua dimensi menggunakan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Pengolahan citra digital adalah disiplin ilmu yang mempelajari berbagai aspek terkait peningkatan kualitas gambar (seperti peningkatan kontras, transformasi warna, dan restorasi gambar), transformasi gambar (seperti rotasi, translasi, skala, dan transformasi geometris), pemilihan fitur gambar yang optimal untuk analisis,

ekstraksi informasi atau deskripsi objek serta pengenalan objek dalam gambar, serta kompresi atau reduksi data untuk penyimpanan, transmisi, dan pemrosesan data. Input dari pengolahan citra adalah gambar, sedangkan outputnya adalah gambar hasil pengolahan. (Avif, 2019).



Gambar 2. 4 Posisi Kordinat Citra (www.agussetiawanpcd.wordpress.com)

2.3.1 Google Colabolatory

Google Colaboratory adalah platform yang terinspirasi dari *Jupyter Notebooks*. *Jupyter Notebooks* adalah alat open source berbasis browser yang menggabungkan bahasa pemrograman, pustaka, dan alat visualisasi. *Jupyter Notebooks* dapat berfungsi baik secara lokal maupun di cloud. Setiap dokumen terdiri dari beberapa sel yang berisi skrip atau kode markdown, dengan output yang disematkan dalam dokumen tersebut. Jenis output mencakup teks, tabel, grafik, dan diagram. Google Colaboratory adalah proyek yang bertujuan untuk mendemokratisasikan pendidikan dan penelitian dalam pembelajaran mesin. Notebook Colab didasarkan pada Jupyter dan digunakan seperti Google Documents, memungkinkan notebook untuk dibagikan dan pengguna dapat berkolaborasi pada notebook yang sama.

Colab menyediakan runtime Python 3 dan R yang telah dikonfigurasi sebelumnya dengan pustaka pembelajaran mesin dan AI seperti TensorFlow, Matplotlib, dan Keras. *Virtual Machine* (VM) di bawah runtime akan dinonaktifkan setelah jangka waktu tertentu, sehingga semua data dan konfigurasi pengguna akan

hilang. Namun, notebook memungkinkan transfer file dari hard disk VM ke akun Google Drive pengguna. Layanan Google ini juga menyediakan akselerasi runtime GPU yang telah sepenuhnya dikonfigurasi dengan perangkat lunak terkait. Infrastruktur Google Colaboratory di-host di platform Google Cloud.

2.3.2 *Tensorflow*

Tensorflow adalah platform *end-to-end* yang bersifat *open-source* digunakan untuk aplikasi *machine learning*. Ini adalah *library* simbol matematika yang menggunakan aliran data dan pemrograman yang berbeda untuk melakukan berbagai tugas yang berfokus pada *training* dan *inference deep neural network*. Saat ini *library deep learning* yang paling terkenal adalah *Tensorflow Google*. Produk Google menggunakan *machine learning* untuk meningkatkan mesin pencarian, terjemahan, pemberian keterangan gambar atau rekomendasi. *Library API tensorflow* berisi beberapa struktur deteksi objek yang siap untuk digunakan diantaranya, *SSD (Single Shot Detector)*, *Faster-CNN (Faster Region-based Convolutional Neural Network)*, dan *RFCN (Region-based Fully Convolutional Network)* (Al-Azzo et al, 2018). Beberapa kemampuan umum pada *Tensorflow* yang populer antara lain :

1. Dukungan semua bahasa pemrograman populer seperti Python, C++, Java, R dan Go.
2. *TensorFlow* memungkinkan kemudahan dan penerapan model.
3. *TensorFlow* memiliki dukungan komunitas yang baik.

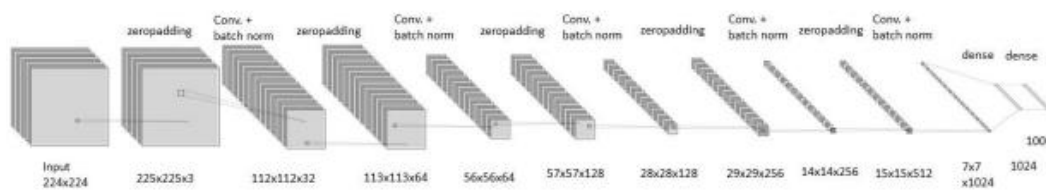
2.3.3 Deteksi Objek

Deteksi Objek adalah metode dari *computer vision* untuk menemukan lokasi objek dalam sebuah gambar atau video. Algoritma deteksi objek biasanya memanfaatkan *machine learning* atau *deep learning* untuk menghasilkan suatu output. Manusia mampu mengenali objek dan menemukan objek menarik dengan cepat. Tujuan dari objek deteksi adalah meniru kecerdasan ini menggunakan komputer. Ada beberapa macam metode yang dapat dilakukan untuk pembacaan fitur-fitur dari seluruh objek. Sistem objek deteksi harus melatih dan menguji data yang sudah dilabeli pada objek disetiap kelasnya untuk proses pengenalan. Ada banyak sekali tipe

model data pada *deep learning* salah satunya yang digunakan pada penelitian ini adalah PASCAL-VOC.

2.3.4 MobileNet

MobileNet merupakan salah satu bagian dari arsitektur *convolutional neural network* (CNN). Arsitektur ini dikembangkan oleh peneliti Google untuk kebutuhan mobile. Perbedaan utama *MobileNet* dengan arsitektur CNN lainnya adalah pada penggunaan lapisan konvolusi dengan ketebalan filter yang disesuaikan dengan input gambar. *MobileNet* juga menggunakan konvolusi *depthwise* dan *pointwise* (Rahman, 2020).



Gambar 2. 5 Arsitektur *MobileNet* (www.researchgate.com)

Seperti namanya mobile, para peneliti dari Google membuat arsitektur CNN yang mampu digunakan untuk ponsel. *MobileNet* merupakan salah satu arsitektur CNN (*Convolutional Neural Network*) yang digunakan untuk memenuhi kebutuhan *resource* berlebih. Perbedaan arsitektur *MobileNet* dan CNN adalah penggunaan *layer* konvolusi atau penggunaan lapisan dengan ketebalan filter yang sesuai dengan ketebalan dari input image. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*. Model dari *MobileNet* didasarkan pada *depthwise separable convolution* yang merupakan bentuk dari *standart convolution* menjadi *depthwise convolution* dan *1x1 convolution* atau disebut juga *pointwise convolution*.

Depthwise convolution menerapkan *single filter* untuk setiap masukan pada *channel*, pada *pointwise convolution* menerapkan *1x1 convolution* untuk menggabungkan output dari *depthwise convolution* dan pada *standart convolution* menggabungkan dan memfilter input menjadi satu set output baru dalam satu

langkah. *Depthwise separable convolution* membagi ini menjadi 2 *layer*, yaitu lapisan pemfilteran dan lapisan penggabungan yang digunakan untuk mengurangi komputasi dan ukuran model secara drastis (Howard et al, 2017).

2.3.5 SSD (Single Shot Multibox Detector)

SSD merupakan kepanjangan dari *Single Shot Multibox Detector* yang didasarkan pada *feed-forward convolutional network* yang menghasilkan kumpulan *bounding box* dan nilai dari setiap kelas objek tersebut kemudian diikuti oleh langkah *non-maximum suppression* untuk menghasilkan deteksi akhir. Lapisan jaringan awal didasarkan pada standar arsitektur yang digunakan untuk klasifikasi gambar berkualitas tinggi yang disebut dengan *base network* (Liu, 2016). SSD termasuk arsitektur *neural network* yang dirancang untuk tujuan deteksi yang berarti lokalisasi (*bounding box*) dan juga bertujuan untuk klasifikasi objek.

2.3.6 SSD-MobileNet

SSD-MobileNet merupakan pengembangan dari *Single Shot MultiBox Detector* (SSD). SSD menggunakan VGG16 sedangkan SSD-MobileNet menggunakan MobileNet sebagai *feature extraction*. Proses MobileNet unggul dalam hal kecepatan karena menggunakan *depthwise separable convolution* (DSC). DSC menggunakan *depthwise convolution* sebagai *filtering* dan *pointwise convolution* yang digabungkan dari hasil *depthwise* untuk menghasilkan beban komputasi yang kecil. Ukuran MobileNet 30 kali lebih kecil dibandingkan VGG16 dan juga memiliki kecepatan 10 kali lebih cepat dengan hasil akurasi yang sama. Perbedaan lainnya SSD-MobileNet menggunakan *ground truth box* untuk proses *training* sama seperti SSD namun berbeda *feature extraction*.

