

**DETEKSI KEMATANGAN CABAI RAWIT UNTUK BENIH DENGAN
MENGGUNAKAN SSD-MOBILENET SECARA REALTIME**

SKRIPSI

RICKY MARTIN ABDI NEGARA PURBA

191402126



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

**DETEKSI KEMATANGAN CABAI RAWIT UNTUK BENIH DENGAN
MENGGUNAKAN SSD-MOBILENET SECARA REALTIME**

SKRIPSI

RICKY MARTIN ABDI NEGARA PURBA

191402126



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

PERSETUJUAN

Judul : DETEKSI KEMATANGAN CABAI RAWIT
 UNTUK BENIH DENGAN MENGGUNAKAN SSD-
 MOBILNET SECARA REALTIME
 Kategori : SKRIPSI
 Nama : RICKY MARTIN ABDI NEGARA PURBA
 Nomor Induk Mahasiswa : 191402126
 Program Studi : TEKNOLOGI INFORMASI
 Fakultas : FAKULTAS ILMU KOMPUTER DAN
 TEKNOLOGI INFORMASI

Medan, 10 Januari 2024

Komisi Pembimbing

Pembimbing 2

Dr. Marischa Elveny S.TI, M.Kom
NIP. 199003272017062001

Pembimbing 1

Prof. Drs. Mahyuddin K. M. Nasution, M.IT., Ph.D
NIP. 196712251998021001

Diketahui/disetujui oleh
Program Studi S1 Teknologi Informasi

Dedy Arisandi, S.Kom, M.Kom
NIP. 197908312009121002

PERNYATAAN

DETEKSI KEMATANGAN CABAI RAWIT UNTUK BENIH DENGAN MENGGUNAKAN SSD-MOBILENET SECARA REALTIME

SKRIPSI

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

Medan, 10 Januari 2024

Ricky Martin Abdi Negara Purba

191402126

UCAPAN TERIMAKASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas kehadiran, rahmat, dan karunia-Nya. Berkat-Nya, penulis berhasil menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi S1 Teknologi Informasi di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Penulis juga sangat berterimakasih kepada pihak-pihak yang mendukung penulis dalam penyelesaian karya tulis ini. Pada kesempatan ini penulis ingin mengucapkan syukur dan terimakasih kepada:

1. Ibu dari penulis Udur Nainggolan S.PAK. yang selalu memberikan motivasi dan dukungan dalam perjalanan dari awal sampai akhir masa perkuliahan penulis.
2. Ayah dari penulis Alm. Drs. Tombang Purba yang memberikan motivasi dalam diri saya sehingga penulis mampu menyelesaikan perkuliahan ini.
3. Keluarga penulis, Meilani Purba, Prengky Purba, Dian Purba, dan Cindi Purba yang tiada henti memberikan dukungan dan doa untuk penulis dalam kondisi apapun
4. Bapak Dr. Muryanto Amin, S.Sos., M.Si., selaku Rektor Universitas Sumatera Utara.
5. Bapak Prof. Drs. Mahyuddin M.IT., Ph.D. selaku Dosen Pembimbing 1 penulis yang telah memberikan arahan dan motivasi kepada penulis dalam penyempurnaan skripsi ini.
6. Ibu Dr. Marischa Elveny S.TI., M.Kom. selaku Dosen Pembimbing 2 penulis yang juga ambil bagian dalam membantu penulis dalam penggerjaan skripsi.
7. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc., selaku Dekan Fasilkom-TI Universitas Sumatera Utara.
8. Bapak Dedy Arisandi S.T., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara
9. Seluruh Dosen Program Studi S1 Teknologi informasi dan Ilmu Komputer Universitas Sumatera Utara yang telah memberikan pembelajaran dan memfasilitasi seluruh aspek perkuliahan dari awal semester sampai akhir semester.

10. Senior-senior penulis yang telah memberikan arahan selama masa perkuliahan saya di USU.
11. Teman-teman Teknologi Informasi 2019, terkhusus kepada (Monang, Timothy, Brian, Sheren, Grace O, Legi dan anggota grup Kemana Lagi) yang selalu bersama dalam menjalani perkuliahan.
12. Teman teman Icha-Icha Tactics (William S, Franssedo S, Sutan S, Harlen M, Miranda dan Lainnya), Kasdata SMA Negeri 2 Lintongnihuta yang juga berperan dalam memberikan semangat dan doa untuk menyelesaikan penulisan ini.
13. Teman- teman Sanbes (Santa Lusia 19) terlebih kepada Angeli Silaban, Kevin, Natanael, William, Eben, Nahum, Josua, Mangampu yang sama sama berkuliah di medan dan telah menemani penulis dalam masa perkuliahan.

Medan, 10 Januari 2024

Penulis

DETEKSI KEMATANGAN CABAI RAWIT UNTUK BENIH DENGAN MENGGUNAKAN SSD-MOBILENET SECARA REALTIME

ABSTRAK

Budidaya cabai rawit sangat banyak diminati masyarakat Indonesia walaupun demikian masih banyak petani Indonesia yang mengalami kesulitan dalam pembudidayaan cabai rawit tersebut salah satu diantaranya faktor benih itu sendiri. Kesulitan dalam faktor benih yang dialami petani dalam pembudidayaan cabai rawit dikarenakan harga benih yang mahal dan juga tidak dapat dipastikannya kualitas benih itu sendiri, sehingga sangat dibutuhkan kemampuan dari petani tersebut dalam membuat benih sendiri. Petani diharuskan mampu dalam memilih buah yang bagus untuk dijadikan benih. Masalah yang muncul dalam pemilihan buah untuk benih tersebut sangat konvensional dan memakan waktu untuk orang yang baru ingin melakukan budidaya cabai rawit. Dalam perkembangan dunia teknologi, di bidang Computer Vision dan *Artificial Intelligence* dimanfaatkan sebagai alat untuk pendekripsi buah cabai rawit yang dapat dilakukan pada perangkat seluler. Penelitian ini bertujuan agar sistem mampu mengklasifikasikan buah cabai rawit berdasarkan tingkat kematangannya secara *real-time* dengan menerapkan *SSD-MobileNet* sebagai model arsitektur jaringannya. Sistem yang digunakan berbasis mobile. Data yang digunakan dalam penelitian ini sebanyak 1210 data yang terdiri dari 880 data sebagai data training dan 220 data sebagai data validation. Data testing sebanyak 110 data yang diambil secara *real-time* dengan menggunakan kamera *smartphone*. Penggunaan metode *SSD-MobileNet* dalam mengklasifikasikan tingkat kematangan buah cabai rawit mendapatkan akurasi sebesar 93,6%. Hasil tersebut menunjukkan bahwa sistem yang dibuat dengan menggunakan metode *SSD-MobileNet* telah berhasil dengan baik dalam mengklasifikasikan tingkat kematangan buah cabai rawit untuk dijadikan benih.

Kata Kunci: Cabai Rawit, Budidaya, *Tensorflow Lite*, *SSD-MobileNet*, *Real Time*

**REAL-TIME DETECTION OF RIPENESS LEVEL IN BIRD'S EYE CHILI FOR
SEED SELECTION USING SSD-MOBILENET**

ABSTRACT

The cultivation of bird's eye chili is highly sought after by the Indonesian community. However, despite this popularity, many Indonesian farmers face difficulties in cultivating bird's eye chili, with one of the primary challenges being the seed factor. Farmers encounter challenges related to seed issues in bird's eye chili cultivation due to the high cost of seeds and the uncertainty regarding their quality. Consequently, farmers need the ability to produce their own seeds. Farmers are required to select high-quality fruits to use as seeds, and the process of fruit selection for seed purposes is conventional and time-consuming, especially for those new to bird's eye chili cultivation. With the advancement of technology in the fields of Computer Vision and Artificial Intelligence, these tools are leveraged for fruit detection, making it feasible on mobile devices. The objective of this research is to develop a system capable of real-time classification of bird's eye chili fruits based on their ripeness levels. This is achieved by implementing SSD-MobileNet as the neural network architecture. The system is designed to be mobile-based. The research employs a dataset consisting of 1210 entries, with 880 used as training data and 220 as validation data. Additionally, 110 real-time data points are gathered for testing using a smartphone camera. The application of the SSD-MobileNet method in classifying the ripeness levels of bird's eye chili fruits resulted in an accuracy of 93.6%. This outcome indicates that the system, developed using the SSD-MobileNet method, has successfully and effectively classified the ripeness levels of bird's eye chili fruits for seed purposes.

DAFTAR ISI

PERSETUJUAN	ii
UCAPAN TERIMAKASIH	iii
ABSTRAK	v
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB 1	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Tujuan Penelitian	5
1.4. Batasan Masalah	5
1.5. Manfaat Penelitian	5
1.6. Metodologi Penelitian	6
1.7. Sistematika Penulisan	6
BAB 2	9
LANDASAN TEORI	9
2.1. Tanaman Cabai Rawit	9
2.1.1. Benih Cabai Rawit	9
2.1.2. Tingkat Kematangan Cabai Rawit untuk Benih	9
2.2. Citra Digital	11
2.3. <i>Computer Vision</i>	14
2.4. Tensorflow	15
2.5. Deteksi Objek	15
2.6. Single Shot Detector (SSD)	15
2.7. MobileNet	16
2.8. SSD-MobileNet	17
2.9. TFLite Model	17
2.10. <i>Confusion Matrix</i>	18
2.11. Penelitian Terdahulu	19
BAB 3	24
ANALISIS DAN PERANCANGAN SISTEM	24
3.1. Data yang digunakan	24

3.2. Analisis Sistem	23
3.2.1. <i>Image Acquisition</i>	26
3.2.2. <i>Pre-Processing</i>	26
3.2.3. <i>Image Classification</i>	30
3.2.4. <i>Training Model</i>	33
3.2.5. <i>Learned Model</i>	33
3.2.6. <i>TFLite Model</i>	34
3.2.7. <i>Testing Process</i>	34
3.2.8. <i>Output</i>	34
3.3. Rancangan <i>Interface</i>	34
BAB 4	40
IMPLEMENTASI DAN PENGUJIAN SISTEM	40
4.1. Implementasi Sistem	40
4.1.1. Spesifikasi Perangkat Keras	40
4.1.2. Spesifikasi perangkat lunak	40
4.1.3. Implementasi Data	40
4.2. Pelatihan Sistem	42
4.3. Implementasi Perancangan antarmuka	45
4.4. Prosedur Operasional	47
4.5. Pengujian Sistem	48
BAB 5	70
KESIMPULAN DAN SARAN	70
5.1. Kesimpulan	70
5.2. Saran	70
DAFTAR PUSTAKA	72

DAFTAR TABEL

Tabel 2. 1 <i>Confusion Matrix</i>	18
Tabel 2. 2 Penelitian Terdahulu	21
Tabel 3. 1 Data Training dan Data Validasi	23
Tabel 4. 1 Spesifikasi Perangkat Keras	40
Tabel 4. 2 Spesifikasi Perangkat Lunak	40
Tabel 4. 3 Parameter Proses <i>Training</i>	43
Tabel 4. 4 Pengujian Sistem	49
Tabel 4. 5 Pengujian Berdasarkan Tingkat Pencahayaan	61
Tabel 4. 6 Pengujian Berdasarkan Jarak	64
Tabel 4. 7 Perhitungan <i>Confussion Matrix</i>	67
Tabel 4. 8 Tabel Nilai TP, FP dan FN	67
Tabel 4. 9 Hasil Perhitungan <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i>	68

DAFTAR GAMBAR

Gambar 2.2 Cabai Rawit Setengah Matang (Sumber: https://shopee.co.id/)	10
Gambar 2.2 Cabai Rawit Setengah Matang (Sumber: https://shopee.co.id/)	10
Gambar 2.3 Cabai Rawit Matang Sempurna (Sumber: https://spizeguru.com/)	11
Gambar 2.4 Koordinat Citra Digital	12
Gambar 2.5 Citra Digital Bentuk Matriks	12
Gambar 2.6 Citra Biner	13
Gambar 2.7 Citra Grayscale	13
Gambar 2.8 Citra Warna	14
Gambar 2.9 Arsitektur Single Shot Detector	16
Gambar 3.1 Buah Matang Sempurna	24
Gambar 3.2 Buah Setengah Matang	23
Gambar 3.3 Buah Mentah	23
Gambar 3.4 Arsitektur Umum	25
Gambar 3.6 Sebelum di resize	26
Gambar 3.6 Setelah di resize	26
Gambar 3.7 Pemberian Label	27
Gambar 3.8 Pemberian Bounding Box	27
Gambar 3.9 Hasil Pelabelan Berupa File .xml	28
Gambar 3.10 Data .xml Convert menjadi .csv	29
Gambar 3.11 file train.record	30
Gambar 3.12 Sebelum dan Sesudah Extract Feature Map	31
Gambar 3.13 Operasi konvolusi antara dua matriks	32
Gambar 3.15 Sebelum dan Sesudah Non Maximum Suppression	33
Gambar 3.17 Wireframe Splash Screen	36
Gambar 3.19 Wireframe Deskripsi	37
Gambar 3.20 Wireframe Panduan Aplikasi	38
Gambar 4.1 Cabai Rawit Matang Sempurna	41
Gambar 4.2 Cabai Rawit Setengah Matang	41
Gambar 4.7 Splashscreen	45
Gambar 4.8 Home	46
Gambar 4.10 Halaman Deskripsi	47

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Cabai rawit merupakan jenis tanaman hortikultura dari jenis sayuran yang buahnya berukuran kecil dan memiliki rasa yang pedas. Cabai jenis ini sangat banyak di budidayakan masyarakat khususnya masyarakat Indonesia dikarenakan kebutuhan pasar untuk komoditas ini sangat tinggi, dan bukan hanya diproduksi di dalam negeri, cabai rawit ini sudah masuk pada komoditas *export*. Cabai banyak dibudidayakan masyarakat Indonesia karena harga jual yang tinggi dan iklim di Indonesia sangat cocok untuk budidaya cabai dan juga dapat tumbuh pada dataran tinggi dan juga dataran rendah.

Permintaan cabai rawit terus meningkat dari tahun ke tahun, bersamaan dengan meningkatnya jumlah penduduk dan industri makanan yang membutuhkan bahan dapur dari komoditas cabai rawit. Badan Pusat Statistik (2021) mengeluarkan data bahwa produksi cabai rawit mencapai 1,36 juta ton. Jumlah tersebut turun 8.09% dibandingkan pada tahun 2020 yang sampai menyentuh 1,5 juta ton. Produksi terbesar ada pada Provinsi Jawa Timur dan Provinsi Sumatera Utara pada posisi ke 4 yang dimana masih memungkinkan meningkat jika masyarakat dibantu dalam proses budidaya cabai rawit.

Komoditas cabai rawit merupakan komoditas primer yang banyak dibutuhkan. Namun dalam budidaya tersebut memiliki beberapa kendala diantarnya: Kualitas benih yang tidak bagus, serangan hama dan penyakit, kondisi iklim yang berubah, dan teknis budidaya (Wahyu, 2021). Dari keempat kendala tersebut ada faktor kualitas benih yang dimana sangat diperlukan untuk memperhatikan dalam pemilihan benih yang akan digunakan. Kualitas benih dipengaruhi oleh tingkat kematangan buah, yang cenderung bervariasi pada tanaman. Kematangan buah tidak terjadi serentak, melainkan ada buah yang matang lebih awal, yang setengah matang, yang masih muda, dan bahkan yang terlalu muda. Benih yang dihasilkan dari buah yang telah mencapai kematangan penuh dengan warna merah yang jelas cenderung memiliki bobot yang lebih

tinggi. Proses pemanenan untuk keperluan perbenihan selalu dilakukan secara bertahap, mengingat perbedaan tingkat kematangan pada buah-buah tersebut. (Kartasaputra, 1989).

Proses pemilihan benih dari buah cabai rawit biasanya dilakukan dengan memilih langsung dengan menggunakan mata telanjang. Walaupun cara tersebut dikatakan sederhana tetapi cara tersebut akan memakan waktu yang cukup lama dan terkadang mengalami kendala dalam memilih buah yang benar. Kendala tersebut dipengaruhi oleh faktor penilaian warna setiap manusia yang bersifat subjektif dan penglihatan visual petani tidak konsisten dan juga adanya permasalahan pada kondisi mata. Cara pemilihan diatas akan berdampak pada kesalahan pemilihan benih yang tidak tepat sasaran. Dari masalah tersebut pemilihan benih dari tingkat kematangannya sangat dibutuhkan dan akan lebih baik jika pemilihan tersebut dapat dilakukan secara akurat. Buah yang dipanen saat mencapai masak secara alami menunjukkan pertumbuhan dan produktivitas yang lebih baik, sementara buah yang dipanen sebelum atau setelah mencapai masak secara alami cenderung kurang optimal untuk pertumbuhan dan produksi. Penurunan kualitas tersebut mungkin disebabkan oleh benih yang belum matang sepenuhnya (dipanen sebelum masak secara alami) atau yang telah mencapai masa tua (dipanen setelah masak secara alami)

Deteksi objek merupakan salah satu masalah paling penting dalam bidang jaringan nirkabel *computer vision*. Ini merupakan dasar dari tugas-tugas visi yang kompleks seperti pelacakan target dan pemahaman pemandangan, serta banyak digunakan dalam jaringan nirkabel. Tugas deteksi objek adalah untuk menentukan apakah ada objek yang termasuk dalam kategori yang ditentukan dalam gambar tersebut. Jika ada, maka tugas selanjutnya adalah mengidentifikasi kategori dan informasi lokasinya. Algoritma deteksi objek tradisional biasanya difokuskan pada deteksi beberapa jenis target, seperti deteksi pejalan kaki. (Y. Zhong, *et al.* 2017). Dari penelitian yang dilakukan (Ken T. Murata, 2018) Hasilnya menunjukkan akurasi deteksi yang cukup baik dari semua model. Akurasi *Faster R-CNN* dalam mendekripsi keberadaan bayi tampaknya lebih baik daripada kinerja *SSD*. Namun, dari pengamatan mereka, *SSD* lebih cepat dan memberikan *bounding box* dengan kualitas yang lebih tinggi daripada *Faster R-CNN*.

