

**IMPLEMENTASI ARSITEKTUR EFFICIENTNET UNTUK
MENGIDENTIFKASI RASA BUAH JERUK BERASTAGI
BERDASARKAN CITRA JERUK BERASTAGI
BERBASIS ANDROID**

SKRIPSI

KARVIN HALIM

191402118



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI FAKULTAS ILMU
KOMPUTER DAN TEKNOLOGI INFORMASI**

UNIVERSITAS SUMATERA UTARA

MEDAN

2023

**IMPLEMENTASI ARSITEKTUR EFFICIENTNET UNTUK
MENGIDENTIFKASI RASA BUAH JERUK BERASTAGI
BERDASARKAN CITRA JERUK BERASTAGI
BERBASIS ANDROID**

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

KARVIN HALIM

191402118



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI FAKULTAS ILMU
KOMPUTER DAN TEKNOLOGI INFORMASI**

UNIVERSITAS SUMATERA UTARA

MEDAN

2023

PERSETUJUAN

PERSETUJUAN

Judul : IMPLEMENTASI ARSITEKTUR EFFICIENTNET
UNTUK MENGIDENTIFIKASI RASA BUAH JERUK
BERASTAGI BERDASARKAN CITRA JERUK
BERASTAGI BERBASIS ANDROID

Kategori : Skripsi

Nama Mahasiswa : KARVIN HALIM

Nomor Induk Mahasiswa : 191402118

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara

Medan, 10 Januari 2024

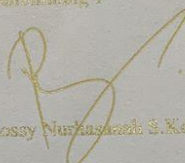
Komisi Pembimbing:

Pembimbing 2



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

Pembimbing 1



Rossy Nurhasanah S.Kom., M.Kom.
NIP. 1987070119619032016

Diketahui/disetujui oleh
Program Studi S-1 Teknologi Informasi

Ketua



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

PERNYATAAN**IMPLEMENTASI ARSITEKTUR EFFICIENTNET UNTUK
MENGIDENTIFKASI RASA BUAH JERUK BERASTAGI BERDASARKAN
CITRA JERUK BERASTAGI BERBASIS ANDROID****SKRIPSI**

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

Medan, November 2023

A handwritten signature in black ink, consisting of a large, stylized 'K' followed by a vertical line and a small horizontal stroke at the bottom.

Karvin Halim

191402118

UCAPAN TERIMAKASIH

Puji dan syukur penulis sampaikan atas kehadiran Tuhan Yang Maha Esa yang senantiasa memberikan berkah, rezeki, dan rahmat. Dengan anugerah-Nya, penulis berhasil menyelesaikan skripsi ini sebagai salah satu persyaratan untuk meraih gelar Sarjana Komputer pada program studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Penulis juga mengucapkan rasa terima kasih yang besar kepada semua pihak yang telah memberikan dukungan, bantuan, dan doa selama proses penulisan skripsi ini. Pada kesempatan ini, penulis ingin menyatakan rasa syukur dan terima kasih kepada:

1. Keluarga penulis, Ayah Lim Bak Chin, Ibu It Li, dan Abang Darwin Halim yang selalu memberikan doa dan dukungan kepada penulis dalam menyelesaikan skripsi ini.
2. Ibu Rossy Nurhasanah, S.Kom., M.Kom. dan Bapak Prof. Drs. Mahyuddin K. M. Nasution, M.IT., Ph.D selaku dosen pembimbing 1 dan 2 penulis yang telah menjadi pembimbing terbaik bagi penulis serta senantiasa memberikan kritik, saran, dan motivasi kepada penulis.
3. Bapak Dedy Arisandi ST., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Ivan Jaya S.Si., M.Kom. selaku Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
5. Seluruh Dosen Program Studi S1 Teknologi Informasi yang telah memberikan ilmu kepada penulis selama masa perkuliahan.
6. Seluruh staff dan pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang membantu dalam segala urusan administrasi selama masa perkuliahan penulis.
7. Teman-teman dekat penulis yaitu Zen, Tazrian, Alwi, Iqbal, Andi, Saddam, Masayu, Lira, Wanda, Devi dan Lais yang bersama-sama berjuang dalam menyelesaikan perkuliahan.

8. Teman penulis dari SMA yaitu Aldo, Gilbert, Carlos, Feri, Nando, Jojo, Mario, Dolin, dan Figo yang menjadi teman ngobrol dan bermain setiap malam selama masa perkuliahan meskipun berkuliah di universitas yang berbeda.
9. Teman-teman HIMATIF yang memberikan kesempatan dan pengalaman berorganisasi kepada penulis selama masa perkuliahan.
10. Teman-teman angkatan 2019 yang sudah melewati masa-masa perkuliahan bersama penulis.
11. Seluruh senior dan junior penulis yang ditemui selama masa perkuliahan.

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini. Maka dari itu, penulis mengharapkan kritik dan saran untuk penyempurnaannya.

Medan, November 2023



Penulis

ABSTRAK

Jeruk (*Citrus spp.*) merupakan salah satu buah budidaya di Indonesia, dengan produksi mencapai 2.551.999,00 ton pada tahun 2022 menurut BPS. Buah jeruk sangat populer dikonsumsi karena kaya manfaat. Selain menyediakan gizi dan energi, jeruk juga dianggap sebagai sumber nutrisi dan suplemen kesehatan yang berharga. Kualitas rasa jeruk memiliki peran penting dalam keberlanjutan industri pemeliharaan buah jeruk, dipengaruhi oleh faktor-faktor seperti varietas jeruk, musim panen, metode pemeliharaan, dan faktor lingkungan. Rasa jeruk sulit diketahui tanpa merusak buahnya. Hal ini dapat menyebabkan limbah ketika membuang buah jeruk yang tidak enak setelah dicicipi. Arsitektur *EfficientNet* merupakan jenis arsitektur *Convolutional Neural Network (CNN)* yang pintar dalam menggabungkan teknik scaling seperti lebar, kedalaman, dan resolusi gambar. Dalam penelitian ini, digunakan *dataset* sebanyak 608 data, terbagi menjadi 424 data untuk pelatihan, 120 data untuk validasi dan 64 data untuk pengujian. Data tersebut mengalami tahap *pre-processing*, melibatkan perubahan ukuran, pemotongan, pembalikan, serta rotasi gambar sebesar 45 dan 90 derajat. Setelah tahap *pre-processing*, data tersebut disampaikan ke algoritma *Convolutional Neural Network (CNN)* dengan arsitektur *EfficientNet-B4*. Pada proses pelatihan model, digunakan 40 *epoch* dan *batch size* sebesar 30. Hasil penggunaan algoritma CNN dengan arsitektur *EfficientNet-B4* menunjukkan tingkat akurasi sebesar 96.88%.

Kata kunci: Jeruk, *EfficientNet*, Convolutional Neural Network.

ABSTRACT

Orange (*Citrus spp.*) is one of the cultivated fruits in Indonesia, with a production reaching 2,551,999.00 tons in 2022 according to the Central Statistics Agency (BPS). Oranges are highly popular for consumption due to their rich nutritional content. Besides providing essential nutrients and energy, oranges are considered valuable sources of nutrition and health supplements. The quality taste of oranges plays a crucial role in the sustainability of the orange cultivation industry, influenced by factors such as orange varieties, harvesting seasons, cultivation methods, and environmental factors. The taste of an orange is hard to determine without damaging the fruit. This can lead to waste when discarding oranges that don't taste good after being sampled. EfficientNet architecture is a type of Convolutional Neural Network (CNN) architecture that intelligently combines scaling techniques such as width, depth, and image resolution. In this study, a dataset consisting of 608 data points was utilized, divided into 424 data points for training, 120 for validation, and 64 for testing. The data underwent pre-processing stages, including resizing, cropping, flipping, as well as rotating images by 45 and 90 degrees. After pre-processing, the data was fed into the Convolutional Neural Network (CNN) algorithm with EfficientNet-B4 architecture. The training process of the model involved 40 epochs with a batch size of 30. The results of using the CNN algorithm with EfficientNet-B4 architecture showed an accuracy rate of 96.88%.

Keyword: Orange, *EfficientNet*, Convolutional Neural Network.

DAFTAR ISI

PERSETUJUAN	i
PERNYATAAN.....	ii
UCAPAN TERIMAKASIH	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Batasan Penelitian	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	5
BAB 2 LANDASAN TEORI	8
2.1. Jeruk Berastagi	8
2.2. Cara Membedakan Jeruk Yang Manis dan Tidak Manis	9
2.3. EfficientNet	9
2.4. CNN (Convolutional Neural Network)	10
2.5. <i>Deep Learning</i>	11
2.6. Confusion Matrix	11
1) Accuracy	11
2) Precission	12
3) Recall	12
4) F1-Score	12
2.7. Penelitian Terdahulu	13

BAB 3 ANALISIS DAN PERANCANGAN	18
3.1. Data yang digunakan	18
3.2. Arsitektur Umum.....	19
3.2.1. <i>Data acquisition</i>	20
3.2.2. <i>Splitting Data</i>	20
3.2.3. <i>Image preprocessing</i>	21
3.2.3.1 <i>Labelling</i>	21
3.2.3.2 <i>Cropping</i>	21
3.2.3.3 <i>Resizing Image</i>	22
3.2.3.4 <i>Augmentasi</i>	23
3.2.4. <i>Image Classifier</i>	24
3.2.4.1. <i>Convolutional Layer</i>	24
3.2.4.2. <i>Feature Map</i>	27
3.2.4.3. <i>Global Average Polling 2D (GAP)</i>	28
3.2.4.4. <i>Fully Connected Layer</i>	29
3.2.5. <i>Learned Model</i>	30
3.2.6. <i>Training Data</i>	31
3.2.7. <i>Testing Data</i>	32
3.3. Perancangan Antarmuka Sistem.....	32
3.3.1. Rancangan Tampilan <i>Splashscreen</i>	33
3.3.2. Rancangan Tampilan <i>Home Page</i>	33
3.3.3. Rancangan Tampilan Untuk Mendeteksi	34
3.3.4. Rancangan Tampilan Hasil Identifikasi	34
3.3.5. Rancangan Tampilan Halaman <i>Tutorial</i>	35
3.3.6. Rancangan Tampilan Halaman Tentang Aplikasi	35
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM	36
4.1. Implementasi Sistem	36
4.1.1. Perangkat Keras dan Perangkat Lunak	36
4.1.2. Implementasi Data	36
4.1.3. Implementasi Desain Antarmuka Aplikasi	37
4.2. Prosedur Operasional	41

4.3.	Pelatihan Sistem	42
4.4.	Pengujian Sistem	46
4.5.	Kendala Penelitian.....	52
4.6.	Perbedaan Penelitian	53
4.7.	Kontribusi Penelitian.....	55
BAB 5 KESIMPULAN DAN SARAN.....		57
5.1.	Kesimpulan.....	57
5.2.	Saran.....	57
DAFTAR PUSTAKA.....		59
Lampiran 1. Tabel Hasil Pengujian Sistem		63

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	14
Tabel 3.1 Pembagian Data Training, Data Validation dan Data Testing.....	18
Tabel 4.1 Pembagian Data Training, Data Validation dan Data Testing Setelah Augmentasi.....	42
Tabel 4.2 Tabel Pelatihan.....	43
Tabel 4.3 Tabel Epoch.....	44
Tabel 4.4 Tabel Hasil Pengujian Sistem.....	46
Tabel 4.5 Confusion Matrix.....	50
Tabel 4.6 Nilai TP, FP, dan FN.....	51
Tabel 4.7 Tabel Nilai Precission, Recall, dan F1-Score.....	52
Tabel 4.8 Tabel Perbedaan Penelitian.....	54

DAFTAR GAMBAR

Gambar 2.1 (a) Jeruk Berastagi, (b) Jeruk Sunkist, dan (c) Jeruk Mandarin...	8
Gambar 2.2 Arsitektur Umum EfficientNet (Tan & Le, 2019).....	10
Gambar 2.3 Arsitektur CNN (O'Shea & Nash, 2015).....	11
Gambar 3.1 Jeruk Berastagi manis dan tidak manis.....	18
Gambar 3.2 Arsitektur umum.....	20
Gambar 3.3 Labelling Data.....	21
Gambar 3.4 Citra sebelum dan sesudah cropping.....	22
Gambar 3.5 Nilai matrix setelah proses resize.....	23
Gambar 3.6 Citra Jeruk sebelum dan sesudah resizing.....	23
Gambar 3.7 Sebelum dan sesudah flip horizontal.....	24
Gambar 3.8 Rotasi 45 derajat dan -45 derajat.....	24
Gambar 3.9 Rotasi 90 derajat dan -90 derajat.....	24
Gambar 3.10 Contoh perhitungan filter convolutional layer.....	26
Gambar 3.11 Posisi filter pada Proses Konvolusi.....	26
Gambar 3.12 Feature Map.....	28
Gambar 3.13 Rancangan Tampilan Splashscreen.....	33
Gambar 3.14 Rancangan Tampilan Home Page.....	33
Gambar 3.15 Rancangan Tampilan Hasil Training.....	34
Gambar 3.16 Rancangan Tampilan Hasil Identifikasi.....	34
Gambar 3.17 Rancangan Tampilan Tutorial.....	35
Gambar 3.18 Rancangan Tampilan Tentang Aplikasi.....	35
Gambar 4.1 Contoh Jeruk Manis.....	37
Gambar 4.2 Contoh Jeruk Tidak Manis.....	37
Gambar 4.3 Tampilan Splashscreen.....	38
Gambar 4.4 Tampilan Halaman Homescreen.....	38
Gambar 4.5 Tampilan Halaman Identifikasi.....	39
Gambar 4.6 Tampilan Halaman Hasil Identifikasi.....	40
Gambar 4.7 Tampilan Halaman Tutorial.....	40
Gambar 4.8 Tampilan Halaman Tentang Aplikasi.....	41

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Jeruk (*Citrus spp.*) merupakan salah satu buah yang kerap kali dijadikan konsumsi karena memiliki banyak manfaat. Jeruk dapat menjadi sumber gizi, sumber energi, nutrisi, dan suplemen kesehatan. Jeruk adalah buah budidaya di Indonesia yang produksinya mencapai 2.551.999 ton pada tahun 2022 (BPS, 2022). Keberlangsungan industri pemeliharaan buah jeruk sangat dipengaruhi oleh kualitas buah yang dipasarkan. Kualitas buah jeruk dipengaruhi oleh banyak faktor, seperti varietas, musim panen, metode pemeliharaan, dan faktor lingkungan lainnya.

Untuk menjamin kualitas buah yang dipasarkan dan memenuhi harapan konsumen, penting untuk memastikan kualitas buah jeruk sebelum dipasarkan. Identifikasi rasa buah jeruk secara visual dapat dilakukan oleh manusia, tetapi memiliki beberapa keterbatasan, antara lain subjektivitas penilaian, kesulitan menilai rasa buah jeruk yang manis dan tidak manis, dan kelelahan mata. Untuk mengatasi keterbatasan tersebut, perlu dikembangkan suatu metode identifikasi rasa buah jeruk secara otomatis. Salah satu metode yang dapat diterapkan adalah menggunakan teknologi citra digital. Penggunaan teknologi citra digital untuk mendeteksi kualitas buah mempunyai beberapa keuntungan. Pertama, metode ini tidak mempengaruhi integritas buah, sehingga buah masih dapat dijual dan dikonsumsi. Kedua, metode ini memberikan informasi yang lebih akurat dan dapat dipantau secara real-time. Ketiga, metode ini mempermudah pengambilan keputusan dalam proses pemasarannya.

