

**IDENTIFIKASI JENIS BUAH SALAK MENGGUNAKAN  
ALGORITMA SSD-MOBILENET**

**SKRIPSI**

**TALITHA ASHVI RAYHAN**

**171402083**



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

# IDENTIFIKASI JENIS BUAH SALAK MENGGUNAKAN ALGORITMA SSD-MOBILENET

## SKRIPSI

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

TALITHA ASHVI RAYHAN

171402083



PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

## PERSETUJUAN

Judul : Identifikasi Jenis Buah Salak Menggunakan Algoritma MobileNet  
Kategori : Skripsi  
Nama Mahasiswa : Talitha Ashvika Rayhan  
Nomor Induk Mahasiswa : 171402083  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Fakultas : Ilmu Komputer dan Teknologi Informasi  
Universitas Sumatera Utara

Kedua, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2,

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

Pembimbing 1,

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

Diketahui/disetujui oleh  
Program Studi S1 Teknologi Informasi  
Ketua,

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

**PERNYATAAN****IDENTIFIKASI JENIS BUAH SALAK MENGGUNAKAN  
ALGORITMA SSD-MOBILENET****SKRIPSI**

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

Medan, 12 Juli 2024

Talitha Ashvi Rayhan  
171402083

## UCAPAN TERIMA KASIH

Puji dan syukur kepada Allah SWT, karena rahmat dan seizinnya yang telah mengiringi hari-hari penulis dalam menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana dari Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.

Penulisan skripsi ini tidak akan selesai tanpa adanya doa, dukungan dan dorongan dari berbagai pihak. Ucapan terima kasih yang sangat besar penulis sampaikan kepada:

1. Diri sendiri yang sudah sabar, tangguh, tidak berhenti mencari motivasi dan tidak menyerah selama proses pengerjaan skripsi sehingga akhirnya dapat dilalui.
2. Kedua orang tua penulis, Bapak Ir. Abdullah Hadely dan Ibu Elvi Agustina yang selalu memberikan doa yang tidak putus dikirimkan sepanjang hidup penulis.
3. Ibu Dr. Maya Sivi Lydia B.SC., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Dedy Arisandi ST., M.Kom. selaku Ketua Prodi S1 Teknologi Infommasi Universitas Sumatera Utara.
5. Bapak Ivan Jaya S.Si, M.Kom. selaku Sekretaris Prodi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Bapak Dedy Arisandi, S.T., M.Kom. selaku dosen pembimbing pertama penulis yang telah banyak membimbing dan membantu penulis dalam penelitian serta penulisan skripsi ini.
7. Bapak Romi Fadillah Rahmat B.Comp.Sc., M.Sc.selaku dosen pembimbing kedua penulis yang telah banyak membimbing dan membantu penulis dalam penelitian serta penulisan skripsi ini.
8. Bapak Ainul Hizriadi, S.Kom., M.Sc. selaku dosen penguji pertama yang banyak memberikan bimbingan dan masukan dalam penulisan skripsi ini.
9. Ibu Umayya Ramadhani Putri Nasution, S.TI., M.Kom. selaku dosen penguji kedua yang sudah banyak memberikan ilmu dan arahan terkait penulisan skripsi ini.
10. Seluruh dosen dan staf pegawai di lingkungan Program Studi Teknologi Informasi dan Fasilkom-TI USU, yang telah membantu selama masa perkuliahan penulis.
11. Keluarga besar penulis yang telah memberikan doa, motivasi, dukungan dan semangat kepada penulis.
12. Uci yang selalu menyemangati, tidak pernah berhenti mengingatkan untuk berdoa

dan tidak berhenti mendoakan penulis agar segala prosesnya berjalan lancar.

13. Muhammad Bung Alhavi dan Hasby Ibrahim Movic selaku adik penulis yang selalu membantu dan menyemangati penulis.
14. Jean Wahda Aischara selaku adik penulis yang tidak berhenti memberikan motivasi dan memberikan masukan kepada penulis.
15. Kepada penghuni grup “Galaxy”, Rezky Febrydawanti, Nadia Nasywa Lubis, Nabila Azzahra, Bella Olivia Putrisanni, Vania Putri Saryandra, Fakhirah Mentaya dan Fadhilah Annisa selaku teman yang selalu ada dan memberikan banyak dukungan dan motivasi kepada penulis selama masa perkuliahan.
16. Penghuni grup “Bukan Galaxy”, Dinul, Teha, Gilbert, Fajar dan Rafif yang senantiasa membantu dan menyemangati selama masa perkuliahan.
17. Raisya Karin Amelda Lubis yang selalu bersedia dan bertahan menjadi tempat berkeluh kesah selama ini.
18. Dina Unzila, teman penulis yang mengerti, mendengarkan, menghibur dan memberikan banyak sekali dukungan serta semangat kepada penulis.
19. Sita Kirana Atikah yang sudah selalu ada dan menemani melewati masa senang dan sulitnya penulis.
20. Kak Mel, Kak Pika, Kak Riri dan Kak Dhita yang selalu setia mendukung.
21. Teman-teman dari program studi Teknologi Informasi, khususnya angkatan 2017 Kom B yang sudah banyak membantu dan menemani penulis selama masa perkuliahan.

Penulis mengetahui bahwa skripsi ini masih belum sempurna dan masih memiliki banyak kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk memperbaiki skripsi pada masa yang akan datang. Penulis juga berharap bahwa skripsi ini dapat bermanfaat bagi semua pihak yang membacanya

Medan, 12 Juli 2024

Penulis

## **IDENTIFIKASI JENIS BUAH SALAK MENGGUNAKAN ALGORITMA SSD-MOBILENET**

### **ABSTRAK**

Salak merupakan buah tropis yang memiliki berbagai varietas dengan perbedaan morfologi yang subtil namun signifikan. Penelitian ini bertujuan untuk mengidentifikasi jenis salak menggunakan metode SSD Mobilenet dalam analisis citra digital. Metode yang dikembangkan dalam penelitian ini mampu membedakan tiga jenis salak yang umum ditemukan di Indonesia, yaitu salak pondoh, salak sidempuan, dan salak bali, berdasarkan karakteristik fisik seperti warna kulit, ukuran dan bentuk buah. Pengujian dilakukan dengan menggunakan dataset citra salak yang telah dikumpulkan secara eksklusif, dan hasilnya menunjukkan bahwa SSD Mobilenet mencapai akurasi yang memuaskan. Rata-rata akurasi keseluruhan yang dicapai adalah 91,6%. Temuan ini memiliki aplikasi yang potensial dalam industri pertanian untuk pengelolaan varietas tanaman yang lebih efisien dan akurat, serta dalam industri makanan untuk memastikan keaslian produk. Teknologi pengenalan objek seperti yang dikembangkan dalam penelitian ini dapat digunakan untuk meningkatkan presisi dalam identifikasi buah-buahan, mempercepat proses sortasi serta pengidentifikasian.