Realtime merujuk pada pengelolaan data atau informasi yang terjadi secara segera atau dalam waktu nyata. Ini berarti bahwa saat data atau informasi dibuat, dikumpulkan, atau diterima, proses pengolahan atau analisisnya dilakukan secepat mungkin tanpa ada penundaan yang signifikan. Penerapan realtime pada deteksi objek sangat membantu para penggunaanya dikarenakan pengguna mendapatkan informasi secara langsung dan informasi tersebut diperbarui sesuai dengan perubahan yang terjadi pada lingkungan atau sistem yang berjalan.

(Zilvanhisna, *et al.* 2020) melakukan penelitian untuk mengklasifikasikan tingkat kematangan cabai rawit yang berjudul “Penentuan Tingkat Kematangan Cabe Rawit (*Capsicum frutescens L.*) Berdasarkan *Gray Level Co-Occurrence Matrix*”. Perancangan sistem ini menggunakan metode *Backpropagation* dengan menggunakan ekstraksi fitur warna dan tekstur. Ekstraksi fitur nya berdasarkan warna yang diambil dari nilai rata rata saturasi, sedangkan untuk ekstraksi fitur berdasarkan tekstur menggunakan nilai fitur *Gray Level Co-occurrence Matrix* (GLCM) mampu mengklasifikasikan tingkat kematangan cabai rawit ke dalam 4 kelas yaitu cabai rawit matang, cabai rawit orange, cabai rawit matang hijau tua dan cabai rawit tidak matang. Hasil dari pelatihan sistem menunjukkan kemampuan untuk mengklasifikasikan tingkat kematangan cabai rawit dengan tingkat akurasi sebesar 81,4%, dan tingkat akurasi proses pendekripsi cabai rawit sebesar 74,2%. Kemudian penelitian oleh (Febri Liantoni, *et al.* 2018) meneliti kematangan cabai rawit, pada penelitian ini mereka menggunakan *Fuzzy K-Nearest Neighbor* berdasarkan fitur HSV Citra. Hasil dari penelitian ini mendapatkan akurasi sebesar 86,66% dengan jumlah data sebanyak 60 dan dibagi menjadi 4 kelas diantaranya cabai matang, cabai mentah, cabai setengah matang, dan cabai busuk.

Penelitian sebelumnya pernah menggunakan *SSD MobilenetV2* mengidentifikasi gerakan dan tindakan di rumah pintar yang dilakukan oleh (Thi Thu *et al.* 2021). Dalam sistem ini, data diambil dari kamera perangkat seluler yang digunakan untuk mendekripsi objek. Hasilnya adalah objek pada bingkai gambar dengan kotak pembatas (*bounding box*). Hasilnya menunjukkan bahwa sistem ini memenuhi persyaratan dengan akurasi lebih dari 90%, yang cocok untuk aplikasi nyata. Dari penelitian yang dilakukan (Preeti Nagrath, *et al.* 2021)

Model *SSDMNV2* dibandingkan dengan model-model yang sudah ada sebelumnya seperti *LeNet-5*, *AlexNet*, *VGG-16*, dan *ResNet-50* dengan melatih mereka pada dataset yang sama, dan model yang diusulkan unggul dalam hal akurasi, skor F1, dan parameter FPS. Akibatnya, model *SSDMNV2* mudah diimplementasikan pada perangkat embedded yang tidak mungkin dilakukan dengan model-model berat dan digunakan untuk deteksi real-time menggunakan model-model tersebut yang memerlukan daya komputasi yang baik. (W. Rahmiantar, *et al.* 2021) juga melakukan penelitian dengan menerapkan beberapa model untuk mendekripsi manusia dengan menggunakan *NVIDIA jetson* untuk mempercepat penerapan *deep learning*. Penelitian tersebut menyorot kemampuan model dari *PedNet*, *multiped*, *SSD mobileNetV1*, *SSD MobileNetV2*, dan *SSD inceptionV2*. Dalam percobaan yg dilakukan menunjukkan hasil bahwa model *SSDMobileNet* memberikan akurasi tertinggi dengan waktu komputasi tercepat dibandingkan dengan model lain pada dataset video dengan berbagai scenario yg mereka buat. Waktu komputasi yang cepat pada implementasi menggunakan *SSD MobileNet V2* memungkinkan untuk merancang aplikasi online dengan kinerja real-time pada sistem tertanam.

Maka, pada penelitian ini akan mengembangkan sebuah aplikasi untuk mengidentifikasi kematangan cabai rawit untuk keperluan benih menggunakan pengelolaan citra secara *realtime* berbasis mobile dengan menggunakan *SSD-MobileNet*. *SSD-MobileNet* terdiri *SSD(Single Shot Detector)* yang berperan sebagai model dasar dan *MobileNet* yang berperan sebagai model jaringan.

Dari masalah tersebut penulis akan melakukan penelitian dengan membangun sistem agar mampu mendekripsi tingkat kematangan cabai rawit yang akan digunakan oleh petani untuk dijadikan kembali menjadi benih yang baik dengan judul penelitian “**Deteksi Kematangan Cabai Rawit Untuk Dijadikan Benih Dengan Menggunakan Ssd-Mobilenet Secara Realtime**”.

1.2. Rumusan Masalah

Tahap pemilihan buah cabai rawit yang ingin dijadikan benih menjadi faktor penting karena dapat mempengaruhi kualitas benih yang nanti akan dihasilkan, oleh sebab itu petani harus memperhatikan proses pemilihan yang benar. Selama ini proses pemilihan buah calon benih yang dilakukan petani secara tradisional dimana petani melakukan pemilihan secara manual, seperti

memilih satu per satu buah yang dikiranya sudah matang tanpa ada acuan yang tetap dalam proses pemilihan. Hal tersebut dapat menyebabkan benih yang akan dihasilkan tidak memiliki kualitas yang sama, bukan hanya di sisi kualitas, waktu yang termakan dalam proses pemilihan membutuhkan waktu yang banyak. Hal tersebut dipengaruhi oleh kemampuan visual manusia tidak konsisten dan faktor kelelahan yg bisa mengganggu penglihatan. Untuk itu, dibutuhkan sistem yang dapat menyelesaikan masalah tersebut sehingga petani mampu mengidentifikasi benih yang baik untuk dijadikan benih.

1.3. Tujuan Penelitian

Tujuan yang didapatkan dari penelitian ini yaitu untuk menentukan kematangan buah cabai rawit yang paling optimal untuk dijadikan benih dengan menerapkan metode *SSD-MobileNet* secara *realtime* dan mengetahui kemampuan *SSD-MobileNet* dalam mendekripsi objek.

1.4. Batasan Masalah

Batasan masalah ini bertujuan agar penelitian ini dapat berfokus pada tujuan penelitian yang ingin didapat dengan batasan masalah seperti berikut:

1. Objek penelitian ini menggunakan citra cabai yang tumbuh di Doloksanggul, Kab.Humbang Hasundutan, Sumatera Utara.
2. Citra buah cabai rawit tidak terserang penyakit.
3. Citra buah cabai rawit digunakan dengan ekstensi JPG/JPEG.
4. Citra buah cabai rawit yang digunakan menggunakan kamera android.
5. Deteksi kematangan cabai rawit untuk keperluan benih dibagi menjadi 3 kategori cabai umur Setengah matang, Matang Sempurna, dan Mentah.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Membantu petani untuk menentukan tingkat kematangan cabai rawit sehingga pemilihan buah cabai rawit untuk dijadikan benih akan lebih maksimal.
2. Menjadi satu bahan referensi untuk penelitian-penelitian berikutnya yang berkaitan dengan kajian di bidang ini.

1.6. Metodologi Penelitian

Tahapan tahapan yang akan dilakukan dalam penelitian ini ialah:

1. Studi Literatur

Studi literatur melibatkan serangkaian kegiatan yang terkait dengan cara mengumpulkan data dari berbagai sumber tertulis, membaca dan mencatat informasi, serta menyusun materi untuk keperluan penelitian. Pada proses ini penulis mengumpulkan informasi terkait tingkat kematangan buah cabai rawit dan metode *SSD-MobileNet* dari beberapa sumber diantaranya buku, jurnal, skripsi, dan berbagai referensi lainnya.

2. Analisis Masalah

Setelah semua informasi terkumpul, penulis akan melakukan proses analisis masalah sehingga dapat mempertimbangkan metode yang tepat.

3. Perancangan Sistem

Setelah tahap studi literatur dan analisis masalah dilakukan penulis merancang arsitektur umum, mengumpulkan data, menentukan data *training* dan *testing*, dan merancang antarmuka sistem.

4. Pembangunan Sistem

Pembangunan sistem pada tahap ini dengan merealisasikan sistem yang sudah dirancang pada tahap perancangan sistem.

5. Pengujian Sistem

Dalam fase ini, sistem yang telah dibangun akan diuji. Pengujian ini bertujuan untuk memverifikasi bahwa sistem beroperasi secara efisien dan dapat mendekripsi tingkat kematangan cabai rawit.

6. Dokumentasi dan Penyusunan Laporan

Tahap ini merupakan tahapan dokumentasi selama proses penelitian dan menyusun laporan dari proses analisis sampai dengan proses pengujian sistem.

1.7. Sistematika Penulisan

Dalam sistematika penelitian ini dibagi menjadi 5 bab yang dijabarkan sebagai berikut:

BAB 1. PENDAHULUAN

Pendahuluan menjabarkan apa yang menjadi latar belakang pemilihan judul yang difokuskan pada “Deteksi Kematangan Cabai Rawit Untuk Benih Dengan

Menggunakan *Ssd-MobileNet* Secara *Realtime*”. Selain itu, juga disampaikan rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2. LANDASAN TEORI

Landasan teori membahas tentang isu penting terkait pada penelitian. Teori penting yang dibutuhkan dan berkaitan dengan skripsi ini seperti tanaman cabai rawit, benih cabai rawit, jenis tingkat kematangan buah cabai rawit, pengolahan citra digital, *single shot detector* (SSD), *MobileNet* dan beberapa penelitian terkait yang dijadikan sebagai referensi.

BAB 3. ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan akan membahas mengenai arsitektur umum dari metode *SSD-MobileNet* yang akan digunakan untuk memproses citra buah cabai rawit sampai menghasilkan output hasil deteksi tingkat kematangan buah cabai rawit. Dalam pembahasan ini juga akan menampilkan *wireframe* dari tampilan yang dibangun.

BAB 4. IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini, akan diuraikan implementasi sistem yang telah dirancang untuk menyajikan hasil dari pengujian sistem.

BAB 5. KESIMPULAN DAN SARAN

Pada bagian ini, disajikan kesimpulan dari penelitian serta rekomendasi yang dapat diimplementasikan pada penelitian berikutnya.

BAB 2

LANDASAN TEORI

2.1. Tanaman Cabai Rawit

Cabai rawit (*Capsicum frutescens L.*) adalah salah satu tanaman hortikultura dari famili Solanaceae yang tidak hanya memiliki nilai ekonomi tinggi, tetapi juga diketahui karena buahnya yang memiliki kombinasi warna, rasa, dan nilai nutrisi yang lengkap (Kouassi *et al.*, 2012). Tanaman cabai berasal dari daerah tropis dan subtropis di Benua Amerika, terutama Kolombia, Amerika Selatan, dan menyebar luas hingga mencapai Amerika Latin. Penyebaran cabai ke seluruh dunia, termasuk negara-negara di Asia seperti Indonesia, diprakarsai oleh pedagang Spanyol dan Portugis. Diperkirakan terdapat sekitar 20 jenis cabai, yang sebagian besar tumbuh dan berkembang di Benua Amerika. Meskipun demikian, di Indonesia, masyarakat umumnya hanya mengenal beberapa jenis cabai, seperti cabai besar, cabai keriting, cabai rawit, dan paprika (Harpenas dan Dermawan, 2010).

Menurut Warisno dan Dahana (2010), taksonomi tanaman cabai rawit sebagai berikut:

Kingdom	: Plantae
Divisi	: Spermatophyta
Kelas	: Dicotyledonae
Ordo	: Solanales
Famili	: Solanaceae
Genus	: <i>Capsicum</i>
Spesies	: <i>Capsicum frutescens L</i>

Cabai rawit merupakan bahan makanan yang sangat banyak dikonsumsi oleh rakyat Indonesia karena disegala kalangan menggunakan cabai rawit untuk kebutuhan makanan sehari-hari. Cabai rawit mengandung banyak mineral dan vitamin. Buah cabai rawit mengandung vitamin A yang lebih banyak dibandingkan dengan cabai jenis lain. (Triatun, 2019).

2.1.1. Benih Cabai Rawit

Menurut Sarpian (1999), buah dari pohon cabai rawit tidak semuanya layak dijadikan benih kembali. Buah yang akan diambil kembali dan dijadikan benih harus dipilih kembali. Beberapa syarat buah yang baik untuk diambil bijinya dan dijadikan benih sebagai berikut:

1. Pada cabang tanaman paling bawah yang memiliki buah dapat dijadikan sebagai benih dikarenakan lebih cepat tumbuh setelah disemaikan.
2. Buah yang sudah masak dan memiliki kulit yang mengkilap
3. Ukurannya yang lebih besar dari ukuran normalnya.
4. Buah bersih dan tidak terinfeksi penyakit.

Untuk mendapatkan buah yang memenuhi syarat tersebut dapat ditemukan pada tanaman induk yang benar benar baik.

2.1.2. Tingkat Kematangan Cabai Rawit untuk Benih

Kematangan cabai rawit yang ideal untuk benih tergantung pada varietas yang akan dibudidayakan. Untuk varietas yang biasanya tumbuh pada dataran tinggi berikut beberapa pedoman dalam pengambilan buah cabai rawit:

a) Kematangan Fisiologis.

Kematangan fisiologis yang dimaksud yaitu buah sudah mencapai tahap perkembangan normal dan siap untuk dipanen.

b) Warna Buah.

Kematangan cabai rawit akan mengalami perubahan warna mencapai kematangannya. Perubahan warna cabai rawit dari warna hijau sampai dengan warna merah mengkilap.

c) Tekstur kulit.

Kulit yang sudah siap biasanya sudah mengkerut dan keriput tetapi tetap keras

d) Ukuran Buah

Buah yang memiliki ukuran lebih besar dari normalnya.

Berikut tingkat kematangan buah cabai rawit:

1. Buah Cabai Rawit Mentah

Buah cabai rawit yang mentah biasanya memiliki warna hijau dan memiliki ukuran yang lebih kecil. Pada tahap ini buah cabai rawit tersebut tidak dapat dijadikan benih walaupun demikian buah yang berwarna hijau

memiliki tingkat ketahan pada suhu ruangan lebih lama dibandingkan buah yang hampir matang dan yang matang sempurna. Contoh gambar cabai rawit berwarna hijau.



Gambar 2.2 Cabai Rawit Setengah Matang (Sumber: <https://shopee.co.id/>)

2. Buah Cabai Rawit Setengah Matang

Buah cabai rawit yang setengah matang belum memasuki masa masak fisiologis walaupun pada buah nya sudah tampak perubahan warna dari sebelumnya. Buah yang setengah matang ini mempunyai ciri fisik yang mempunyai warna orange hampir merah dan ada juga yang mempunyai warna kuning. Pada masa ini buah ini masih belum optimal jika dijadikan benih dikarenakan buah tersebut masih mengandung banyak air dan kemampuannya dalam berkecambah kurang optimal. Contoh gambar buah cabai rawit yang setengah matang:



Gambar 2.2 Cabai Rawit Setengah Matang (Sumber: <https://shopee.co.id/>)

3. Buah Cabai Rawit Matang Sempurna

Buah cabai rawit matang sempurna yang sudah mencapai matang fisiologis. Buah yang matang ditandai dengan warna buah yang berwarna merah dan mempunyai tekstur yang keriput. Pada tahap inilah benih yang bagus dapat dihasilkan karena pada tahap ini buah kadar air pada buah paling rendah dan kemampuan biji nya dalam berkecambah sangat baik. Walaupun demikian buah yang sudah matang tidak dapat bertahan lama oleh karena itu banyak petani yang mengeringkan buah tersebut agar dapat bertahan lama. Contoh gambar buah matang sempurna:

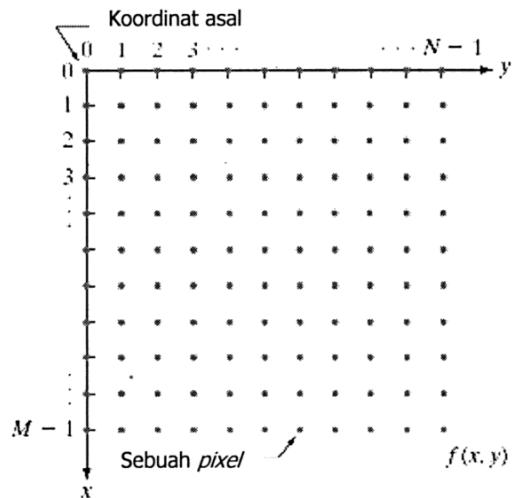


Gambar 2.3 Cabai Rawit Matang Sempurna (Sumber: <https://spizeguru.com/>)

2.2. Citra Digital

Citra digital merupakan representasi kumpulan nilai-nilai yang merepresentasikan bilangan real dan kompleks melalui penggunaan urutan bit tertentu. Suatu citra dapat dianggap sebagai fungsi $f(x, y)$ dengan dimensi M baris dan N kolom, di mana x dan y merepresentasikan koordinat spasial, dan nilai f pada koordinat (x, y) dikenal sebagai tingkat intensitas atau tingkat keabuan citra pada titik tersebut. Apabila nilai-nilai x, y, dan f secara keseluruhan bersifat terbatas dan diskrit, maka citra tersebut dapat

dikategorikan sebagai citra digital. Ilustrasi di bawah memberikan gambaran tentang lokasi koordinat dalam konteks citra digital.