2.4 Citra

Citra adalah representasi atau kesan terlihat yang diperoleh oleh kamera, teleskop, mikroskop, atau perangkat lain, atau yang ditampilkan pada layar komputer (Oxford Dictionaries, n.d.). Gambar digital adalah representasi gambar nyata yang terdiri dari sekumpulan angka yang dapat disimpan dan ditangani oleh komputer

digital. Gambar digital terdiri dari area kecil atau dikenal sebagai piksel (elemen gambar). Di setiap piksel terdiri dari angka yang menggambarkan property seperti kecerahan atau warna. Gambar digital memiliki beberapa karakteristik salah satunya gambar berwarna. Gambar berwarna dapat memiliki tiga warna yaitu RGB (*Red, Green, Blue*) yang biasanya digunakan dalam monitor atau pemindai komputer, atau empat warna yaitu CYMK (*Cyan, Magenta, Yellow, Black*) yang biasanya digunakan dalam printer berwarna. (*Encyclopedia, n.d.*). Citra digital dibagi menjadi 3 jenis yaitu RGB image, grayscale image dan binary image.

2.4.1 RGB Image

RGB Image adalah citra yang terdiri dari Merah (*Red*), Hijau (*Green*) dan Biru (*Blue*) sebagai komponen warna utamanya. Citra digital RGB memiliki penyimpanan 24 bit dimana masing-masing komponen warna menampung 8 bit sehingga dapat mengekspresikan 16 juta lebih variasi warna.

2.4.2 Grayscale Image

Grayscale Image adalah citra skala abu-abu dimana setiap pikselnya memiliki warna gradasi mulai dari putih sampai hitam sehingga citra grayscale sangat cocok untuk pengolahan file gambar. Secara umum, citra grayscale disimpan dalam format 8 bit, memungkinkan 256 tingkatan skala abu-abu. Setiap piksel dalam citra ini memiliki intensitas antara 0 hingga 255, dengan 0 merepresentasikan hitam dan 255 merepresentasikan putih. (Kusumanto et al., 2011).

2.4.3 Binary Image

Binary Image adalah citra yang hanya memiliki dua warna yaitu hitam dan putih sehingga nilai intensitasnya hanya memiliki 2 nilai yaitu 0 dan 1. *Binary Image* dapat diperoleh dari proses *thresholding* dari citra *grayscale*. (Kusumanto et al., 2011).

2.5 Penelitian Terdahulu

Mariana (2019) bertujuan untuk mengembangkan sistem pakar berbasis *web* untuk diagnosis penyakit semangka menggunakan metode *certainty factor*. Untuk tujuan ini, data gejala dan penyakit dikumpulkan untuk digunakan sebagai data input sistem pakar. Berdasarkan data penyakit yang telah diperoleh, langkah berikutnya adalah menentukan kode untuk setiap gejala awal dari masing-masing penyakit tersebut. Langkah selanjutnya adalah membuat aturan (*rule*) yang akan digunakan dalam sistem pakar, termasuk data relasi yang menghubungkan gejala dengan penyakit tanaman semangka. Alurnya adalah sebagai berikut: pertama, menampilkan data gejala penyakit yang sesuai dengan pilihan pengguna, kemudian membuat *rule base*. Jika jumlah gejala kurang dari atau sama dengan 6, proses selanjutnya adalah perhitungan nilai *certainty factor* (cf). Hasil diagnosis akhir akan ditampilkan sebagai penyakit beserta nilai cf-nya. Dalam pengujian sistem untuk menguji keakuratan perhitungan dan analisis manual, ditemukan bahwa tingkat kesalahan sebesar 0,11%. Dalam pengujian dengan 6 pertanyaan terhadap 11 responden (10 petani dan 1 pakar), hasilnya menunjukkan bahwa 97% petani setuju, 3% kurang setuju, dan 0% tidak setuju. Sementara itu, pakar memberikan persetujuan 100%, menunjukkan bahwa sistem ini sesuai dan mudah digunakan.

Chan et al. (2022) bertujuan untuk membuat sistem deteksi wajah dengan menggunakan algoritma *SSD MobileNet V2*. Metode ini diusulkan dikarenakan untuk mengatasi permasalahan *over-detection* dalam gambar statis dan *misdetetection* dalam gambar yang tidak mengandung wajah dengan memanfaatkan *two-step method* yakni *SSD MobileNet V2* dan *Geometrical Algorithm* untuk mendeteksi *region* dari wajah pada gambar.

Prima (2023) bertujuan untuk membuat sistem pendeteksi aneka ragam buah menggunakan jaringan *SSD MobileNet V2*. Penelitian ini menggunakan 8 jenis buah sebagai objek penelitian yaitu Jambu Merah, Pir, Apel, Apel Manis, Pisang, Nanas dan Buah Naga. Beberapa tahapan yang dilalui pada proses pelatihan model yaitu *initializing data*, *resizing data*, dan konversi gambar ke *NumPy* array. Dari hasil pengujian dengan beberapa skenario, didapati bahwa tingkat akurasi rata rata diatas 80%.

Banerjee et al. (2023) bertujuan untuk membuat sistem klasifikasi penyakit pada tanaman semangka. Penyakit yang dideteksi yakni *Powdery Mildew*, *Anthrachnose*, *Downy Mildew*, *Bacterial Fruits Blotch*, *Gummy Rope Blight*, *Alternatives Leaves Spot*, *Myrothecium Leaves Spot*, *Cercospora Leaves Spot*, dan *Bacterial Fruit Blotch*. Model yang digunakan terdiri dari 4 *convolutional layers*, dan 4 *max pooling level*, serta 1 *fully connected layer*. Untuk tiap-tiap penyakit, dianalisa tingkat *recall*, *precision*, dan skor F1 nya untuk mendapatkan performa dari model tersebut. Hasil yang didapatkan menunjukkan adanya potensi akurasi dan efikasi dalam klasifikasi penyakit. Tingkat akurasi bervariasi berdasarkan penyakit yang dideteksi, dengan akurasi pada *Anthrachnose* (76.00%), dan pada *Gummy Trunk Blight* sebesar (75.56%). Secara umum, model ini berjalan dengan baik terhadap perolehan skor F1, terutama untuk *Anthrachnose* (73.17%), dan *Gummy Trunk Blight* (73.12%). Selanjutnya, tingkat *micro-average precision* (70.29%) dan *weighting-average precision* (70.26%).

Alfarizi et al. (2023) bertujuan untuk mengembangkan sistem untuk deteksi kucing secara real-time. Metode ini diusulkan dikarenakan sistem yang ada dipasaran sekarang hanya menggunakan sensor PIR dan tidak dapat mengidentifikasi objek tertentu. Sistem ini menggunakan jaringan *SSD MobileNet V2* dengan parameter *precision*, *recall*, dan *loss*. *Node-Red platform* dan *library TensorFlow.js* digunakan untuk mengimplementasikan model dengan menggunakan *Raspberry Pi 4*. Hasil yang didapatkan adalah tingkat akurasi sebesar 0.91, kemudian tingkat *recall* sebesar 0.89, diikuti skor F1 sebesar 0.92. Berdasarkan hasil pengujian, model yang dibuat dengan metode *transfer learning* menggunakan model *pre-trained* dari *SSD MobileNet V2* mencapai tingkat akurasi sebesar 82%. Jarak, kecerahan, dan resolusi kamera dapat mempengaruhi efektifitas model yang digunakan.

Suryavanshi et al. (2023) bertujuan untuk mengidentifikasi dan mengkategorikan penyakit pada tanaman *Cucurbitaceae* dengan menggunakan algoritma CNN. *Dataset* yang digunakan dikelompokkan berdasarkan 4 tingkat keparahan penyakit, yakni 1%, 25%, 50%, 75%, dan 100%. Hasil menunjukkan

model yang digunakan memiliki performa yang baik, dengan tingkat akurasi, *recall*, dan skor F1 berkisar diantara 85.93% hingga 94.85%.

Arora et al. (2024) bertujuan untuk mengembangkan sistem deteksi keseriusan penyakit WMV pada tanaman semangka. Metode yang digunakan adalah gabungan antara RNN dan *Random Forest*. Hasil yang didapati sebesar 99.8% dengan lebih dari 16.000 gambar dengan tingkat keparahan WMV yang berbeda-beda. Kelebihan model yang digunakan dalam mensortir dengan tepat 5 level keseriusan penyakit WMV adalah dengan melakukan analisis *Confusion Matrix*.