Dalam teknologi citra digital, terdapat algoritma *CNN (Convolutional Neural Network)* yang memiliki kemampuan yang baik dalam melakukan analisis citra dan membedakan objek. Penelitian yang dilakukan oleh (Tan & Le, 2019) mengajukan arsitektur yang dinamakan *EfficientNet*, yang dirancang dengan jumlah parameter yang lebih sedikit, sehingga arsitektur ini membutuhkan waktu

pelatihan yang lebih sedikit. Ada tiga parameter utama yang mempengaruhi akurasi dari *EfficientNet*, yaitu *depth* (d), *width* (w), dan *resolution* (r). *EfficientNet* menggabungkan *scaling depth*, *scaling width*, dan *scaling resolution* untuk meningkatkan akurasi model dan efisiensi penggunaan sumber daya komputasinya. Teknik ini dapat membantu dalam menemukan *trade-off* yang optimal antara akurasi dan kompleksitas model. Teknik ini juga menjadi salah satu karakteristik pembeda antara *EfficientNet* dengan arsitektur yang lain.

Sebelumnya sudah ada beberapa penelitian yang melakukan pendeteksian kualitas buah dengan mengimplementasikan teknologi komputasi. Pada tahun 2021, (Yanfei Li et al., 2021) melakukan penelitian dengan melakukan identifikasi dan klasifikasi kualitas apel dengan menggunakan model berbasis CNN dan penelitian ini memperoleh akurasi hingga sebesar 95.33%. Penelitian yang dilakukan oleh (Kholiq Prajatama et al., 2019) menguji algoritma *Naive Bayes* untuk mendeteksi kualitas buah Apel Malang Manalagi. Peneliti membagi kualitas buah apel menjadi 3 kelas yaitu segar, kurang segar, dan busuk. Penelitian ini mendapatkan akurasi sebesar 63%. (Khairullah & Putra, 2021) melakukan penelitian untuk menguji Operasi Morfologi (*Opening dan Closing*) dan Metode *Backpropagation* untuk mengidentifikasi kematangan cabai. Dengan menggunakan *confusion matrix*, peneliti mendapatkan hasil akurasi sebesar 70%.

Penelitian yang dilakukan oleh (Atila et al., 2021). Menggunakan model *EfficientNet* mengidentifikasi penyakit daun tanaman Peneliti membandingkan metode yang digunakan Peneliti menggunakan data uji original dan data uji augmentasi saat pengujian dan mendapatkan hasil akurasi sebesar 98,31% dan 99,38%. (Rifqi & Suharjito, 2021) melakukan deteksi kematangan tandan buah segar kelapa sawit berdasarkan analisis warna menggunakan model *EfficientNet-B0* dan *EfficientNet-B1*. Peneliti mendapatkan hasil evaluasi model sebesar 98,93% untuk model *EfficientNet-B0* dan 99,49% untuk model *EfficientNet-B1*.

Berdasarkan penelitian-penelitian diatas yang telah saya jabarkan, dapat disimpulkan bahwa mendeteksi kualitas suatu buah dapat dilakukan dengan menggunakan algoritma deep learning. Dan berdasarkan referensi-referensi yang telah dibaca dan ditelaah, penulis ingin melakukan penelitian dengan judul

“Implementasi Arsitektur *EfficientNet* Untuk Menidentifikasi Rasa Pada Buah Jeruk Berdasarkan Citra Jeruk Berastagi Berbasis Android”.

1.2. Rumusan Masalah

Jeruk adalah salah satu buah yang banyak dikonsumsi di Indonesia. Rasa dari buah jeruk sulit diketahui tanpa mencicipinya secara langsung. Hal ini dapat menyebabkan meningkatnya limbah ketika kita membuang jeruk yang rasanya tidak enak. Oleh karena itu, diperlukanlah sebuah aplikasi yang dapat membantu untuk mengetahui rasa dari sebuah jeruk tanpa merusak jeruk untuk menjaga keutuhan dari jeruk tersebut.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah mengidentifikasi rasa pada buah Jeruk Berastagi dengan memanfaatkan arsitektur *EfficientNet-B4*. Pada penelitian ini, arsitektur *EfficientNet-B4* juga akan dievaluasi untuk mencari tahu seberapa efektif arsitektur ini digunakan untuk mengidentifikasi rasa buah Jeruk Berastagi.

1.4. Batasan Penelitian

Adapun batasan penelitian ini adalah :

- 1) Data yang dikumpulkan berupa citra buah Jeruk Berastagi.
- 2) Pengambilan citra buah Jeruk Berastagi dilakukan menggunakan kamera digital dengan resolusi minimum 12 MP.
- 3) Data yang digunakan dalam penelitian ini berbentuk file .jpg dan .jpeg.
- 4) Hasil klasifikasi adalah “manis” dan “tidak manis”
- 5) Rasa “tidak manis” merupakan rasa yang diluar rasa manis (asam, pahit, asin, hambar, dan sebagainya).

1.5. Manfaat Penelitian

Berikut adalah manfaat yang dapat diperoleh dari penelitian ini:

- 1) Dapat membantu dalam mengetahui kualitas buah Jeruk Berastagi dengan membagi ke 2 kelas yaitu Manis dan Tidak Manis.

- 2) Penelitian ini dapat dijadikan sebagai referensi untuk penelitian yang akan datang.

1.6. Metodologi Penelitian

Terdapat beberapa tahapan metode yang akan dilakukan dalam penelitian ini, Berikut merupakan tahapan-tahapan yang akan dilakukan dalam penelitian ini :

- 1) Studi Literatur

Pada tahap ini, peneliti akan mengumpulkan informasi mengenai penelitian tentang kualitas buah jeruk, pengolahan citra dan arsitektur EfficientNet. Informasi akan dicari dari berbagai sumber seperti jurnal, skripsi terdahulu, informasi dari pakar dan sumber informasi yang lain.

- 2) Analisis Permasalahan

Pada tahap ini, peneliti mencari informasi mengenai deteksi kualitas buah jeruk berdasarkan informasi yang didapatkan dari tahap studi literatur.

- 3) Perancangan Sistem

Menurut analisis tahap sebelumnya, dibuat perancangan sistem yang berupa perancangan arsitektur umum, penentuan *training*, *validation* dan *testing* data.

- 4) Implementasi

Perancangan sistem yang dibuat sebelumnya lalu diimplementasikan maka dari itu menghasilkan sistem yang sesuai berdasarkan tujuan penelitian.

- 5) Pengujian Sistem

Pada tahapan ini, sistem yang telah dibuat dilakukan pengujian untuk memastikan bahwa sistem yang telah dibuat dapat digunakan dan untuk mendapatkan nilai akurasi dari penerapan arsitektur *EfficientNet* dalam melakukan identifikasi rasa buah jeruk berastagi berdasarkan citra jeruk berastagi.

- 6) Dokumentasi dan Penyusunan Laporan

Pada tahap ini dilakukan penyusunan laporan yang mempresentasikan hasil dari penelitian yang dilakukan.

1.7. Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi 5 bagian, di antaranya :

Bab 1: Pendahuluan

Bab ini mencakup beberapa bagian penting dalam penelitian, yaitu latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, metodologi penelitian, dan struktur atau sistematika penulisan dalam skripsi ini.

Bab 2: Landasan Teori

Bab dua berisi teori yang berhubungan dengan penelitian serta berkaitan dengan masalah dalam penelitian. Beberapa teori diantaranya adalah teori terkait Jeruk Berastagi, *EfficientNet*, *CNN*, dan *Deep Learning*.

Bab 3: Analisis dan Perancangan

Bab tiga berisi analisis data, perancangan arsitektur umum beserta penjelasan dari tahapan – tahapan yang terdapat didalam arsitektur umum penelitian serta melakukan perancangan pada sistem yang akan dibangun.

Bab 4: Implementasi dan Pengujian

Bab ini berisi implementasi dari hasil analisis data dan perancangan model yang telah dijelaskan pada bab sebelumnya.

Bab 5: Kesimpulan dan Saran

Pada bab ini, dijelaskan kesimpulan dari pelaksanaan penelitian dan juga saran dari penulis yang bisa digunakan sebagai referensi untuk penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1. Jeruk Berastagi

Jeruk Berastagi, yang termasuk dalam jenis jeruk WNO (Washington Navel Orange), merupakan varietas jeruk yang banyak ditanam di Berastagi, Sumatera Utara. Di kalangan masyarakat setempat, jeruk ini dikenal sebagai jeruk manis Berastagi karena popularitasnya yang tinggi di daerah tersebut. Buahnya berbentuk bulat dengan permukaan yang agak halus, memiliki ujung buah bundar, dan terdapat pola berpusar. Kulit buahnya berwarna kuning mengilat dan sulit dikupas ketika sudah matang. Daging buahnya bertekstur lunak, kaya akan kandungan air, dan berwarna kekuningan. Rasa daging buahnya sangat manis dan memiliki aroma harum. Umumnya, jeruk manis Berastagi tidak memiliki biji.

Jeruk Berastagi umumnya dikonsumsi dalam keadaan segar. Namun, di Brasil dan Florida, sekitar 90% dari produksi jeruk Berastagi diolah menjadi jus buah; keduanya merupakan produsen terbesar. Selain untuk konsumsi manusia, daging buah dan molase dari jeruk Berastagi juga dimanfaatkan sebagai pakan ternak. Selain itu, kulit buahnya digunakan untuk membuat pektin dan minyak atsiri (Winda Adelindriani, 2022).



(a)



(b)



(c)

Gambar 2.1 (a) Jeruk Berastagi, (b) Jeruk Sunkist, dan (c) Jeruk Mandarin.

2.2. Cara Membedakan Jeruk Yang Manis dan Tidak Manis

Dalam membedakan antara jeruk yang manis dan yang tidak manis, terdapat beberapa hal yang perlu diperhatikan, yaitu:

- **Warna Kulit**
Jeruk yang manis memiliki warna kulit yang kuning kemerahan. Jeruk yang tidak manis memiliki warna kulit yang hijau (Adnan, 2011).
- **Tekstur Kulit**
Jeruk yang manis memiliki tekstur kulit yang halus dan sedikit kenyal. Jeruk yang tidak manis memiliki tekstur kulit yang kasar dan keras (Adnan, 2011).
- **Berat**
Jeruk yang manis memiliki berat yang lebih berat daripada jeruk yang tidak manis. Hal ini disebabkan oleh kandungan air dan gula yang tinggi dalam jeruk (Yolanda et al., 2022).
- **Aroma**
Jeruk yang manis memiliki aroma yang harum. Jeruk yang tidak manis memiliki aroma yang kurang harum (Yolanda et al., 2022).

Namun perlu diingat bahwa faktor lain seperti kondisi tanah, cuaca, varietas, dan perawatan tanaman juga dapat mempengaruhi rasa buah jeruk itu sendiri.

2.3. EfficientNet

EfficientNet adalah sebuah arsitektur jaringan saraf tiruan (deep neural network) yang dikembangkan oleh Mingxing Tan dan Quoc V. Le pada tahun 2019. *EfficientNet* dirancang dengan jumlah parameter yang lebih sedikit, sehingga arsitektur ini membutuhkan waktu pelatihan yang lebih sedikit. Ada tiga parameter utama yang mempengaruhi akurasi dari *EfficientNet*, yaitu *depth* (d), *width* (w), dan *resolution* (r). *Depth* (d) mengacu pada jumlah blok pada arsitektur CNN, *width* (w) mengacu pada lebar setiap blok (jumlah filter pada setiap layer), dan *resolution* (r) mengacu pada ukuran input gambar. Salah satu karakteristik pembeda *EfficientNet* dengan arsitektur lain adalah teknik scaling yang digunakan dalam desain arsitekturnya. Untuk meningkatkan akurasi model dan efisiensi penggunaan sumber daya komputasi, *EfficientNet* menggabungkan *scaling depth*,

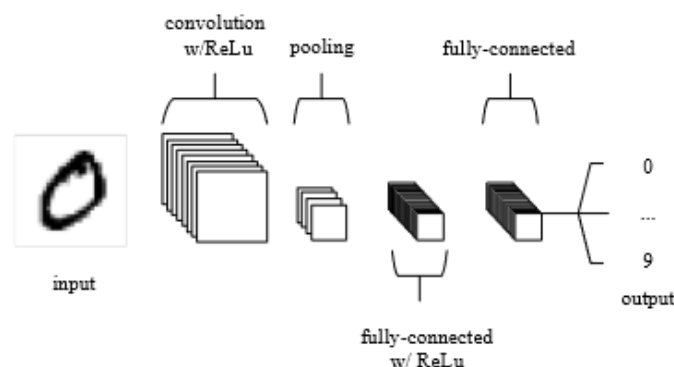
scaling width, dan *scaling resolution* secara efisien. Teknik *scaling* yang digunakan dalam *EfficientNet* dapat membantu menemukan *trade-off* yang optimal antara akurasi dan kompleksitas model.



Gambar 2.2 Arsitektur Umum *EfficientNet* (Tan & Le, 2019)

2.4. CNN (Convolutional Neural Network)

Convolutional Neural Network (CNN) adalah salah satu arsitektur jaringan saraf tiruan yang secara khusus dirancang untuk memproses data citra. CNN terdiri dari tiga jenis lapisan utama, yaitu lapisan konvolusi, lapisan *pooling*, dan lapisan *fully connected*. Lapisan konvolusi berperan dalam mengekstraksi fitur-fitur dari citra, sedangkan lapisan *pooling* berfungsi untuk mereduksi ukuran citra yang telah diekstraksi.. Setelah itu, hasilnya akan diaplikasikan pada lapisan-lapisan *fully connected* untuk menghasilkan output yang diinginkan, seperti klasifikasi atau segmentasi citra. CNN telah menjadi arsitektur yang sangat populer dalam berbagai aplikasi pengolahan citra, termasuk deteksi objek, klasifikasi citra, dan segmentasi citra (O'Shea & Nash, 2015).



Gambar 2.3 Arsitektur CNN (O'Shea & Nash, 2015)

2.5. *Deep Learning*

Deep learning adalah bagian dari *machine learning* yang memanfaatkan arsitektur jaringan saraf tiruan dengan banyak lapisan (*deep neural network*) untuk melaksanakan berbagai tugas pembelajaran mesin. Dalam *deep learning*, untuk melakukan ekstraksi fitur-fitur yang penting data yang masuk akan melewati beberapa layer terlebih dahulu agar dapat mempelajari pola-pola yang rumit dan model yang diciptakan lebih akurat. *Deep learning* memiliki kemampuan untuk mempelajari fitur-fitur yang lebih kompleks dan abstrak, serta kemampuan untuk meningkatkan performa dengan penambahan data latih. Namun, *deep learning* juga memiliki beberapa tantangan, seperti kebutuhan akan sumber daya komputasi yang besar dan data latih yang cukup (Arnold et al., 2011).

2.6. Confusion Matrix

Confusion Matrix adalah sebuah metode evaluasi yang digunakan untuk mengukur hasil klasifikasi suatu permasalahan. Metode ini membandingkan prediksi output yang dihasilkan oleh model dengan nilai yang sebenarnya atau diinginkan, memberikan gambaran yang lebih rinci tentang performa model pada setiap kelas atau kategori. Dengan memanfaatkan *Confusion Matrix*, kita dapat mengukur tingkat akurasi dari model klasifikasi serta memperoleh informasi mengenai jenis kesalahan yang terjadi, seperti *false positive* dan *false negative*. Ada empat nilai yang akan terbentuk dalam *confusion matrix*, diantaranya yaitu:

- *True Positive (TP)*: Data dengan nilai positif yang diprediksi dengan benar sebagai positif.
- *True Negative (TN)*: Data dengan nilai negatif yang diprediksi dengan benar sebagai negatif.
- *False Positive (FP)*: Data dengan nilai negatif yang diprediksi sebagai positif.
- *False Negative (FN)*: Data dengan nilai positif yang diprediksi sebagai negatif.