Gambar 2.4 Koordinat Citra Digital

Citra digital dalam bentuk matriks berukurab $N \times M$.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \dots\dots (2.1)$$

Gambar 2.5 Citra Digital Bentuk Matriks

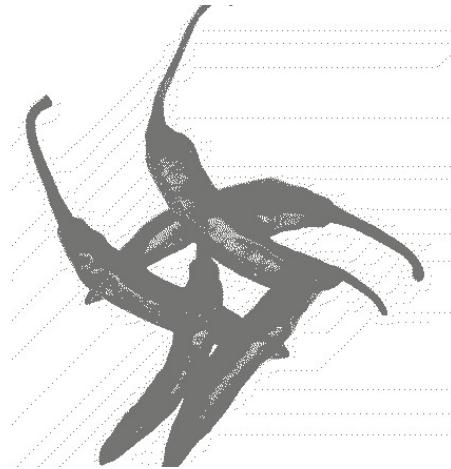
2.2.1. Pengelolaan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah disiplin ilmu yang fokus pada teknik-teknik pengolahan citra, mencakup gambar diam (foto) dan gambar bergerak (yang diambil dari webcam), dan dilakukan secara digital menggunakan komputer (Sutoyo, et al. 2009). Secara matematis, citra dianggap sebagai fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi. Untuk dapat diolah oleh komputer digital, citra perlu direpresentasikan secara numerik dengan nilai-nilai diskrit. Proses transformasi dari representasi fungsi kontinu ke nilai-nilai diskrit ini disebut digitalisasi citra.

Berikut merupakan jenis-jenis citra berdasarkan nilai pixelnya.

- a. Citra Biner

Citra biner adalah citra digital yang memiliki dua opsi nilai pixel, yakni hitam dan putih. Citra biner sering disebut sebagai citra *monokrom*, di mana setiap pixel dalam citra biner hanya memerlukan 1 bit untuk merepresentasikan nilainya.



Gambar 2.6 Citra Biner

b. Citra *Grayscale*

Citra *grayscale* adalah citra digital yang memiliki nilai kanal yang sama pada setiap *pixel*, yang berarti nilai *RED=GREEN=BLUE*. Warna-warna yang terdapat dalam gambar ini melibatkan hitam, berbagai tingkat keabuan, dan putih. Variasi tingkatan keabuan dalam citra *grayscale* mencakup rentang dari hitam hingga mendekati putih, dengan tingkat keabuan yang mewakili warna abu. Contohnya, citra *grayscale* berikut memiliki kedalaman warna 8 bit, yang berarti terdapat 256 kombinasi warna keabuan yang berbeda.



Gambar 2.7 Citra *Grayscale*

c. Citra Warna

Citra warna terbentuk dari tiga komponen warna dasar, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*). Secara umum, komputer menampilkan citra warna dengan menggunakan 8, 16, atau 24 bit untuk setiap pixelnya. Jumlah bit pada setiap pixel menentukan seberapa banyak warna yang dapat ditampilkan. Masing-masing komponen warna dasar menggunakan penyimpanan sebanyak 8 bit (1 *byte*), sehingga setiap warna memiliki rentang nilai antara 0 hingga 255. Dengan menggunakan 8 bit untuk setiap komponen warna dasar, setiap pixel dapat memiliki kombinasi warna sebanyak $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16.777.216$ warna.



Gambar 2.8 Citra Warna

Adapun teknik pengelolaan citra yang digunakan pada penelitian ini ialah *resizing*.

a. *Resizing*

Teknik dalam pengelolaan citra yang bertujuan untuk mengubah ukuran suatu citra digital, baik dalam memperbesar atau memperkecil dengan merubah nilai pada tiap pixel yang membangun suatu citra agar mempunyai data yang seragam.

2.3. Computer Vision

Computer vision adalah teknologi yang memungkinkan komputer untuk secara otomatis melihat dan mengenali objek di sekitarnya, mirip dengan kemampuan visual manusia. Proses ini melibatkan serangkaian tahap, termasuk akuisisi citra, pemrosesan citra, klasifikasi, pengenalan, dan pengambilan keputusan. Tujuan utama dari *computer vision* adalah mengkomputerisasi kemampuan penglihatan manusia, yang berarti

menciptakan citra digital yang mencerminkan apa yang dapat dilihat oleh manusia. Dengan demikian, input dari computer vision berupa citra yang merepresentasikan penglihatan manusia, sementara outputnya berupa citra digital yang telah melalui berbagai tahap pemrosesan.

2.4. Tensorflow

Tensorflow adalah sebuah platform open-source yang menyediakan solusi lengkap untuk aplikasi machine learning. Ini berfungsi sebagai sebuah *library* yang menggunakan aliran data dan berbagai bahasa pemrograman untuk menjalankan beragam tugas yang terkait dengan pelatihan dan penggunaan jaringan neural yang mendalam. *Tensorflow*, yang dikembangkan oleh Google, adalah salah satu perpustakaan *deep learning* yang paling terkenal saat ini. Google menggunakan teknologi *machine learning* ini untuk meningkatkan berbagai aspek seperti mesin pencarian, terjemahan, analisis gambar, dan rekomendasi.

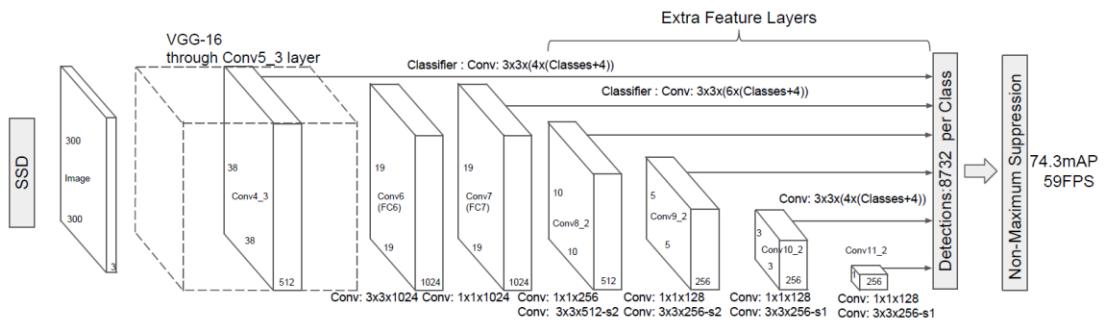
2.5. Deteksi Objek

Deteksi objek merupakan salah satu tugas penting di dalam domain *computer vision* yang bertujuan untuk mengenali serta memahami posisi dan jenis objek dalam gambar atau video. Dalam proses deteksi objek harus melatih dan menguji data yang sudah di labeli pada objek disetiap kelasnya untuk pengenalannya. Ada beberapa tipe model yang biasa digunakan pada *deep learning* seperti *PASCAL-VOC*, *YOLO*, *CreateML*. Pada penelitian ini menggunakan *PASCAL-VOC*.

2.6. Single Shot Detector (SSD)

Single shot detector merupakan metode deteksi objek dalam deep learning yang digunakan untuk mendeteksi objek dalam gambar. SSD dirancang untuk mendeteksi objek dalam berbagai kategori dengan cepat dan akurat. *YOLO*, *MobileNet + SSD*, *SqueezeDet* adalah beragam model yang termasuk dalam keluarga detektor tembakan tunggal. Perbedaan utama antara varian ini adalah pembuat enkodenya dan juga konfigurasi spesifik dari jangkar yang telah ditentukan. Model *MobileNet+SSD* menampilkan *encoder* berbasis *MobileNet*, *SqueezeDet* meminjam *encoder* *SqueezeNet*, dan model *YOLO* juga menampilkan arsitektur konvolusionalnya. Nah dengan begitu SSD merupakan pilihan tepat untuk model yang ditujukan untuk perangkat seluler atau tertanam.

SSD bekerja seperti berikut, setiap gambar input ditandai dengan area bounding box dengan berbagai ukuran dan pada setiap kisi, pendeksi dilakukan pada kelas dan rasio aspek yang berbeda. SSD melakukan perbandingan objek terhadap default bounding box dengan berbagai rasio selama masa training.



Gambar 2.9 Arsitektur Single Shot Detector

SSD menampilkan arsitektur yang terlihat pada Gambar 2.9. Pada langkah awal, SSD mengubah dimensi setiap gambar dalam dataset menjadi 300x300 piksel. Proses konvolusi kemudian diterapkan pada gambar tersebut melalui layer *Mobilenet*, yang bertindak sebagai *ekstraktor* fitur. Selanjutnya, lima layer berikutnya memainkan peran sentral dalam struktur SSD. Pada lapisan-lapisan tersebut, SSD dilengkapi dengan *Multi-Scale Feature Map* untuk meningkatkan akurasi dalam proses pendeksi. Dalam langkah eliminasi *default box*, SSD memanfaatkan metode *Non-Maximum Suppression*. Keseluruhan proses ini menghasilkan 8732 prediksi, dengan tingkat akurasi yang mencapai nilai tinggi, sementara kecepatannya mencapai 59 *frame per second* (FPS).

2.7. MobileNet

MobileNet adalah suatu model arsitektur *Convolutional Neural Network (CNN)* yang telah menjalani tahap *pretraining* dengan menggunakan konvolusi yang dapat dipisahkan secara mendalam. Dirancang khusus untuk membangun jaringan saraf yang ringan dan memiliki kinerja cepat, *MobileNet* terutama difokuskan untuk aplikasi seluler. *Google Inc.* adalah pihak yang mengembangkan model ini.

MobileNet menjadi pilihan yang baik ketika diperlukan sumber daya komputasi yang terbatas. Model ini menonjol karena memiliki daya komputasi yang rendah, membuatnya cocok untuk implementasi transfer learning. Kelebihan ini sangat bermanfaat untuk perangkat seluler, sistem tersemat, dan komputer yang tidak

dilengkapi dengan GPU atau memiliki efisiensi komputasi yang terbatas, meskipun dengan pengorbanan sedikit akurasi hasil. Struktur lapisan konvolusi *MobileNet* terdiri dari sepuluh blok. Blok pertama menggunakan konvolusi standar, menghasilkan 32 fitur. Blok-blok berikutnya mengadopsi metode konvolusi yang dapat dipisahkan, diikuti oleh *down-sampling* dengan *max pooling*. Peta skematik dari fitur-fitur tersebut mengalami peningkatan secara eksponensial hingga mencapai 1024 fitur pada blok terakhir, yang dicapai melalui operasi perkalian biner.

Menggunakan lebih banyak layer tidak serta merta membuat model lebih efisien karena berkaitan dengan ukuran dan kecepatan, karena di dunia nyata seperti robotika, mobil *self-driving*, dan *augmented reality* memerlukan waktu sebaik mungkin mengingat kemampuan komputasi perangkat yang terbatas. *MobileNet* adalah model arsitektur CNN yang dirancang secara efektif dengan dua set *hyper-parameter* untuk membuat model latensi yang sangat kecil dan rendah yang mudah diimplementasikan sesuai dengan kebutuhan aplikasi seluler dan *embedded applications*. *MobileNet* dibangun di *depthwise separable convolutions* untuk mengurangi perhitungan di lapisan awal.

2.8. SSD-MobileNet

SSD-MobileNet mengintegrasikan konsep dari SSD dalam mendekripsi objek dengan keunggulan dalam efisiensi komputasi yang dimiliki oleh *MobileNet*. Hasilnya adalah model yang mampu dengan cepat dan tepat mengenali objek di berbagai jenis perangkat, termasuk perangkat mobile. Kinerja dan efisiensi yang tinggi ini membuat *SSD-MobileNet* sering menjadi pilihan yang sangat baik untuk tugas deteksi objek dalam berbagai skala dan berbagai jenis objek, dan juga sangat cocok untuk aplikasi yang memerlukan pemrosesan *real-time* serta beroperasi pada perangkat dengan sumber daya komputasi yang terbatas.

2.9. TFLite Model

TensorFlow Lite (TFLite) adalah kumpulan alat untuk mengonversi dan mengoptimalkan model *TensorFlow* agar berjalan di perangkat seluler. *TensorFlow Lite* menggunakan model *tensorflow* yang diubah menjadi format model *machine learning* yang lebih kecil dan efisien.

2.10. Confusion Matrix

Evaluasi teknik klasifikasi dapat diperoleh dalam hal kebenaran dengan menghitung ukuran statistik, yaitu *True Positives (TP)*, *True Negatives (TN)*, *False Positive (FP)*, dan *False Negatives (FN)*. Komponen-komponen ini membentuk Matriks Konfusi seperti yang ditunjukkan dalam Tabel 2.1. *Confusion Matrix* adalah tabel yang dapat dihasilkan untuk suatu pengklasifikasi pada sebuah set data biner dan dapat digunakan untuk menggambarkan kinerja pengklasifikasi tersebut.

Tabel 2. 1 *Confusion Matrix*

		Prediksi	
		<i>Positif</i>	<i>Negatif</i>
Aktual	<i>Positif</i>	TP	FN
	<i>Negatif</i>	FP	TN

Matriks ini didasarkan pada ketentuan sebagai berikut :

- a. TP (*True Positive*): Merujuk pada aktual ya dan prediksi ya
- b. FN (*False Negative*): Merujuk pada aktual ya dan prediksi tidak
- c. FP (*False Positive*): Merujuk pada aktual tidak prefiksi ya
- d. TN (*True Negative*): Merujuk pada aktual tidak prefiksi tidak

Ada beberapa *performance matrix* umum yang dapat diukur seperti *accuracy*, *precision* dan *recall*.

1. Accuracy

Akurasi adalah suatu metode evaluasi yang mengukur sejauh mana kesamaan antara nilai yang diprediksi oleh suatu model dan nilai yang sebenarnya. Tingkat kebenaran dari prediksi dapat diukur dengan menghitung jumlah data yang diklasifikasikan dengan tepat oleh model tersebut.

$$\text{Accuracy} = \frac{\text{jumlah data uji benar}}{\text{jumlah keseluruhan data uji}} \times 100\%$$

2. *Precision*

Precision adalah sebuah metode evaluasi yang membandingkan jumlah data yang relevan yang berhasil diidentifikasi oleh sistem dengan keseluruhan data yang ditemukan oleh sistem, tanpa memandang apakah data tersebut relevan atau tidak.

$$\text{Precision} = \frac{\sum TP}{\sum TP + FP} \times 100\%$$

3. *Recall*

Recall, yang sering disebut sebagai tingkat kemampuan model dalam menangkap informasi yang relevan, adalah metrik yang mengindikasikan sejauh mana prediksi positif yang akurat dibandingkan dengan total data positif yang sebenarnya.

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN} \times 100\%$$

4. *F1-Score*

F1-Score adalah suatu metrik yang menyatukan presisi dan recall dalam suatu perbandingan. Rumus yang digunakan untuk menghitung *F1-Score* adalah:

$$F1Score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

2.11. Penelitian Terdahulu

Pada konteks masalah ini, beberapa penelitian telah menginvestigasi klasifikasi tanaman cabai rawit. Sebagai contoh, penelitian yang dilakukan oleh (Febri & Fitri, 2018) fokus pada kematangan cabai rawit. Dalam penelitian tersebut, mereka menerapkan metode *Fuzzy K-Nearest Neighbor* berdasarkan fitur HSV citra. Hasilnya menunjukkan akurasi sebesar 86,66% dengan menggunakan jumlah data sebanyak 60, yang dibagi menjadi empat kelas, yaitu cabai matang, cabai mentah, cabai setengah matang, dan cabai busuk.

Penelitian lain yang membahas tingkat kematangan cabai rawit dilakukan oleh (Zilvanhisna et al., 2020). Mereka meneliti tingkat kematangan cabai rawit dengan memanfaatkan *Gray Level Co-Occurrence Matrix* dan menerapkan metode backpropagation. Hasil penelitian ini mencapai tingkat akurasi sebesar 74,2%. Pada penelitian ini, tangkai cabai rawit ditampilkan, berbeda dengan penelitian sebelumnya yang tidak menampilkan tangkai tersebut.

Selanjutnya, dalam penelitian yang membahas klasifikasi cabai, (Ilyas et al., 2020) melakukan penelitian terhadap jenis cabai, termasuk cabai gunung, cabai rawit

Taiwan, cabai keriting merah, cabai keriting hijau, dan cabai rawit putih. Pengujian sistem dilakukan dengan menggunakan metode *Convolutional Neural Network* (CNN) untuk proses klasifikasi jenis cabai. Hasil dari penelitian ini menunjukkan akurasi sebesar 80%, yang masih dapat ditingkatkan dengan menggunakan lebih banyak data dalam proses pengolahan.

Selanjutnya, (Nufus et al., 2021) telah mengembangkan sebuah sistem pendekripsi pejalan kaki menggunakan gambar 360° yang telah dinormalisasi, berbasis pada *SSD MobileNet V2*. Sistem yang dihasilkan menggunakan teknologi *deep learning*. Hasil pengujian offline pada 548 gambar 360° pada kondisi siang hari menunjukkan bahwa 60,40% gambar terdeteksi dengan sempurna. Pada pengujian yang sama untuk 514 gambar 360° ternormalisasi pada kondisi sore hari, diperoleh hasil bahwa 62,25% gambar terdeteksi dengan sempurna. Pada pengujian secara real-time, diperoleh tingkat kepercayaan (*confident level*) sebesar 90% untuk pendekripsi pejalan kaki pada siang hari, dan 85% pada sore hari.

Selanjutnya, Huong Nguyen Thi Thu, *et al* yang melakukan kombinasi SSD dengan *MobilenetV2* untuk mengidentifikasi bentuk Gerakan pada penerapan rumah pintar. Penelitian menggunakan 11 jenis gerakan yang dominan dan data tersebut diambil dengan menggunakan kamera perangkat seluler. Hasilnya menunjukkan bahwa sistem memenuhi syarat dengan akurasi 90% yang cocok digunakan pada dunia nyata.

Selanjutnya, (Fadli, 2021) melaporkan implementasi *SSD-MobileNet* dalam mengklasifikasikan jenis ras kucing. Proses klasifikasi dilakukan pada perangkat desktop, dan jenis ras kucing yang diklasifikasikan meliputi sphynx, Himalaya, bengal, persia, dan moggy. Hasil penelitian menunjukkan bahwa sistem mencapai akurasi sebesar 93,8%.