**Kata Kunci:** Identifikasi jenis salak, SSD-MobileNet, algoritma deteksi, akurasi

## **IDENTIFICATION OF SALAK TYPE USING SSD-MOBILENET ALGORITHM**

### **ABSTRACT**

*Salak is a tropical fruit that has various varieties with subtle but significant morphological differences. This research aims to identify snake fruit types using the SSD Mobilenet method in digital image analysis. The method developed in this research is able to distinguish three types of salak commonly found in Indonesia, namely pondoh salak, sidempuan salak, and Balinese salak, based on physical characteristics such as skin color, size and shape of the fruit. Tests were carried out using a dataset of salak images that had been collected exclusively, and the results showed that SSD Mobilenet achieved satisfactory accuracy. The overall average accuracy achieved was 91.6%. These findings have potential applications in the agricultural industry for more efficient and accurate management of crop varieties, as well as in the food industry to ensure product authenticity. Object recognition technology such as that developed in this research can be used to increase precision in fruit identification, speed up the sorting and identification process..*

**Keywords:** *salak identification, SSD-MobileNet, detection algorithm, accuracy*



## DAFTAR ISI

<b>PERSETUJUAN</b>	<b>i</b>
<b>PERNYATAAN</b>	<b>ii</b>
<b>UCAPAN TERIMA KASIH</b>	<b>iii</b>
<b>ABSTRAK</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR TABEL</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>x</b>
<b>BAB 1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	3
1.7 Sistematika Penulisan	4
<b>BAB 2 LANDASAN TEORI</b>	<b>6</b>
2.1 Salak	6
2.2 Jenis Salak	7
2.2.1 Salak Pondoh	7
2.2.2 Salak Sidempuan	8
2.2.3 Salak Bali	8
2.3 Pengolahan Citra	9
2.3.1 Google Colabotatory	10
2.3.2 Tensorflow	11
2.3.3 Deteksi Objek	11
2.3.4 MobileNet	12
2.3.5 SSD (Single Shot Multibox Detector)	13
2.3.6 SSD-MobileNet	13

2.4	Citra	13
2.4.1	RGB Image	14
2.4.2	Grayscale Image	14
2.4.3	Binary Image	15
2.5	Penelitian Terdahulu	15
<b>BAB 3 ANALISIS DAN PERANCANGAN SISTEM</b>		<b>21</b>
3.1	Data yang Digunakan	21
3.2	Arsitektur Umum	22
3.2.1	Image Acquisition	23
3.2.2	Image Preprocessing	24
3.2.3	Image Identification	26
3.2.4	Learned Model	28
3.2.5	TF Lite Model	28
3.2.6	Proses Training	29
3.2.7	Proses Testing	31
3.2.8	Output	31
3.3	Flowchart Sistem	31
3.4	Perancangan Antarmuka	32
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM</b>		<b>35</b>
4.1	Implementasi Sistem	35
4.1.1	Perangkat Keras dan Perangkat Lunak	35
4.1.2	Implementasi Data	35
4.1.3	Implementasi Antarmuka	36
4.2	Prosedur Operasional	38
4.3	Pengujian Sistem	38
<b>BAB 5 PENUTUP</b>		<b>46</b>
5.1	Kesimpulan	46
5.2	Saran	46
<b>DAFTAR PUSTAKA</b>		<b>47</b>

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	17
<b>Tabel 3. 1</b> Pembagian Dataset.....	21
<b>Tabel 4. 1</b> Hasil Pengujian .....	39
<b>Tabel 4. 2</b> Confusion Matrix .....	44
<b>Tabel 4. 3</b> Nilai TP, FP dan FN dari Salak .....	44
<b>Tabel 4. 4</b> Nilai Precission, Recall dan F1-Score .....	45

## DAFTAR GAMBAR

Gambar 2. 1 Salak Pondoh	7
Gambar 2. 2 Salak Sidempuan	8
Gambar 2. 3 Salak Bali	9
Gambar 2. 4 Posisi Kordinat Citra	10
Gambar 2. 5 Arsitektur MobileNet	12
Gambar 2. 6 RGB image	14
Gambar 2. 7 Grayscale image	15
Gambar 2. 8 Binary Image	15
<b>Gambar 3. 1</b> Data Salak .....	21
<b>Gambar 3. 2</b> Arsitektur Umum.....	23
<b>Gambar 3. 3</b> Data .xml setelah diconvert ke dalam tabel .csv .....	26
<b>Gambar 3. 4</b> Flowchart Sistem.....	32
<b>Gambar 3. 5</b> Rancangan Halaman Dashboard .....	33
<b>Gambar 3. 6</b> Rancangan Halaman Identifikasi .....	34

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Indonesia merupakan negara dengan keanekaragaman jenis tertinggi di dunia karena terletak di daerah katulistiwa yang mempunyai tipe hutan hujan tropik cukup unik. Kekayaan jenis tumbuhan di hutan Indonesia sampai sekarang paling tidak terdapat 30.000 jenis tumbuhan yang sebagian besar masih tumbuh liar di hutan-hutan. Saat ini baru sekitar 4.000 jenis saja yang diketahui telah dimanfaatkan langsung oleh penduduk dan hanya sekitar seperempatnya yang telah dibudidayakan bahkan mungkin kurang dari 10 persennya (Williams, *et al*, 1975). Sehingga tidak kurang dari 329 jenis buah-buahan (terdiri dari 61 suku dan 148 marga) baik yang merupakan jenis asli Indonesia maupun pendatang (introduksi) dapat ditemukan di Indonesia (Rifai, 1986). Keanekaragaman jenis buah di Indonesia dapat dibedakan berdasarkan rasanya yang manis, asam, sepat, maupun pahit. Bentuknya yang bulat maupun lonjong, ukurannya yang kecil maupun besar, tekstur kulit luarnya yang mulus, berlekuk, maupun berduri, bahkan warnanya yang hijau, kuning, maupun merah (Hijang, 2014).