Penelitian sebelumnya yang sudah dijelaskan akan disajikan secara ringkas dalam Tabel 2.1

Tabel 2. 1 Penelitian Terdahulu

No.	Peneliti	Tahun	Judul	Hasil
1	Dinesh Arora, Kapil Mehta, Amresh Kumar, Shweta Lamba	2024	Evaluating Watermelon Mosaic Virus Seriousness with Hybrid RNN and Random Forest Model: A Five-Degree Approach	Mengembangkan sistem deteksi tingkat keseriusan WMV pada semangka dengan menggabungkan RNN dan Random Forest untuk 16.000 gambar dengan akurasi 99.8%.
2	Ankita Suryavanshi, Vinay Kukreja, Prateek Srivastava, Shiva Mehta, Siddhant Thapliyal	2024	Next-Gen Cucurbit Disease Diagnosis: A Federated Learning CNN Approach	Hasil evaluasi menggunakan algoritma CNN pada tanaman Cucurbitaceae dengan tingkat keseriusan dari 1%, 25%, 50%, 75%, dan 100% dengan hasil Skor F1 sebesar 85.93% hingga 94.85%
3	Salman Alfarizi, Zahra Salsabila, Sania Bahrullah, Hesty Susanti,	2023	Real-Time Cat Detection System	Mengembangkan sistem deteksi kucing secara real-time dengan

	Willy Anugrah Cahyadi		using MobileNet-SSD V2	menggunakan jaringan SSD MobileNet V2. Sistem ini menggunakan Node-Red dan library TensorFlow.js. Skor F1 yang didapat adalah 0.92.
4	Deepak Banerjee, Vinay Kukreja, Amit Gupta, Vijay Singh, Tejinder Pal Singh Brar	2023	CNN and SVM-based Model for Effective Watermelon Disease Classification	Mengembangkan sistem deteksi 8 penyakit pada tanaman semangka dengan menggunakan metode CNN dan SVM. Hasil dari <i>micro-average precision</i> adalah 70.29% dan <i>weighting-average precision</i> senilai 70.26%.
5	Ario Prima	2023	Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNet V2	Metode deteksi aneka ragam buah dengan menggunakan jaringan SSD MobileNet V2 pada delapan kelas buah yaitu Jambu Merah, Pir, Apel, Apel Gula, isang, jeruk, Nanas, dan Buah Naga. Adapun tingkat akurasi rata-rata adalah > 80%.
6	Aifian Adi Sufian Chan, M.F.L Abdullah, Saizalmursidi Md Mustam, Farhana Ahman Poad, Ariffudin Joret	2022	Face Detection in Still Image using SSD MobileNet V2 and Geometrical Algorithm	Mengembangkan sistem untuk deteksi wajah dengan menggunakan jaringan SSD MobileNet V2. Hasil akurasi yang didapatkan senilai

				91.5%, dimana nilai ini masih kalah dibandingkan dengan metode MTCNN dan Dlib meskipun berfungsi lebih baik dalam mengatasi <i>overdetection</i> dan <i>misdetction</i> .
7	Alan Wiedy Mariana	2019	Sistem Pakar Diagnosis Penyakit Tanaman Semangka Menggunakan Metode Certainty Factor	Pengembangan sistem pakar diagnosis menggunakan metode certainty factor. Pengujian dilakukan terhadap 11 responden, terdiri dari 10 petani dan 1 pakar. Hasilnya menunjukkan bahwa 97% petani setuju dan 3% tidak setuju dengan sistem tersebut. Pengujian terhadap pakar menunjukkan tingkat persetujuan 100%.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan membahas tentang analisis dan perancangan yang mencakup data yang digunakan, teknik identifikasi penyakit semangka, perancangan sistem, arsitektur umum, serta alur kerja sistem dalam bentuk antarmuka pengguna dan diagram *use case*.

3.1 Data yang Digunakan

Pada penelitian yang dilakukan ini citra yang dikumpulkan bersumber dari set data *public Kaggle* (Kapadnis, 2023). Citra diambil dalam format JPG dan dengan menggunakan orientasi *portrait* serta *landscape*. Gambar berikut merupakan contoh dari citra daun semangka yang akan melalui diproses untuk klasifikasi pada Gambar 3.1.



Gambar 3. 1 Data Daun Semangka (www.depositphotos.com)

Data secara keseluruhan yang diperoleh adalah sebanyak 600 citra yang kemudian disatukan menjadi set data citra penyakit semangka. Data penyakit semangka *Downy Mildew* berjumlah 200 citra, *Watermelon Mosaic Virus* sebanyak 200 citra, dan daun semangka sehat sebanyak 200 citra. Pada pengujian identifikasi, keseluruhan dari jumlah data diacak dan dibagi dalam dua set data yang berbeda yaitu data uji dan data latih. Bobot nilai data latih dengan data uji adalah senilai

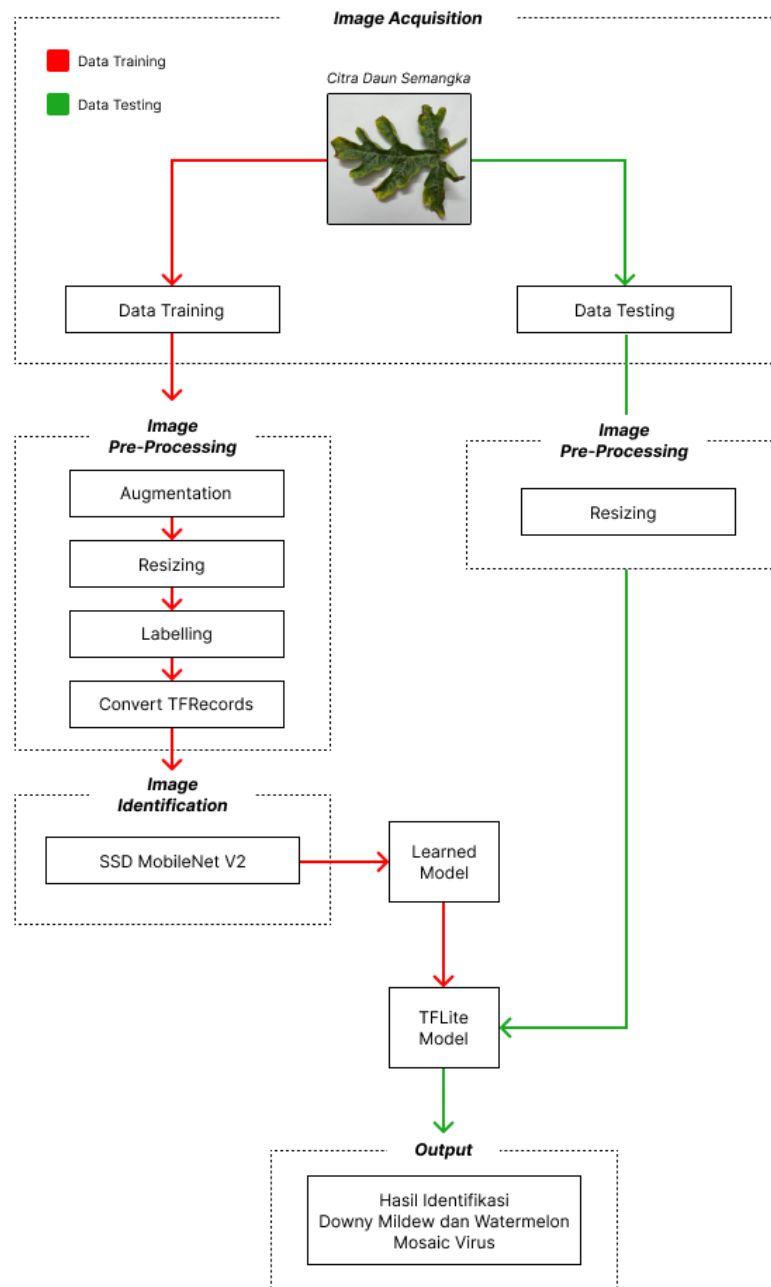
80:20. Pembagian data uji dan data latih untuk indentifikasi jenis penyakit semangka dapat dilihat di Tabel 3.1.

Tabel 3. 1 Jumlah data pelatihan dan data pengujian

No.	Kategori	Data Latih	Data Uji	Jumlah Data
1.	<i>Downy Mildew</i>	160	40	200
2.	<i>Watermelon Mosaic Virus</i>	160	40	200
3.	Healthy (Sehat)	160	40	200
Total		480	120	600

3.2 Arsitektur Umum

Penelitian ini bertujuan untuk mengidentifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada tanaman semangka melalui beberapa tahap, termasuk proses latihan, validasi, dan uji. Proses latihan melibatkan penggunaan kembali model *SSD MobileNet* yang telah dilatih sebelumnya. Data latih diambil dari citra yang diperoleh melalui *Google Image Search* atau *Kaggle*. Tahap pre-processing mencakup *augmentation*, *resizing* dan *labelling* data. Selanjutnya, citra input diproses menggunakan arsitektur *MobileNet* sebagai ekstraktor fitur. Ciri-ciri yang diekstrak kemudian diklasifikasikan oleh model *MobileNet* yang telah dilatih sebelumnya, dan diidentifikasi menggunakan arsitektur jaringan SSD untuk mendeteksi jenis penyakit. Outputnya berupa kotak pembatas, label, dan persentase akurasi dari jenis penyakit yang terdeteksi. Selain itu, akan dilakukan pencatatan berupa rekapitulasi dari proses identifikasi, dalam bentuk csv yang berisikan *timestamp* beserta penyakit yang dideteksi.