Tabel 2. 2 Penelitian Terdahulu

No.	Penulis	Judul	Tahun	Keterangan
1.	Febri Liantoni & Fitri Nur Annisa	Fuzzy K-Nearest Neighbor Pada Klasifikasi Kematangan Cabai Berdasarkan Fitur Hsv Citra	2018	Penerapan metode <i>Fuzzy K-Nearest Neighbor</i> berdasarkan fitur HSV citra dalam penelitian ini menghasilkan tingkat akurasi sebesar 86,66%.
2.	Zilvanhisna E F, et al.	Penentuan Tingkat Kematangan Cabe Rawit (<i>Capsicum frutescens L.</i>) Berdasarkan Gray Level Co-Occurrence Matrix	2020	Penelitian ini mengadopsi metode <i>backpropagation</i> dengan memanfaatkan <i>Gray Level Co-Occurrence Matrix</i> untuk melakukan klasifikasi pada buah cabai rawit. Hasil evaluasi menunjukkan tingkat akurasi sebesar 74,2%.
3.	Ilyas Perlindungan & Risnawati	Pengenalan Tanaman Cabai Dengan Teknik Klasifikasi Menggunakan Metode Cnn	2020	Penelitian ini menggunakan metode CNN untuk mengklasifikasikan jenis cabai dengan mendapatkan akurasi sebesar 80%.
4.	Huong Nguyen Thi Thu, et al	Proposing a Recognition System of Gestures Using MobilenetV2 Combining Single Shot Detector Network for Smart-Home Applications	2021	Menggunakan <i>SSD-MobileNet</i> untuk mengidentifikasi Gerakan dan tindakan pada rumah pintar. Penelitian ini mendapatkan akurasi lebih dari 90%.
5.	N. Nufus, et al.	Sistem Pendekripsi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi	2021	Penggunaan <i>SSD-MobileNet</i> dalam pengujian secara real-time menghasilkan tingkat kepercayaan sebesar 90% untuk pendekripsi pejalan kaki pada siang hari dan 85% pada sore hari.
6.	Muhammad Fadli	Klasifikasi Jenis Ras Kucing Menggunakan Ssd-Mobilenet Secara Real-Time Berbasis Desktop0070	2021	Menggunakan <i>SSD-MobileNet</i> pada klasifikasi kucing yang dibagi menjadi 5 kelas dan mendapatkan akurasi sebesar 93,8%.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data yang digunakan

Penelitian ini menggunakan citra buah cabai rawit yang tumbuh di area pertanian di Kecamatan Doloksanggul, Kabupaten Humbang Hasundutan. Citra buah diambil dengan menggunakan kamera pada perangkat POCO X3 NFC yang memiliki resolusi 64 MP. Saat pengambilan citra buah, jarak antara kamera dengan objek adalah 10-15 cm. Citra-citra ini memiliki format .jpg dengan ukuran pixel rata-rata 3472 * 3472, dan sebanyak 1100 data. Selanjutnya, data yang terkumpul dibagi menjadi dua bagian, yakni data training sebanyak 880 data dan data validasi sebanyak 220 data. Sebagai contoh, berikut adalah citra buah cabai rawit:



Gambar 3.1 Buah Matang Sempurna



Gambar 3.2
Buah Setengah Matang



Gambar 3.3 Buah Mentah

Terdapat tiga tingkatan kematangan yang dibutuhkan untuk mendukung penelitian ini yaitu citra buah matang sempurna, buah setengah matang, buah mentah. Tabel 3.1 Besar rasio perbandingan antara data latih dan data validasi sebesar 80 : 20.

Tabel 3. 1 Data Training dan Data Validasi

Data Training	Data Validasi
880	220

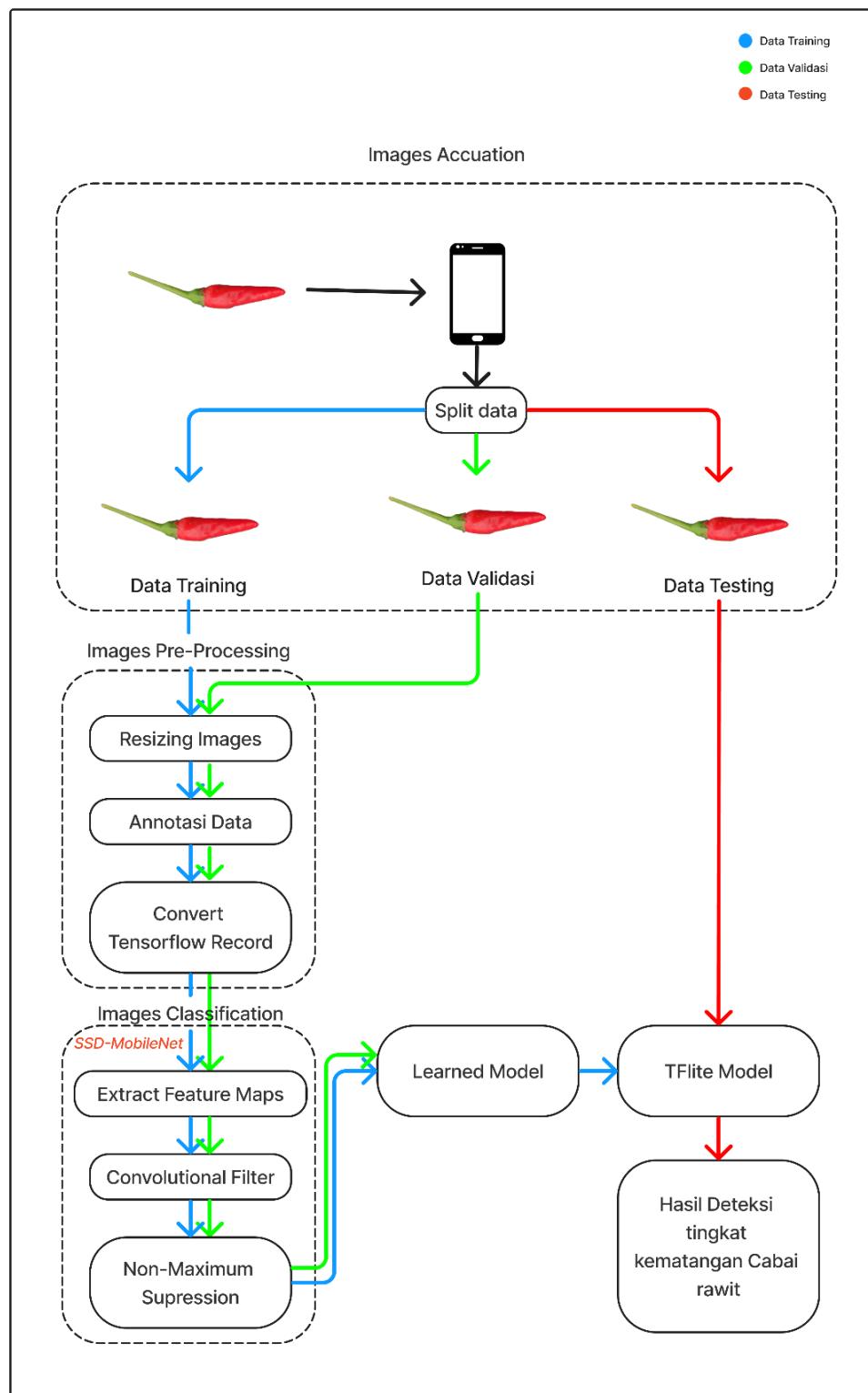
3.2. Analisis Sistem

Penelitian ini akan melibatkan beberapa tahapan. Tahapan pertama ialah pengumpulan data atau *image acquisition*, yang melibatkan citra buah cabai rawit yang akan digunakan sebagai data *training* dan validasi. Setelah itu, data citra akan masuk ke

dalam tahap yang disebut *pre-processing* atau tahap persiapan data. Pada tahap ini, data akan mengalami proses *resize* citra, di mana ukuran pixelnya diubah untuk mengurangi beban dalam proses pelatihan. Setelah tahap *pre-processing* data tidak langsung masuk proses *training*, data yang dikumpulkan masuk dalam proses annotasi data yaitu pemberian label pada setiap citra untuk menetukan *ground truth box*-nya. Data annotasi yang sudah dihasilkan berformat .xml akan diubah ke dalam format .csv.

Data dengan format .csv tersebut diubah ke dalam data *TFRecord* agar dapat diimplementasikan dengan menggunakan *library Tensorflow*. Setelah data dalam bentuk *TFRecord* data sudah bisa masuk dalam proses *training*, tapi sebelum masuk tahap *training* perlu untuk penyesuaian jumlah kelas, *batch_size*, *num_steps*, *fine_tune_checkpoint_type*, dan selanjutnya mengubah letak *path* dari direktori *checkpoint* dan file *TFRecord* dan juga file *labelmap*-nya selanjutnya proses *training* dilakukan. Dengan demikian, model yang dihasilkan akan di *convert* ke dalam format *TF-lite*, agar dapat digunakan pada sistem *android*.

Setelah melewati semua proses tersebut, citra buah cabai rawit berhasil dikategorikan berdasarkan kelasnya. Tahapannya dapat dilihat pada arsitektur di bawah pada gambar 3.4.

**Gambar 3.4** Arsitektur Umum

3.2.1. *Image Acquisition*

Pada tahap ini, awal dari sistem dimulai dengan mengumpulkan citra buah cabai rawit. Citra ini diperoleh dengan menggunakan kamera smartphone dan diambil dalam format gambar JPEG/JPG. Citra yang sudah dikumpulkan akan dibagi menjadi dua kelompok, yaitu data training dan data validasi.

3.2.2. *Pre-Processing*

Tahap ini merupakan persiapan awal terhadap citra yang telah dikumpulkan untuk mempermudah proses selanjutnya. Berikut adalah tahapan dalam proses *pre-processing*:

1. *Resizing*

Resizing dilakukan guna mengubah ukuran citra baik memperbesar maupun memperkecil ukuran citra. Tujuan dari proses *resizing* ini untuk menyeragamkan ukuran citra yang akan digunakan selama proses *training* dan *testing*. Ukuran citra yang dikumpulkan berukuran 3472×3472 pixels dan memiliki ukuran data rata-rata 5 MB. Pada proses ini data di *resizing* menjadi 640×640 pixels.



Gambar 3.6 Sebelum di *resize*

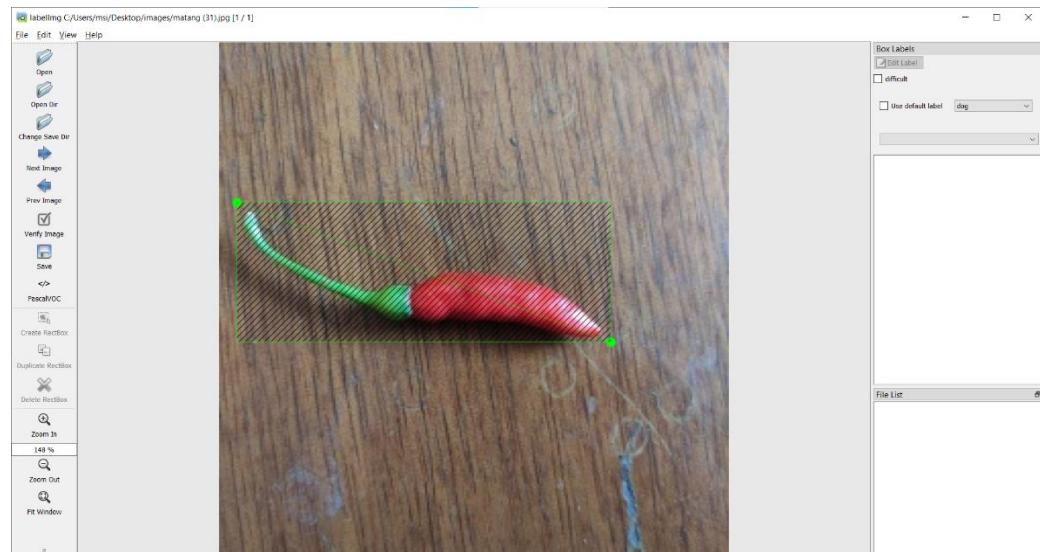


Gambar 3.6 Setelah di *resize*

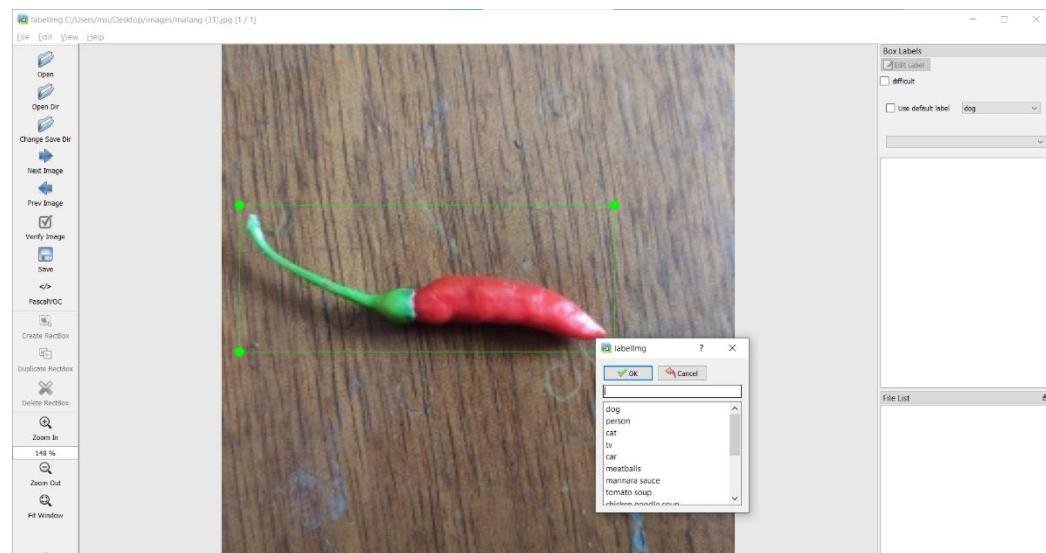
2. *Data Annotation*

Tahap ini dilakukan dengan proses pelabelan pada seluruh citra untuk menentukan *ground box*, yang bertujuan memberikan penanda untuk setiap kelas tertentu. Proses ini menghasilkan representasi suatu wilayah dalam bentuk daftar titik koordinat dalam kurva yang tertutup, mencakup deskripsi luas dan perimeter wilayah tersebut.

Hasil pelabelan pada anotasi data ini berformat .xml dari *PASCALVOC* yang berjumlah 1100 data. Informasi yang ada pada data hasil anotasinya ialah titik koordinat (xmax, ymin, xmin, ymax) pada label citra juga hasil. Proses pelabelan ini memakan waktu selama kurang lebih 5 jam dan dimana pelabelannya dilakukan dalam 1 hari tersebut dengan istirahat 15 menit setiap satu jam. Gambar 3.7 dan gambar 3.8 merupakan proses pemberian anotasi gambar menggunakan *labelImg*.

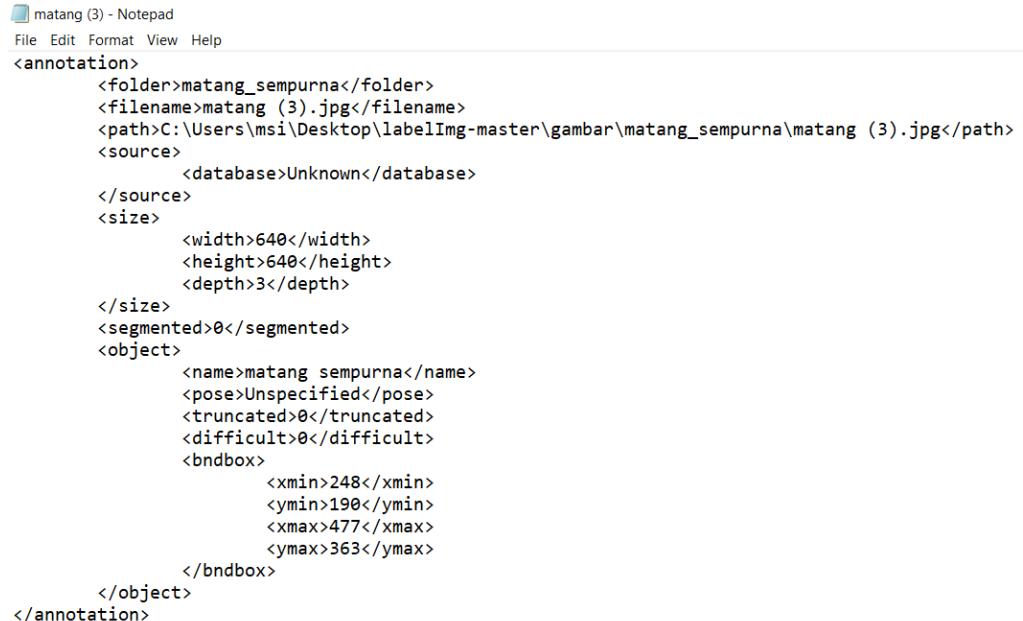


Gambar 3.8 Pemberian *Bounding Box*



Gambar 3.7 Pemberian Label

Setelah akses ke citra sudah berhasil, tahap selanjutnya merupakan proses pemberian *bounding box* seperti pada gambar 3.7. Setelah seleksi *bounding box* dilakukan, dilanjutkan dengan pemberian kelas pada objek yang dimana pada penelitian ini menggunakan 3 kelas yaitu matang, setengah matang, matang sempurna. Berikut isi file anotasi tersebut.



```

matang (3) - Notepad
File Edit Format View Help
<annotation>
    <folder>matang_sempurna</folder>
    <filename>matang (3).jpg</filename>
    <path>C:\Users\msi\Desktop\labelImg-master\gambar\matang_sempurna\matang (3).jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>640</width>
        <height>640</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>matang sempurna</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>248</xmin>
            <ymin>190</ymin>
            <xmax>477</xmax>
            <ymax>363</ymax>
        </bndbox>
    </object>
</annotation>

```

Gambar 3.9 Hasil Pelabelan Berupa File .xml

3. Generate CSV File

Data yang telah diberikan label akan disimpan dalam format .xml. Selanjutnya file hasil labeling dengan formal .xml tersebut akan di konversi ke dalam formal CSV. *Generate CSV File* bertujuan agar file digabungkan menjadi satu yang berekstensi .csv. berikut tampilan isi file csv nya.