Salah satu buah yang banyak ditemui di Indonesia adalah salak (*Salacca zalacca*). Daerah asal tanaman salak tidak jelas diketahui secara pasti, tetapi diperkirakan berasal dari Indonesia, Thailand, dan Malaysia. Ada juga yang mengatakan bahwa tanaman salak adalah tanaman asli Indonesia yang berasal dari Pulau Jawa. Di Indonesia, bercocok tanam salak sudah dikenal sejak zaman kolonial Belanda. Tanaman salak banyak memiliki varietas yang diantaranya memiliki sifat unggul baik dari segi rasa maupun penampilan buahnya (Mandiri, 2010).

Salak mempunyai nilai ekonomis dan peluang pasar yang cukup luas baik di dalam negeri maupun ekspor. Varietas salak dibedakan berdasarkan tekstur daging buah, warna kulit buah, tekstur kulit, besar buah, aroma dan rasa daging buah, serta habitus (Harahap, 2013).

Pengolahan hasil perkebunan salak pada saat ini sudah berkembang pesat dikarenakan sudah banyak industri yang menggunakan teknologi canggih dalam tiap proses pengolahannya. Salah satu tahap pengolahan pada hasil perkebunan salak ialah pengelompokan hasil panen salak berdasarkan jenisnya. Dalam pengelompokan jenis buah salak ditentukan oleh beberapa parameter, diantaranya berat, ukuran, ciri, warna dan masih banyak lagi. Pengelompokan jenis buah salak dari sisi bentuk dan warna kulit buah merupakan faktor penting dalam proses pengidentifikasian.

Pada tahap mengidentifikasi jenis buah salak secara manual oleh mata manusia memiliki persepsi yang cenderung subyektif dikarenakan faktor komposisi warna. Maka dari itu, dibutuhkan suatu alat dengan sebuah sistem yang dapat melakukan pemilihan buah salak berdasarkan jenisnya. *SSD-MobileNet* merupakan gabungan antara model *MobileNet* yang digunakan sebagai ekstraksi fitur dan model SSD akan digunakan sebagai deteksi objek pada citra. *MobileNet* merupakan model yang berbasis arsitektur *streamlined* yang menggunakan *depthwise separable convolutions* dalam membangun *deep neural network*. *Depthwise seperable convolution* terdiri dari dua layer yaitu *depthwise convolution* yang berfungsi untuk menerapkan filter pada citra input, dan *pointwise convolution* digunakan untuk menggabungkan hasil *output* dari *depthwise convolution* sehingga menghasilkan beban komputasi lebih ringan (Howard et al. 2017).

Berdasarkan latar belakang serta pertimbangan menggunakan metode *SSD-MobileNet* maka penulis mengajukan judul dalam penelitian ini “**Identifikasi Jenis Buah Salak Menggunakan SSD-MobileNet**”.

## 1.2 Rumusan Masalah

Salak merupakan salah satu jenis komoditas buah yang sangat mudah dan sering dijumpai di Indonesia. Dalam mengidentifikasi jenis buah salak secara manual dan *real time* memiliki persepsi yang cenderung subyektif dikarenakan faktor komposisi warna dan bentuk. Maka dari itu, dibutuhkan suatu sistem pada alat dalam menentukan jenis buah salak secara efisien dan cepat menggunakan *SSD-Mobilnet* yang berdasarkan warna dan bentuk.

### 1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini antara lain :

1. Jenis buah salak yang digunakan pada penelitian ini adalah salak pondoh, salak sidempuan dan salak bali.
2. Sistem hanya memproses citra digital (gambar) yang diambil menggunakan *smartphone* secara *real time*.
3. Gambar yang digunakan untuk *labeling* berformat *JPG*.

### 1.4 Tujuan Penelitian

Tujuan pada penelitian ini berguna untuk identifikasi jenis buah salak dengan menggunakan algoritma *SSD-MobileNet*.

### 1.5 Manfaat Penelitian

Adapun manfaat pada penelitian ini yaitu:

1. Memberikan informasi kepada masyarakat tentang bagaimana mengenali jenis-jenis buah salak yang ditemukan di indonesia.
2. Mengetahui hasil identifikasi jenis-jenis buah salak dengan menggunakan *SSD-MobileNet*.
3. Dapat menjadi referensi pada penelitian bidang *Image Processing*, khususnya dalam penerapan.

### 1.6 Metode Penelitian

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

1. Studi Pustaka dan Literatur

Pada tahap studi literatur ini dilakukan dengan mengumpulkan data citra dari buah salak dan mempelajari referensi tentang *SSD-MobileNet*, ciri-ciri dari tiap jenis buah salak yang diperoleh dengan mengumpulkan bahan dari jurnal, skripsi, artikel, buku, panduan dan sumber referensi lainnya untuk

proses pengumpulan bahan.

## 2. Analisis

Pada tahap ini dilakukan analisis terhadap tahapan yang ada dilakukan sebelumnya untuk mendapatkan penelitian dari berbagai informasi yang terkait dari penelitian beberapa sumber yang didapatkan.

## 3. Perancangan Sistem

Setelah dilakukan analisis, dilakukan tahap selanjutnya yakni perancangan sistem. Tahap ini merupakan tahap yang dilakukan untuk perancangan arsitektur umum dan *user interface* penelitian.

## 4. Implementasi

Tahap dimana dilakukan implementasi dan penerapan dari sistem yang telah dibangun.

## 5. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun apakah sistem berjalan sesuai yang diinginkan atau tidak.

## 6. Dokumentasi

Merupakan tahap akhir dari seluruh tahap penelitian. Tahap ini dilakukan untuk penyusunan laporan serta mengambil kesimpulan dari penelitian yang telah dilakukan.

# 1.7 Sistematika Penulisan

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

## Bab 1: Pendahuluan

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

## Bab 2: Landasan Teori

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



**Bab 3: Analisis dan Perancangan**

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

**Bab 4: Implementasi dan Pengujian**

Pada bab ini, akan dijelaskan implementasi dan pembahasan dari rancangan aplikasi dari penelitian yang telah dibuat pada bab 3. Hasil dari pengujian aplikasi dan implementasi juga akan dijabarkan pada bab ini.