Berikut merupakan metode perhitungan *confusion matrix* antara lain:

- 1) Accuracy

Nilai akurasi merupakan nilai yang didapat dari jumlah data yang bernilai positif dengan prediksi yang bernilai positif juga dan data yang bernilai negatif dengan prediksi yang bernilai negatif juga. Setelah itu, nilai tersebut dibagi dengan jumlah dari seluruh data pada penelitian. Semakin banyak nilai data dengan diprediksi benar maka semakin tinggi nilai akurasi dari suatu penelitian.

$$\text{Akurasi} = \frac{\text{Jumlah data dengan prediksi benar}}{\text{Jumlah keseluruhan data}} \quad (2.4)$$

2) Precision

Nilai *precision* merupakan rasio antara jumlah prediksi positif yang benar (*True Positive*) dengan total prediksi positif yang dilakukan (*True Positive + False Positive*). Dalam kata lain, presisi mengukur sejauh mana hasil prediksi positif yang diperoleh konsisten dan akurat. Adapun rumus mencari nilai *precision* dapat dilihat pada persamaan 2.5.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.5)$$

3) Recall

Nilai *recall* menggambarkan sejauh mana model dapat mengidentifikasi dan memperoleh informasi yang benar dari keseluruhan data yang sebenarnya bernilai positif. *Recall* merupakan ukuran presentase keberhasilan model dalam mendeteksi dan mengambil data positif yang relevan. Adapun rumus mencari nilai *recall* terdapat pada persamaan 2.5.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.6)$$

4) F1-Score

F1-score adalah suatu metrik yang menggabungkan nilai *recall* dan *precision* menjadi sebuah perbandingan atau rata-rata. *F1-score* menyajikan informasi tentang sejauh mana model dapat mencapai keseimbangan antara mengenali data yang benar positif (*recall*) dan memberikan prediksi yang benar positif

(*precision*). Adapun rumus untuk mendapatkan nilai F1-Score terdapat pada persamaan 2.7.

$$F1-Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.7)$$

2.7. Penelitian Terdahulu

Sebelumnya sudah ada beberapa penelitian yang melakukan pendeteksian kualitas buah dengan mengimplementasikan teknologi komputasi. Pada tahun 2021, Yanfei Li et al. melakukan penelitian dengan melakukan identifikasi dan klasifikasi kualitas apel dengan image processing berdasarkan *Convolutional Neural Network*. Peneliti membagi kualitas apel menjadi 3 jenis yaitu *premium grade*, *middle grade*, dan *poor grade*. Peneliti memperoleh akurasi hingga sebesar 95.33%.

Pada tahun 2019, Kholiq Prajatama et al. melakukan penelitian dengan menguji algoritma *Naive Bayes* untuk mendeteksi kualitas buah Apel Malang Manalagi. Peneliti membagi klasifikasi kualitas buah apel menjadi 3 kelas yaitu segar, kurang segar, dan busuk. Dari 30 buah apel yang diuji terdapat 11 kesalahan sehingga tingkat akurasi yang didapat oleh peneliti hanya sebesar 63%.

Muhammad Rifqi dan Suharjito melakukan melakukan deteksi kematangan tandan buah segar (TBS) kelapa sawit berdasarkan analisis warna. Pada penelitian ini, model deep learning yang mereka gunakan adalah *EfficientNet-B0* dan *EfficientNet-B1*. Peneliti membagi kelas dataset berdasarkan tingkat kematangannya yaitu TBS kurang masak, TBS masak, TBS mentah, TBS terlalu mentah, TBS upnormal, dan janjang kosong. Dari model *EfficientNet* yang dibangun, didapat hasil evaluasi model *EfficientNet-B0* sebesar 98.93% dan model *EfficientNet-B1* 99.49%.

Pada tahun 2021, Ümit Atila et al. melakukan penelitian untuk mengidentifikasi penyakit daun tanaman menggunakan model deep learning *EfficientNet*. Peneliti membandingkan metode yang digunakan dengan metode-metode lain seperti *AlexNet*, *ResNet50*, *VGG16*, dan *Inception V3*. Peneliti

mendapatkan hasil yang menunjukkan bahwa EfficientNet memiliki hasil yang lebih baik dibandingkan metode-metode lain dimana pada saat testing data dengan dataset yang original *EfficientNet-B5* mendapatkan persentase akurasi tertinggi yaitu sebesar 98.31%. Dan pada saat testing dengan dataset yang telah di augmentasi *EfficientNet-B4* mendapatkan persentasi akurasi tertinggi yaitu sebesar 99.38%.

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Judul	Tahun	Keterangan
1.	Yanfei Li et al.	Apple quality identification and classification by image processing based on convolutional neural networks	2021	Peneliti menggunakan CNN untuk melakukan klasifikasi dan identifikasi kualitas buah apel.
2.	Ümit Atila et al.	Plant leaf disease classification using EfficientNet deep learning model	2021	Peneliti melakukan penelitian untuk mengidentifikasi penyakit daun tanaman menggunakan model deep learning EfficientNet. Peneliti juga membandingkan EfficientNet dengan metode lain seperti AlexNet, ResNet50, VGG16, dan Inception V3.

3.	Kholiq Prajatama et al.	Deteksi Kualitas Buah Apel Malang Manalagi Menggunakan Algoritma Naive Bayes	2019	Peneliti menggunakan Algoritma Naïve Bayes untuk membuat sistem deteksi kualitas Buah Apel Malang Manalagi. Peneliti membagi kelas apel menjadi apel segar dan busuk.
4.	Khairullah, Erwin Dwika Putra	Identifikasi Kematangan Cabai Menggunakan Operasi Morfologi (Opening dan Closing) dan Metode Backpropagation	2021	Peneliti menguji Operasi Morfologi (Opening dan Closing) dan Metode Backpropagation untuk mengidentifikasi kematangan cabai. Peneliti juga mengungkapkan bahwa kualitas dan ukuran citra sangat berpengaruh untuk proses testing dan uji data. Kualitas cahaya saat mengambil gambar mempengaruhi warna dari citra cabai karena bayangan yang dipantulkan.
5.	Muhammad Rifqi, Suharjito	Deteksi Kematangan Tandan Buah Segar (TBS) Kelapa Sawit Berdasarkan Komposisi Warna Menggunakan Deep	2021	Peneliti melakukan deteksi kematangan tandan buah segar (TDS) kelapa sawit berdasarkan warna menggunakan deep learning. Model deep

		Learning		learning yang mereka gunakan adalah EfficientNet B0 dan B1
6.	Siricharoen et al.	Recognizing the sweet and sour taste of pineapple fruits using residual networks and green-relative color transformation attached with Mask R-CNN	2023	Peneliti menggunakan ResNet, transformasi ruang warna relatif hijau dan Mask R-CNN untuk melakukan identifikasi rasa buah nanas. Kelas rasa dibagi menjadi manis, asam, dan asam-manis.
7.	Asriny et al.	Orange Fruit Images Classification using Convolutional Neural Networks	2020	Peneliti menggunakan model berbasis CNN untuk melakukan klasifikasi kualitas buah jeruk. Peneliti membagi kelas buah jeruk keprok/siam menjadi 5 kelas yaitu jeruk bagus kelas 1, jeruk bagus kelas 2, jeruk rusak, jeruk busuk, dan jeruk belum matang.
8.	Adelkhani et al., 2013	Taste characterization of orange using image processing combined with ANFIS	2013	Peneliti menggunakan ANFIS untuk menganalisis rasa jeruk dari varietas Bam, Khooni, dan Thompson. Hasilnya menunjukkan tingkat keberhasilan klasifikasi

				rasa untuk masing-masing varietas jeruk Bam, Khooni, dan Thompson adalah 96,6%, 93,3%, dan 93,3%
--	--	--	--	--

BAB 3

ANALISIS DAN PERANCANGAN

3.1. Data yang digunakan

Data penelitian ini berupa gambar jeruk Berastagi yang diambil dari Kebun Jeruk di Berastagi dan jeruk Berastagi yang dibeli dari pasar lokal di Kota Medan. Data diambil dengan menggunakan kamera smartphone dengan resolusi 12 MP secara langsung dan berekstensi .jpg/.jpeg. Total jumlah data citra Jeruk Berastagi yang diambil sebanyak 608 data. Terdapat 2 tingkat mutu data citra Jeruk Berastagi pada penelitian ini yaitu jeruk berastagi yang manis dan jeruk berastagi yang tidak manis yang masing-masing berjumlah 304 data dan 304 data. Sebagai contoh data penelitian, terdapat ilustrasi pada gambar 3.1.



Gambar 3.1 Jeruk Berastagi manis dan tidak manis

Kedua rasa tersebut, yakni rasa manis dan tidak manis, selanjutnya dibagi menjadi tiga bagian, dengan 424 data digunakan sebagai data latih, 120 data sebagai data validasi, dan 64 data sebagai data uji. Tabel 3.1 menyajikan pembagian data yang digunakan dalam penelitian ini.

Tabel 3.1 Pembagian Data Training, Data Validation dan Data Testing

No.	Data	Data Training	Data Validation	Data Testing	Jumlah
1.	Citra Jeruk Berastagi Manis	212	60	32	304
2.	Citra Jeruk Berastagi Tidak Manis	212	60	32	304
Total Data		424	120	64	608

Rasa jeruk divalidasi dengan cara mencicipi langsung setelah jeruk diambil fotonya. Proses validasi ini melibatkan partisipasi dari 5 orang yang mencicipi jeruk tersebut, dan hasilnya diambil melalui mekanisme voting.

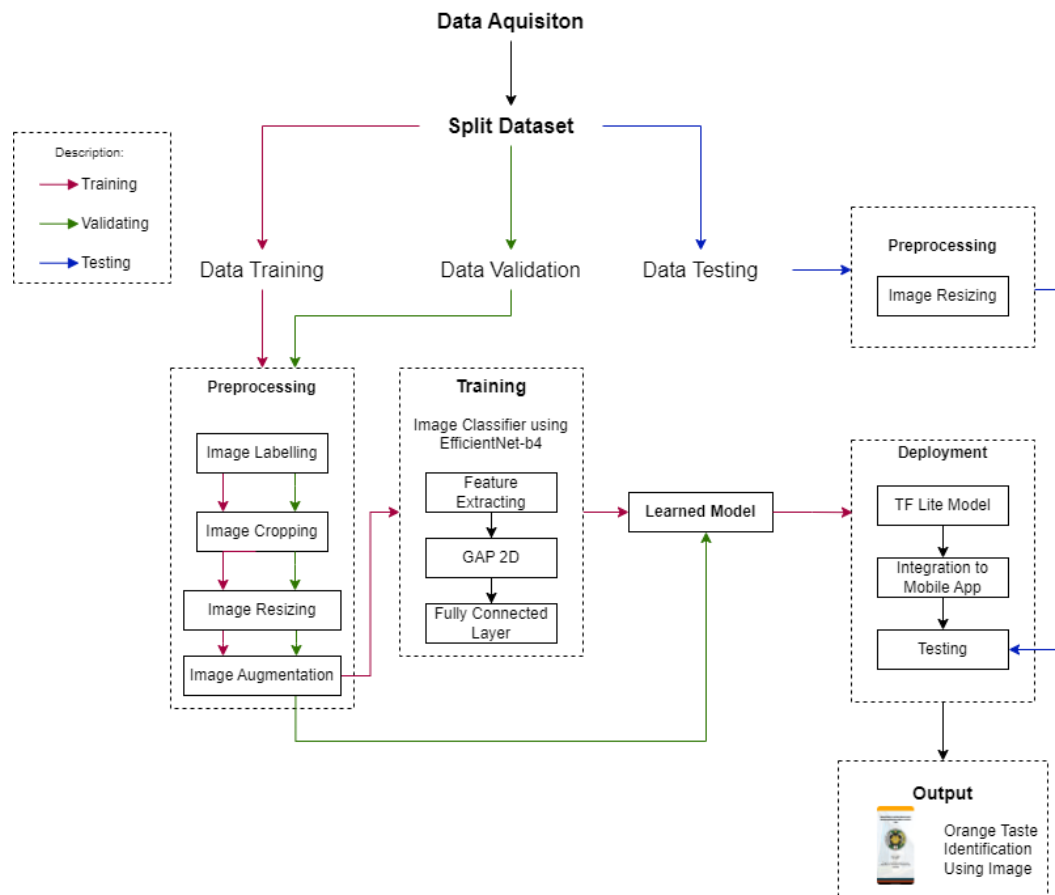
Pendekatan ini bertujuan untuk memastikan bahwa evaluasi rasa jeruk dilakukan secara kolektif dan memberikan pandangan yang lebih komprehensif. Dengan melibatkan sejumlah individu, diharapkan hasil voting dapat memberikan gambaran yang lebih objektif mengenai rasa jeruk yang dinilai.

3.2. Arsitektur Umum

Langkah-langkah yang diterapkan dalam penelitian ini mencakup beberapa tahap. Tahap pertama merupakan proses pengambilan data citra Jeruk Berastagi yang memiliki rasa manis dan tidak manis. Pengambilan data citra tersebut diambil langsung menggunakan kamera smartphone dengan resolusi 12MP.

Tahap berikutnya merupakan tahap *pre-processing* antara lain, *labeling* untuk memberikan penamaan pada citra jeruk, *cropping* untuk membuang bagian yang tidak diperlukan dalam citra dan proses *resizing* untuk merubah ukuran pixel setiap gambar jeruk menjadi ukuran 380x380 pixel. Berikutnya citra melewati proses augmentasi yaitu dengan melakukan *flipping* foto, rotasi secara vertikal dan horizontal serta rotasi gambar sebesar 45 derajat. Citra yang melewati tahap *pre-processing* kemudian dijadikan data latih dan data uji.

Tahap terakhir merupakan tahap pengklasifikasian data citra dimana pada penelitian ini menggunakan fitur ekstraksi yang terdapat pada algoritma CNN dengan arsitektur *EfficientNet*. Adapun output yang dihasilkan setelah tahap klasifikasi berupa 2 tingkatan yaitu Jeruk Manis dan Jeruk Tidak Manis.. Seluruh tahapan ini tergambar dan dapat dilihat pada arsitektur yang ada pada gambar 3.2 berikut ini.



Gambar 3.2 Arsitektur umum

3.2.1. Data acquisition

Dalam langkah ini, terjadi pengumpulan data yang berupa gambar Buah Jeruk Berastagi. Gambar jeruk tersebut dikelompokkan menjadi dua kategori, yaitu jeruk dengan rasa manis dan jeruk dengan rasa tidak manis. Pengambilan gambar dilakukan dengan menggunakan kamera smartphone beresolusi 12MP.