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	matang (1).jpg	640	640	matang sempuma	19	104	473	363
3	matang (10).jpg	640	640	matang sempuma	206	108	313	394
4	matang (100).jpg	640	640	matang sempuma	154	313	411	436
5	matang (101).jpg	640	640	matang sempuma	229	294	445	430
6	matang (102).jpg	640	640	matang sempuma	237	311	412	408
7	matang (103).jpg	640	640	matang sempuma	254	325	488	438
8	matang (104).jpg	640	640	matang sempuma	222	275	491	385
9	matang (105).jpg	640	640	matang sempuma	159	308	469	443
10	matang (106).jpg	640	640	matang sempuma	222	300	419	429
11	matang (107).jpg	640	640	matang sempuma	277	267	501	385
12	matang (108).jpg	640	640	matang sempuma	260	286	439	395
13	matang (109).jpg	640	640	matang sempuma	273	260	446	348
14	matang (11).jpg	640	640	matang sempuma	293	124	387	368
15	matang (111).jpg	640	640	matang sempuma	254	290	489	368
16	matang (112).jpg	640	640	matang sempuma	252	248	456	339
17	matang (113).jpg	640	640	matang sempuma	217	263	446	403
18	matang (114).jpg	640	640	matang sempuma	255	251	477	373
19	matang (115).jpg	640	640	matang sempuma	261	284	461	383
20	matang (117).jpg	640	640	matang sempuma	285	342	471	461

Gambar 3.10 Data .xml Convert menjadi .csv

4. Convert *TFRecords*

Pada tahap *Convert TFRecords*, hasil labeling yang berupa file csv akan diubah menjadi bentuk *TensorFlow Records (TFRecords)*. Hal tersebut dilakukan untuk meminimalisis data yang terkumpul. Data yang disimpan pada tahap ini akan disimpan kedalam bentuk string biner.

Gambar 3.11 *file train.record*

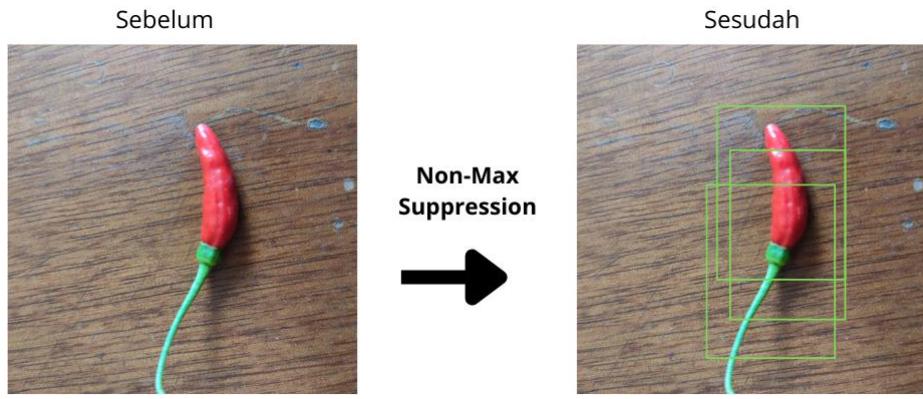
3.2.3. Image Classification

Pada tahap ini, data latih akan melewati proses klasifikasi jenis tingkat kematangan buah cabai rawit menggunakan *SSD-MobileNet*, yang dimana *SSD-MobileNet* dibagi menjadi 2 bagian yaitu *SSD* sebagai base model dan *MobileNet* akan membantu dalam tahap deteksi objek. Algoritma *SSD* juga digunakan untuk mendeteksi objek dengan *Bounding Box* untuk memperkirakan lokasi objek yang akan dideteksi. Model ini memiliki beberapa komponen penting, termasuk seperti *Extract Feature Map*, *Convolutional Filter*, dan *Non-Maximum Suppression*.

a. Extract Feature Map

Pada tahap ini, jaringan *MobileNet* digunakan untuk mengekstrak fitur dari gambar input. *SSD-MobileNet* melibatkan langkah-langkah konvolusi dan pooling pada lapisan-lapisan jaringan *MobileNet* untuk menghasilkan peta fitur dari gambar input. Jaringan ini terdiri dari banyak lapisan (biasanya disebut *convolution layer*) yang bertanggung jawab untuk mengidentifikasi berbagai fitur pada gambar. Misalnya, pada lapisan awal,

mereka dapat mendeteksi fitur sederhana seperti tepi dan sudut, sedangkan pada lapisan yang lebih dalam, mereka dapat mengidentifikasi fitur yang lebih kompleks seperti bentuk dan tekstur. Informasi tersebut dijadikan sebagai dasar dalam pembuatan *bounding box* serta kelas objek yang ada dalam gambar.



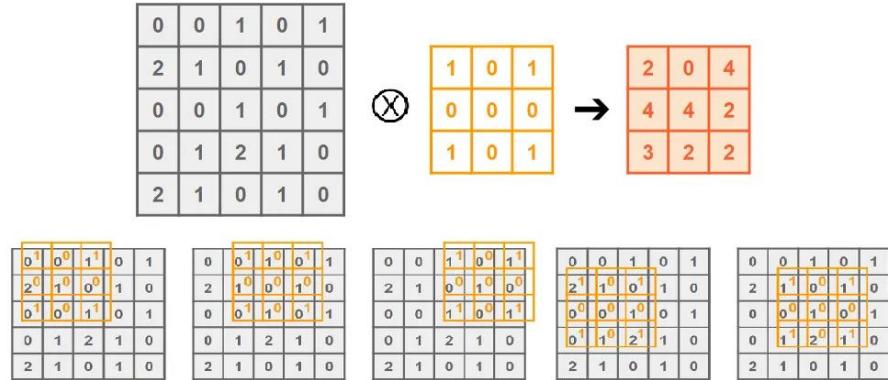
Gambar 3.12 Sebelum dan Sesudah Extract Feature Map

b. Convolutional Filter

Gambar yang sudah di proses kemudian diteruskan melalui convolution layer dalam *Mobilenet*. Setiap lapisan terdiri dari convolution filter yang berbeda untuk mendeteksi fitur fitur yang berbeda pada gambar. Setiap filter ini digunakan untuk mendeteksi pola atau fitur tertentu dalam peta fitur. Setiap filter memiliki ukuranya tertentu (misalnya, 3×3). Proses konvolusi ini melibatkan perkalian antara nilai-nilai filter dan nilai-nilai di area kecil pada peta fitur. Operasi konvolusi dilakukan berulang kali menggunakan berbagai filter pada berbagai lapisan dalam jaringan. Hal ini memungkinkan jaringan untuk membangun representasi yang semakin abstrak dan kompleks dari gambar saat informasi melalui lapisan-lapisan tersebut. Lapisan ini adalah blok fundamental dari *Convolutional Neural Network*. Istilah konvolusi mengimplementasikan kombinasi matematis dari dua fungsi untuk mendapatkan fungsi ketiga. Ia bekerja pada mekanisme jendela geser, yang membantu dalam mengekstraksi fitur dari suatu gambar. Ini membantu dalam pembuatan peta fitur. Konvolusi dua matriks fungsional, yang satu adalah matriks gambar masukan A dan yang lainnya adalah kernel konvolusional B, menghasilkan keluaran C sebagai:

$$C(T) = (A * B)(x) = \int_{-\infty}^{\infty} A(T) \times B(T - x) \, dT$$

Gambar 3.13 Operasi konvolusi antara dua matriks



Gambar 3.14 Ilustrasi Proses Konvolusi (Jiwon Jeong, 2019)

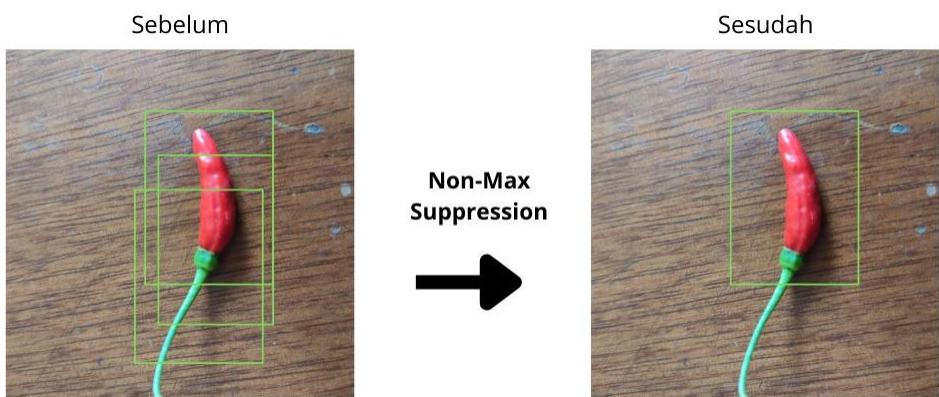
Pada gambar 3.13 menunjukkan ilustrasi pada proses dalam arsitektur jaringan pada penelitian ini. Oleh karena input berukuran 640*640 pixel sehingga terbatasnya kemampuan penulis dalam memvisualisasikan proses kalkulasi, maka peneliti menampilkan ilustrasi dengan *input feature map* mempunyai matriks berukuran 5*5 dengan menggunakan matriks 3*3 sebagai filter. Proses tersebut menggunakan stride 1, yang dapat diartikan proses pergeseran filter sejauh 1 pixel secara horizontal dan vertikal pada *convolution layer*. Proses tersebut biasa disebut *sliding widow*. Pada gambar 3.14 menunjukkan operasi yang digunakan dalam konvolusi dengan 2 matriks. Berikut rincian perhitungan *dot product* yang dilakukan.

- I. $(0 \times 1) + (0 \times 0) + (1 \times 1) + (2 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 2$
- II. $(0 \times 1) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 1) = 0$
- III. $(1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 4$
- IV. $(2 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (2 \times 1) = 4$
- V. $(1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (2 \times 0) + (1 \times 1) = 4$
- VI. $(0 \times 1) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 0) + (1 \times 0) + (2 \times 1) + (1 \times 0) + (0 \times 1) = 2$
- VII. $(0 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 0) + (2 \times 0) + (2 \times 1) + (1 \times 0) + (0 \times 1) = 3$
- VIII. $(0 \times 1) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (2 \times 0) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 2$

$$\text{IX. } (1 \times 1) + (0 \times 0) + (1 \times 1) + (2 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 1) = 2$$

c. Non-Maximum Suppression

SSD menggunakan *Non-Maximum Suppression* untuk menghapus prediksi duplikat yang menunjuk ke objek yang sama. SSD mengurutkan prediksi berdasarkan *confidence score*. Mulai dari prediksi kepercayaan teratas, SSD mengevaluasi apakah kotak batas yang diprediksi sebelumnya memiliki IoU (*Intersection over Union*) lebih tinggi dari yang ditentukan dengan prediksi saat ini untuk kelas yang sama. Jika ditemukan, prediksi paling rendah akan diabaikan.



Gambar 3.15 Sebelum dan Sesudah *Non Maximum Suppression*

3.2.4. Training Model

Sebelum proses training dilakukan, hal yang harus dilakukan yaitu menentukan nilai nilai *num_class*, *batch size*, *num_step*, *checkpoint path*, *test.record path*, dan *train.record path* yang diatur pada konfigurasi *pipeline*. Sebelum proses konfigurasi, diperlukan file *labelmap* yang berisi 3 kategori pada penelitian ini yaitu mentah, setengah matang, matang sempurna. Proses ini memakan waktu kurang lebih 6 jam.

3.2.5. Learned Model

Setelah melewati proses *training* maka akan didapatkan file *savedmodel* dengan ekstensi .pb. file tersebut akan dikonversi kedalam format yang berekstensi *tflite* agar sistem yang dibangun dapat berjalan di perangkat android. Model inilah yang akan digunakan untuk mendeteksi objek yang akan diuji. Sebelum proses convert model ke dalam bentuk *tflite* agar dapat digunakan pada perangkat android, model yang tercipta terlebih dahulu akan

melalui proses validasi dengan menggunakan data validasi. Proses validasi ini merupakan proses pengujian untuk memvalidasi sistem apakah sistem berjalan sesuai dengan yang diinginkan.

3.2.6. *TFLite Model*

Pada tahap ini, menggunakan *TensorFlow Lite*, model yang dihasilkan dari proses training data dengan ekstensi .pb akan dikonversi menjadi format yang memiliki ekstensi tflite agar sistem yang dibangun dapat berjalan di perangkat Android. *Tensorflow Lite* menyediakan dasar untuk model TensorFlow yang telah di-*train* sepenuhnya dan kemudian dapat dikompresi untuk digunakan pada perangkat seluler.

3.2.7. *Testing Process*

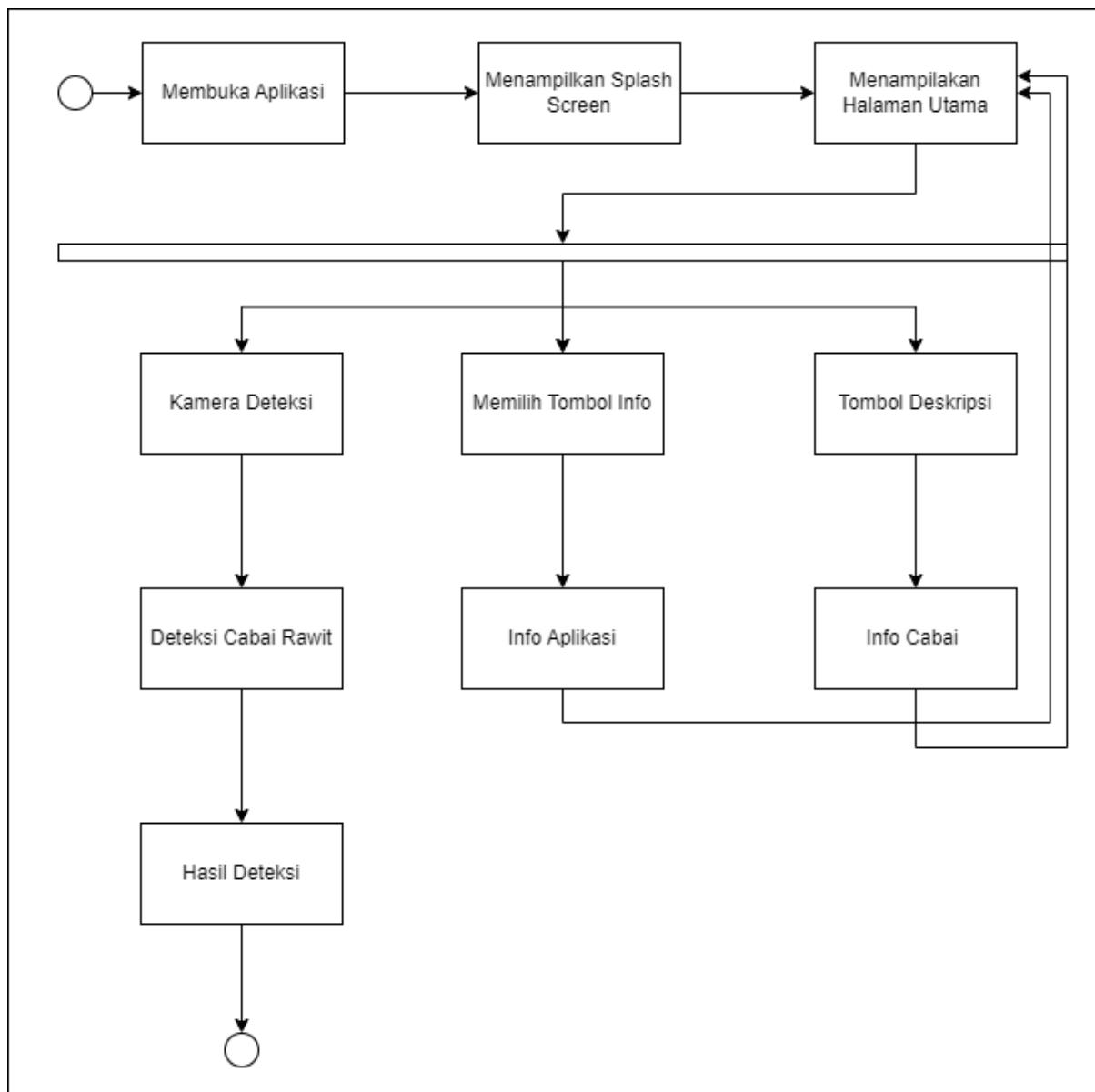
Pada langkah selanjutnya akan dilakukan proses pengujian kepada model yang dihasilkan dari proses training. Kegiatan ini dilakukan dengan tujuan untuk memastikan bahwa kinerja model yang dilatih berhasil mendekripsi objek cabai rawit menjadi tiga tingkatan kematangan yaitu matang, setengah matang dan mentah. Dalam tahap ini juga dapat melihat keakuratan dari penggunaan metode *SSD-MobileNet*.

3.2.8. *Output*

Tahap ini akan menampilkan *bounding box* beserta label dari hasil citra cabai rawit yang telah diklasifikasikan yang dibagi menjadi 3 kelas yaitu mentah, mengkal, dan matang.

3.3. Rancangan Interface

Pada tahap ini membahas tentang perancangan dan mendeskripsikan antarmuka dari sistem. Sistem dibangun dengan menggunakan *Tensorflow Lite* yang dijalankan pada perangkat mobile dalam bentuk aplikasi. Aplikasi ini akan mendekripsi dan mengklasifikasikan citra buah cabai rawit yang ditangkap oleh kamera secara langsung dengan menampilkan *bounding box*. Dengan demikian pengguna dapat melihat hasil deteksi dengan bantuan *bounding box* pada objek yang dikenali oleh sistem saat kamera perangkat aktif.