**Bab 5: Kesimpulan dan Saran**

Pada bab ini, berisi kesimpulan dan ringkasan dari rancangan yang dibahas pada bab 3 dan hasil penelitian yang dijelaskan pada bab 4. Pada bab ini juga dimuat saran–saran untuk pengembangan penelitian yang selanjutnya.

## **BAB 2**

### **LANDASAN TEORI**

Pada bab ini membahas tentang teori-teori yang berkaitan dalam memahami permasalahan-permasalahan yang diangkat dalam penelitian ini. Teori-teori yang terdapat pada bab ini adalah pembahasan tentang jenis salak, Pengolahan citra, *MobileNet* dan penelitian-penelitian terdahulu sebagai pendukung penelitian.

#### **2.1 Salak**

Salak (*Salacca zalacca*) merupakan tanaman buah asli dari Indonesia. Buah ini tumbuh subur di daerah tropis. Tanaman ini termasuk dalam keluarga Palmae yang diduga dari Pulau Jawa. Ternyata tidak hanya di Indonesia, salak juga dapat tumbuh dan menyebar di Malaysia, Filipina, Brunei, dan Thailand (Widyastuti, 1996).

Tanaman salak ini tumbuh secara berumpun dan tinggi tanamannya dapat mencapai 7 m, tetapi rata-rata yang tumbuh tidak lebih dari 4,5 m. Tanaman ini merupakan tanaman berumah dua yang dapat menghasilkan bunga jantan terpisah dengan tanaman yang menghasilkan bunga betina. Batang berduri hampir tidak terlihat karena tertutup oleh pelepah daun yang tumbuh rapat. Daun tersusun berbentuk roset seperti pedang dengan panjang antara 2,5 – 7 m. Bunga jantan dan bunga betina merupakan bunga majemuk yang masing-masing tersusun dalam bunga tongkol. Buah tersusun dalam tandan yang masing-masing muncul dari ketiak daunnya. Buah yang dihasilkan biasanya berbentuk bulat atau bulat telur terbalik dengan bagian pangkal meruncing. Tiap buah salak terdiri dari 3 septa daging buah. Rasanya bervariasi, ada yang manis, asam, sepat atau kombinasi dari ketiganya (Widyastuti, 1996).

Tanaman salak dapat tumbuh hampir di seluruh daerah di Indonesia. Akan tetapi, untuk dapat tumbuh dengan produktif tanaman ini membutuhkan lingkungan yang ideal. Ketinggian tempat yang diinginkan berkisar antara 1 – 400 m di atas permukaan laut dengan curah hujan rata-rata 200 – 400 mm /bulan. Suhu udara harian daerah antara 20° – 30°C dan terkena sinar matahari antara 50 – 70% menjadi

tempat yang baik untuk pertumbuhannya. Jenis tanah yang ideal adalah tanah yang gembur, mengandung bahan organik, dengan air tanah yang dangkal, dan mampu menyimpan air tetapi tidak mudah tergenang (Widyastuti, 1996).

Salak yang sudah mencapai umur 6 – 7 bulan umumnya sudah dapat dipanen sejak hari penyerbukan. Buah yang dipetik pada umur tersebut sudah masak, rasanya manis, beraroma salak dan masir. Cara pemanenan buah salak biasanya dilakukan dengan memotong tangkai tandannya menggunakan sabit. Buah salak dalam satu tandan memiliki kematangan yang tidak seragam, maka dari itu dilakukan petik pilih dari tandannya (Mandiri, 2010). Buah salak yang sudah matang ditandai dengan sisik yang jarang, warna kulit buah merah kehitaman atau kuning tua dan bulu-bulunya telah hilang. Ujung kulit buah (bagian buah yang meruncing) bila ditekan terasa lunak, warnanya mengkilat dan mudah terlepas bila dipetik dari tandannya (Mandiri, 2010).

## **2.2 Jenis Salak**

### **2.2.1 Salak Pondoh**

Salak pondoh tersusun dari tiga bagian utama, yaitu kulit, daging buah dan bagian biji. Bagian kulit terdiri dari sisik-sisik yang tersusun seperti genting dan kulit ari yang langsung menyelimuti daging buah dengan warna putih transparan.



**Gambar 2. 1** Salak Pondoh

### 2.2.2 Salak Sidempuan

Salak Sidempuan adalah salah satu varietas buah salak yang berasal dari daerah Sidempuan di Sumatera Utara, Indonesia. Buah ini terkenal dengan cita rasanya yang manis dan sedikit asam, serta teksturnya yang renyah. Kulit Salak Sidempuan memiliki sisik yang halus dan berwarna coklat kemerahan. Daging buahnya berwarna putih kekuningan, tebal, dan renyah, dengan rasa dominan manis dan sedikit sentuhan asam. Selain itu, Salak Sidempuan dikenal karena tidak terlalu banyak serat, sehingga lebih nyaman dikonsumsi. Buahnya juga mudah dipisahkan dari bijinya, menambah kenyamanan saat dimakan.



**Gambar 2. 2** Salak Sidempuan

### 2.2.3 Salak Bali

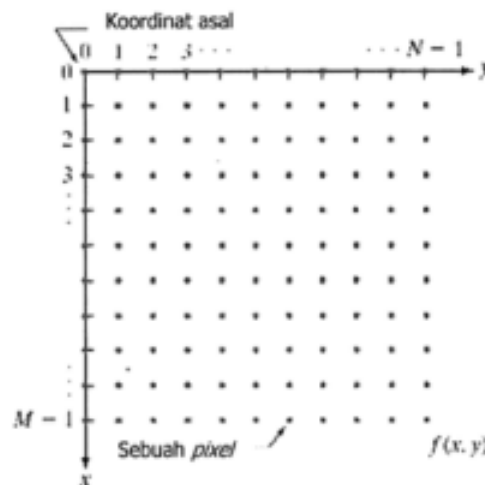
Salak Bali adalah salah satu varietas buah salak yang terkenal dan berasal dari Pulau Bali, Indonesia. Buah ini sangat diminati karena rasa dan teksturnya yang khas. Kulit Salak Bali memiliki sisik yang lebih besar dan kasar dibandingkan dengan varietas salak lainnya, dengan warna coklat tua hingga hitam kemerahan. Daging buah Salak Bali berwarna putih gading hingga kekuningan, tebal, dan renyah. Rasanya dominan manis dengan sedikit rasa asam, memberikan sensasi segar saat dikonsumsi. Salak Bali juga memiliki aroma yang khas dan harum, yang menjadi daya tarik tersendiri. Biji dari Salak Bali relatif lebih kecil, sehingga daging buah yang bisa dimakan lebih banyak.