3.2.2. Splitting Data

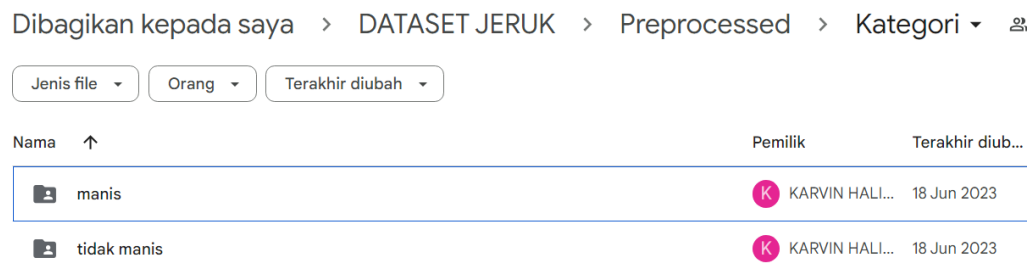
Setelah proses pengumpulan data citra buah jeruk Berastagi, data-data tersebut selanjutnya disusun ke dalam tiga kategori, yaitu data latih, data validasi, dan data uji. Data latih digunakan untuk melatih model mesin, sedangkan data validasi digunakan untuk mengevaluasi kinerja model selama proses pelatihan. Sementara itu, data uji berperan dalam menguji kinerja model yang telah dilatih.

3.2.3. *Image preprocessing*

Proses ini melibatkan pengolahan citra jeruk untuk memperoleh hasil yang optimal sehingga dapat diolah lebih lanjut pada tahap berikutnya. Langkah-langkah preprocessing yang akan diterapkan dalam penelitian ini mencakup *Labeling*, *Cropping*, *Resizing*, dan Augmentasi data.

3.2.3.1 *Labelling*

Proses *labeling* dilakukan untuk menamai atau membuat label pada data yang akan digunakan untuk *training* sistem. Proses pelabelan citra ini dilakukan dengan mengelompokkan gambar ke dalam folder pada *google drive* yang masing-masing telah diberi nama sesuai tingkatan klasifikasinya yaitu manis dan tidak manis. Gambar pada folder-folder tersebut kemudian akan dimuat oleh sistem sesuai labelnya masing-masing. Adapun proses labeling terdapat pada gambar 3.3 dibawah ini.



Gambar 3.3 *Labelling Data*

3.2.3.2 *Cropping*

Cropping merupakan metode yang digunakan untuk memotong atau mengurangi ukuran gambar, sehingga hanya bagian yang relevan atau menarik yang tetap dipertahankan. Proses ini dilakukan dengan menghapus sebagian area yang tidak diinginkan atau tidak perlu dari gambar asli. Tujuan dari teknik pemotongan gambar ini adalah untuk meningkatkan fokus pada objek atau area tertentu dalam gambar, menghilangkan gangguan atau elemen yang tidak relevan, serta

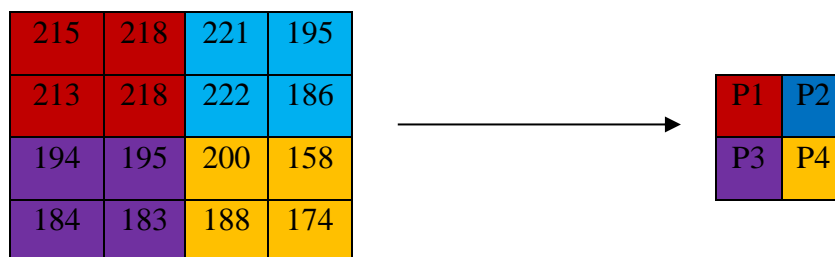
memperbaiki komposisi visual gambar. Contoh citra sebelum dan setelah proses cropping dapat dilihat pada gambar 3.4 di bawah ini.



Gambar 3.4 Citra sebelum dan sesudah *cropping*

3.2.3.3 *Resizing Image*

Proses *resizing* dilakukan dengan tujuan mengubah dimensi gambar sehingga semua gambar yang masuk ke dalam sistem memiliki ukuran yang seragam. Dalam penelitian ini, dimensi citra diubah menjadi 380 x 380 pixel. Berikut adalah contoh dari proses *resizing* pada penelitian ini:



Citra berukuran 4x4 piksel kemudian diubah dimensinya menjadi citra berukuran 2x2 piksel dengan cara mengambil nilai rata-rata dari empat piksel yang berdekatan. Nilai piksel baru tersebut akan menjadi nilai dari gambar yang telah diubah ukurannya. Adapun perhitungannya adalah:

$$P1 = (215 + 218 + 213 + 218) \div 4 = 216$$

$$P2 = (221 + 195 + 222 + 186) \div 4 = 206$$

$$P3 = (194 + 195 + 184 + 183) \div 4 = 189$$

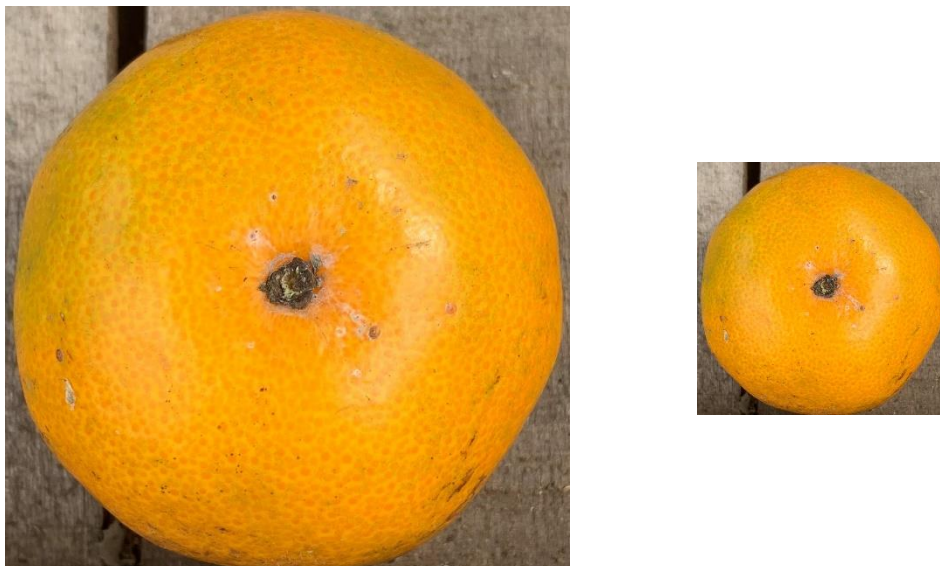
$$P4 = (200 + 158 + 188 + 174) \div 4 = 180$$

Piksel baru yang merupakan hasil *resizing* dengan matriks tersebut dapat ditemukan pada gambar 3.5 di bawah ini.

216	206
189	180

Gambar 3.5 Nilai matrix setelah proses *resize*

Adapun contoh citra sebelum dan sesudah *resizing* terdapat pada gambar dibawah ini.



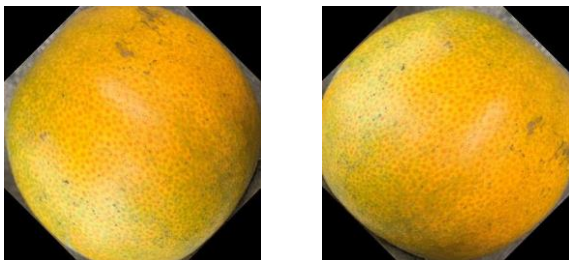
Gambar 3.6 Gambar jeruk sebelum dan setelah proses *resizing*

3.2.3.4 Augmentasi

Augmentasi yaitu memanipulasi atau memodifikasi citra sehingga citra yang asli akan berubah bentuk dan posisi. Augmentasi dilakukan agar mesin dapat mengenali citra objek dalam bentuk dan posisi berbeda. Selain itu, augmentasi juga dapat memperbanyak jumlah data yang akan diproses. Augmentasi pada penelitian ini adalah *flip* dan *rotate*. *Flip* yang dilakukan pada penelitian ini adalah *flip* secara *horizontal* dan *rotate* yang digunakan adalah rotasi sebesar 45 derajat dan 90 derajat. Adapun citra jeruk sebelum dan sesudah augmentasi terdapat pada gambar 3.7, 3.8, dan 3.9 dibawah ini



Gambar 3.7 Sebelum dan sesudah *flip horizontal*



Gambar 3.8 Rotasi 45 derajat dan -45 derajat



Gambar 3.9 Rotasi 90 derajat dan -90 derajat

3.2.4. *Image Classifier*

Dalam penelitian ini, klasifikasi dilakukan dengan menerapkan algoritma *Convolutional Neural Network (CNN)* menggunakan *Google Colab* dan *TensorFlow* sebagai alat bantu. Pembagian klasifikasi terdiri dari tiga langkah, yakni tahap pelatihan, validasi, dan pengujian. Adapun tahapan dari klasifikasi *image EfficientNet* adalah sebagai berikut:

3.2.4.1. *Convolutional Layer*

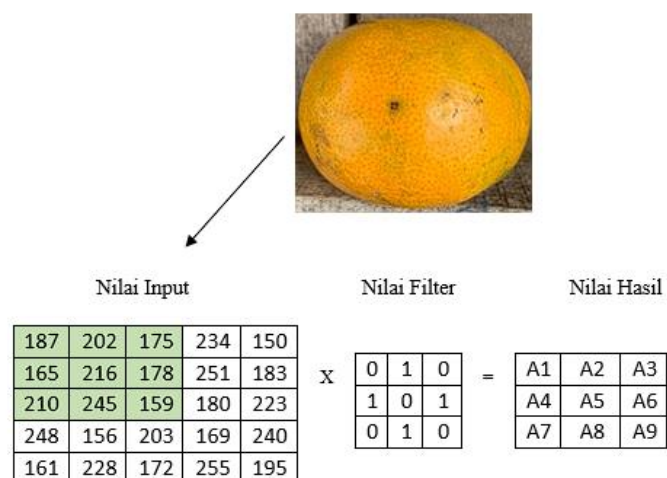
Convolutional layer menggunakan filter untuk mengenali dan mengekstraksi ciri visual dalam objek jeruk yang sedang diidentifikasi. Filter dalam lapisan konvolusi terdiri dari sejumlah neuron yang tersusun dalam matriks dengan dimensi tinggi dan lebar tertentu (dalam piksel). Proses konvolusi melibatkan

ekstraksi informasi penting dari input dengan mengalikan bobot pada setiap piksel dan menjumlahkannya. Hasil konvolusi ini kemudian diteruskan ke lapisan berikutnya untuk pemrosesan lebih lanjut.

Arsitektur *EfficientNet* menghadirkan filter-filter yang dirancang secara khusus untuk mencapai tingkat efisiensi dan kinerja optimal dalam pemrosesan citra. Dengan menggeser filter ke seluruh gambar, tercipta peta aktivasi yang memvisualisasikan di mana fitur-fitur penting berada dalam gambar. Lapisan konvolusi memiliki peran dalam melaksanakan operasi konvolusi pada citra dan menghasilkan feature map yang menyoroti pola-pola penting dalam citra jeruk. Adapun contoh matriks gambar *input* pada penelitian ini adalah:

187	202	175	234	150
165	216	175	251	183
210	245	159	180	223
248	156	203	169	240
161	228	172	255	195

Kemudian matriks dari *input* gambar dihitung dengan menggunakan filter yang terdapat pada *convolutional layer*. *Activation map* dihasilkan dari pengulangan proses perhitungan ini untuk setiap elemen dari gambar *input*. Adapun contoh salah satu perhitungan *filter* pada matriks gambar penelitian ini adalah sebagai berikut:



Gambar 3.10 Contoh perhitungan *filter convolutional layer*

Gambar 3.10 menggambarkan cara perhitungan konvolusi dilakukan pada lapisan konvolusi dengan menggunakan filter berukuran 3×3 dan *stride* sebesar 1. *Stride*, yang menentukan seberapa jauh filter digeser pada matriks input setiap langkahnya, terlihat dalam proses penggeseran filter seperti yang ditunjukkan dalam Gambar 3.11.

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

187	202	175	234	150
165	216	178	214	183
210	224	159	180	223
230	156	203	169	240
161	228	172	206	195

Gambar 3.11 Posisi *filter* pada Proses Konvolusi

Dalam Gambar 3.11, tergambar proses perhitungan (*dot product*), di mana hasil perkalian antara matriks deteksi fitur atau filter dengan matriks input menjadi nilai dari *feature map*. Selanjutnya, berikut adalah perincian perhitungan lapisan konvolusi pada penelitian ini:

$$A1 = (187 \times 0) + (202 \times 1) + (175 \times 0) + (165 \times 1) + (216 \times 0) + (178 \times 1) + (210 \times 0) + (224 \times 1) + (159 \times 0) = 769$$

$$A2 = (202 \times 0) + (175 \times 1) + (234 \times 0) + (216 \times 1) + (178 \times 0) + (214 \times 1) + (224 \times 0) + (159 \times 1) + (180 \times 0) = 764$$

$$A3 = (175 \times 0) + (234 \times 1) + (150 \times 0) + (178 \times 1) + (214 \times 0) + (183 \times 1) + (159 \times 0) + (180 \times 1) + (223 \times 0) = 775$$

$$A4 = (165 \times 0) + (216 \times 1) + (178 \times 0) + (210 \times 1) + (224 \times 0) + (159 \times 1) + (230 \times 0) + (156 \times 1) + (203 \times 0) = 741$$

$$A5 = (216 \times 0) + (178 \times 1) + (214 \times 0) + (224 \times 1) + (159 \times 0) + (180 \times 1) + (156 \times 0) + (203 \times 1) + (169 \times 0) = 785$$

$$A6 = (178 \times 0) + (214 \times 1) + (183 \times 0) + (159 \times 1) + (180 \times 0) + (223 \times 1) + (203 \times 0) + (169 \times 1) + (240 \times 0) = 765$$

$$A7 = (210 \times 0) + (224 \times 1) + (159 \times 0) + (230 \times 1) + (156 \times 0) + (203 \times 1) + (161 \times 0) + (228 \times 1) + (172 \times 0) = 885$$

$$A8 = (224 \times 0) + (159 \times 1) + (180 \times 0) + (156 \times 1) + (203 \times 0) + (169 \times 1) + (228 \times 0) + (172 \times 1) + (206 \times 0) = 656$$

$$A9 = (159 \times 0) + (180 \times 1) + (223 \times 0) + (203 \times 1) + (169 \times 0) + (240 \times 1) + (172 \times 0) + (206 \times 1) + (195 \times 0) = 829$$

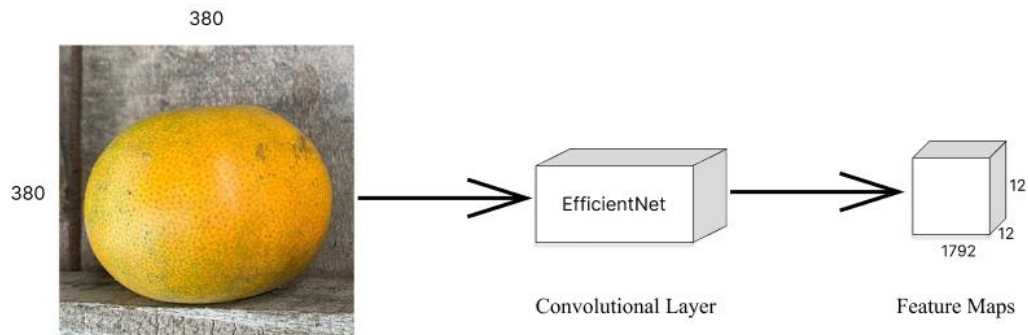
Dengan perhitungan tersebut, maka didapatkan contoh hasil matriks penelitian ini yang sudah diproses pada convolutional layer yaitu sebagai berikut

769	764	775
741	785	765
885	656	829

3.2.4.2. Feature Map

Feature map adalah hasil dari serangkaian konvolusi pada berbagai lapisan konvolusi dalam jaringan. Proses ini dimulai dengan input gambar yang melewati lapisan-lapisan konvolusi menggunakan filter kecil untuk melakukan konvolusi pada gambar. Setiap filter menghasilkan *feature map* yang menyoroti bagian-bagian penting dari gambar yang relevan dengan pola yang diidentifikasi, dan filter tersebut memiliki bobot yang dapat disesuaikan selama pelatihan model. Proses konvolusi ini menghasilkan *feature map* dengan ukuran yang lebih kecil tetapi dengan kedalaman (depth) yang lebih besar. Masing-masing lapisan konvolusi memiliki tanggung jawab untuk mengekstraksi dan meningkatkan fitur-fitur penting seperti tepi, tekstur, dan pola visual dari gambar. Dalam arsitektur *EfficientNet*, pendekatan scaling koefisien (*width*, *depth*, dan *resolution*) serta struktur blok yang terhubung secara khusus membantu menciptakan *feature map* yang efisien dan berkualitas tinggi. Dimensi awal gambar dalam *feature map* adalah $380 \times 380 \times 3$, namun berubah menjadi $12 \times 12 \times 1792$ pada model yang

telah dilatih menggunakan *EfficientNet-B4*. Gambar 3.12 dibawah ini merupakan visualisasi dari fitur map.