Gambar 3.16 Activity Diagram

3.3.1. Activity Diagram

Activity Diagram dibuat untuk mengilustrasikan tahapan aktifitas atau tindakan yang berlangsung pada aplikasi. Dengan diagram tersebut alur tiap langkah dapat dijelaskan melalui tahapan-tahapan yang terjadi pada aplikasi. *Activity Diagram* pada sistem dapat dilihat pada gambar Gambar 3.14

3.3.2. Tampilan *Splash Screen*

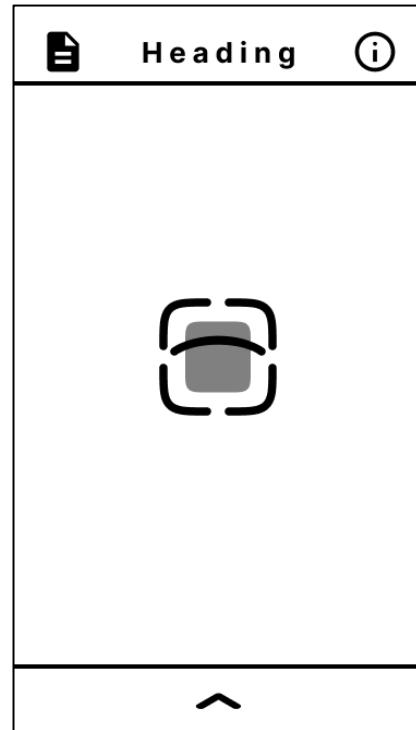
Halaman *Splash Screen* merupakan tampilan pertama yang tertampilkan. Pada tampilan splash screen ini, logo, nama aplikasi, dan nama pembuat aplikasi akan ditampilkan.



Gambar 3.17 Wireframe *Splash Screen*

3.3.3. Tampilan Halaman Deteksi/*Home*

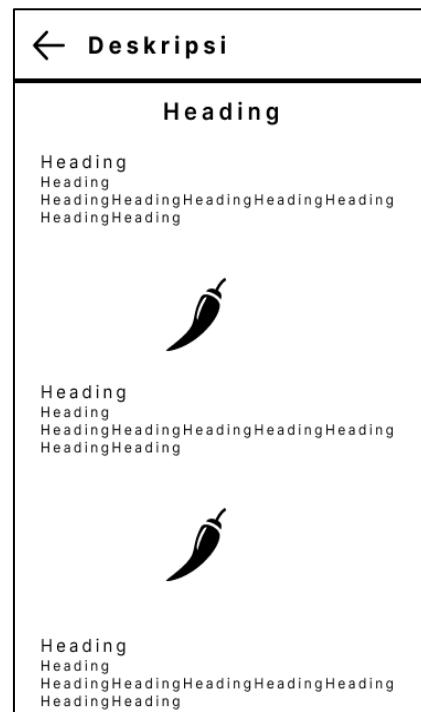
Setelah menampilkan *Splash Screen* aplikasi akan menampilkan halaman beranda dan sekaligus mengaktifkan kamera *smartphone* untuk melakukan deteksi tingkat kematangan cabai rawit.



Gambar 3.18 Wireframe Home Screen

3.3.4. Tampilan Deskripsi

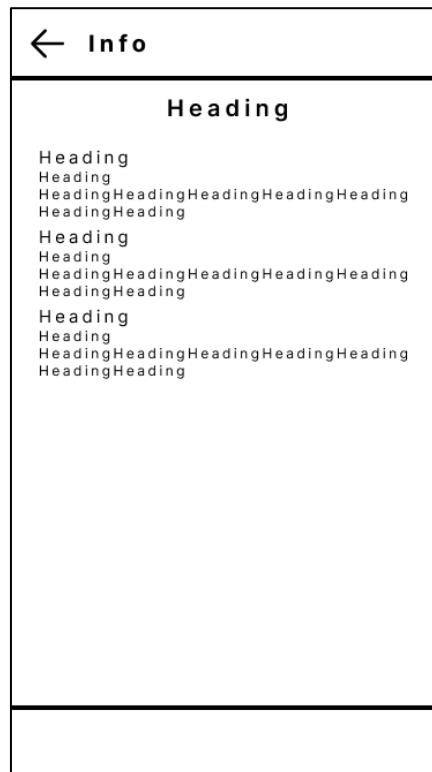
Pada halaman ini akan menampilkan informasi tentang cabai rawit, seperti tingkat kematangannya, cara pembuatan benih cabai rawit, dan lainnya yang menyangkut tentang cabai rawit.



Gambar 3.19 Wireframe Deskripsi

3.3.5. Tampilan Halaman Panduan

Halaman ini berfungsi untuk menjelaskan tentang panduan dalam penggunaan aplikasi seperti tombol tombol pada aplikasi.



Gambar 3.20 Wireframe Panduan Aplikasi

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Langkah implementasi untuk mendeteksi tingkat kematangan cabai rawit menggunakan *SSD-MobileNet* membutuhkan beberapa perangkat pendukung, di antaranya:

4.1. Implementasi Sistem

Pada implementasi *SSD-Mobilenet* dalam melakukan deteksi tingkat kematangan cabai rawit untuk dijadikan benih membutuhkan beberapa spesifikasi perangkat pendukung dalam melakukan penelitian ini. Perangkat tersebut meliputi perangkat keras maupun perangkat lunak, seperti:

4.1.1. Spesifikasi Perangkat Keras

Deskripsi lengkap perangkat keras yang digunakan dalam pengembangan sistem terdapat pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Perangkat Keras

Nama Perangkat Keras	Spesifikasi
MSI	GPU: <i>AMD Ryzen 3 4300U with Radeon 2.70 GHz</i>
	RAM: 8 GB
	<i>OS: Windows 10 Home Single Language</i>
	ROM SSD: 512 GB
POCO X3 NFC	GPU: <i>Octa-core Max2.30GHz</i>
	RAM: 8 + 3 GB
	<i>Android 11</i>
	Kamera Belakang 64 MP

4.1.2. Spesifikasi Perangkat Lunak

Informasi mengenai perangkat lunak yang digunakan dalam pembangunan sistem terdapat pada tabel 4.

Tabel 4. 2 Spesifikasi Perangkat Lunak

Nama Perangkat	Perangkat Lunak
MSI	<i>Python</i>
	<i>Anaconda</i>
	<i>Tensorflow</i>
	<i>LabelImg</i>
	<i>Google Colab</i>
	<i>Android Studio</i>

4.1.3. Implementasi Data

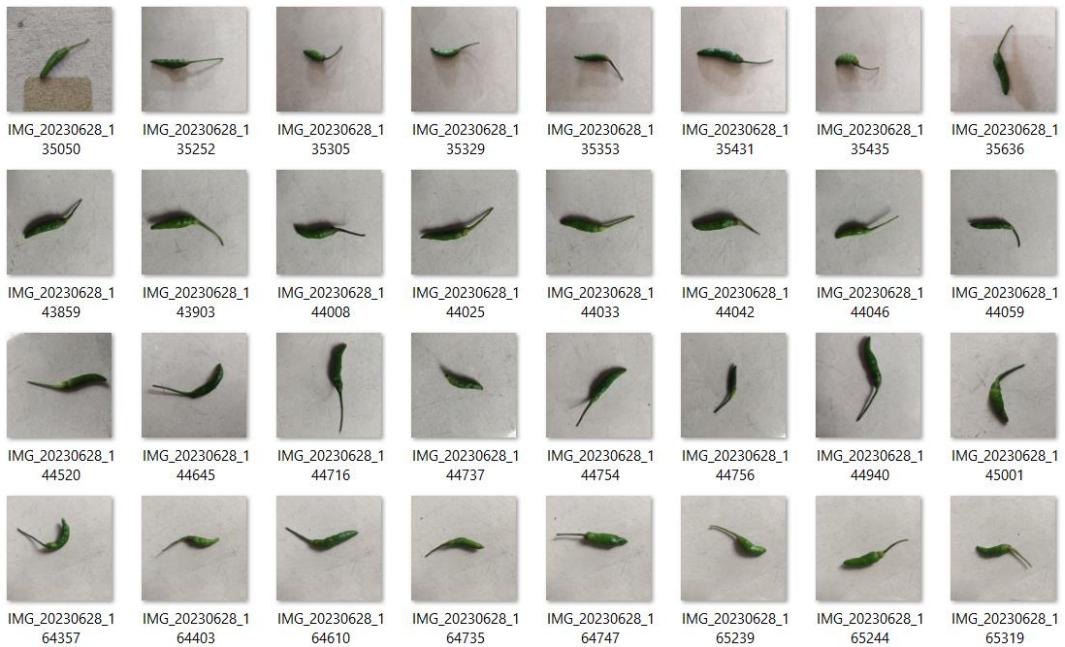
Dalam pembagunan sistem ini, citra buah cabai rawit dikumpulkan dengan menggunakan kamera *smartphone* yang beresolusi 64 MP. Proses pengumpulan citra buah cabai rawit di ambil dari lokasi pembibitan Roberkat di daerah Kecamatan Doloksanggul, Kabupaten Humbang Hasundutan Provinsi Sumatera Utara. Citra cabai rawit dikumpulkan dalam 5 waktu berbeda tiap minggu nya. Dalam proses pengambilan data, citra dibagi menjadi menjadi tiga kategori yaitu mentah, setengah matang, matang sempurna. Contoh citra cabai rawit yang dikumpulkan tertampil pada gambar 4.1, 4.2, 4.3.



Gambar 4.1 Cabai Rawit Matang Sempurna



Gambar 4.2 Cabai Rawit Setengah Matang



Gambar 4.3 Cabai Rawit Mentah

4.2. Pelatihan Sistem

Setelah data melalui proses preprocessing, model dilatih untuk memahami ciri-ciri unik pada setiap objek menggunakan metode *SSD MobileNetV2*. Proses ini dilakukan di platform *Google Colab*.

```
[ ] #download pretrained model ssdmobilenetv2
!wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
!wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8.tar.gz

2023-09-26 10:59:10 - http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 142.250.1.207, 108.177.120.207, 142.250.103.207, ...
Connecting to download.tensorflow.org (download.tensorflow.org)|142.250.1.207|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46042990 (44M) [application/x-tar]
Saving to: 'ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz'

2023-09-26 10:59:11 (70.2 MB/s) - 'ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz' saved [46842990/46042990]

2023-09-26 10:59:11 (70.2 MB/s) - 'ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8.tar.gz'
Resolving download.tensorflow.org (download.tensorflow.org)... 142.250.1.207, 108.177.120.207, 142.250.103.207, ...
Connecting to download.tensorflow.org (download.tensorflow.org)|142.250.1.207|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20518283 (20M) [application/x-tar]
Saving to: 'ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8.tar.gz'

2023-09-26 10:59:11 (45.9 MB/s) - 'ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8.tar.gz' saved [20518283/20518283]
```

Gambar 4.4 Google Colab

Google Colab menyediakan akses ke GPU Nvidia Tesla K80, T4, P4, atau P100 secara gratis, akses RAM yang tersedia 12,7GB. Pada penggunaan *google colab* gratis

mempunyai batasan komputasi terbatas dalam satu sesi yaitu 12 jam nonaktif. Jika ingin menggunakan RAM dan GPU lebih maka bisa diupgrade ke *google colab pro*.

Parameter konfigurasi utama saat training sebagaimana tertampil pada tabel 4.4.

Tabel 4. 3 Parameter Proses *Training*

Parameter	Keterangan
<i>Num Classes</i>	3
<i>Batch Size</i>	8
Total Rate	50.000

Num Classes 3 dapat diartikan bahwa sistem yang dibangun untuk mendekripsi objek yang dibagi menjadi 3 kelas. Pemilihan *batch size* 8 pada pelatihan model yang di terapkan di perangkat *mobile* agar penggunaan kapasitas memori yang lebih kecil. Ukuran *batch* yang besar menyebabkan konvergensi ke fungsi pelatihan yang memperkecil tajam, sedangkan ukuran *batch* yang kecil menyebabkan konvergensi minimalisir datar karena gangguan dalam estimasi gradien. (Evgin & Ali, 2018). *Total Rate* sebanyak 50.000 dipilih berdasarkan perbandingan yang dilakukan peneliti dengan melihat hasil *total loss* yang dihasilkan menunjukkan pada *total rate* sebanyak 50.000 menunjukkan nilai *total loss* yang lebih kecil dari yang lainnya.

Dengan parameter yang ditunjukkan pada tabel 4.3, proses *training* yang dilakukan memakan waktu kurang lebih 6 jam untuk menghasilkan *learned model SSD-Mobilenet v2*. Gambaran proses *training data* sebagaimana terlihat pada gambar 4.5.

```

'learning_rate': 0.0015495324}
INFO:tensorflow:Step 9300 per-step time 0.410s
I0929 12:45:24.751312 133386906058752 model_lib_v2.py:705] Step 9300 per-step time 0.410s
INFO:tensorflow:{'Loss/classification_loss': 0.071529716,
'Loss/localization_loss': 0.021649286,
'Loss/regularization_loss': 0.1300242,
'Loss/total_loss': 0.2232032,
'learning_rate': 0.0011881685}
I0929 12:45:24.751691 133386906058752 model_lib_v2.py:708] {'Loss/classification_loss': 0.071529716,
'Loss/localization_loss': 0.021649286,
'Loss/regularization_loss': 0.1300242,
'Loss/total_loss': 0.2232032,
'learning_rate': 0.0011881685}
INFO:tensorflow:Step 9400 per-step time 0.414s
I0929 12:46:06.135688 133386906058752 model_lib_v2.py:705] Step 9400 per-step time 0.414s
INFO:tensorflow:{'Loss/classification_loss': 0.04609518,
'Loss/localization_loss': 0.016921567,
'Loss/regularization_loss': 0.13001327,
'Loss/total_loss': 0.19303003,
'learning_rate': 0.00087409496}
I0929 12:46:06.135998 133386906058752 model_lib_v2.py:708] {'Loss/classification_loss': 0.04609518,
'Loss/localization_loss': 0.016921567,
'Loss/regularization_loss': 0.13001327,
'Loss/total_loss': 0.19303003,
'learning_rate': 0.00087409496}
INFO:tensorflow:Step 9500 per-step time 0.410s
I0929 12:46:47.154151 133386906058752 model_lib_v2.py:705] Step 9500 per-step time 0.410s
INFO:tensorflow:{'Loss/classification_loss': 0.055654306,
'Loss/localization_loss': 0.016921567,
'Loss/regularization_loss': 0.13001327,
'Loss/total_loss': 0.19303003,
'learning_rate': 0.00087409496}

```

Gambar 4.5 Proses Training

Pada gambar 4.5 menunjukkan proses *training* pada setiap *step* yang dilakukan menghasilkan *loss* dan *per-step-time* yang berbeda. Jumlah *loss* dijadikan tolak-ukur dari *error* atau kesalahan dari suatu model sehingga *loss* berbanding terbalik dengan akurasi hasil. Semakin besar *total_loss* maka akurasi hasilnya akan semakin kecil begitu sebaliknya. Hasil akhir *loss* dan *per-step-time* yang dilakukan pada total 50.000 *step* seperti pada gambar 4.6.

```

INFO:tensorflow:Step 49900 per-step time 0.422s
I1020 07:35:51.943598 132916825288704 model_lib_v2.py:705] Step 49900 per-step time 0.422s
INFO:tensorflow:{'Loss/classification_loss': 0.027497485,
'Loss/localization_loss': 0.0048733205,
'Loss/regularization_loss': 0.073031075,
'Loss/total_loss': 0.10540188,
'learning_rate': 8.2254405e-07}
I1020 07:35:51.943906 132916825288704 model_lib_v2.py:708] {'Loss/classification_loss': 0.027497485,
'Loss/localization_loss': 0.0048733205,
'Loss/regularization_loss': 0.073031075,
'Loss/total_loss': 0.10540188,
'learning_rate': 8.2254405e-07}
INFO:tensorflow:Step 50000 per-step time 0.422s
I1020 07:36:34.123810 132916825288704 model_lib_v2.py:705] Step 50000 per-step time 0.422s
INFO:tensorflow:{'Loss/classification_loss': 0.027724773,
'Loss/localization_loss': 0.0029021804,
'Loss/regularization_loss': 0.073031075,
'Loss/total_loss': 0.10365803,
'learning_rate': 0.0}
I1020 07:36:34.124173 132916825288704 model_lib_v2.py:708] {'Loss/classification_loss': 0.027724773,
'Loss/localization_loss': 0.0029021804,
'Loss/regularization_loss': 0.073031075,
'Loss/total_loss': 0.10365803,
'learning_rate': 0.0}

```

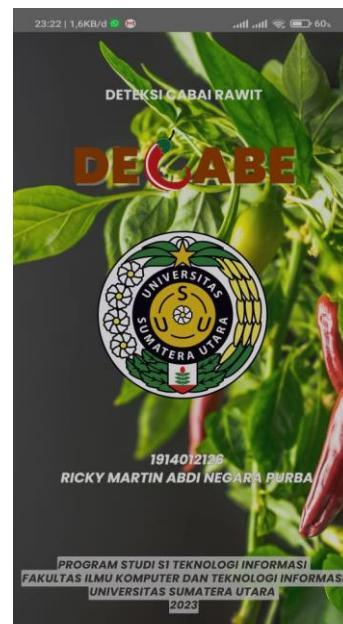
Gambar 4.6 Hasil Akhir Proses Training

Pada gambar 4.6 menunjukkan hasil akhir jumlah loss dan per-*step-time*. Proses *training* yang dilakukan dengan paramater total step 50.000 *total loss* sebesar 0.10365803 dan *classification_loss* 0.027724773 yang menunjukkan bahwa model sudah baik dalam mengklasifikasikan objek ke dalam kelas yg tepat. Dapat dilihat bahwa semakin banyak *total step* yang dilakukan maka jumlah *loss* juga akan semakin kecil. Setelah tahap *training* selesai maka akan menghasilkan *learned model SSD-MobileNet v2*. Untuk menghasilkan sebuah *learned model* dilakukan lagi proses *exported* agar model yang telah dilatih dapat digunakan. *Learned model* yang telah di-*export* akan menjadi *file format saved_model.pb*. *Format* inilah yang digunakan untuk klasifikasi buah cabai rawit. Proses *export model* hasilnya terdiri dari 2 *file* yaitu *saved_model.pb* dan *file checkpoint* yang dapat digunakan jika ingin melanjutkan dan menambah jumlah *step* pada proses *training*.