**Gambar 2. 3** Salak Bali

### **2.3 Pengolahan Citra**

Pengolahan citra (*Image Processing*) merupakan ilmu yang melakukan olahan citra berdasarkan *input* maupun *output*. Bisa saja *output* dari pengolahan citra berupa citra atau sekumpulan karakteristik atau parameter yang berhubungan dengan citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan (Avif, 2019).



**Gambar 2. 4** Posisi Kordinat Citra (Avif,2019)

### 2.3.1 Google Colabulatory

*Google Colaboratory* merupakan sebuah alat yang mirip dan berdasarkan dari *Jupyter Notebooks*. *Jupyter* adalah alat *open source* berbasis browser yang mengintegrasikan bahasa pemrograman, *library*, dan alat visualisasi. *Jupyter Notebook* berfungsi baik secara lokal maupun di *cloud*. Setiap dokumen terdiri dari beberapa sel, di mana setiap sel berisi bahasa skrip atau kode *markdown*, dan *output*-nya disematkan dalam dokumen. Tipe *output* termasuk teks, tabel dan grafik. *Google Colaboratory (Colab)* adalah proyek yang bertujuan untuk menyebarluaskan pendidikan dan penelitian pembelajaran mesin. *Notebook Colab* didasarkan pada *Jupyter* dan digunakan sebagai objek *Google Documents*: 17 notebook dapat dibagikan dan pengguna dapat berkolaborasi pada notebook yang sama.

Kolaborasi menyediakan runtime *Python 2* dan *3* yang telah dikonfigurasi sebelumnya dengan pembelajaran mesin dan *library AI*, seperti *TensorFlow*, *Matplotlib*, dan *Keras*. Mesin virtual di bawah runtime (VM) dinonaktifkan setelah jangka waktu tertentu, dan semua data dan konfigurasi pengguna hilang. Namun, *notebook* memungkinkan untuk mentransfer *file* dari *hard disk* VM ke akun *Google Drive* pengguna. Layanan *Google* ini menyediakan akselerasi *GPU runtime*, juga sepenuhnya dikonfigurasi dengan perangkat lunak yang sebelumnya diuraikan. Infrastruktur *Google Colaboratory* di-host di *platform Google Cloud*.

### 2.3.2 Tensorflow

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

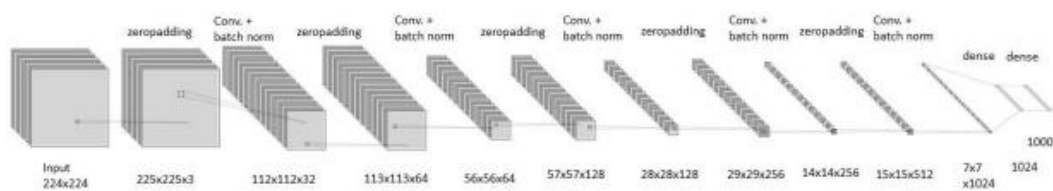
1. *Support* semua bahasa pemrograman populer seperti *Python, C++, Java, R* dan *Go*.
2. *TensorFlow* memungkinkan kemudahan dan penerapan model.
3. *TensorFlow* memiliki dukungan komunitas yang baik.

### 2.3.3 Deteksi Objek

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

### 2.3.4 MobileNet

*MobileNet* merupakan salah satu arsitektur dari *convolutional neural network* (CNN). Para peneliti dari google membuat arsitektur tersebut untuk kebutuhan *mobile*. Perbedaan mendasar dari *MobileNet* dengan arsitektur CNN lainnya ialah pada penggunaan *layer* konvolusi dengan ketebalan filter yang sesuai dengan input gambar. *MobileNet* juga digunakan untuk konvolusi *depthwise* dan *pointwise* (Rahman, 2020).



**Gambar 2. 5** Arsitektur MobileNet (Rahman, 2020)

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

*Depthwise convolution* menerapkan *single filter* untuk setiap masukan pada *channel*, pada *pointwise convolution* menerapkan *1x1 convolution* untuk menggabungkan *output* dari *depthwise convolution* dan pada *standart convolution* menggabungkan dan mem-filter *input* menjadi satu *set output* baru dalam satu langkah. *Depthwise separable convolution* membagi ini menjadi 2 *layer*, yaitu lapisan pem-filter-an dan lapisan penggabungan yang digunakan untuk mengurangi komputasi dan ukuran model secara drastis (Howard et al, 2017).



### 2.3.5 SSD (Single Shot Multibox Detector)

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

### 2.3.6 SSD-MobileNet

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

## 2.4 Citra

Citra adalah representasi atau kesan terlihat yang diperoleh oleh kamera, teleskop, mikroskop, atau perangkat lain, atau yang ditampilkan pada layar. Gambar digital adalah representasi gambar nyata yang terdiri dari sekumpulan angka yang dapat disimpan dan ditangani oleh komputer digital. Gambar digital terdiri dari area kecil atau dikenal sebagai piksel (elemen gambar). Di setiap piksel terdiri dari angka yang menggambarkan *property* seperti kecerahan atau warna. Gambar digital

memiliki beberapa karakteristik salah satunya gambar berwarna. Gambar berwarna dapat memiliki tiga warna yaitu *RGB* (*Red, Green, Blue*) yang biasanya digunakan dalam *monitor* atau pemindai komputer, atau empat warna yaitu *CYMK* (*Cyan, Magenta, Yellow, Black*) yang biasanya digunakan dalam *printer* berwarna. Citra digital dibagi menjadi 3 jenis yaitu *RGB image*, *grayscale image* dan *binary image*.

#### 2.4.1 RGB Image

*RGB Image* adalah citra yang terdiri dari Merah (*Red*), Hijau (*Green*) dan Biru (*Blue*) sebagai komponen warna utamanya. Citra digital *RGB* memiliki penyimpanan 24 bit dimana masing-masing komponen warna menampung 8 bit sehingga dapat mengekspresikan 16 juta lebih variasi warna. Contoh *RGB image* dapat dilihat pada gambar 2.6 (Kumar, 2010).