Gambar 3.12 *Feature Map*

3.2.4.3. *Global Average Pooling 2D (GAP)*

Metode Global Average Pooling 2D (GAP) dalam model EfficientNet-B4 merupakan teknik yang dimanfaatkan untuk mengurangi dimensi spasial dari feature map pada lapisan sebelumnya. Pada model EfficientNet-B4, feature map yang dihasilkan memiliki dimensi yang cukup besar, misalnya $12 \times 12 \times 1792$. GAP bertujuan untuk mentransformasi feature map tersebut menjadi representasi yang lebih sederhana, yaitu $1 \times 1 \times 1792$.

Proses *Global Average Pooling (GAP)* dilaksanakan dengan menghitung nilai rata-rata dari setiap kanal pada peta fitur (*feature map*). Dengan lebih rinci, GAP melakukan perhitungan rata-rata untuk semua nilai piksel pada setiap kanal secara terpisah. Misalnya, untuk setiap kanal, GAP akan menjumlahkan nilai piksel di dalam kanal tersebut, lalu membaginya dengan jumlah total piksel. Hasil akhir adalah nilai rata-rata dari masing-masing kanal. "Kanal" dalam hal ini merujuk pada dimensi atau lapisan data dalam sebuah feature map atau gambar. Sebagai contoh, pada citra berwarna dalam format RGB, setiap kanal mewakili informasi intensitas warna yang berbeda. Karenanya, citra RGB memiliki tiga kanal yang berbeda. Dalam proses pengolahan gambar menggunakan Convolutional Neural Network (CNN), tiap lapisan konvolusi dapat menghasilkan berbagai kanal yang menggambarkan fitur-fitur beragam dalam tingkat kompleksitas yang berubah-ubah. Di lapisan awal, kanal mungkin mewujudkan fitur-fitur dasar seperti tepi atau garis sederhana, sedangkan di lapisan lebih

dalam, kanal akan mewujudkan fitur-fitur yang lebih kompleks seperti tekstur atau bentuk objek. Dengan menerapkan metode GAP, informasi penting dalam feature map tetap terjaga, walaupun dalam bentuk representasi yang lebih sederhana. Hasilnya adalah feature map dengan dimensi spasial 1×1 , yang dapat dianggap sebagai gambaran keseluruhan yang global dari fitur-fitur lokal dalam feature map sebelumnya.

Metode GAP membawa beberapa kelebihan. Pertama, GAP secara signifikan mengurangi dimensi spasial, yang pada gilirannya mengurangi jumlah parameter yang perlu diatur selama proses pelatihan. Ini membantu menghindari overfitting dan mempercepat komputasi. Kedua, GAP mampu memberikan representasi yang lebih tahan terhadap pergeseran dan deformasi pada objek dalam citra. Dengan mengambil nilai rata-rata, posisi piksel menjadi kurang relevan, sehingga feature map menjadi lebih invarian terhadap perubahan spasial. Dalam konteks EfficientNet-B4, metode GAP diterapkan setelah lapisan konvolusi untuk menghasilkan feature map yang lebih padat sebelum melanjutkan dengan lapisan-lapisan berikutnya, seperti lapisan fully connected yang digunakan untuk tujuan klasifikasi

3.2.4.4. *Fully Connected Layer*

Dalam arsitektur *Convolutional Neural Network (CNN)*, *Fully Connected Layer* merupakan langkah di mana setiap *neuron* pada lapisan sebelumnya dihubungkan dengan setiap *neuron* pada lapisan berikutnya. *Fully Connected Layer* memiliki fungsi utama sebagai klasifikasi, di mana ia menggunakan fitur-fitur yang telah diambil dari gambar jeruk sebelumnya dan menghasilkan keluaran untuk keperluan klasifikasi.

Tahapan dalam fully connected layer dimulai dengan merubah feature map hasil dari langkah sebelumnya menjadi vektor satu dimensi. Setiap elemen dalam vektor ini mewakili fitur-fitur yang telah diekstraksi. Vektor ini kemudian mengalir melalui lapisan yang terdiri dari neuron-neuron fully connected, di mana setiap neuron memiliki bobot dan bias yang unik. Tiap neuron dalam lapisan ini melakukan komputasi aktivasi menggunakan fungsi Rectified Linear Unit (ReLU) pada seluruh fitur dari vektor input. Fungsi ReLU memberikan karakteristik non-

linear pada aktivasi, hanya memungkinkan aktivasi terjadi ketika nilai input positif. Dengan menghapus nilai negatif, fungsi ReLU membantu percepatan komputasi serta pengurangan kompleksitas model. Ini juga membantu dalam mematikan neuron yang tidak aktif, memungkinkan hanya fitur-fitur yang penting dari gambar jeruk yang diaktifkan. Proses ini diulang untuk setiap neuron pada Fully Connected Layer, dan keluaran dari setiap neuron dihitung.

Selanjutnya, keluaran dari neuron-neuron ini diarahkan menuju lapisan output, yang menggunakan fungsi aktivasi yang cocok dengan riset ini, yaitu fungsi sigmoid untuk tugas klasifikasi biner (2 kelas). Fungsi sigmoid digunakan di lapisan output untuk memberikan probabilitas prediksi pada kelas yang sesuai, seperti manis atau tidak manis. Output sigmoid berada dalam rentang 0 hingga 1, dapat diartikan sebagai probabilitas. Dengan menggunakan sigmoid, model dapat memberikan prediksi berbasis probabilitas untuk masing-masing kelas, di mana nilai mendekati 1 menunjukkan probabilitas yang lebih tinggi untuk kelas tertentu. Ini memungkinkan pengambilan keputusan yang lebih akurat berdasarkan probabilitas, serta memudahkan penggunaan ambang batas (threshold) untuk klasifikasi biner. Pada tahap pelatihan, bobot dan bias dalam lapisan FC diperbarui menggunakan algoritma optimasi Adam, yang menggabungkan adaptasi gradien dan momentum untuk mengoptimalkan bobot dan bias dalam jaringan saraf tiruan. Optimasi Adam membantu dalam percepatan konvergensi yang efisien. Secara keseluruhan, tahap fully connected layer memungkinkan model untuk memahami korelasi yang kompleks antara fitur-fitur yang telah diekstraksi dan output klasifikasi yang diinginkan.

3.2.5. *Learned Model*

Proses klasifikasi citra saat melatih data menghasilkan sebuah model yang telah belajar, yang akan digunakan dalam aplikasi. Model yang telah belajar ini, yang juga dikenal sebagai "*saved model*" berisi struktur grafik dan parameter bobot yang dihasilkan selama pelatihan dari data tersebut. Model ini terdiri dari dua klasifikasi citra jeruk berastagi, yaitu Manis dan Tidak Manis. Selanjutnya, model yang telah belajar ini akan diubah formatnya menjadi *Tensorflow Lite* agar dapat digunakan di platform *Android*.

3.2.6. *Training Data*

Proses *training data* melibatkan sistem dalam mempelajari citra-citra yang diinput dan menghasilkan model dengan menggunakan algoritma CNN berbasis arsitektur *EfficientNet-B4*. Jumlah data *training* ditingkatkan dibandingkan dengan data *testing* untuk meningkatkan pemahaman sistem terhadap pemetaan data yang lebih kompleks. Data training berjumlah sekitar 70% dari total data penelitian, yakni sebanyak 424 data. Pada tahap training, data diproses dalam 20 iterasi (*epoch*) dengan nilai batch sebesar 10. Setelah penyelesaian proses pelatihan, model dihasilkan dan akan digunakan dalam tahap pengujian. Berikut ini adalah *pseudocode* yang menggambarkan langkah-langkah dalam proses training data

```

IMPORT library
DEFINISIKAN train_data_augmentation sebagai ImageDataGenerator dengan
parameter augmentasi rescaling, rotasi, pergeseran (shifting), shear, zoom, dan
flipping
DEFINISIKAN valid_data_augmentation sebagai ImageDataGenerator dengan
parameter augmentasi rescaling
BUAT train_datagen dengan menerapkan train_data_augmentation ke direktori
data pelatihan
BUAT valid_datagen dengan menerapkan valid_data_augmentation ke direktori
data validasi
DEFINISIKAN checkpoint_filepath sebagai jalur untuk menyimpan titik
kontrol dengan tempat penampungan epoch
DEFINISIKAN checkpoint_callback menggunakan ModelCheckpoint dengan
checkpoint_filepath, pemantauan kehilangan validasi, mode, dan tingkat
ketercakupan
DEFINISIKAN reduce_lr_callback menggunakan ReduceLROnPlateau dengan
pemantauan kehilangan validasi, faktor, kesabaran, mode, dan tingkat
ketercakupan
MUAT model EfficientNetB4 yang telah dilatih sebelumnya dengan bobot dari
'imagenet' dan tentukan bentuk inputnya
CETAK jumlah lapisan dalam model dasar

```

DEFINISIKAN jumlah lapisan yang akan disesuaikan

BEKUK lapisan dalam model dasar hingga lapisan penyesuaian yang ditentukan

BUAT model dengan menambahkan lapisan secara berurutan: `base_model`, `GlobalAveragePooling2D`, `Dense`, `Dense`

KOMPILASI model menggunakan pengoptimal Adam, kerugian cross-entropy biner, dan metrik akurasi

CETAK ringkasan model

ATUR jumlah epoch, langkah pelatihan per epoch, dan langkah validasi per epoch

LATIH model dengan menggunakan metode fit, dengan menspesifikasikan `train_generator`, `validation_data`, dan callbacks

3.2.7. *Testing Data*

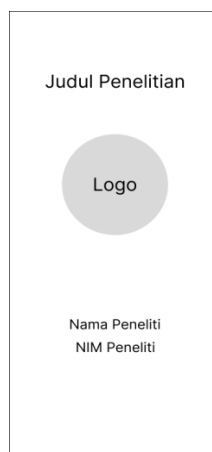
Proses *testing* data merupakan proses untuk menguji sistem dengan model yang sudah terbentuk pada proses *training* data. Proses ini bertujuan untuk mengetahui apakah sistem dapat mengenali klasifikasi rasa buah jeruk dengan menginput citra jeruknya. Pada proses ini dapat dilihat seberapa tinggi pemahaman sistem untuk mengklasifikasikan data yang akan diinput, hal tersebut dapat dilihat dari akurasi testing data.

3.3. Perancangan Antarmuka Sistem

Pada langkah ini, fokus diberikan pada rancangan antarmuka atau tampilan aplikasi pendeteksi rasa Jeruk Berastagi. Hal ini bertujuan untuk memberikan gambaran menyeluruh tentang aplikasi kepada pengguna dan memfasilitasi pemahaman serta penggunaan yang lebih mudah. Rancangan antarmuka ini dirancang agar pengguna dapat dengan jelas memahami fitur-fitur aplikasi dan merasa nyaman dalam menggunakannya.

3.3.1. Rancangan Tampilan *Splashscreen*

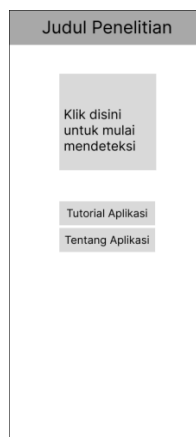
Splashscreen adalah layar pertama yang muncul saat aplikasi dijalankan untuk pertama kali. Tampilan splashscreen ini mencakup judul aplikasi (penelitian), logo, detil peneliti, dan institusi penelitian. Gambar 3.13 dibawah adalah rancangan dari tampilan *splashscreen*.



Gambar 3.13 Rancangan Tampilan *Splashscreen*

3.3.2. Rancangan Tampilan *Home Page*

Home page berisikan judul penelitian yang ditulis pada bagian atas halaman. Lalu pada halaman ini juga terdapat tombol untuk memulai aplikasi, tutorial aplikasi, dan tentang aplikasi. Gambar 3.14 dibawah adalah rancangan dari tampilan *home page*.



Gambar 3.14 Rancangan Tampilan *Home Page*

3.3.3. Rancangan Tampilan Untuk Mendeteksi

Halaman ini akan muncul jika tombol “Klik disini untuk mulai mendeteksi” ditekan pada halaman utama sebelumnya. Pada halaman ini terdapat tombol untuk mengambil gambar dari galeri dan tombol untuk mengambil gambar dengan menggunakan kamera. Gambar 3.15 dibawah adalah rancangan dari tampilan untuk mendeteksi.



Gambar 3.15 Rancangan Tampilan Hasil *Training*

3.3.4. Rancangan Tampilan Hasil Identifikasi

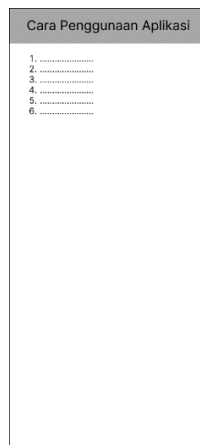
Hasil klasifikasi dari gambar yang sebelumnya telah dimasukkan oleh pengguna di halaman menu utama akan ditampilkan pada halaman ini. Gambar 3.16 dibawah adalah rancangan dari tampilan hasil identifikasi.

Hasil Identifikasi
<p>Hasil Identifikasi Citra</p> <p>Saran dan penjelasan hasil identifikasi:</p> <p>1.</p> <p>2.</p> <p>3.</p> <p>Catatan dari peneliti:</p> <p>.....</p>

Gambar 3.16 Rancangan Tampilan Hasil Identifikasi

3.3.5. Rancangan Tampilan Halaman *Tutorial*

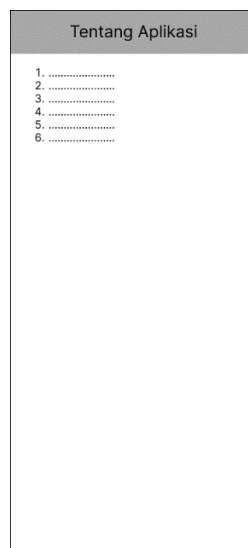
Halaman ini berisikan informasi penggunaan aplikasi berisi tata cara penggunaan aplikasi yang ditulis ke dalam kalimat. Gambar 3.17 dibawah adalah rancangan dari tampilan halaman *tutorial*.