4.3. Implementasi Perancangan antarmuka

Tampilan antarmuka dari aplikasi ini hanya terdiri dari 5 halaman yaitu tampilan *splash screen*, halaman deteksi/*home*, halaman deskripsi, dan halaman panduan. Ketika pengguna menjalankan aplikasi, maka tampilan halaman pertama yang akan tampil adalah *splash screen*.

- Layar pertama yang muncul saat aplikasi dibuka adalah layar *splash screen*, seperti yang dapat terlihat pada gambar. 4.7.



Gambar 4.7 *Splashscreen*

- b. Beberapa detik setelah halaman splashscreen tampil, akan muncul tampilan *home/deteksi* seperti pada gambar 4.8. Tampilan *home/deteksi* merupakan tampilan untuk mendeteksi objek citra cabai rawit yang dapat ditangkap oleh kamera *smartphone* seperti pada gambar 4.8. Pada saat kamera diarahkan pada objek cabai rawit maka sistem akan mengeluarkan *bounding box* beserta *confident score* pada objek yang dikenali oleh sistem.



Gambar 4.8 Home

- c. Pada halaman *home/deteksi* terdapat beberapa menu seperti menu panduan dan menu deskripsi. Menu panduan berisi informasi tentang tata cara menggunakan aplikasi. Mulai dari penjelasan tentang tombol-tombol dan menu pada aplikasi seperti yang terdapat pada gambar 4.9.



Gambar 4.9 Panduan

- d. Menu deskripsi berisi tentang informasi tentang cabai rawit. Informasi yang dimuat seperti tingkat kematangan cabai rawit yang bagus untuk dijadikan benih, panduan dalam pembuatan benih cabai rawit, dan lainnya. Halaman ini dapat dilihat pada gambar 4.10



Gambar 4.10 Halaman Deskripsi

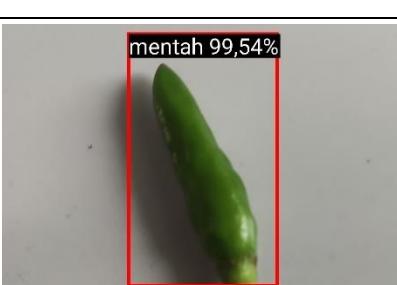
4.4. Prosedur Operasional

Tahap awal yang harus dilakukan *user* ketika ingin menggunakan aplikasi adalah menjalankan aplikasi dengan cara menekan *icon launcher* aplikasi. Kemudian akan muncul splash screen yang berisi nama aplikasi, logo Universitas Sumatera Utara dan nama penulis. Setelah beberapa saat, aplikasi akan meminta izin akses pada kamera *smartphone* agar dapat menjalankan aplikasi. Setelah izin akses diberikan akan muncul tampilan *home/deteksi* berisi juga tombol deskripsi dan panduan aplikasi. Menu panduan merupakan menu yang menampilkan arahan dan informasi terkait tata cara menggunakan aplikasi. Menu deskripsi cabai rawit berisi informasi tentang buah cabai rawit dan informasi tentang pembuatan benih cabai rawit. Pada bagian *home/deteksi* pengguna dapat mengarahkan kamera *smartphone* ke objek cabai rawit yang akan di deteksi. Setelah itu, aplikasi akan memberikan *bounding box* yang berisi informasi tentang tingkat kematangan pada buah cabai rawit dan nilai akurasinya.

4.5. Pengujian Sistem

Tahap pengujian sistem ini berguna agar dapat mengevaluasi kinerja sistem dan performa penggunaan *SSD-MobileNet* untuk mendeteksi tingkat kematangan cabai rawit yang dibagi ke dalam 3 kategori.

Tabel 4. 4 Pengujian Sistem

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
1		Setengah Matang	Setengah Matang	Benar
2		Setengah Matang	Setengah Matang	Benar
3		Mentah	Mentah	Benar
4		Matang Sempurna	Matang Sempurna	Benar
5		Mentah	Mentah	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
6		Setengah Matang	Setengah Matang	Benar
7		Setengah Matang	Setengah Matang	Benar
8		Setengah Matang	Setengah Matang	Benar
9		Matang Sempurna	Matang Sempurna	Benar
10		Setengah Matang	Setengah Matang	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
11		Matang Sempurna	Matang Sempurna	Benar
12		Setengah Matang	Setengah Matang	Benar
13		Setengah Matang	Setengah Matang	Benar
14		Setengah Matang	Setengah Matang	Benar
15		Setengah Matang	Setengah Matang	Benar

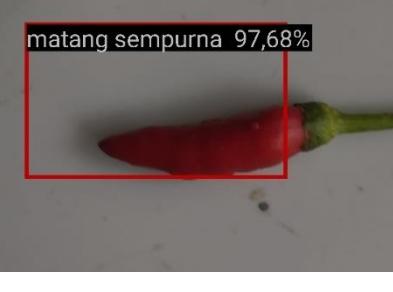
Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
16		Setengah Matang	Setengah Matang	Benar
17		Matang Sempurna	Matang Sempurna	Benar
18		Setengah Matang	Setengah Matang	Benar
19		Setengah Matang	Setengah Matang	Benar
20		Setengah Matang	Setengah Matang	Benar

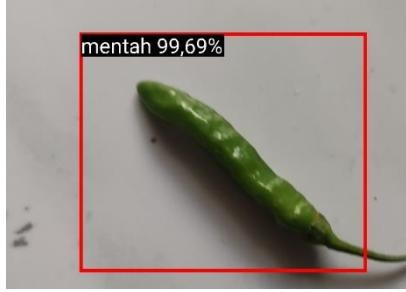
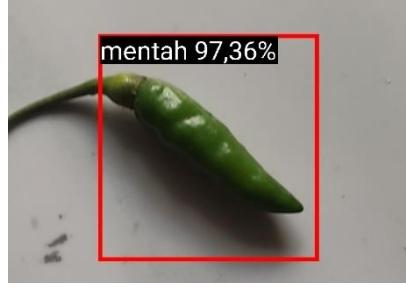
Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
21		Mentah	Mentah	Benar
22		Setengah Matang	Setengah Matang	Benar
23		Setengah Matang	Setengah Matang	Benar
24		Setengah Matang	Setengah Matang	Benar
25		Matang Sempurna	Matang Sempurna	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
26		Setengah Matang	Matang Sempurna	Salah
27		Setengah Matang	Matang Sempurna	Salah
28		Setengah Matang	Matang Sempurna	Salah
29		Matang Sempurna	Matang Sempurna	Benar
30		Setengah Matang	Matang Sempurna	Salah

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
31		Setengah Matang	Setengah Matang	Benar
32		Setengah Matang	Setengah Matang	Benar
33		Mentah	Mentah	Benar
34		Mentah	Mentah	Benar
35		Mentah	Mentah	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
36		Mentah	Mentah	Benar
37		Mentah	Mentah	Benar
38		Mentah	Mentah	Benar
39		Mentah	Mentah	Benar
40		Matang Sempurna	Matang Sempurna	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
41		Matang Sempurna	Matang Sempurna	Benar
42		Matang Sempurna	Matang Sempurna	Benar
43		Mentah	Mentah	Benar
44		Mentah	Mentah	Benar
45		Mentah	Mentah	Benar

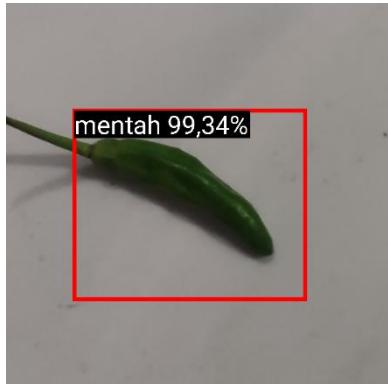
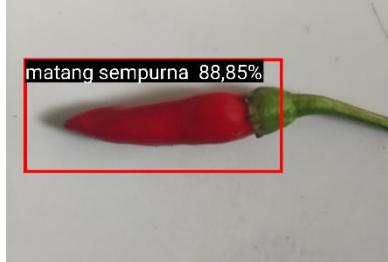
Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
46		Mentah	Mentah	Benar
47		Mentah	Mentah	Benar
48		Matang Sempurna	Matang Sempurna	Benar
49		Matang Sempurna	Matang Sempurna	Benar
50		Setengah Matang	Setengah Matang	Benar

Tabel 4. 4 Pengujian Sistem (Lanjutan)

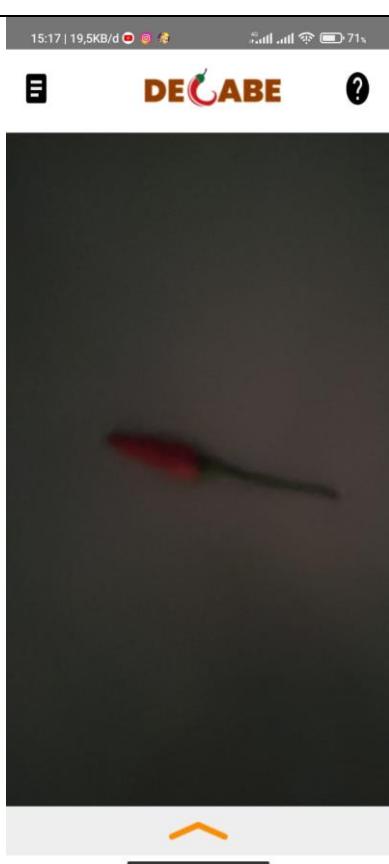
No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
51		Setengah Matang	Setengah Matang	Benar
52		Setengah Matang	Setengah Matang	Benar
53		Setengah Matang	Setengah Matang	Benar
54		Setengah Matang	Setengah Matang	Benar
55		Setengah Matang	Matang Sempurna	Salah

Tabel 4. 4 Pengujian Sistem (Lanjutan)

No	Citra	<i>Actual Output</i>	<i>Desired Output</i>	Status
56		Matang Sempurna	Matang Sempurna	Benar
57		Mentah	Mentah	Benar
58		Matang Sempurna	Matang Sempurna	Benar
59		Matang Sempurna	Setengah Matang	Salah
60		Setengah Matang	Matang Sempurna	Salah

Untuk melihat seberapa jauh mana kemampuan model pada sistem dalam mendeteksi citra buah cabai rawit, maka dilakukan pengujian dengan menjadikan pencahayaan dan jarak sebagai parameter. Jarak yang ditentukan sebagai acuan dalam pengujian ini adalah 8cm, 15cm, dan 20cm, sementara untuk pencahayaan di uji berdasarkan pencahayaan minim, lebih, berimbang

Tabel 4. 5 Pengujian Berdasarkan Tingkat Pencahayaan

No.	Pencahayaan	Gambar	Keterangan
1.	Minim		Sistem tidak mampu mendeteksi buah cabai rawit

Tabel 4. 5 Pengujian Berdasarkan Tingkat Pencahayaan (Lanjutan)

No.	Pencahayaan	Gambar	Keterangan
2.	Lebih		Sistem mampu mendeteksi buah cabai rawit

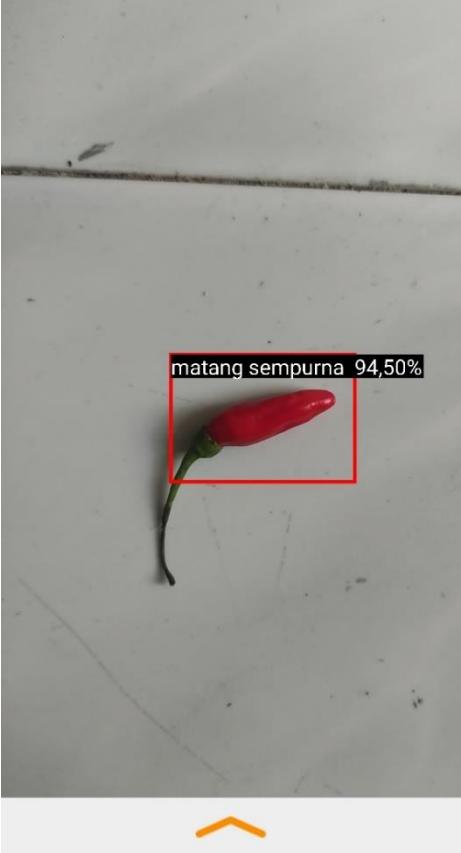
Tabel 4. 5 Pengujian Berdasarkan Tingkat Pencahayaan (Lanjutan)

No.	Pencahayaan	Gambar	Keterangan
3.	Berimbang		Sistem mampu mendeteksi buah cabai rawit

Tabel 4. 6 Pengujian Berdasarkan Jarak

No.	Jarak	Gambar	Keterangan
1.	8 cm		Sistem mampu melakukan deteksi dan klasifikasi dengan benar

Tabel 4. 6 Pengujian Berdasarkan Jarak (Lanjutan)

No.	Jarak	Gambar	Keterangan
2.	15 cm		Sistem mampu melakukan dekripsi dan klasifikasi dengan benar

Tabel 4. 6 Pengujian Berdasarkan Jarak (Lanjutan)

No.	Jarak	Gambar	Keterangan
3.	30 cm		Sistem tidak mampu melakukan deteksi dan klasifikasi.

Berdasarkan proses pengujian seperti pada tabel 4.5 dan 4.6 maka dapat diberikan simpulan bahwa sistem tidak dapat mendeteksi objek yang jarak nya melebihi 30cm dan pada keadaan cahaya yang minim dan lebih. Berdasarkan pengujian pada tabel 4.4, ditemukan 7 kesalahan hasil deteksi tingkat kematangan buah cabai rawit dari 110 data cabai rawit yang diuji. Penyebab terjadinya kesalahan deteksi tersebut dikarenakan perbedaan antara buah cabai rawit yang setengah matang memiliki tingkat kemiripan dengan buah matang sempurna sehingga dibutuhkan jumlah dataset yang lebih banyak pada buah matang sempurna. Jarak pengambilan citra juga mempengaruhi kemampuan sistem dalam mendeteksi objek.

Dikarenakan tingkat kemiripan antara buah yang matang sempurna dengan beberapa buah setengah matang dari segi warna buah tersebut. Hal tersebut membuat

sistem terkadang salah melakukan deteksi terhadap kedua jenis tersebut. Untuk menghitung kemampuan kinerja model pada sistem dapat digunakan *Confusion Matrix*.

Tabel 4. 7 Perhitungan *Confussion Matrix*

Actual		Prediksi			Total
		Mentah	Setengah Matang	Matang Sempurna	
			Matang	Sempurna	
	Mentah	30	-	-	30
	Setengah Matang	-	36	4	40
	Matang Sempurna	-	1	39	40
	Total	30	37	43	110

Dari tabel 4.7 sistem melakukan beberapa kesalahan pada buah setengah matang dan matang sempurna sementara pada uji coba buah mentah sistem tidak melakukan kesalahan. Pada buah setengah matang sistem melakukan 4 kesalahan yang dimana sistem mendeteksinya menjadi buah matang sempurna. Pada buah matang sempurna sistem melakukan 1 kesalahan saat ujicoba sistem yang dimana sistem mendeteksi buah tersebut menjadi setengah matang. Hal tersebut terjadi dikarenakan kemiripan pada buah cabai rawit matang sempurna mirip dengan setengah matang.

Tabel 4. 8 Tabel Nilai TP, FP dan FN

	TP	FP	FN
Mentah	30	0	0
Setengah Matang	36	4	1
Matang Sempurna	39	1	4
Total	105	5	5

Dari data nilai TP, FP, dan FN dapat kita tentukan nilai dari performance dari sistem tersebut. Adapun perhitungan untuk mendapatkan nilai dari performance tersebut seperti akurasi, presisi, *recall*, dan *F-1 score*-nya sebagai berikut:

a. *Precision*

$$\text{Precision Mentah} = \frac{30}{30+0} \times 100\% = 100\%$$

$$\text{Precision Setengah Matang} = \frac{36}{36+1} \times 100\% = 97\%$$

$$\text{Precision Matang Sempurna} = \frac{39}{39+6} \times 100\% = 87\%$$

b. *Recall*

$$\text{Recall Mentah} = \frac{30}{30+0} \times 100\% = 100\%$$

$$\text{Recall Setengah Matang} = \frac{36}{36+6} \times 100\% = 86\%$$

$$\text{Recall Matang Sempurna} = \frac{39}{39+1} \times 100\% = 98\%$$

c. *F1-Score*

$$\text{F1Score Mentah} = 2 \times \frac{100\% \times 100\%}{100\% + 100\%} = 100\%$$

$$\text{F1Score Setengah Matang} = 2 \times \frac{97\% \times 86\%}{97\% + 86\%} = 91\%$$

$$\text{F1Score Matang Sempurna} = 2 \times \frac{87\% \times 98\%}{87\% + 98\%} = 92\%$$

d. *Accuracy*

$$\text{Accuracy} = \frac{103}{110} \times 100\% = 93,6\%$$

Tabel 4.9 Hasil Perhitungan *Precision*, *Recall*, dan *F1-Score*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Mentah	100%	100%	100%
Setengah Matang	97%	86%	91%
Matang Sempurna	87%	98%	92%

Penelitian Deteksi Kematangan Cabai Rawit menggunakan SSD-MobileNet yang fokuskan pada pembuatan benih mampu mengklasifikasikan tingkat kematangan buah cabai rawit dengan tingkat akurasi sebesar 93,6%.

Pengujian sistem untuk mengetahui seberapa baik dan sesuai sistem yang dibuat terhadap pihak petani. Pengujian sistem yang dilakukan meliputi pengujian fitur dari aplikasi. Adapun fitur dari aplikasi “Decabe”:

1. Fitur Deteksi
2. Informasi kematangan buah cabai rawit

3. Panduan penggunaan aplikasi

Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah wawancara, peneliti melakukan wawancara kepada pihak petani dan ahli pembibitan cabai rawit. Proses wawancara dilakukan secara langsung ke pihak pihak yang terkait dan yang akan menjadi pengguna aplikasi. Hasil wawancara terhadap pihak pihak terkait dirangkum untuk dapat dijadikan evaluasi pada sistem ini.