**Gambar 2. 6** RGB image (Kumar,2010)

#### 2.4.2 Grayscale Image

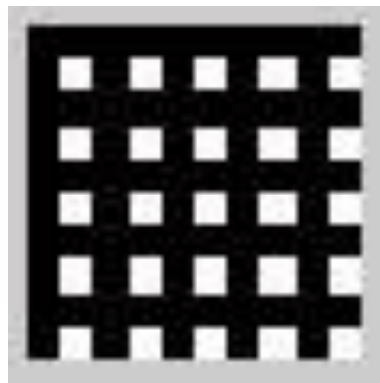
*Grayscale Image* adalah citra skala abu-abu dimana setiap pikselnya memiliki warna gradasi mulai dari putih sampai hitam sehingga citra *grayscale* sangat cocok untuk pengolahan *file* gambar. Pada umumnya citra *grayscale* memiliki penyimpanan 8 bit sehingga dapat menampung 256 tingkatan skala abu-abu dimana tiap piksel memiliki intensitas 0 hingga 255 dengan 0 menjadi hitam dan 255 menjadi putih (Kusumanto,2016).



**Gambar 2. 7** Grayscale image (Kusumanto,2016)

#### 2.4.3 Binary Image

*Binary Image* adalah citra yang hanya memiliki dua warna yaitu hitam dan putih sehingga nilai intensitasnya hanya memiliki 2 nilai yaitu 0 dan 255 (Kusumanto, 2016).



**Gambar 2. 8** Binary Image (Kusumanto,2016)

### 2.5 Penelitian Terdahulu

Nazil Ilham Burhanudin *et.al* (2023) menjelaskan bahwa penggunaan model pra-terlatih SSD dengan arsitektur *MobileNet* untuk mengenali objek dalam aplikasi kasir otomatis, khususnya fokus pada tiga kelas produk soft drink: *Coca-Cola*, *Floridina*, dan *Good Day*. Pengambilan data dilakukan menggunakan kamera *smartphone*, dengan total 203 data yang dikumpulkan. Hasil penelitian menunjukkan bahwa proses pengenalan objek produk mencapai akurasi sebesar 82,9%, dengan presisi 97,5% dan *recall* 84,7%. Waktu yang dibutuhkan untuk

proses pengenalan produk berkisar antara 365 ms hingga 827 ms, dengan rata-rata 695 ms (0,69 detik).

Yusuf Amrozi *et.al* (2022) yang menggunakan algoritma *Support Vector Machine (SVM)* untuk mengklasifikasi jenis pisang berdasarkan citra warna, dengan fokus pada Pisang Ambon dan Pisang Lady Finger. Data yang terdiri dari 1256 citra pisang. Hasil penelitian ditampilkan dalam confusion matrix dengan nilai *True Positive (TP)* sebesar 0,82 dan *False Positive (FP)* sebesar 0,18, serta *False Negative (FN)* sebesar 0,02 dan *True Negative (TN)* sebesar 0,98. Hasil tersebut menunjukkan bahwa algoritma *SVM* cukup efektif dalam mengklasifikasi jenis pisang, memberikan jaminan mutu atas produk yang dihasilkan oleh petani.

Nikotesa Eko Rianto Pah (2021) bertujuan untuk mengklasifikasi jenis apel merah berdasarkan ciri bentuk dan warna, mengatasi kesulitan dalam membedakan jenis-jenis apel merah yang mirip secara visual. Data citra yang digunakan adalah data sekunder dengan ukuran 100 x 100 piksel dalam format *\*JPG*. Metode ekstraksi fitur mencakup ekstraksi ciri warna *Mean HSV* (dengan nilai output 3) dan ciri bentuk *Moment Invariant* (dengan nilai output 7), sehingga setiap citra direpresentasikan dengan 10 nilai. Klasifikasi dilakukan menggunakan *Euclidean Distance*, dengan skenario pengujian menggunakan *K-Fold Cross Validation*, membagi 1.710 data citra menjadi 10 subset. Dari 50 kali pengujian, rata-rata akurasi mencapai 98,82%, dengan akurasi tertinggi sebesar 99,12% pada pengujian ke-46, dan akurasi terendah sebesar 98,54% pada pengujian ke-48.

Muhammad Fadli (2021) menggunakan teknologi *Computer Vision* dan *Artificial Intelligence* untuk mengklasifikasikan jenis kucing secara *real-time* dengan model *SSD-MobileNet* pada *platform desktop*. Kucing yang diklasifikasikan termasuk Sphynx, Himalaya, Bengal, Persia dan Moggy, dengan uji coba dilakukan pada jarak tidak kurang dari 40 cm dari kamera dan dalam posisi yang berbeda-beda untuk setiap jenisnya. Hasil penelitian menunjukkan bahwa sistem mencapai akurasi sebesar 93.8%, menandakan kemampuan dalam mengenali dan mengklasifikasikan jenis kucing dengan baik menggunakan teknologi *Computer Vision* dan *Artificial Intelligence*.

Evy Kamilah Ratnasari *et.al*(2016) mengeksplorasi pengenalan buah melalui

citra *RGB* menggunakan fitur warna dan tekstur sebagai alternatif yang lebih handal dan efektif daripada menggunakan fitur bentuk dan ukuran. Metode pengenalan ini didasarkan pada *K-Nearest Neighbor* berdasarkan fitur warna dan tekstur *co-occurrence*. Dengan *dataset* mencakup 1882 citra buah untuk 12 kelas yang berbeda, hasil eksperimen menunjukkan kemampuan yang baik dalam mengenali buah, dengan akurasi tertinggi mencapai 92%. Hal ini menunjukkan potensi penggunaan fitur warna dan tekstur sebagai pendekatan yang dapat diandalkan dalam pengenalan buah melalui visi komputer.

Yulia Saraswati (2010) mengembangkan perangkat lunak untuk mengklasifikasikan tomat secara otomatis berdasarkan bentuk, ukuran, dan kematangan warna kulitnya. Sistem menggunakan dua *webcam* yang ditempatkan pada sudut 90° terhadap tomat, dengan jarak 20 cm dari objek. Pengujian dilakukan menggunakan pencahayaan lampu dan sinar matahari, dengan pengolahan citra digital untuk menghitung diameter, luas piksel, dan luas piksel warna matang tomat. Evaluasi sistem dilakukan menggunakan *Mean Opinion Score (MOS)* dengan akurasi tertinggi untuk klasifikasi bentuk (86,17%), ukuran (84,04%), dan kematangan (80,85%). Akurasi keseluruhan mencapai 54,26%. Penelitian ini diharapkan dapat membantu petani dalam menentukan kualitas tomat yang diproduksi.