Gambar 3. 2 Arsitektur Umum

Pada penelitian ini, arsitektur umum dari perancangan sistem ditunjukkan pada Gambar 3.2, dimana kerja sistem ini dibagi menjadi beberapa bagian:

3.2.1 *Image Acquisition*

Tahap ini adalah ketika data citra dari daun semangka diambil untuk digunakan

Sebagai input dalam pembangunan sistem. Citra-citra ini didapatkan dari platform Kaggle, memiliki format JPG dengan dimensi 640×640 piksel.

3.2.2 *Image Preprocessing*

Selanjutnya, citra daun semangka yang telah dipisahkan menjadi data pelatihan akan masuk ke tahap preprocessing gambar, di mana citra-citra ini akan menjalani tiga proses.. Tahapannya adalah sebagai berikut.

3.2.2.1 *Augmentation*

Pada tahapan pertama, gambar melalui proses *augmentation*, dimana proses ini dilakukan dengan cara sedikit mengubah citra asli gambar dengan 5 jenis perubahan yaitu *brightened*, *shifted*, *zoomed*, *flipped*, dan *scaled*. Proses *augmentation* sendiri dilakukan dengan tujuan mengurangi probabilitas *overfitting* ataupun *underfitting* dari data yang akan dilatih.

3.2.2.2 *Resizing*

Proses kedua yang akan dilakukan pada tahap *pre-processing* adalah *resizing*. Pada tahap ini, dilakukan proses sederhana mengubah ukuran citra daun semangka dari ukuran 3072×3072 menjadi berukuran 640×640 piksel. Hal ini dilakukan agar citra yang ada dapat diinput ke dalam sistem dan mengurangi besaran data supaya sistem tidak terlalu berat. Proses dari *resizing* ini dapat dilihat pada *pseudocode* dibawah ini :

```
BEGIN
  DEFINE ORIGINAL IMAGE FOLDER PATH
  DEFINE RESIZED IMAGE FOLDER PATH
  DEFINE NEW PIXEL

  FUNC resize_image(image_path, output_path, target_size):
    READ IMAGE DATA FROM image_path
    RESIZE IMAGE DATA TO target_size
    SAVE RESIZED DATA TO output_path

  FOREACH image file IN ORIGINAL_FOLDER_PATH:
    CALL FUNC resize_image(image_path, output_path, target_size)

END
```

Pseudocode di atas mendefinisikan proses untuk mengubah ukuran citra dalam sebuah folder. Proses dimulai dengan mendefinisikan *path* folder untuk citra asli dan folder untuk menyimpan citra yang telah di-*resize*, serta menentukan ukuran *pixel* baru. Sebuah fungsi ``resize_image`` dibuat untuk membaca citra dari ``image_path``, mengubah ukurannya menjadi ``target_size``, dan menyimpan citra yang telah diubah ukurannya ke ``output_path``. *Pseudocode* ini kemudian melakukan iterasi melalui setiap *file* citra dalam *folder* citra asli dan memanggil fungsi ``resize_image`` dengan parameter *path file* citra asli, *path folder output*, dan ukuran *pixel* baru untuk setiap citra tersebut.

3.2.2.3 Labeling

Tahapan selanjutnya setelah citra selesai melalui tahap *resizing* adalah *labeling*. Pada tahap ini, dilakukan proses memberikan sebuah label kepada citra untuk mendapat objek khusus yang akan mampu dibaca oleh sistem. *Labeling* yang dilakukan pada citra dibuat dengan memanfaatkan aplikasi *LabelImg* yang memiliki fungsi untuk melakukan *labeling* pada citra daun semangka berdasarkan kategori yang sudah ditentukan. *Output* yang dikeluarkan adalah citra daun semangka yang sudah dilabel dalam bentuk xml. Informasi yang tersimpan dalam *file* xml tersebut berupa identifikasi dari objek khusus yang sudah ditandai beserta koordinat dari objek yang berguna pada proses selanjutnya.

3.2.2.4 Convert TFRecord

Langkah berikutnya adalah konversi *TFRecord*. Pada tahap ini, data label dalam format XML akan diubah menjadi format *TFRecord* untuk efisiensi penyimpanan. Hal ini dilakukan karena volume data yang besar dapat memperlambat proses pelatihan. Dengan *TFRecord*, data disimpan sebagai *string biner*, yang dapat meningkatkan kecepatan proses pelatihan. Data XML dan citra daun semangka akan dikonversi menjadi *file TFRecord*. Penggunaan *TFRecord* memungkinkan penghematan ruang *disk* dan mempercepat proses sistem yang akan dilakukan selanjutnya.

Sebelumnya, citra daun semangka yang sudah berbentuk xml akan di-*convert* ke dalam tabel .csv untuk membuat sistem dapat memetakan hasil citra yang sudah melalui proses pelabelan. Selanjutnya, *file* yang sudah di-*convert* akan di-*convert* kembali ke dalam bentuk *TFRecord* yang akan digunakan untuk proses *training* selanjutnya. Langkah atau prosedur dalam mengkonversi ke bentuk *TFRecord* dapat dilihat secara seksama pada *pseudocode* dibawah ini:

```
BEGIN
  DEFINE ORIGINAL IMAGE FOLDER PATH
  DEFINE FOLDER PATH FOR TFRecord
  DEFINE NEW PIXEL

  FUNC create_tfrecord(image_path, output_path):
    READ IMAGE DATA FROM image_path
    CREATE NEW TFRecord FILE IN output_path
    FOREACH IMAGE DATA:
      CONVERT IMAGE DATA TO protobuff
      CREATE protobuff example INTO TFRecord File
    SAVE RESIZED DATA TO output_path

  FOREACH image file IN ORIGINAL_FOLDER_PATH:
    CALL FUNC create_tfrecord(image_path, output_path)

END
```

3.2.3 Image Identification

Selanjutnya pada tahap ini, data *training* yang sudah ada akan melalui tahap identifikasi penyakit tanaman semangka dengan menggunakan *Mobilenet-SSD*. *Mobilenet-SSD* terdiri dari SSD yang berperan sebagai *base model*, dan *Mobilenet* sebagai *network model*. SSD akan melakukan *adjustment* terhadap pendeteksian objek dengan *bounding box* sementara *MobileNet* akan bekerja untuk mengekstrak fitur yang nantinya akan diklasifikasi. Dua fitur utama dalam *Mobilenet-SSD* adalah sebagai berikut:

3.2.3.1 Extract Feature Maps

SSD menggunakan VGG16 untuk mengekstrak *feature maps* yang kemudian mendeteksi objek menggunakan *Conv4_3 layer*. SSD akan memprediksi *anchor box offset* yang telah ditentukan sebelumnya untuk setiap lokasi *feature maps*.

Setiap kotak akan memiliki ukuran dan posisi yang tetap pada sel. *Feature Maps* akan diskalakan ulang sehingga hanya bertanggung jawab untuk objek pada skala tertentu.

```
BEGIN

# Load VGG16 model for feature extraction
VGG16_model = load_VGG16()

# Extract feature maps from Conv4_3 layer
feature_maps = extract_feature_maps(VGG16_model, layer='Conv4_3')

# Initialize SSD (Single Shot MultiBox Detector)
SSD = initialize_SSD()

# Iterate over each location in the feature maps
FOR each location IN feature_maps:

    # Predict anchor box offsets for the current location
    anchor_box_offsets = SSD.predict_offsets(location)

    # Define predefined anchor boxes with fixed sizes and positions
    anchor_boxes = define_anchor_boxes(location)

    # Update anchor boxes with the predicted offsets
    updated_boxes = update_boxes(anchor_boxes, anchor_box_offsets)

    # Assign each updated box to a specific cell
    assign_boxes_to_cells(updated_boxes, location)

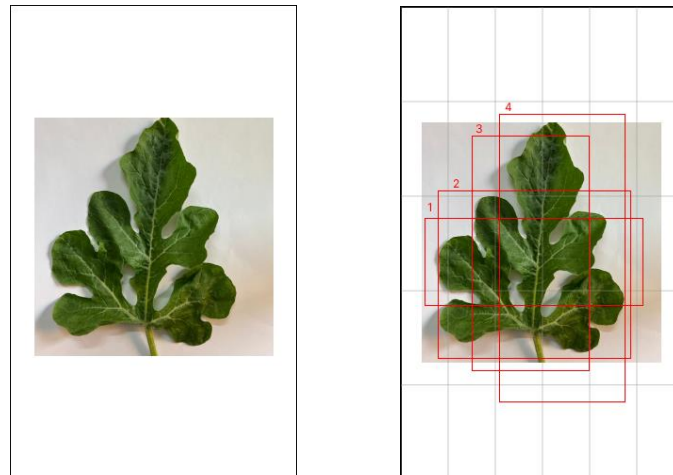
# Scale the feature maps so that each map is responsible for objects of a specific
scale
scaled_feature_maps = scale_feature_maps(feature_maps)

# SSD will use the scaled feature maps for object detection
SSD.detect_objects(scaled_feature_maps)

END
```

SSD memprediksi *bounding box* pada area yang ditandai menggunakan *default bounding box* melalui berbagai skala dan rasio di setiap lokasi pada peta fitur. Selama pelatihan, SSD membandingkan objek dengan *default bounding box* yang memiliki berbagai rasio. Setiap *default box* dengan IoU > 0.5 dianggap cocok.