Pihak petani yang terlibat dalam proses pengujian dan penilaian terhadap sistem, yakni:

1. 2 Petani
2. 3 Pihak pembibitan

Para penilai aplikasi diminta untuk mencoba aplikasi “Decabe” dan dimintai untuk memberikan penilaian tentang aplikasi. Penilaian dilakukan pada 2 aspek yakni *understandability* dan aspek *fungsionalitas* tertera sebagai berikut:

1. *Understandability*

Understandability merupakan aspek yang menunjukkan tingkat pemahaman pengguna terhadap output yang dikembalikan oleh aplikasi.

2. *Fungsionalitas*

Fungsionalitas merupakan aspek yang menunjukkan tingkat kegunaan output yang dikembalikan oleh fitur aplikasi kepada pengguna.

Pertanyaan yang diberikan kepada pengguna aplikasi untuk menilai fitur pada aplikasi “Decabe” yang dinilai dari aspek *understandability* dan *functionality* adalah sebagai berikut:

1. Pertanyaan penilaian fitur berdasarkan aspek *understandability*, yakni:
 - a) Apakah instruksi penggunaan aplikasi “Decabe” mudah dimengerti?
 - b) Apakah informasi yang diberikan tentang tingkat kematangan buah cabai rawit sudah termasuk informatif?

- c) Apakah pengguna merasa tertarik dengan tampilan dari aplikasi “Decabe”
2. Pertanyaan penilaian fitur berdasarkan aspek *understandability*, yakni:
 - a) Apakah Instruksi penggunaan aplikasi “Decabe” berguna?
 - b) Apakah fitur pada aplikasi “Decabe” berguna dalam meningkatkan pengetahuan terhadap pemilihan buah cabai rawit untuk benih?
 - c) Apakah aplikasi “Decabe” sudah cukup membantu dalam menentukan tingkat kematangan buah cabai rawit untuk dijadikan benih?
 - d) Apakah aplikasi “Decabe” dapat secara efektif digunakan untuk membantu dalam proses pemilihan buah cabai rawit untuk dijadikan benih?

Berikut rangkuman dari jawaban jawaban pengguna dari pertanyaan terkait aplikasi “Decabe” dari aspek *understandability* dan *functionality*, dari aspek *understandability* dapat disimpulkan bahwa, instruksi dapat dengan mudah dimengerti, dan informasi terkait tingkat kematangan buah cabai rawit sudah cukup detail dan pengguna merasa tertarik dalam menggunakan aplikasi ‘Decabe’ dikarenakan aplikasi yang mudah digunakan dan informatif. Jawaban dari pertanyaan dari aspek *functionality* dapat disimpulkan bahwasanya, fitur dari aplikasi “Decabe” sudah cukup mampu dalam menambah pengetahuan pengguna terkait tingkat kematanganan buah cabai rawit dan untuk kemampuan sistem dalam mendeteksi kematangan buah cabai rawit sangat membantu dikarenakan sistem dapat memangkas cukup banyak waktu dalam pemilihan buah cabai rawit untuk dijadikan benih.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini untuk mendeteksi tingkat kematangan cabai rawit menggunakan metode *SSD-MobileNet* menyimpulkan bahwa:

1. SSD-MobileNet mampu mendeteksi dan mengklasifikasikan 3 tingkat kematangan bauh cabai rawit yang terdiri dari mentah, setengah matang, dan matang sempurna.
2. Sistem mampu mendapatkan akurasi sebesar 93,6%
3. Faktor yang mempengaruhi kegagalan dalam mendeteksi objek dikarenakan adanya kemiripan antara buah cabai rawit matang sempurna dengan setengah matang.
4. Kemampuan model mendeteksi lebih baik saat jarak antar kamera smartphone 8-15 cm dan pada intensitas cahaya yang seimbang dengan objek dengan menggunakan kamera smartphone POCO X3 NFC.
5. Model yang di bentuk dari metode *SSD-MobileNet* terbukti mampu dijalankan di perangkat *mobile*.

5.2. Saran

Saran berikut untuk penelitian selanjutnya agar menjadi pedoman dan referensi pada penelitian selanjutnya:

1. Penambahan jumlah data latih untuk hasil yang maksimal dalam deteksi objek.
2. Penggunaan metode lain guna untuk mencari metode terbaik dalam mendeteksi buah cabai rawit untuk benih.
3. Sistem dapat dioptimalkan sehingga sistem mampu mendeteksi pada jarak diatas 15 cm.
4. Sistem dapat dioptimalkan dengan menggunakan sensor cahaya untuk menyelesaikan masalah pada sistem yang kurang mampu mendeteksi pada keadaan minim cahaya.

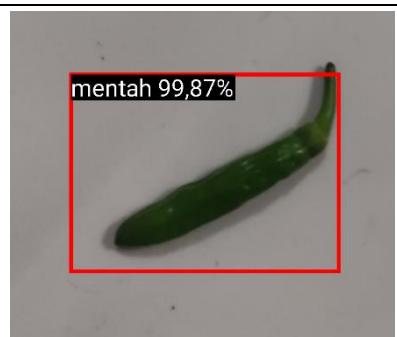
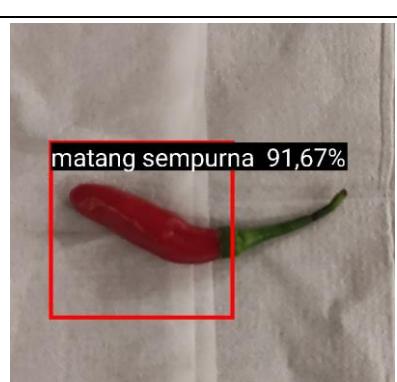
DAFTAR PUSTAKA

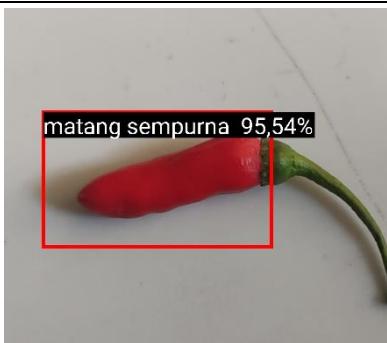
- Adyakbar. (2019). Budidaya Tanaman Cabai. Diakses pada 27 Februari 2023, dari <http://cybex.pertanian.go.id/mobile/artikel/84918/BUDIDAYA-TANAMAN-CABAI/>
- Ashworth, S. (2002). Seed to Seed: Seed Saving and Growing Techniques for Vegetable Gardeners. Seed Savers Exchange, Decorah, IA.
- Darmawan, A. C., Respatijarti, & Soetopo, L. (2014). Pengaruh tingkat kematangan benih terhadap pertumbuhan dan produksi cabai rawit (*Capsicum frutescens* L.) varietas Comexio. Jurnal Produksi Tanaman, 2(4), 339-346.
- Fuadi, A., & Suharso, A. (2022). Perbandingan arsitektur MobileNet dan NASNet-Mobile untuk klasifikasi penyakit pada citra daun kentang. Jurnal Ilmiah Penelitian Informatika (JIPI), 7(3), 701-710.
- Goceri, Evgen & Gooya, Ali. (2018). On The Importance of Batch Size for Deep Learning. An Istanbul Meeting for World Mathematicians, Minisymposium on Approximation Theory & Minisymposium on Math Education, Istanbul
- Howard, A. G., Wang, W., Zhu, M., Weyand, T., Chen, B., Kalenichenko, D., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. Diakses pada 1 Maret 2020 dari <https://arxiv.org/pdf/1704.04861>.
- Huong Nguyen Thi, Quang Tran Minh. (2021). "Proposing a Recognition System of Gestures Using MobilenetV2 Combining Single Shot Detector Network for Smart-Home Applications." Journal of Electrical and Computer Engineering, Hindawi.
- Khandelwal, R. (2019). SSD : Single Shot Detector for object detection using MultiBox. Retrieved from Towards Data Science diakses pada 23 Februari 2023, dari <https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca>
- Kouassi, C. K., Koffi-Nevry, R., Guillaume, L. Y., dkk. 2012. Profiles of bioactive compounds of some pepper fruit (*Capsicum* L.) Varieties grown in Côte d'Ivoire. Innovative Romanian Food Biotechnol, 11, 23-31.
- Nagrat, P. (2021). Corrigendum to "SSDMNV2: A real-time DNN-based face mask detection system using single-shot multibox detector and MobileNetV2" [Sustain. Cities Soc. 66 (2021) 102692].

- Narein T, Adith. (2022). Single Shot Detector (SSD) + Architecture of SSD diakses pada 23 Februari 2023, dari <https://iq.opengenus.org/single-shot-detector/>
- Palwankar, T., & Kothari, K. (2022). Real Time Object Detection using SSD and MobileNet. International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 10, Issue III. Department of Computer Engineering, Rajiv Gandhi Institute of Technology.
- Phon-Amnuaisuk, S., Murata, K. T., dkk. (2018). Exploring the Applications of Faster R-CNN and Single-Shot Multi-box Detection in a Smart Nursery Domain. Cornell University.
- Prasetyo, E., Purbaningtyas, R., Adityo, R. D., Suciati, N., & Faticahah, C. (2022). Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes. Information Processing in Agriculture, 9, 485-496. doi: 10.1016/j.inpa.2022.06.003.
- Putra, Darma. (2010). Pengolahan Citra Digital. Andi, Yogyakarta.
- Ristarina, Ristarina (2018) PENGARUH PEMBERIAN AMPAS TEH TERHADAP PERTUMBUHAN TANAMAN CABAI RAWIT (*Capsicum frutescens L.*) DAN SUMBANGSIHNYA PADA MATERI BIOLOGI DI SMA. Undergraduate Thesis thesis, UIN RADEN FATAH PALEMBANG.
- Sutoyo. T, Mulyanto etc. (2009). Teori Pengolahan Citra Digital. UDINUS Semarang: Andi Yogyakarta.
- W. L. D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A. C. Berg. 2015. Ssd:Single Shot Multibox Detector.
- Wahyu Rahmani^{1*}and Ari Hernawan²Department of Electrical Engineering, National Central University,Zhongli, Taiwan²Department of Computer Science and Information Engineering, National Central University, Zhongli, TaiwanA. M. R. Purnomo and T. B. Adji, "Face Attribute Detection with MobileNetV2 and NasNet-Mobile," International Journal of Electrical and Computer Engineering (IJECE), 9(3), 1846-1853.
- Zhong, Y., Yang, Y., dkk. (2017). Device-free sensing for personnel detection in a foliage environment. IEEE Geoscience and Remote Sensing Letters, 14(6), 921–925.

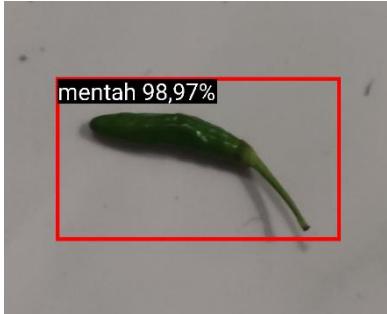
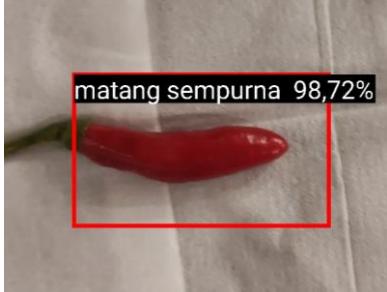
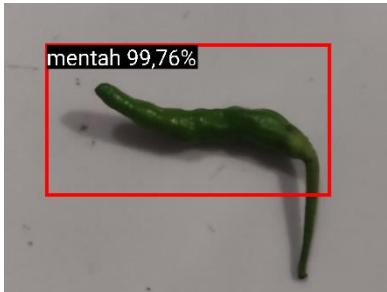
LAMPIRAN

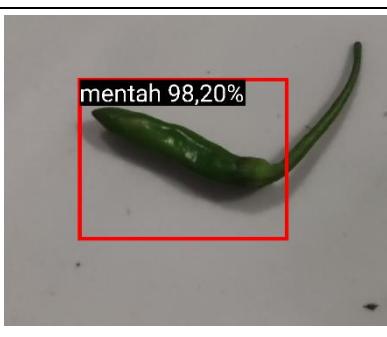
No	Citra	Actual Output	Desired Output	Status
1		Setengah Matang	Setengah Matang	Benar
2		Setengah Matang	Setengah Matang	Benar
3		Matang Sempurna	Matang Sempurna	Benar
4		Matang Sempurna	Matang Sempurna	Benar
5		Setengah Matang	Setengah Matang	Benar

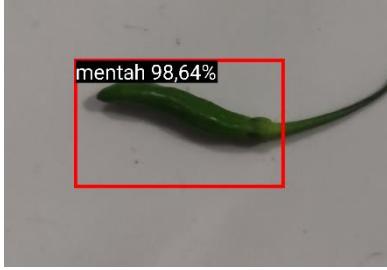
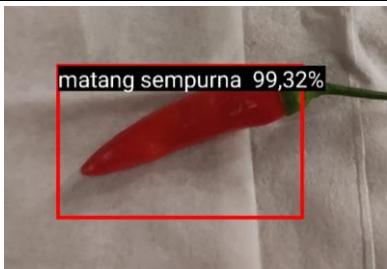
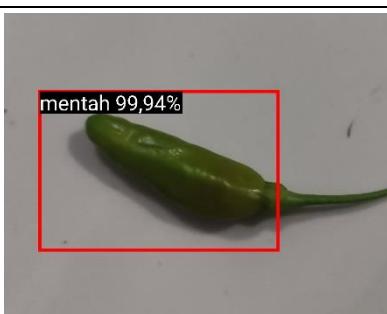
6		Setengah Matang	Setengah Matang	Benar
7		Setengah Matang	Setengah Matang	Benar
8		Mentah	Mentah	Benar
9		Setengah Matang	Setengah Matang	Benar
10		Matang Sempurna	Matang Sempurna	Benar

11		Matang Sempurna	Matang Sempurna	Benar
12		Matang Sempurna	Matang Sempurna	Benar
13		Matang Sempurna	Matang Sempurna	Benar
14		Matang Sempurna	Matang Sempurna	Benar
15		Setengah Matang	Setengah Matang	Benar

16		Mentah	Mentah	Benar
17		Mentah	Mentah	Benar
18		Matang Sempurna	Matang Sempurna	Benar
19		Matang Sempurna	Matang Sempurna	Benar
20		Mentah	Mentah	Benar

21		Mentah	Mentah	Benar
22		Matang Sempurna	Matang Sempurna	Benar
23		Matang Sempurna	Matang Sempurna	Benar
24		Mentah	Mentah	Benar
25		Mentah	Mentah	Benar

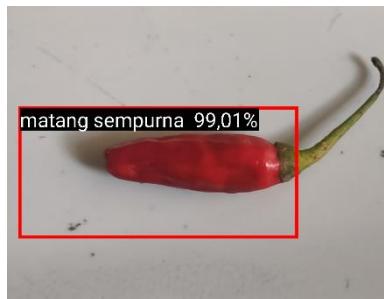
26	 A green chili pepper is shown against a light gray background. A red rectangular box highlights the upper portion of the pepper, containing the text "mentah 68,60%".	Mentah	Mentah	Benar	
27	 A green chili pepper is shown against a light gray background. A red rectangular box highlights the upper portion of the pepper, containing the text "mentah 95,51%".	Mentah	Mentah	Benar	
28	 A green chili pepper is shown against a light gray background. A red rectangular box highlights the upper portion of the pepper, containing the text "mentah 99,84%".	Mentah	Mentah	Benar	
29	 A green chili pepper is shown against a light gray background. A red rectangular box highlights the upper portion of the pepper, containing the text "mentah 98,20%".	Mentah	Mentah	Benar	
30	 A red chili pepper is shown against a light gray background. A red rectangular box highlights the upper portion of the pepper, containing the text "matang sempurna 52,24%".	Matang Sempurna	Matang Sempurna	Benar	

31		Matang Sempurna	Matang Sempurna	Benar
32		Mentah	Mentah	Benar
33		Matang Sempurna	Matang Sempurna	Benar
34		Mentah	Mentah	Benar
35		Mentah	Mentah	Benar

36		Matang Sempurna	Matang Sempurna	Benar
37		Matang Sempurna	Matang Sempurna	Benar
38		Matang Sempurna	Matang Sempurna	Benar
39		Matang Sempurna	Matang Sempurna	Benar
40		Setengah Matang	Setengah Matang	Benar

41		Setengah Matang	Setengah Matang	Benar
42		Matang Sempurna	Matang Sempurna	Benar
43		Matang Sempurna	Matang Sempurna	Benar
44		Matang Sempurna	Matang Sempurna	Benar
45		Matang Sempurna	Matang Sempurna	Benar

46



Matang Sempurna Matang Sempurna Benar

47



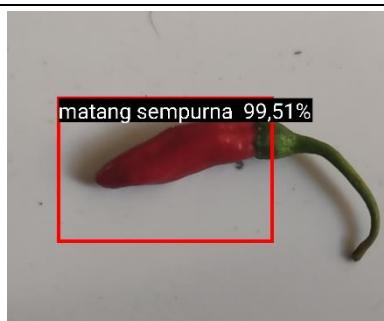
Matang Sempurna Matang Sempurna Benar

48



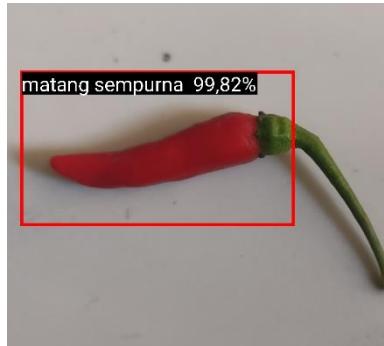
Matang Sempurna Matang Sempurna Benar

49



Matang Sempurna Matang Sempurna Benar

50



Matang Sempurna Matang Sempurna Benar