**Tabel 2. 1** Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Keterangan
1	Nazil Ilham Burhanudi n, Arif Dwi Laksito, Acihmah, Sidauruk,	<i>Object Recognition with SSD MobileNet Pre- Trained Model in The Cashier Application</i>	Penelitian menggunakan model pra-terlatih <i>SSD MobileNet</i> untuk mengenali objek pada aplikasi kasir otomatis, khususnya produk <i>softdrink</i> seperti <i>coca-cola</i> , <i>floridina</i> , dan <i>good day</i> . Data diambil dari kamera <i>smartphone</i> dengan total 203 data. Hasilnya

	Muhamma d Resa, Arif Yudianto, Alfie Nur Rahmi (2023)		menunjukkan akurasi pengenalan objek sebesar 82,9%, dengan presisi 97,5% dan <i>recall</i> 84,7%.
2	Yusuf Amrozi, Dian Yulianti, Agung Susilo, Nur Novianto, Rizka Ramadhan (2022)	Klasifikasi Jenis Buah Pisang Berdasarkan Citra Warna dengan Metode SVM	Penelitian ini menggunakan algoritma <i>Support Vector Machine (SVM)</i> untuk mengklasifikasi jenis pisang berdasarkan citra warna, dengan fokus pada Pisang Ambon dan Pisang Lady Finger. Data yang digunakan mencakup 1256 citra pisang. Hasil penelitian menunjukkan <i>confusion matrix</i> dengan nilai <i>True Positive (TP)</i> = 0,82, <i>False Positive (FP)</i> = 0,18, <i>False Negative (FN)</i> = 0,02, dan <i>True Negative (TN)</i> = 0,98.
3	Nikotesa Eko Rianto Pah (2021)	Pengolahan Cita Digital untuk Klasifikasi Jenis Buah Apel Menggunakan Ciri Warna HSV dan Ciri Bentuk Moment Invariant	Penelitian ini menggunakan metode klasifikasi untuk membedakan jenis-jenis apel merah berdasarkan ciri bentuk dan warna, dengan data citra sekunder berukuran 100 x 100 piksel. Klasifikasi menggunakan <i>Euclidean Distance</i> dan <i>K-Fold Cross Validation</i> pada 1.710 data citra. Hasilnya menunjukkan rata-rata akurasi sebesar 98,82%,

		dengan nilai tertinggi 99,12% dan terendah 98,54%.
4	Muhammadd Fadli (2021)	<p>Klasifikasi Jenis Ras Kucing Menggunakan <i>SSD-MobileNet</i> Secara Real-Time Berbasis Desktop</p> <p>Penelitian ini menggunakan teknologi <i>Computer Vision</i> dan <i>Artificial Intelligence</i> untuk mengklasifikasikan jenis kucing secara <i>real-time</i> dengan model <i>SSD-MobileNet</i> pada <i>platform desktop</i>. Kucing yang diklasifikasikan termasuk Sphynx, Himalaya, Bengal, Persia dan Moggy, dengan akurasi sistem mencapai 93.8%.</p>
5	Evy Kamilah Ratnasari dan Anggit Wikaningrum (2016)	<p>Pengenalan Jenis Buah Pada Citra Menggunakan Pendekatan Klasifikasi Berdasarkan Fitur Warna Lab dan Tekstur Co-Occurrence</p> <p>Penelitian ini menggunakan metode pengenalan buah berdasarkan fitur warna dan tekstur, mengatasi keterbatasan pengenalan menggunakan fitur bentuk dan ukuran pada citra RGB. Klasifikasi dilakukan dengan <i>K-Nearest Neighbor</i>. Eksperimen pada dataset 1882 citra buah menunjukkan akurasi tertinggi 92%, menandakan potensi metode ini dalam berbagai aplikasi.</p>
6	Yulia Saraswati (2010)	<p>Sistem Klasifikasi Jenis Buah Tomat Berdasarkan Bentuk dan Ukuran serta Warna Permukaan</p> <p>Penelitian ini menghasilkan perangkat lunak untuk mengklasifikasikan tomat secara otomatis berdasarkan bentuk, ukuran, dan kematangan warna</p>

---

Kulit Buah Berbasis Pengolahan Citra Digital	kulitnya. Dengan menggunakan dua <i>webcam</i> , sistem dapat mencapai akurasi tinggi dalam klasifikasi bentuk (86,17%), ukuran (84,04%), dan kematangan (80,85%) tomat. Ini diharapkan dapat membantu petani meningkatkan kualitas produksi tomat mereka.
--	--

---



## BAB 3

### ANALISIS DAN PERANCANGAN SISTEM

Pada bab analisis dan perancangan berisi tentang data yang digunakan, teknik yang digunakan untuk identifikasi jenis salak, perancangan sistem, arsitektur umum, alur kerja sistem dalam bentuk *use case diagram* dan *user interface*.

#### 3.1 Data yang Digunakan

Dalam penelitian ini citra yang dikumpulkan melalui *Kaggle* dan belum di *labeling* untuk datanya serta diambil secara langsung. Citra diambil dalam format *JPG* dan pengaturan orientasi *portrait* maupun *landscape*. Berikut adalah contoh citra dari jenis salak yang akan melalui proses identifikasi pada Gambar 3.1.