Setiap sel atau lokasi pada peta fitur dapat memprediksi empat objek identifikasi. Setiap identifikasi mencakup *bounding box* dan kelas objek yang dibatasi.



Gambar 3. 3 Foto Asli (Kiri), 4 Objek Prediksi Tiap Sel atau Lokasi (Kanan)

3.2.3.2 *Convolution Filters*

SSD dalam prosesnya tidak menggunakan jaringan proposal *region* yang didelegasikan. Sebaliknya, SSD menggunakan metode yang sangat sederhana, yaitu menghitung sel atau lokasi dan *confidence scores* menggunakan *filter* konvolusi yang kecil. Setelah melakukan *Extract Feature Maps*, SSD akan menerapkan *filter* konvolusi 3×3 untuk setiap sel atau lokasi yang berguna untuk membuat prediksi. Proses ini akan bekerja untuk menyesuaikan posisi *bounding box* dengan *ground truth boxes* yang asli. Penyesuaian ini jauh lebih baik daripada melakukan prediksi dengan koordinat yang acak. Untuk itu *Multibox* akan bekerja sebagai prediksi untuk memastikan *bounding box* berada di posisi yang dekat dengan *ground truth boxes* yang asli.

3.2.4 *Learned Model*

Setelah proses *training*, kita akan mendapatkan sebuah *file learned model* yang bernama *savedmodel* dengan ekstensi *.pb* dikarenakan menggunakan *Tensorflow V2*. Informasi yang ada di dalam *file* tersebut berupa sebuah grafik serta bobot model hasil dari proses *training*, dimana pada daun semangka mempunyai 2 jenis

penyakit yaitu *Downy Mildew* dan *Watermelon Mosaic Virus*. Hasil identifikasi data *training* yang ada akan menjadi *learned model*. *Learned model* tersebut merupakan sebuah data yang sudah dilatih untuk selanjutnya akan di-convert ke dalam bentuk *Tensorflow Lite* agar bisa digunakan pada aplikasi *mobile* berbasis *android*.

```
BEGIN

# Train the model on the dataset
model = train_model(dataset)

# Save the learned model as a TensorFlow saved model with .pb extension
save_model(model, 'savedmodel.pb')

# The learned model contains the training graph and weights
learned_model = load_saved_model('savedmodel.pb')

# Define the types of diseases for watermelon leaves
disease_types = ['Downy Mildew', 'Watermelon Mosaic Virus']

# The learned model will identify and classify these diseases
classified_results = classify_diseases(learned_model, disease_types)

# Convert the learned model to TensorFlow Lite format for mobile application usage
tflite_model = convert_to_tflite(learned_model)

# Save the TensorFlow Lite model
save_tflite_model(tflite_model, 'model.tflite')

END
```

3.2.5 TFLite Model

Pada tahap sebelumnya, hasil *training* mengeluarkan *output* berupa *savedmodel* yang berekstensi *.pb*. Pada tahap ini, *file* tersebut akan di-convert ke dalam bentuk *tensorflow lite*. *File* di-convert ke dalam bentuk *TFLite* dikarenakan sistem yang akan dibangun akan berbasis *android*. Model *TensorFlow* yang sudah selesai dilatih akan disediakan oleh *TFLite* untuk selanjutnya akan dikompresi lalu diaplikasikan di perangkat *mobile*, hal ini menyebabkan *file* yang telah di-convert ke dalam bentuk *TFLite* akan berubah menjadi ukuran biner yang lebih kecil untuk bisa masuk

ke dalam perangkat *mobile*. Proses *convert* ke dalam bentuk *TFLite* dapat dilihat pada code dibawah ini:

```
BEGIN
  DEFINE ORIGINAL PATH TensorFlow model
  DEFINE PATH TO SAVE TFLite Model

  FUNC convert_to_tflite(model_path, output_path):
    LOAD TensorFlow model FROM model_path
    CREATE converter TFLite FROM TensorFlow model
    CONVERT model USING converter
    SAVE CONVERTED MODEL TO output_path

  CALL FUNC convert_to_tflite(model_path, output_path)
END
```

Pseudocode di atas menjelaskan langkah-langkah untuk mengonversi model *TensorFlow* ke format *TensorFlow Lite (TFLite)*. Proses dimulai dengan mendefinisikan *path file* model *TensorFlow* asli dan *path* untuk menyimpan *file* model yang telah di-konversi ke format *TFLite*. Sebuah fungsi bernama ``convert_to_tflite`` dibuat untuk menangani proses konversi ini. Fungsi tersebut pertama-tama memuat model *TensorFlow* dari *path* yang ditentukan. Kemudian, fungsi ini membuat objek *converter TFLite* dari model yang telah dimuat. Dengan menggunakan objek *converter* tersebut, fungsi mengonversi model ke format *TFLite*. Setelah itu, model yang telah di-konversi disimpan ke *path output* yang telah ditentukan. Langkah terakhir dalam proses adalah memanggil fungsi ``convert_to_tflite`` dengan parameter *path file* model *TensorFlow* asli dan *path* untuk menyimpan *file* model *TFLite*.

3.2.6 Proses Training

Proses *training* merupakan sebuah tahapan implementasi menggunakan metode *Mobilenet-SSD*. Seluruh data yang ada akan diberi label dan menghasilkan sebuah *file xml* yang kemudian akan di-*convert* menjadi *file train.record* dan akan digunakan sebagai data *input* pada proses *training*.

Setelah semua data untuk proses *training* telah ada, maka selanjutnya adalah menentukan nilai seperti nilai *input* citra, bobot, *batch size*, dan jumlah *steps*

maksimal yang akan diatur dalam konfigurasi *pipeline*. Sebelum melakukan konfigurasi, diperlukan sebuah *labelmap* sebagai data identifikasi objek yang akan digunakan oleh sistem. Sebuah *file* dengan nama *labelmap.txt* yang berisikan banyak kategori dan nama dari kategori tersebut. Pada penelitian ini, proses *training* dilakukan dengan menggunakan *SSD MobileNet V2* dengan bantuan *Tensorflow-GPU* dapat dilihat pada *pseudocode* dibawah ini :

```
BEGIN
  DEFINE the folder path for the original image data
  DEFINE the folder path to save the XML file
  DEFINE the folder path to store the train.record file
  DEFINE the labelmap.txt file
  DEFINE the pipeline.config file
  DEFINE a path to save the training model results

  FUNC label_data(data_path, output_xml_path):
    FOREACH image file in the data_path folder:
      Label the image
      Save the labels as an XML file in output_xml_path

  FUNC convert_to_tfrecord(xml_folder, output_record_path):
    Read all XML files from xml_folder
    Convert XML file to train.record file
    Save the train.record file to output_record_path

  CALL FUNC label_data(data_path, output_xml_path)
  CALL FUNC convert_to_tfrecord(xml_folder, output_record_path)

  FUNC create_labelmap(labelmap_path):
    Determine the number of categories and the name of each category
    Save the object identification data to labelmap_path as labelmap.txt

  CALL FUNC create_labelmap(labelmap_path)

  DEFINE pipeline configuration:
  Specify the path for the train.record file
  Specify the path for the labelmap.txt file
  Specify the path for the MobileNet SSD pretrained model file
  Determine the input image value
  Determine the initial weight
  Determine the batch size
  Determine the maximum number of steps
  Save the configuration to the pipeline.config file
```

```

FUNC train_model(pipeline_config_path, output_model_path):
    Load pipeline configuration from pipeline_config_path
    Load dataset and labelmap
    Load MobileNet SSD pretrained model
    Start the training process
    Save the training results model to output_model_path

CALL FUNC train_model(pipeline_config_path, output_model_path)

END

```

Setelah proses training selesai, akan menghasilkan sebuah model dengan ekstensi .pb yang kemudian akan dikonversi ke dalam bentuk tflite agar dapat digunakan pada perangkat mobile.