Gambar 3.17 Rancangan Tampilan Tutorial

3.3.6. Rancangan Tampilan Halaman Tentang Aplikasi

Halaman Tentang Aplikasi berisikan informasi mengenai aplikasi seperti deskripsi dan tujuan aplikasi. Gambaran dari tampilan tentang aplikasi terlihat pada Gambar 3.18.



Gambar 3.18 Rancangan Tampilan *Tentang Aplikasi*

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Penerapan identifikasi rasa Buah Jeruk Berastagi dengan memanfaatkan arsitektur *EfficientNet-B4* dilaksanakan dalam suatu sistem yang membutuhkan komponen perangkat keras, perangkat lunak, dan sumber daya data pendukung, termasuk.:

4.1.1. Perangkat Keras dan Perangkat Lunak

Perangkat keras dan perangkat lunak yang digunakan untuk mengimplementasikan aplikasi ini melibatkan:

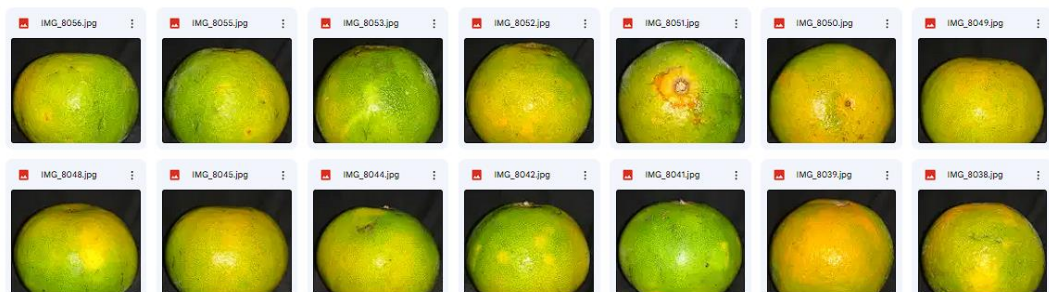
1. *Processor AMD® Ryzen™ 5 2400G*
2. *Storage SSD 256GB dan HDD 1TB*
3. *RAM 18GB*
4. *OS Windows 10 Home 64-bit*
5. *Android Studio*
6. *Google Colab Notebooks*
7. *Tensorflow Lite*

4.1.2. Implementasi Data

Penelitian ini menggunakan data yang terdiri dari citra jeruk Berastagi yang diambil dari Kebun Jeruk di Berastagi dan jeruk Berastagi yang dibeli dari pasar lokal di Kota Medan. Proses pengambilan citra menggunakan kamera smartphone dengan kualitas 12MP sebagai data penelitian. Sebanyak 608 citra jeruk diambil untuk penelitian ini. Validasi rasa jeruk dilakukan dengan mencicipi jeruk setelah pengambilan citra. Proses validasi melibatkan partisipasi dari 5 orang, dan dilakukan voting untuk memastikan konsistensi penilaian terhadap rasa jeruk yang dicicipi. Adapun beberapa contoh data pada masing-masing klasifikasi terdapat pada gambar 4.1 dan 4.2.



Gambar 4.1 Contoh Jeruk Manis



Gambar 4.2 Contoh Jeruk Tidak Manis

4.1.3. Implementasi Desain Antarmuka Aplikasi

Implementasi desain antarmuka aplikasi didasari oleh rancangan desain yang sudah dilakukan pada bab sebelumnya, yaitu:

1. Tampilan *Splashscreen*

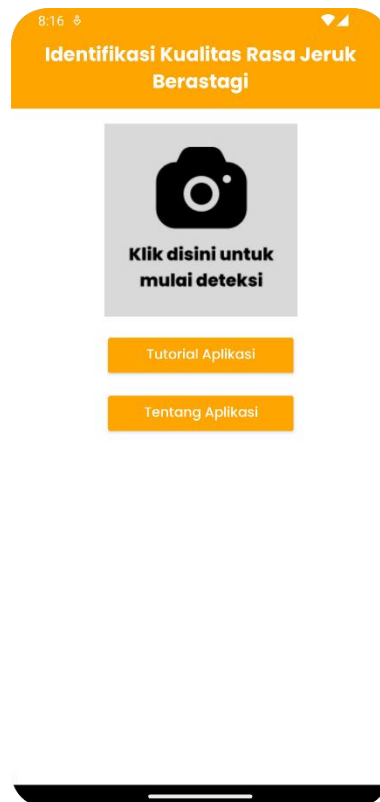
Halaman *splashscreen* adalah halaman yang muncul saat pengguna memulai aplikasi. Halaman ini mencakup judul penelitian, logo universitas, dan informasi identitas peneliti, termasuk nama peneliti, NIM, fakultas, dan instansi peneliti. Setelah beberapa saat, pengguna akan dialihkan secara otomatis ke halaman beranda. Gambar 4.3 dibawah merupakan tampilan *splashscreen*.



Gambar 4.3 Tampilan *Splashscreen*

2. Tampilan Halaman *Homescreen*

Halaman *homescreen* berisikan *icon* untuk memulai identifikasi, tombol tutorial aplikasi, dan tombol tentang aplikasi. Gambar 4.4 dibawah merupakan tampilan halaman *homescreen*.



Gambar 4.4 Tampilan Halaman *Homescreen*

3. Tampilan Halaman Identifikasi

Halaman identifikasi adalah halaman dimana identifikasi dilakukan. Di dalam halaman ini, terdapat sebuah tombol untuk melaksanakan identifikasi dan dua tombol tambahan untuk memilih metode pengambilan gambar yang akan digunakan. Gambar dapat dimasukan dengan cara mengambil gambar secara langsung melalui kamera atau memilih gambar yang telah ada dari galeri. Gambar yang dipilih akan ditampilkan dihalaman ini. Tampilan halaman identifikasi terdapat pada Gambar 4.5.



Gambar 4.5 Tampilan Halaman Identifikasi

4. Tampilan Halaman Hasil Identifikasi

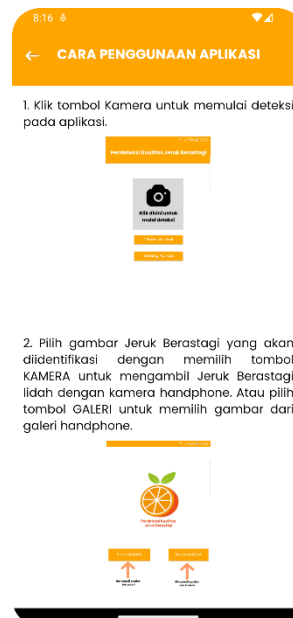
Halaman Hasil Identifikasi berisikan hasil identifikasi rasa jeruk berdasarkan gambar yang dimasukan. Hasil akan menunjukkan bahwa buah jeruk tersebut memiliki rasa yang manis atau tidak manis. Pada halaman ini juga terdapat cara untuk memilih buah Jeruk Berastagi yang manis. Tampilan halaman hasil identifikasi bisa dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan Halaman Hasil Identifikasi

5. Tampilan Halaman *Tutorial*

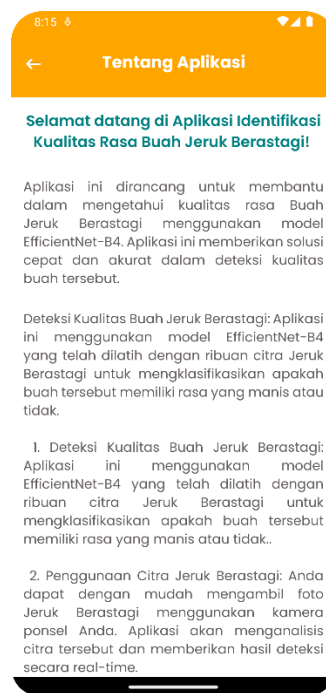
Halaman tutorial adalah halaman yang berisikan mengenai tata cara penggunaan aplikasi. Halaman ini dapat diakses dengan menekan tombol “Tutorial Aplikasi” yang terdapat pada halaman utama. Tampilan halaman *tutorial* bisa dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan Halaman Tutorial

6. Tampilan Halaman Tentang Aplikasi

Halaman Tentang Aplikasi adalah halaman yang berisikan deskripsi aplikasi, tujuan aplikasi, arsitektur yang digunakan, dan cara memilih buah jeruk berastagi yang manis. Tampilan halaman tentang aplikasi bisa dilihat pada Gambar 4.8.



Gambar 4.8 Tampilan Halaman Tentang Aplikasi

4.2. Prosedur Operasional

Aplikasi ini memiliki beberapa tampilan utama, termasuk *splashscreen*, halaman *home*, halaman hasil identifikasi, halaman tutorial, dan halaman informasi tentang aplikasi. Saat pengguna pertama kali membuka aplikasi, mereka akan melihat tampilan *splashscreen*. Setelah itu, mereka akan diarahkan ke halaman *home*. Di sana, pengguna dapat menemukan tombol untuk memulai proses identifikasi. Setelah menekan tombol tersebut, mereka akan diarahkan ke halaman identifikasi rasa jeruk. Pada halaman ini, pengguna memiliki pilihan untuk mengimpor citra jeruk dari galeri atau mengambil foto dengan kamera. Pengguna bisa menekan tombol “pilih gambar” untuk mengakses galeri ataupun tombol “mulai kamera”

untuk memasukan citra jeruk melalui kamera. Setelah citra jeruk dimasukan, pengguna dapat menekan tombol “deteksi” untuk melihat apakah jeruk tersebut memiliki rasa yang manis atau tidak manis.

Petunjuk pengguna mengenai penggunaan aplikasi dapat diakses dengan menekan tombol "tutorial aplikasi" yang terdapat di halaman utama. Selain itu, informasi tambahan mengenai aplikasi dapat ditemukan dengan menekan tombol "tentang aplikasi" di halaman utama.

4.3. Pelatihan Sistem

Proses pelatihan dilaksanakan dengan memanfaatkan algoritma *Convolutional Neural Network (CNN)* dengan arsitektur *EfficientNet-B4*. Jumlah data citra Jeruk Berastagi yang digunakan saat pelatihan sebanyak 4352 citra Jeruk Berastagi. Data yang digunakan saat pelatihan didapatkan dari 90% dari total data penelitian yaitu sebanyak 544 citra yang kemudian diaugmentasi sebanyak 6 kali. Sedangkan untuk data testing ada sebanyak 64 buah. Data testing tidak mengalami augmentasi dan akan dijadikan sebagai data uji setelah model siap digunakan. Rincian pembagian data dapat dilihat pada Tabel 4.1.

Tabel 4.1 Pembagian Data Training, Data Validation dan Data Testing Setelah Augmentasi

No.	Data	Data Training	Data Validating	Data Testing	Jumlah
1	Citra Jeruk Berastagi Manis	1696	480	32	2208
2	Citra Jeruk Berastagi Tidak Manis	1696	480	32	2208
Total Data		3392	960	64	4416

Saat proses pelatihan model, dilakukan beberapa percobaan dengan melakukan *hyperparameter tuning* untuk mencari dan mendapatkan model pelatihan terbaik yang akan diimplementasikan ke android. *Hyperparameter tuning* dilakukan dengan cara memodifikasi parameter-parameter seperti jumlah

epoch, ukuran *batch*, dan tingkat pembelajaran (*learning rate*) dimodifikasi untuk mencapai model pelatihan yang optimal berdasarkan akurasi validasi (*validation accuracy*) dan kerugian validasi (*validation loss*).

Pada percobaan pertama dengan *epoch* 20 dan *batch size* 10, diperoleh *validation accuracy* sebesar 96.09% dan *validation loss* sebesar 13.61%. Percobaan berikutnya dengan *epoch* 20 dan *batch size* 20 menghasilkan *validation accuracy* sebesar 95.83 %, namun *validation loss* meningkat menjadi 14.48%. Selanjutnya, pada percobaan dengan *epoch* 20 dan *batch size* 30, *validation accuracy* menurun menjadi 94.53%, dengan *validation loss* sebesar 11.99%. Percobaan menggunakan *epoch* 20 dan *batch size* 40 menghasilkan *validation accuracy* sebesar 95.83%, dan *validation loss* sebesar 14.02%.

Percobaan dilanjutkan dengan menggunakan *epoch* sebesar 40 dan *batch size* sebesar 10 menghasilkan *validation accuracy* sebesar 96.88% dan *validation loss* sebesar 11.65%. Selanjutnya percobaan menggunakan *epoch* sebesar 40 dan *batch size* sebesar 20 menghasilkan *validation accuracy* yang sama yaitu sebesar 96.88% dan *validation loss* yang lebih tinggi yaitu sebesar 15.38%. Percobaan selanjutnya menggunakan *epoch* sebesar 40 dan *batch size* sebesar 30 menghasilkan *validation accuracy* yang lebih tinggi yaitu sebesar 97.66% dan *validation loss* menurun hingga sebesar 9.86%. Kemudian percobaan selanjutnya menggunakan *epoch* sebesar 40 dan *batch size* sebesar 40 menghasilkan *validation accuracy* sebesar 95.83% dan *validation loss* sebesar 15.89%.

Tabel 4.2 Tabel Pelatihan

epoch	batch size	val_Acc (%)	val_loss (%)
20	10	96,09	13,61
20	20	95,83	14,48
20	30	94,53	11,99
20	40	95,83	14,02
40	10	96,88	11,65
40	20	96,88	15,38
40	30	97,66	9,86
40	40	95,83	15,89

Berdasarkan eksperimen yang dilakukan, didapatkan bahwa percobaan dengan 40 *epoch* dan ukuran *batch* sebesar 30 memiliki hasil yang paling baik. Percobaan dengan *epoch* sebesar 40 dan *batch size* sebesar 30 mendapatkan *validation accuracy* sebesar 97.66% dan *validation loss* sebesar 9.86%. Keseimbangan yang baik antara akurasi dan loss validasi, menunjukkan kemampuan yang lebih baik dalam menggeneralisasi ke data yang tidak terlihat selama pelatihan (Ghnemat et al., 2023).

Setelah pelatihan, didapatkan bahwa *epoch* ke-19 menghasilkan *validation accuracy* sebesar 96.61% dan *validation loss* sebesar 8.90%. Informasi tentang setiap *epoch* yang dapat diamati tertera dalam tabel 4.3.