3.2.7 Proses Testing

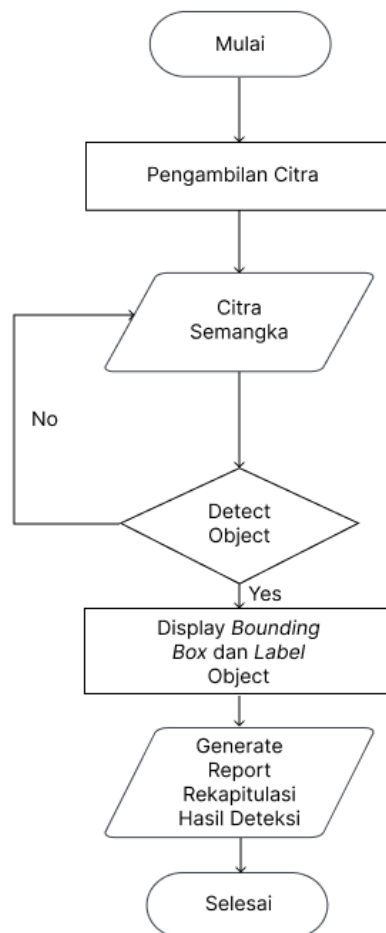
Selanjutnya adalah tahap pengujian, yang merupakan proses untuk mengevaluasi model yang telah dilatih pada tahap sebelumnya. Proses ini bertujuan untuk memastikan bahwa model dapat diintegrasikan dengan sistem, mampu mengenali objek penyakit tanaman semangka yang telah dibuat, serta dapat mengidentifikasi penyakit semangka menjadi tiga kategori: *Downy Mildew*, *Mosaic Virus*, dan *Healthy*. Selain itu, tahap ini juga untuk menghitung tingkat akurasi dari metode *Mobilenet-SSD* yang diterapkan dalam identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dalam penelitian ini.

3.2.8 Output

Hasil akhir dari sistem yang telah dibuat yaitu berupa informasi akurasi dan *bounding box* yang menunjukkan hasil identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dengan 3 kategori yaitu *Downy Mildew*, *Mosaic Virus*, dan *Healthy* yang dilakukan pada saat proses *testing*.

3.3 Flowchart Sistem

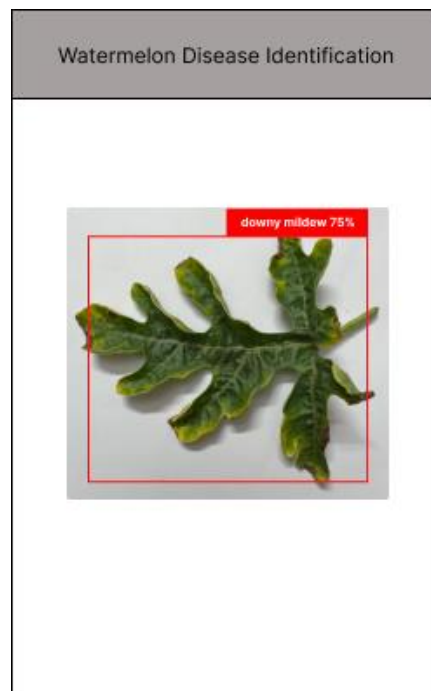
Pada bagian ini, dijelaskan proses kerja sistem beserta teknik-teknik yang digunakan sebagai solusi dalam perancangan sistem. Detail mengenai cara kerja sistem dapat ditemukan dalam Gambar 3.4.



Gambar 3. 4 *Flowchart System*

3.4 Perancangan Antarmuka

Perancangan antarmuka adalah tahap yang menjelaskan alur kerja sistem, yang bertujuan untuk meningkatkan minat pengguna dan memberikan kemudahan dalam penggunaan sistem. Desain antarmuka yang terlihat pada Gambar 3.5 merupakan tampilan yang digunakan pengguna untuk mendeteksi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus*. Aplikasi ini menggunakan *Tensorflow Lite* dan hanya memiliki satu tampilan utama. Pada antarmuka utama aplikasi, sistem secara otomatis mengidentifikasi objek saat diarahkan ke objek yang dituju, ditandai dengan munculnya kotak pembatas yang memberikan informasi tentang penyakit tanaman semangka yang sedang diperiksa.



Gambar 3. 5 Rancangan Antarmuka Aplikasi

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan membahas implementasi *SSD MobileNet V2* dalam identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dari bab sebelumnya serta membahas hasil pengujian terhadap sistem yang telah dirancang.

4.1 Implementasi Sistem

Pada tahap ini, implementasi *SSD MobileNet V2* dalam identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* secara *realtime* menggunakan perangkat *android*. Oleh karena itu, sistem ini memerlukan dukungan perangkat keras dan perangkat lunak pendukung, yaitu:

4.1.1 Perangkat Keras dan Perangkat Lunak

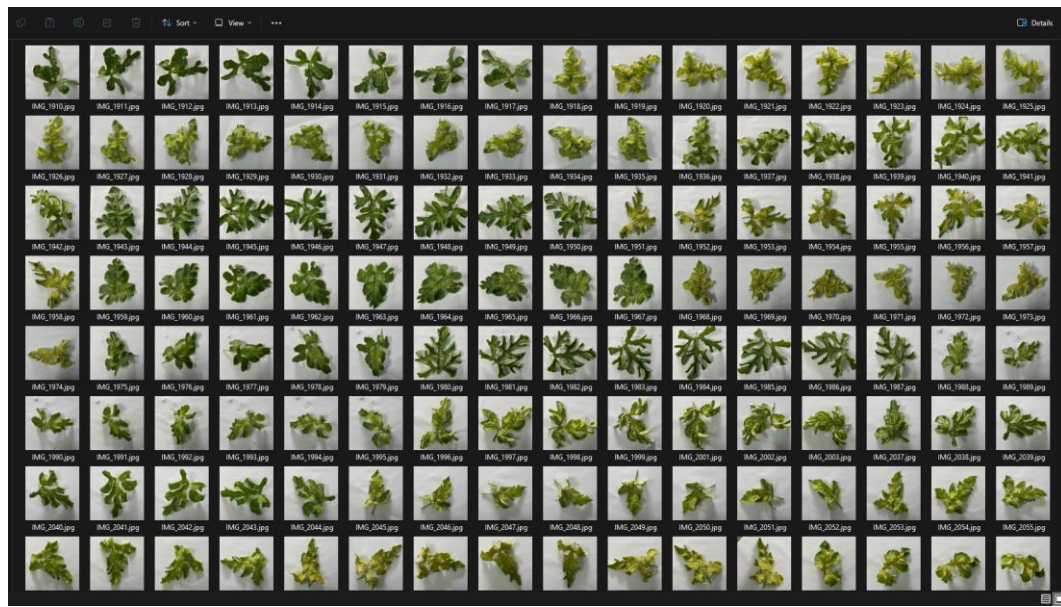
Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* ini adalah:

1. *Processor* 3,9 GHz Ryzen 5 5600G
2. Penyimpanan Memori RAM sebesar 32GB
3. Penyimpanan SSD sebesar 1TB
4. Sistem operasi menggunakan Windows 11
5. Bahasa pemrograman Python versi 3.1.10
6. *Library* tf-models-official versi 2.14.0

4.1.2 Implementasi Data

Data yang digunakan dalam penelitian ini menggunakan dua data yaitu data *train* dan *data test*. *Dataset* yang ada berdasarkan data yang diambil dari internet yaitu *Kaggle* dan secara langsung dengan tiga (3) yaitu *Downy Mildew*, *Watermelon Mosaic Virus*, dan *Healthy* sebanyak 600 citra dengan 200 *Downy Mildew*, 200

Watermelon Mosaic Virus dan 200 *Healthy*. Contoh data penyakit tanaman semangka yang sudah di-labeling bisa dilihat dengan seksama Gambar 4.1.



Gambar 4. 1 Data Citra Daun Semangka

4.1.3 Implementasi Antarmuka

Implementasi antarmuka dibuat berdasarkan perancangan antarmuka sistem yang telah dibahas sebelumnya pada bab 3. Tampilan sistem terdiri dari satu (1), yaitu tampilan identifikasi.

1. Tampilan Identifikasi

Tampilan Identifikasi merupakan halaman awal saat pengguna pertama kali membuka atau menjalankan aplikasi. Pada tampilan ini pengguna langsung bisa menggunakan aplikasi dengan cara mengarahkan ke objek yang ingin diidentifikasi. Tampilan Identifikasi ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Tampilan Halaman Identifikasi

4.2 Prosedur Operasional

Tampilan aplikasi yang dibangun terdiri dari halaman identifikasi. Halaman identifikasi yang dibangun dapat dilihat pada Gambar 4.2 dimana pengguna dapat mengarahkan kamera pada objek daun semangka yang ingin diketahui penyakitnya yaitu *Downy Mildew*, *Watermelon Mosaic Virus*, atau *Healthy*.