Tabel 4.3 Tabel Epoch

Epoch	Loss	Accuracy	Val_Loss	Val_Acc
1	0.5665	0.7655	1.4613	0.4609
2	0.4797	0.7788	0.3955	0.8281
3	0.4548	0.8186	2.7501	0.5703
4	0.4588	0.8164	0.4584	0.8203
5	0.4454	0.8363	0.2177	0.9219
6	0.4071	0.8142	0.8651	0.7344
7	0.3689	0.8650	0.4228	0.8906
8	0.3480	0.8606	0.2813	0.9141
9	0.3057	0.8783	0.1908	0.9453
10	0.3049	0.8783	0.1651	0.9453
11	0.2594	0.8960	0.1965	0.9453
12	0.3471	0.8673	0.2050	0.9141
13	0.2875	0.8695	0.2972	0.8750
14	0.2151	0.9358	0.1597	0.9375
15	0.2732	0.8960	0.1574	0.9531
16	0.2407	0.9049	0.1774	0.9219
17	0.3253	0.8695	0.1254	0.9453
18	0.2341	0.9049	0.2516	0.9062
19	0.2291	0.9181	0.2044	0.9219
20	0.2281	0.9204	0.1462	0.9531
21	0.2343	0.9049	0.2063	0.9453
22	0.2755	0.8938	0.1597	0.9375
23	0.2586	0.9071	0.1762	0.9375
24	0.2616	0.8872	0.1296	0.9453

25	0.2582	0.9137	0.2353	0.9062
26	0.2646	0.9027	0.1265	0.9609
27	0.1930	0.9425	0.2287	0.9141
28	0.3303	0.8628	0.0986	0.9766
29	0.2570	0.9093	0.2314	0.9297
30	0.1989	0.9270	0.2204	0.9219
31	0.2724	0.8938	0.1061	0.9844
32	0.2508	0.9093	0.1521	0.9453
33	0.2504	0.9027	0.1845	0.9297
34	0.2567	0.8938	0.1623	0.9531
35	0.3319	0.8695	0.2431	0.9141
36	0.2637	0.8916	0.1158	0.9609
37	0.2278	0.9137	0.1338	0.9766
38	0.2806	0.8805	0.1789	0.9453
39	0.2912	0.8805	0.1519	0.9609
40	0.3414	0.8650	0.1367	0.9453

Setelah 40 *epoch*, didapatkan bahwa pada *epoch* ke-28 memiliki hasil akurasi validasi paling tinggi yaitu sebesar 97.66% dan validasi *loss* hanya sebesar 9.86%. *Callback ReduceLROnPlateau* digunakan saat pelatihan model. *Callback* ini mengawasi nilai *val_loss* (kerugian pada data validasi) untuk menilai apakah ada penurunan yang signifikan dalam kinerja model. Jika tidak ada penurunan yang mencolok dalam sejumlah *epoch* tertentu, dalam kasus ini, selama 3 *epoch* sesuai dengan nilai kesabaran (*patience*) yang telah ditetapkan, maka *learning rate* akan disesuaikan menjadi lebih rendah.

Dalam penelitian ini, pada *epoch* ke-9, terjadi penurunan *learning rate* yang disesuaikan menjadi 0.00010000000474974513. Penyesuaian ini dilakukan karena selama 3 *epoch* terakhir, tidak terlihat penurunan yang signifikan pada nilai *val_loss*. Kemudian, pada *epoch* ke-13, terjadi penurunan kembali pada *learning rate* menjadi 1.0000000474974514e-05 karena *val_loss* tidak mengalami penurunan yang signifikan selama 3 *epoch* sebelumnya. Saat mencapai *epoch* ke-20, *learning rate* dikurangi menjadi 1.0000000656873453e-06. Penyesuaian *learning rate* kembali dilakukan pada *epoch* ke-23, menjadi 1.0000001111620805e-07. Pada *epoch* ke-26, terjadi penurunan lagi pada



learning rate menjadi 1.000000082740371e-08. Pada *epoch* ke-31, *learning rate* dikurangi menjadi 1.000000082740371e-09. Setelah itu, pada *epoch* ke-34, terjadi penurunan kembali pada *learning rate* menjadi 1.000000082740371e-10. Pada *epoch* ke-37, *learning rate* diturunkan menjadi 1.000000082740371e-11. Pada *epoch* terakhir, yaitu *epoch* ke-40, *learning rate* dikurangi menjadi 1.000000082740371e-12. Keseluruhan proses penyesuaian *learning rate* ini dilakukan secara otomatis menggunakan *callback ReduceLROnPlateau* dengan memantau nilai *val_loss*






Penggunaan *callback ReduceLROnPlateau* bertujuan untuk mengatur kecepatan belajar model. Model dapat mencoba untuk mencapai hasil yang lebih baik dan akurat selama pelatihan dengan mengurangi *learning rate*. Pengurangan *learning rate* dapat membantu untuk meningkatkan konvergensi model dan memperbaiki kinerja model.






4.4. Pengujian Sistem






Pengujian sistem menggunakan 10% data penelitian yaitu sebanyak 64 data citra jeruk yang belum diargumentasi. Sistem mengidentifikasi citra jeruk ke 2 kelas yaitu manis dan tidak manis sesuai dengan pelatihan sistem. Tabel 4.4 dibawah merupakan beberapa contoh hasil pengujian sistem.

Tabel 4.4 Tabel Hasil Pengujian Sistem

No	Citra	Manual	Sistem	Status
1		Manis	Manis	Benar
2		Manis	Manis	Benar

3		Manis	Manis	Benar
4		Manis	Manis	Benar
5		Manis	Manis	Benar
6		Manis	Manis	Benar
7		Manis	Manis	Benar

8		Manis	Manis	Benar
9		Manis	Manis	Benar
10		Manis	Manis	Benar
11		Tidak Manis	Tidak Manis	Benar
12		Tidak Manis	Tidak Manis	Benar

13		Tidak Manis	Tidak Manis	Benar
14		Tidak Manis	Tidak Manis	Benar
15		Tidak Manis	Tidak Manis	Benar
16		Tidak Manis	Tidak Manis	Benar
17		Tidak Manis	Manis	Salah

18		Tidak Manis	Tidak Manis	Benar
19		Tidak Manis	Tidak Manis	Benar
20		Tidak Manis	Manis	Salah

Tabel di atas mencerminkan hasil dari 20 pengujian aplikasi yang telah dilakukan. Informasi lebih lanjut mengenai pengujian secara menyeluruh dapat ditemukan di lampiran penelitian ini. Dari serangkaian pengujian dengan menggunakan 64 data, terdapat 2 data yang terklasifikasikan dengan kesalahan, khususnya pada kategori jeruk yang tidak manis. Hasil lengkap dari pengujian ini dapat ditemukan dalam confusion matrix yang terdokumentasikan di Tabel 4.5 penelitian ini.

Tabel 4.5 *Confusion Matrix*

		Prediksi	
		Manis	Tidak Manis
Aktual	Manis	32	0
	Tidak Manis	30	2

Dalam Tabel 4.3, terlihat bahwa terdapat dua citra Jeruk Berastagi yang tidak manis yang mengalami kesalahan prediksi. Dari data tersebut, kita dapat menghitung nilai *True Positive (TP)*, *False Positive (FP)*, dan *False Negative (FN)*. Nilai-nilai ini terdokumentasikan dalam Tabel 4.6.

Tabel 4.6 Nilai TP, FP, dan FN

	TP	FP	FN
Manis	32	0	0
Tidak Manis	30	2	0

Berdasarkan hasil pengujian data, kita memiliki 62 data yang merupakan True Positive (TP), 2 data yang merupakan False Positive (FP), dan tidak ada data yang merupakan False Negative (FN). Dengan nilai-nilai ini, kita dapat menghitung nilai Precision, Recall, dan F1-Score untuk setiap tingkatan klasifikasi.

a. *Precision*

Precision dapat dihitung dengan menggunakan rumus yang terdapat dalam persamaan 2.5.

$$\text{Precision Manis} = \frac{32}{32} \times 100\% = 100\%$$

$$\text{Precision Tidak Manis} = \frac{30}{32} \times 100\% = 93.75\%$$

b. *Recall*

Recall dapat dihitung dengan menggunakan rumus yang terdapat dalam persamaan 2.6.

$$\text{Recall Manis} = \frac{32}{32} \times 100\% = 100\%$$

$$\text{Recall Tidak Manis} = \frac{32}{32} \times 100\% = 100\%$$

c. *F1-Score*

F1-Score dapat dihitung dengan menggunakan rumus yang terdapat dalam persamaan 2.7.

$$\begin{aligned}
 F1\text{-Score Manis} &= 2 \times \frac{(1.00 \times 100)}{(1.00 + 1.00)} \times 100\% \\
 &= 2 \times \frac{1.00}{2.00} \times 100\% = 100\% \\
 F1\text{-Score Tidak Manis} &= 2 \times \frac{(1.00 \times 0.9375)}{(1.00 + 0.9375)} \times 100\% \\
 &= 2 \times \frac{0.9375}{1.9375} \times 100\% = 96.88\%
 \end{aligned}$$

Dengan perhitungan yang telah dilakukan, kita dapat membuat tabel nilai sebagai berikut:

Tabel 4.7 Tabel Nilai *Precision*, *Recall*, dan *F1-Score*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Manis	100%	100%	100%
Tidak Manis	93.75%	100%	96.88%

Dari total 64 data yang ada, 62 data diklasifikasikan dengan benar. Dari 64 data tersebut, 32 adalah citra Jeruk Berastagi dengan rasa manis dan 32 lainnya adalah citra Jeruk Berastagi dengan rasa tidak manis. Dengan hasil klasifikasi yang telah didapatkan, dihitunglah nilai akurasi dengan rumus perhitungan yang terdapat pada persamaan 2.4

$$Accuracy = \frac{32 + 30}{64} \times 100\% = \frac{62}{64} \times 100\% = 96.88\%$$

Dari perhitungan yang telah dilakukan, ditemukan bahwa tingkat akurasi penelitian menggunakan algoritma CNN dengan model EfficientNet-B4 mencapai 96.88%.

4.5. Kendala Penelitian

Ketika pengujian, terdapat kesalahan ketika mengidentifikasi rasa jeruk berastagi. Jeruk yang diidentifikasi rasanya tidak manis, ketika dicicipi memberikan rasa

manis. Hal ini terjadi karena hasil identifikasi hanya berdasarkan citra luar dari jeruk yang dianalisis. Ketika memilih jeruk, ada beberapa pertimbangan selain warna dan tekstur yaitu berat dan aroma jeruk. Di samping itu, gambar visual atau citra hanya memberikan representasi visual yang terbatas dan tidak dapat mencakup semua elemen yang terlibat dalam menilai rasa suatu buah. Rasa jeruk, sebagai contoh, dipengaruhi oleh faktor kimia khusus seperti kandungan gula, asam, aroma, dan sebagainya, yang tidak dapat terlihat atau diwakili secara visual dalam citra (Ron Porat et al., 2016).

Penelitian yang dilakukan oleh (Adelkhani et al., 2013) menggunakan ANFIS untuk mengklasifikasi rasa buah jeruk jenis Bam, Khooni, dan Thompson. Dari setiap 30 buah jeruk dengan varietas Khooni, Bam, dan Thompson, masing-masing 28 jeruk Khooni, 28 jeruk Bam, dan 29 jeruk Thompson berhasil diidentifikasi dengan akurasi yang sesuai. Keberhasilan sistem dalam mengenali rasa mencapai 93,33% untuk Khooni, 93,33% untuk Bam, dan 96,6% untuk Thompson, secara berurutan. Peneliti mengungkapkan bahwa kualitas rasa jeruk dipengaruhi oleh sejumlah faktor, dan mengandalkan satu parameter saja tidak memadai untuk mencapai karakterisasi rasa jeruk yang menyeluruh.

Identifikasi rasa pada buah nanas dilakukan oleh (Siricharoen et al., 2023) menggunakan *ResNet*, transformasi ruang warna relatif hijau, dan *Mask R-CNN* untuk mengidentifikasi rasa pada buah nanas, dan berhasil mencapai nilai *mAP* sebesar 0.8351. Hasil penelitian ini menyimpulkan bahwa ciri-ciri yang membedakan antara rasa asam, manis & asam, dan manis terutama terlokalisasi pada beberapa wilayah, seperti tunas nanas atau area di antara tunas dari satu buah. Beberapa poin penting mencakup beberapa wilayah tunas nanas tetapi sudah cukup untuk mengidentifikasi rasa. Namun, penelitian ini tidak mampu menjelaskan karakteristik rasa buah nanas dalam istilah fisik.

4.6. Perbedaan Penelitian

Perbandingan antara penelitian saya yang menggunakan algoritma CNN dengan arsitektur *EfficientNet-B4* untuk mengidentifikasi rasa jeruk Berastagi, dengan penelitian Adelkhani et al. (2013) dan Asriny et al. (2020), menunjukkan variasi

dalam pendekatan dan tujuan penelitian. Penelitian pertama oleh Adelkhani et al. fokus pada klasifikasi rasa jeruk Bam, Khoni, dan Thompson menjadi lima kelas dengan menggunakan kombinasi *Image Processing* dan ANFIS, mencapai tingkat keberhasilan klasifikasi yang tinggi, tetapi dengan skema klasifikasi yang berbeda.

Sementara itu, penelitian kedua oleh Asriny et al. mengeksplorasi penggunaan CNN untuk mengklasifikasi jeruk ke dalam lima kelas berdasarkan kualitas fisik, dengan membandingkan fungsi aktivasi *ReLU* dan *Tanh*. Hasilnya menunjukkan keunggulan fungsi aktivasi *ReLU* dengan tingkat akurasi sebesar 96%. Dalam konteks ini, penelitian saya fokus pada identifikasi rasa jeruk Berastagi dalam dua kelas, yaitu manis dan tidak manis, dengan tingkat akurasi sebesar 96.88%, menunjukkan ketepatan dan efisiensi algoritma CNN dengan model *EfficientNet-B4* dalam penelitian rasa buah. Sementara ketiganya menghadirkan kontribusi yang berharga, perbedaan tujuan klasifikasi, jumlah kelas, dan pendekatan algoritma menunjukkan keragaman dalam metode penelitian yang dapat diaplikasikan dalam domain identifikasi kualitas buah.

Tabel 4.8 Tabel Perbedaan Penelitian

Penelitian	Metode	Objek	Jumlah Kelas	Akurasi
Penelitian Saya	CNN dengan arsitektur <i>EfficientNet-B4</i>	Jeruk Berastagi	2 (Manis dan Tidak Manis)	96.88%
Adelkhani et al. (2013)	Image Processing dengan ANFIS	Jeruk Bam, Khoni, Thompson	5 (Sangat Asam, Asam, Manis-Asam, Manis,	96.6% (BAM), 93.3% (Khoni), 93.3% (Thompson)

			Sangat Manis)	
Asriny et al.	CNN	Jeruk Keprak/Siam	5 (Baik Tingkat 1, Baik Tingkat 2, Rusak, Busuk, Belum Matang)	96% (<i>ReLU</i>), 93.8% (<i>Tanh</i>)

4.7. Kontribusi Penelitian

Penelitian saya yang menggunakan algoritma CNN dengan arsitektur *EfficientNet-B4* untuk mengidentifikasi rasa jeruk Berastagi menandai kontribusi yang pengolahan citra dan analisis rasa buah-buahan. Penelitian ini menjadi yang pertama dalam menerapkan arsitektur *EfficientNet-B4* secara khusus untuk analisis rasa jeruk, menunjukkan inovasi dalam pendekatan algoritma untuk klasifikasi rasa buah. Dengan tingkat akurasi sebesar 96.88%, penelitian ini tidak hanya membuktikan keefektifan model *EfficientNet-B4* dalam tugas klasifikasi tersebut tetapi juga memberikan landasan untuk penelitian lebih lanjut dalam penggunaan arsitektur tersebut dalam konteks analisis rasa buah-buahan secara umum.

Keberhasilan penelitian ini juga menggambarkan potensi penerapan teknologi pemrosesan citra dan jaringan saraf konvolusional untuk meningkatkan kualitas dan ketepatan dalam mengidentifikasi rasa buah. Dengan memfokuskan penelitian pada buah jeruk Berastagi dan mengklasifikasikannya menjadi dua kelas utama, yaitu manis dan tidak manis, hasilnya dapat membantu industri pertanian dan pengolahan makanan dalam pemilihan buah dengan rasa yang diinginkan. Dengan demikian, penelitian ini memberikan sumbangan penting dalam menggabungkan teknologi pengolahan citra dan kecerdasan buatan untuk

meningkatkan pemahaman kita terhadap rasa buah dan dapat menjadi landasan untuk pengembangan lebih lanjut dalam bidang ini.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari keseluruhan pelaksanaan penelitian ini, diperoleh hasil penelitian dan pengujian sistem identifikasi rasa buah Jeruk Berastagi berdasarkan citra jeruk berastagi berbasis android menggunakan model *EffecientNet-B4*. Kesimpulan yang bisa diambil adalah:

1. Pemanfaatan metode *Convolutional Neural Network* dengan menggunakan arsitektur model *EfficientNet-B4* memungkinkan identifikasi rasa jeruk berastagi berdasarkan citra jeruk berastagi, dengan tingkat akurasi pengujian mencapai 96.88%
2. Pengaplikasian *hyperparameter tuning* saat proses pelatihan model dapat meningkatkan hasil akurasi pada sistem.
3. Model yang digunakan dalam implementasi aplikasi adalah model yang telah diuji dan terbukti optimal dalam eksperimen. Model ini dilatih dengan 40 *epoch* dan *batch size* sebesar 30, menghasilkan *validation accuracy* sebesar 97.66% dan *validation loss* sebesar 9.86%.