Gambar 4. 3 Prosedur Operasional

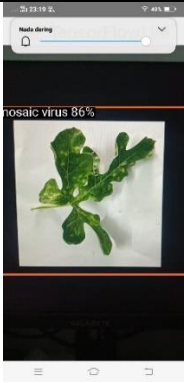
Ketika objek di deteksi menggunakan *smartphone* maka *bounding box* akan terbentuk seperti Gambar 4.3. Pada Gambar 4.3 dapat dilihat bahwasannya objek yang berada di depan kamera teridentifikasi penyakit *Watermelon Mosaic Virus* yang berada di dalam *bounding box* berwarna hijau.





Bounding box akan bergerak mengikuti arah kamera *smartphone* yang akan mengidentifikasi penyakit pada daun semangka yang ada pada *dataset* yang telah dibuat. Sistem yang dibuat juga mampu mengidentifikasi jenis penyakit lainnya yaitu *Downy Mildew*, ataupun jika tanaman semangka dalam kondisi Sehat.


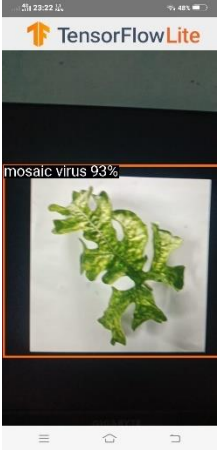

4.3 Pengujian Sistem




Pada tahap ini dilakukan pengujian sistem untuk melihat hasil menggunakan *SSD-MobileNet V2* dalam mengidentifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus*. Tabel pengujian sistem dapat dilihat pada Tabel. 4.1.




Tabel 4. 1 Hasil Pengujian



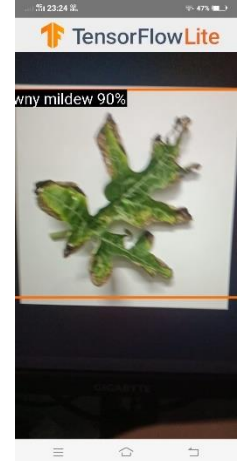
No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Keterangan
1		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil


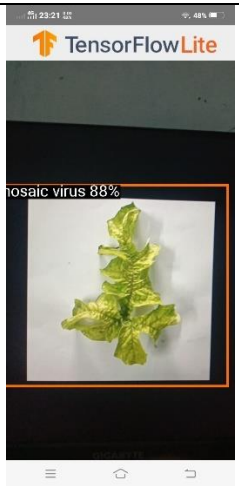
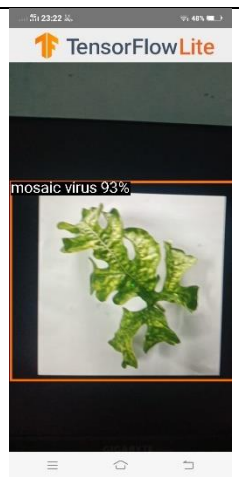
2		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil
3		<i>Mosaic Virus</i>	<i>Healthy</i>	Gagal
4		<i>Mosaic Virus</i>	<i>Downy Mildew</i>	Gagal
5		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil

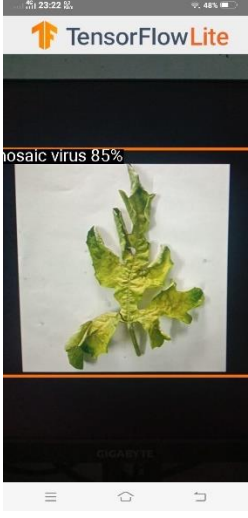
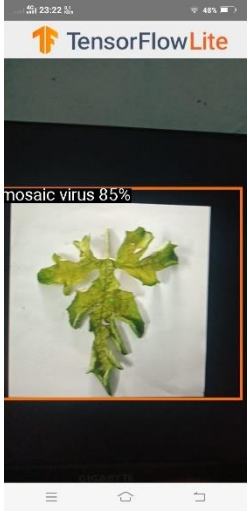

6	 <p>TensorFlow Lite</p> <p>mosaic virus 88%</p>	<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil
7	 <p>TensorFlow Lite</p> <p>mosaic virus 93%</p>	<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil
8	 <p>TensorFlow Lite</p> <p>mosaic virus 85%</p>	<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil




9		<i>Mosaic Virus</i>	<i>Downy Mildew</i>	Gagal
10		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
11		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil




12		<i>Downy Mildew</i>	<i>Mosaic Virus</i>	Gagal
13		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
14		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil


15		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
16		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
17		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil

18		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
19		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil
20		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil

21		<i>Mosaic Virus</i>	<i>Mosaic Virus</i>	Berhasil
22		<i>Mosaic Virus</i>	<i>Downy Mildew</i>	Berhasil
23		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil

24		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
25		<i>Healthy</i>	<i>Healthy</i>	Berhasil
26		<i>Healthy</i>	<i>Healthy</i>	Berhasil

27	 <p>The screenshot shows a mobile app interface with the TensorFlow Lite logo at the top. Below the logo, a green leaf is displayed within a white square frame. Above the leaf, the text "healthy 91%" is visible. The app's status bar at the top shows the time as 23:28 and battery level at 44%.</p>	<i>Healthy</i>	<i>Healthy</i>	Berhasil
28	 <p>The screenshot shows a mobile app interface with the TensorFlow Lite logo at the top. Below the logo, a green leaf with yellowish spots is displayed within a white square frame. Above the leaf, the text "Downy mildew 89%" is visible. The app's status bar at the top shows the time as 23:28 and battery level at 47%.</p>	<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
29	 <p>The screenshot shows a mobile app interface with the TensorFlow Lite logo at the top. Below the logo, a green leaf is displayed within a white square frame. Above the leaf, the text "healthy 91%" is visible. The app's status bar at the top shows the time as 23:28 and battery level at 44%.</p>	<i>Healthy</i>	<i>Heathy</i>	Berhasil

30		<i>Downy Mildew</i>	<i>Downy Mildew</i>	Berhasil
----	---	---------------------	---------------------	----------

Berdasarkan pengujian sistem yang telah dilakukan pada daun semangka, diperoleh nilai akurasi mencapai 93%, terdapat beberapa kesalahan yang dilakukan oleh sistem dalam melakukan identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dari total 120 pengujian yang dilakukan.

Adapun penyebab kesalahan data yang ada dikarenakan banyaknya bentuk ciri khas maupun citra warna daun penyakit yang hampir serupa satu sama lain dalam data yang ada setelah melalui tahap *resizing* sehingga sistem mendapatkan kesulitan yang lebih saat mendeteksi perbedaan antara ketiga kelas yang ada serta faktor cahaya yang juga mempengaruhi dalam mendeteksi jenis penyakit serta tingkat keparahannya.

Tabel 4. 2 *Confusion Matrix*

	<i>WMV</i>	<i>Downy Mildew</i>	<i>Healthy</i>	Total
<i>WMV</i>	36	3	1	40
<i>Downy Mildew</i>	3	36	1	40
<i>Healthy</i>	1	1	38	40
Total	40	40	40	120

Dari Tabel 4.2 didapati bahwa salah satu faktor kesalahan deteksi pada pengujian terjadi antara *WMV*, *Downy Mildew* terjadi dikarenakan warna daun secara keseluruhan cenderung mirip, namun ketika gejala dari penyakit semakin terlihat jelas, deteksi menjadi lebih akurat.

Tabel 4. 3 Nilai TP, FP dan FN dari Tanaman Semangka

	TP	FP	FN
WMV	36	4	4
<i>Downy Mildew</i>	36	4	4
<i>Healthy</i>	38	2	2
Total	110	10	10

Pada Tabel 4.3 dapat dilihat nilai *True Positive (TP)* sebanyak 110 data, dimana data positif dapat diklasifikasikan oleh sistem dengan benar, *False Positive (FP)* sebanyak 10 data, dimana data tidak tepat namun sistem mendeteksinya benar, dan *False Negative (FN)* sebanyak 10 data, dimana sistem salah satu mendeteksi objek sehingga sistem mendeteksinya dengan nilai yang lain.