5.2. Saran

Saran yang dapat diajukan oleh penulis untuk pengembangan penelitian ini adalah:

1. Pada penelitian selanjutnya, diharapkan dapat membuat sistem klasifikasi rasa buah Jeruk Berastagi secara *real time*.
2. Pada penelitian selanjutnya, diharapkan dapat menambahkan parameter lain seperti berat, aroma, dan kandungan dalam buah untuk identifikasi rasa jeruk.
3. Pada penelitian selanjutnya, dapat menguji area sekitar tangkai buah jeruk yang berkerut atau tidak sebagai parameter pembedanya.
4. Pada penelitian selanjutnya, diharapkan dapat menggunakan jeruk dengan varietas lain sebagai objek penelitian.

5. Pada penelitian selanjutnya, diharapkan menggunakan metode klasifikasi lain agar dapat dibandingkan dengan hasil penelitian yang didapatkan dengan menggunakan arsitektur *Effecientnet-B4*.

DAFTAR PUSTAKA






- Adelkhani, A., Beheshti, B., Minaei, S., Javadikia, P., & Ghasemi-Varnamkhasti, M. (2013). Taste characterization of orange using image processing combined with ANFIS. *Measurement*, 46(9), 3573–3580. <https://doi.org/10.1016/j.measurement.2013.06.052>
- Afzalani, A., Muthalib, R. A., Dianita, R., Hoesni, F., Raguati, R., & Musnandar, E. (2021). Evaluasi Suplementasi Indigofera zollingeriana Sebagai Sumber Green Protein concentrate Terhadap Produksi Gas Metan, Amonia dan Sintesis Protein Mikroba Rumen. *Jurnal Ilmiah Universitas Batanghari Jambi*, 21(3), 1455. <https://doi.org/10.33087/jiubj.v21i3.1736>
- Alam, N., Darwin, D., & Firmansyah, U. (2018). Pengembangan Metode Pemilah Kualitas Telur Bebek Berdasarkan Normal/Abnormal Berbasis Image Processing. *Faktor Exacta*, 11(4), 362. <https://doi.org/10.30998/faktorexacta.v11i4.2889>
- Albahry, Adnan. (2011). Karakterisasi Sifat Fisik Jeruk Manis Berdasarkan Tingkat Ketuaan.
- Ali, Md. N. Y., Sorwar, G., & Rahman, Md. L. (2023). Mangoes Taste Detection System Using Deep Learning. *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1–5. <https://doi.org/10.1109/AISP57993.2023.10134902>
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3), 292. <https://doi.org/10.3390/electronics8030292>
- Arnold, L., Rebecchi, S., Chevallier, S., & Paugam-Moisy, H. (2011). An Introduction to Deep Learning. *Computational Intelligence*.
- Asriny, D. M., Rani, S., & Hidayatullah, A. F. (2020). Orange Fruit Images Classification using Convolutional Neural Networks. *IOP Conference Series: Materials Science and Engineering*, 803(1), 012020. <https://doi.org/10.1088/1757-899X/803/1/012020>
- Asy'ari, M. Z., Gouw, A. W., & Limanjaya, D. A. (2023). Implementasi Algoritma Convolutional Neural Network Pada Kendaraan Tanpa Awak Skala Kecil. *Jurnal Asimetrik: Jurnal Ilmiah Rekayasa & Inovasi*, 19–26. <https://doi.org/10.35814/asiimetrik.v5i1.4082>
- Badan Pusat Statistik (BPS). 2022. Produksi Tanaman Buah-buahan 2022. (<https://www.bps.go.id/indicator/55/62/2/produksi-tanaman-buah-buahan.html>). Diakses 25 Mei 2023.






- Ghnemat, R., Alodibat, S., & Abu Al-Haija, Q. (2023). Explainable Artificial Intelligence (XAI) for Deep Learning Based Medical Imaging Classification. *Journal of Imaging*, 9(9), 177. <https://doi.org/10.3390/jimaging9090177>
- Khairi, D. ul, Ahsan, K., Badshah, G., Ali, S. Z., Raza, S. A., Alqahtani, O., & Shiraz, M. (2023). *Comparison Analysis of Machine Learning Classification on Apple Fruit Quality* [Preprint]. In Review. <https://doi.org/10.21203/rs.3.rs-3025343/v1>
- Khairullah, K., & Putra, E. D. (2021). Identifikasi Kematangan Cabai Menggunakan Operasi Morfologi (Opening dan Closing) dan Metode Backpropagation. *SISTEMASI*, 10(1), 96. <https://doi.org/10.32520/stmsi.v10i1.1094>
- Kholiq Prajatama, Fajar Eko Nugroho, Andri Faisal Sentosa, Syifa Fauziah, & Anggit Dwi Hartanto. (2019). *Deteksi Kualitas Buah Apel Malang Manalagi Menggunakan Algoritma Naive Bayes*. 8.
- Lakshmi, J. V. N. (n.d.). *Image Classification Algorithm on oranges to perceive sweetness using Deep Learning techniques*.
- Maier, A., Syben, C., Lasser, T., & Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift Für Medizinische Physik*, 29(2), 86–101. <https://doi.org/10.1016/j.zemedi.2018.12.003>
- Mokarram, M., Amin, H., & Khosravi, M. R. (2019). Using adaptive neuro-fuzzy inference system and multiple linear regression to estimate orange taste. *Food Science & Nutrition*, 7(10), 3176–3184. <https://doi.org/10.1002/fsn3.1149>
- Muchtar, K., Chairuman, Yudha Nurdin, & Afdhal Afdhal. (2021). Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(1), 107–113. <https://doi.org/10.29207/resti.v5i1.2831>
- Oleh, D., & Putra, A. S. J. (2023). *IDENTIFIKASI KEMATANGAN BUAH JERUK MEDAN MENGGUNAKAN METODE KNN (K-NEAREST NEIGHBOR) BERBASIS RED, GREEN, BLUE (RGB)*.
- O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (arXiv:1511.08458). arXiv. <http://arxiv.org/abs/1511.08458>
- Pangkasidhi, A. K., Palit, H. N., & Tjondrowiguno, A. N. (n.d.). *Aplikasi Pendukung Diagnosis COVID-19 Yang Menganalisis Hasil X-Ray Paru-Paru Dengan Model*.






- Pawening, R. E., Shudiq, W. J., & Wahyuni, W. (2020). KLASIFIKASI KUALITAS JERUK LOKAL BERDASARKAN TEKSTUR DAN BENTUK MENGGUNAKAN METODE k-NEAREST NEIGHBOR (k-NN). *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, 1(1), 10–17. <https://doi.org/10.33650/coreai.v1i1.1640>
- Perdani, W. R., Magdalena, R., & Caecar Pratiwi, N. K. (2022). Deep Learning untuk Klasifikasi Glaukoma dengan menggunakan Arsitektur EfficientNet. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), 322. <https://doi.org/10.26760/elkomika.v10i2.322>
- Rifqi, M. & Suharjito. (2021). DETEKSI KEMATANGAN TANDAN BUAH SEGAR (TBS) KELAPA SAWIT BERDASARKAN KOMPOSISI WARNA MENGGUNAKAN DEEP LEARNING. 14(2).
- Rizal, S., Ibrahim, N., Pratiwi, N. K. C., Saidah, S., & Fu'Adah, R. Y. N. (2020). Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNet. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 8(3), 693. <https://doi.org/10.26760/elkomika.v8i3.693>
- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtijas, H. P. A., Yustanti, W., & Prihanto, A. (2021). Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48. <https://doi.org/10.26740/jieet.v5n2.p44-48>
- Ron Porat, Sophie Deterre, Pierre Giampaoli, & Anne Plotto. (2016). *Biotechnology in flavor production* (The Flavor of Citrus Fruit).
- Salamah, U., Anita Ratnasari, & Sarwati Rahayu. (2022). Automated Fruit Classification Menggunakan Model VGG16 dan MobileNetV2. *JSAI (Journal Scientific and Applied Informatics)*, 5(3), 176–181. <https://doi.org/10.36085/jsai.v5i3.3615>
- Siricharoen, P., Yomsatieankul, W., & Bunsri, T. (2023). Recognizing the sweet and sour taste of pineapple fruits using residual networks and green-relative color transformation attached with Mask R-CNN. *Postharvest Biology and Technology*, 196, 112174. <https://doi.org/10.1016/j.postharvbio.2022.112174>
- Syava Aulya Aqhiela. (2022). PENDETEKSI CACAT BUAH JERUK DENGAN IMAGE PROCESSING. *BUANA ILMU*, 6(2), 114–121. <https://doi.org/10.36805/bi.v6i2.2344>






- Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*.
- Taufiq, R. M., Arry, S., & Hanif, Z. (n.d.). *Citrus in Indonesia: Production Perspective for Market Development*.
- Ümit Atila, Murat Uçar, Kemal Akyol, & Emine Uçar. (2021). *Plant leaf disease classification using EfficientNet deep learning model*. <https://doi.org/10.1016/j.ecoinf.2020.101182>
- Winda Adelindriani. (2022). *Analisis Kadar Vitamin C pada Buah Jeruk di Pasar Tradisional Menggunakan Titrasi Iodometri Sytematic Review*.
- Yanfei Li, Xianying Feng, Yandong Liu, & Xingchang Han. (2021). Apple Quality Identification And Classification by Image Processing Based On Convolutional Neural Networks. *Scientific Reports*, 11. <https://www.nature.com/articles/s41598-021-96103-2#citeas>
- Yolanda, Elisa Sianturi and Basuki, Sigit Priyono and Ellys, Yuliarti (2022) ANALISIS PREFERENSI KONSUMEN TERHADAP BUAH JERUK BERASTAGI DAN BUAH JERUK POKAM DI TOKO BUAH HARUM MANIS KOTA PEMATANGSIANTAR. Universitas Bengkulu.
- Zaelani, F., & Miftahuddin, Y. (2022). Perbandingan Metode EfficientNetB3 dan MobileNetV2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun: Metode EfficientNetB3 dan MobileNetv2. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 9(1). <https://doi.org/10.33197/jitter.vol9.iss1.2022.911>
- Zamachsari, F., & Puspitasari, N. (2021). Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(2), 203–212. <https://doi.org/10.29207/resti.v5i2.2952>
- Zeb, A., Qureshi, W. S., Ghafoor, A., Malik, A., Imran, M., Mirza, A., Tiwana, M., & Alanazi, E. (2022). *Towards Sensory Assessment Classification using Short-Wave NIR Spectroscopy for Orange Cultivars* [Preprint]. In Review. <https://doi.org/10.21203/rs.3.rs-1882562/v2>






Lampiran 1. Tabel Hasil Pengujian Sistem






No	Citra	Manual	Sistem	Status
1		Manis	Manis	Benar
2		Manis	Manis	Benar
3		Manis	Manis	Benar
4		Manis	Manis	Benar
5		Manis	Manis	Benar






6		Manis	Manis	Benar
7		Manis	Manis	Benar
8		Manis	Manis	Benar
9		Manis	Manis	Benar
10		Manis	Manis	Benar






11		Manis	Manis	Benar
12		Manis	Manis	Benar
13		Manis	Manis	Benar
14		Manis	Manis	Benar
15		Manis	Manis	Benar







16		Manis	Manis	Benar
17		Manis	Manis	Benar
18		Manis	Manis	Benar
19		Manis	Manis	Benar
20		Manis	Manis	Benar






21		Manis	Manis	Benar
22		Manis	Manis	Benar
23		Manis	Manis	Benar
24		Manis	Manis	Benar
25		Manis	Manis	Benar






26		Manis	Manis	Benar
27		Manis	Manis	Benar
28		Manis	Manis	Benar
29		Manis	Manis	Benar
30		Manis	Manis	Benar






31		Manis	Manis	Benar
32		Manis	Manis	Benar
33		Tidak Manis	Tidak Manis	Benar
34		Tidak Manis	Tidak Manis	Benar
35		Tidak Manis	Tidak Manis	Benar




36		Tidak Manis	Tidak Manis	Benar
37		Tidak Manis	Tidak Manis	Benar
38		Tidak Manis	Tidak Manis	Benar
39		Tidak Manis	Tidak Manis	Benar
40		Tidak Manis	Tidak Manis	Benar

41		Tidak Manis	Tidak Manis	Benar
42		Tidak Manis	Tidak Manis	Benar
43		Tidak Manis	Tidak Manis	Benar
44		Tidak Manis	Tidak Manis	Benar
45		Tidak Manis	Tidak Manis	Benar
46		Tidak Manis	Tidak Manis	Benar

47		Tidak Manis	Tidak Manis	Benar
48		Tidak Manis	Tidak Manis	Benar
49		Tidak Manis	Tidak Manis	Benar
50		Tidak Manis	Tidak Manis	Benar
51		Tidak Manis	Tidak Manis	Benar

52		Tidak Manis	Manis	Salah
53		Tidak Manis	Tidak Manis	Benar
54		Tidak Manis	Tidak Manis	Benar
55		Tidak Manis	Manis	Salah
56		Tidak Manis	Tidak Manis	Benar

57		Tidak Manis	Tidak Manis	Benar
58		Tidak Manis	Tidak Manis	Benar
59		Tidak Manis	Tidak Manis	Benar
60		Tidak Manis	Tidak Manis	Benar
61		Tidak Manis	Tidak Manis	Benar

62		Tidak Manis	Tidak Manis	Benar
63		Tidak Manis	Tidak Manis	Benar
64		Tidak Manis	Tidak Manis	Benar



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasikomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR :231/UN5.2.1.14/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasikom-TI USU tanggal 8 Januari 2024 perihal permohonan ujian skripsi
Nama : KARVIN HALIM
NIM : 191402118
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Implementasi Arsitektur Efficientnet Untuk Mengidentifikasi Rasa Buah Jeruk Berastagi Berdasarkan Citra Jeruk Berastagi Berbasis Android
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hono Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggaraan pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasikom-TI USU Periode 2021-2026
- MEMUTUSKAN
- Menetapkan :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
Ketua : Baihaqi Siregar S.Si., MT.
NIP: 197901082012121002
Sekretaris : Faniada Purnamasari S.TM.IT
NIP: 198908172019032023
Anggota Penguji : Rossy Nurhasunah S.Kem., M.Kom
NIP: 198707012019032016
Anggota Penguji : Prof. Drs. Mahyuddin M.IT., Ph.D
NIP: 196712251998021001
Moderator : -
Panitia : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Ujian Pakai (PNPB) Fasikom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan dipertanggungjawabkan sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.
- Tembusan :
1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip
- Ditetapkan di Medan
Pada tanggal 9 Januari 2024
Dekan
MAYA SIREGAR
NIP. 197401272002122001