Tabel 4. 4 Nilai *Precision*, *Recall* dan *F1-Score*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
WMV	0,90	0,90	0,90
<i>Downy Mildew</i>	0,90	0,90	0,90
<i>Healthy</i>	0,95	0,95	0,95

Pada tabel di atas terdapat *precision* yaitu perbandingan antara nilai prediksi benar dengan seluruh hasil yang benar. Adapun rumusnya yaitu :

$$(TP) / (TP+FP).$$

Recall yaitu perbandingan antara nilai prediksi yang benar dengan seluruh data benar. Adapun rumusnya yaitu

$$Recall = (TP) / (TP + FN)$$

F1-Score merupakan perbandingan rata-rata antara nilai *precision* dengan nilai *recall*. Adapun rumus F1-Score yaitu :

$$F1\ Score = 2 * (Recall*Precision) / (Recall + Precision)$$

Berikut persamaan yang digunakan untuk menghitung akurasi dari pengujian data.

$$Akurasi = \frac{Jumlah\ citra\ yang\ benar}{Jumlah\ keseluruhan\ citra} \times 100\%$$

$$Akurasi = \frac{110}{120} \times 100\% = 91,6\%$$

Penelitian dengan judul Identifikasi Penyakit *Downy Mildew* dan

Watermelon Mosaic Virus pada Tanaman Semangka dengan Menggunakan Algoritma *SSD MobileNet V2* Berbasis *Mobile* mampu mengidentifikasi penyakit pada tanaman semangka dengan mendapatkan tingkat akurasi sebesar 91,6%. Sistem yang ada mampu melakukan identifikasi penyakit tanaman semangka secara *realtime* menggunakan *android*.

Terdapat *error* sebesar 8,4% yang terjadi pada sistem dalam melakukan pendeteksian dikarenakan kemiripan yang terjadi di antara beberapa tingkat keparahan penyakit. Tingkat keparahan penyakit beberapa kali memiliki kemiripan setelah dilakukan proses sehingga membuat sistem memiliki kesulitan lebih dalam melakukan pendeteksian untuk mengidentifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus*.

BAB 5

KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan dari Identifikasi Penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada Tanaman Semangka dengan Menggunakan Algoritma *SSD MobileNet V2* Berbasis *Mobile* dan saran untuk referensi pengembangan penelitian selanjutnya.

5.1 Kesimpulan

Hasil dari pengujian aplikasi untuk mengidentifikasi Identifikasi Penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* pada Tanaman Semangka dengan Menggunakan Algoritma *SSD MobileNet V2* Berbasis *Mobile* didapatkan kesimpulan, yaitu:

1. Metode *SSD-MobileNet* dapat mengidentifikasi secara *real-time* penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dengan hasil akurasi baik sebesar 91,6%
2. Beberapa faktor lain yang menyebabkan tingkat akurasi tidak sempurna pada penelitian yang telah dilakukan ini merupakan kesalahan yang terjadi pada proses identifikasi penyakit *Downy Mildew* dan *Watermelon Mosaic Virus*. Hal itu disebabkan karena terjadinya kemiripan dikarenakan objek daun semangka tersebut memiliki bentuk yang mirip setelah melalui proses *resizing* sehingga membuat sistem memiliki kesulitan lebih saat melakukan proses identifikasi.

5.2 Saran

Dari hasil dan kesimpulan yang terdapat pada penelitian ini maka peneliti memberikan saran agar dapat dilakukannya penelitian selanjutnya:

1. Melakukan implementasi dengan metode yang lain sebagai perbandingan pada hasil identifikasi penyakit pada tanaman semangka dari citra daun semangka yang diperoleh menggunakan metode *MobileNet-SSD*.
2. Diharapkan agar dapat menyempurnakan nilai akurasi untuk mendeteksi

penyakit *Downy Mildew* dan *Watermelon Mosaic Virus* dengan berbagai kondisi dan tingkat keparahan.

3. Menambahkan identifikasi penyakit yang lainnya pada tanaman semangka secara *realtime*.

DAFTAR PUSTAKA

- Alfarizi, S., Salsabila, Z., Bahrullah, S., Susanti, H., & Cahyadi, W. A. (2023). Real-Time Cat Detection System using MobileNet-SSD V2. *2023 8th International Conference on Instrumentation, Control, and Automation (ICA)*, 276–280. <https://doi.org/10.1109/ICA58538.2023.10273134>
- Arora, D., Mehta, K., Kumar, A., & Lamba, S. (2024). Evaluating Watermelon Mosaic Virus Seriousness with Hybrid RNN and Random Forest Model: A Five-Degree Approach. *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1–5. <https://doi.org/10.1109/ICRITO61523.2024.10522240>
- Banerjee, D., Kukreja, V., Gupta, A., Singh, V., & Brar, T. P. S. (2023). CNN and SVM-based Model for Effective Watermelon Disease Classification. *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, 1–6. <https://doi.org/10.1109/ASIANCON58793.2023.10270512>
- Chan, A. A. S., Abdullah, M. F. L., Mustam, S. M., Poad, F. A., & Joret, A. (2022). Face Detection in Still Image using SSD MobileNet V2 and Geometrical Algorithm. *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 227–231. <https://doi.org/10.1109/GECOST55694.2022.10010635>
- Damicone, J. P., Edelson, J. V., Sherwood, J. L., Myers, L. D., & Motes, J. E. (2007). Effects of Border Crops and Intercrops on Control of Cucurbit Virus Diseases. *Plant Disease*, 91(5), 509–516. <https://doi.org/10.1094/PDIS-91-5-0509>
- Daniel, A. (2016). *Semangka Tanpa Biji*. Pustaka Batu Press.
- Handayani, M., Taufiq, & Soegiarto. (2017). *Sistem Pakar Diagnosa Penyakit Tanaman Semangka Menggunakan Metode Dempster Shafer Berbasis Web*.
- Kapadnis, S. (2023). *Watermelon Disease Recognition Dataset*. <https://www.kaggle.com/datasets/sujaykapadnis/watermelon-disease-recognition-dataset>
- Kusumanto, R., Tompunu, A. N., & Pambudi, W. S. (2011). Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV. *Seminar Nasional Teknik Elektro 2011*, 83–87.
- Mariana, A. W. (2019). Sistem Pakar Diagnosis Penyakit Tanaman Semangka Menggunakan Metode Certainty Factor. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 3, Issue 1).
- Prima, A. (2023). Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNetv2. *Jurnal Sistim Informasi Dan Teknologi*, 5(2), 208–215.
- Puspitasari, Y., Purwijantiningih, L. M. E., & Pranata, F. S. (2014). Kualitas Selai Lembaran dengan Kombinasi Albedo Semangka (*Citrullus vulgaris* Schard.) dan Buah Naga Super Merah (*Hylocereus costaricensis*). *Jurnal Teknobiologi*, 1–15.
- Silva Junior, F. B. da, Lacerda, C. F. de, Sousa, G. G. de, Sousa, J. T. M. de, Mendonça, A. de M., Silva, M. G. da, Pereira Filho, J. V., & Santos, M. F. dos. (2023). Nutritional status of watermelon irrigated with brackish water in different planting systems. *Revista Brasileira de Engenharia Agrícola e*

- Ambiental*, 27(5), 367–374. <https://doi.org/10.1590/1807-1929/agriambi.v27n5p367-374>
- Suryavanshi, A., Kukreja, V., Srivastava, P., Mehta, S., & Thapliyal, S. (2023). Next-Gen Cucurbit Disease Diagnosis: A Federated Learning CNN Approach. *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 567–573. <https://doi.org/10.1109/SMART59791.2023.10428560>
- Tahir, M., Heluth, A. C., & Widiastuti, H. (2016). UJI AKTIVITAS ANTIOKSIDAN BUAH SEMANGKA (*Citrullus lanatus*) DENGAN METODE FRAP. *Jurnal Farmasi As-Syifaa*, 8(1), 31–38.
- Tunde, A. M. (2019). Determinants of Watermelon Production and its impact on the farmers in Ifelodun L.G.A, Kwara State, Nigeria. *Ruhuna Journal of Science*, 10(2), 149. <https://doi.org/10.4038/rjs.v10i2.80>



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR : 2737/UN5.2.14.D/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca** : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:
Nama : Fakhri Rizha Ananda
NIM : 171402043
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Identifikasi Penyakit Downy Mildew dan Watermelon Mosaic Virus pada Tanaman Semangka dengan Menggunakan Algoritma SSD MobileNet V2 Berbasis Mobile
- Memperhatikan** : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang** : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat** : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN**
- Menetapkan** :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
- Ketua : Dedy Arisandi ST., M.Kom.
NIP: 197908312009121002
- Sekretaris : Fanindia Purnamasari S.TI,M.IT
NIP: 198908172019032023
- Anggota Penguji : Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP: 198603032010121004
- Anggota Penguji : Ivan Jaya S.Si., M.Kom.
NIP: 198407072015041001
- Moderator : -
- Panitera : -
- Kedua** : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga** : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Tembusan :
1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip

Medan
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001