**Gambar 3. 1** Data Salak

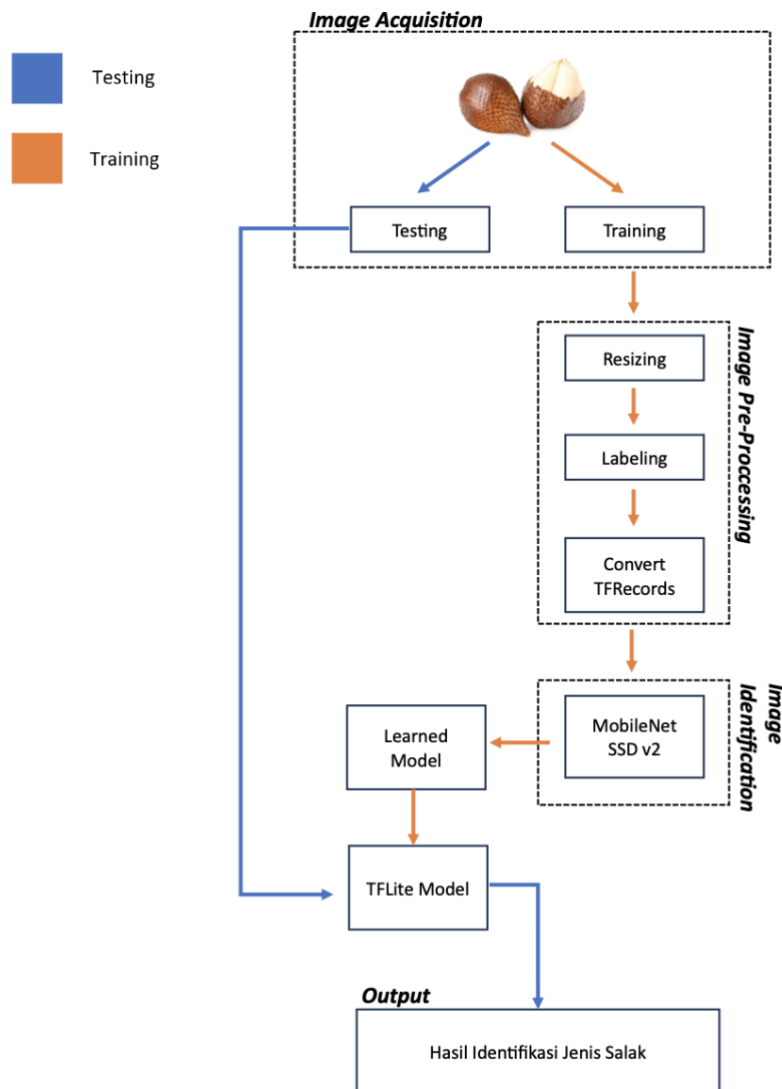
Keseluruhan data yang diperoleh berjumlah 600 citra kemudian digabungkan menjadi *dataset* citra jenis salak. Data citra salak pondoh berjumlah 200 citra, salak sidempuan berjumlah 200 citra, dan salak bali berjumlah 200 citra. Dalam pengujian identifikasi, total seluruh data dibagi secara acak menjadi dua *dataset* berbeda dan dilabeli sebagai *dataset* pelatihan dan *dataset* pengujian. Bobot nilai data latih dengan data uji 80:20. Pembagian *testing dataset* dan *training dataset* untuk identifikasi jenis buah salak terlihat di Tabel 3.1.

**Tabel 3. 1** Pembagian Dataset

No.	Kategori	Data Latih	Data Uji	Jumlah Data
1.	Salak Pondoh	160	40	200
2.	Salak Sidempuan	160	40	200
3.	Salak Bali	160	40	200
<b>Total</b>		<b>480</b>	<b>120</b>	<b>600</b>

### 3.2 Arsitektur Umum

Di dalam penelitian ini, dilakukan beberapa tahapan. Tahapan yang pertama mengumpulkan citra salak yang akan digunakan sebagai data latih. Setelah itu tahap *pre-processing* yaitu *resizing* yang merupakan proses mengubah ukuran citra data menjadi 640 x 640, kemudian data yang telah melewati tahap *resizing* selanjutnya akan melalui tahap *labeling*, yaitu sebuah proses untuk melakukan anotasi atau identifikasi terhadap unsur penting di dalam sebuah citra, proses ini akan membuat citra mampu untuk dibaca oleh sistem yang akan dibangun. Selanjutnya citra yang sudah melalui proses *labeling* akan dilakukan ekstraksi fitur dan identifikasi menggunakan metode *SSD-MobileNet*. Setelah model berhasil terbentuk, model yang ada kemudian akan di-*convert* ke dalam bentuk *TFLite Model* untuk kemudian dapat dimasukkan ke dalam sistem yang akan dibuat. Setelah tahap-tahap tersebut dilakukan, maka citra salak telah berhasil diidentifikasi dan menghasilkan *output* berupa jenis salak. Adapun tahapan yang telah disebutkan terlihat rinciannya dalam bentuk arsitektur umum pada Gambar 3.2.



**Gambar 3. 2** Arsitektur Umum

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

### 3.2.1 Image Acquisition

Tahap ini adalah tahap dimana dilakukan pengambilan data citra salak yang akan digunakan sebagai *input* untuk sistem yang akan dibangun. Citra salak diambil dari *kaggle*. Citra berekstensi *JPG* dengan ukuran 640 x 640 piksel.

### 3.2.2 Image Preprocessing

Berikutnya, citra salak yang sudah dibagi ke dalam data *training* akan masuk ke dalam tahap *image preprocessing*, dimana disini citra salak akan melalui tiga proses. Tahapannya adalah sebagai berikut.

#### 3.2.2.1 Resizing

Proses pertama yang akan dilakukan pada tahap *preprocessing* adalah *resizing*. Pada tahap ini, dilakukan proses sederhana mengubah ukuran citra salak menjadi berukuran 640 x 640 piksel. Hal ini dilakukan agar citra yang ada dapat diinput ke dalam sistem dan mengurangi besaran data supaya sistem tidak terlalu berat. Proses dari *resizing* ini dapat dilihat pada *pseudocode* dibawah ini :

```

Mulai
  Definisikan path folder untuk citra asli
  Definisikan path folder untuk menyimpan citra yang telah di-resize
  Definisikan ukuran pixel baru

  Fungsi resize_image(image_path, output_path, target_size):
    Baca citra dari image_path
    Ubah ukuran citra menjadi target_size
    Simpan citra yang telah di-resize ke output_path

  Untuk setiap file citra dalam folder citra asli:
    Panggil fungsi resize_image dengan parameter (path file citra asli, path
    folder output, ukuran pixel baru)

Selesai

```

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

ukuran *pixel* baru untuk setiap citra tersebut.

#### 3.2.2.2 Labeling

Tahapan selanjutnya setelah citra selesai melalui tahap *resizing* adalah *labeling*. Citra yang awalnya didapat dari *Kaggle* hanya berupa gambar tanpa ada *file xml* nya yang merupakan *label* pada tiap gambar. Pada tahap ini, dilakukan proses memberikan sebuah *label* kepada citra untuk mendapat objek khusus yang akan mampu dibaca oleh sistem. *Labeling* yang dilakukan pada citra dibuat menggunakan aplikasi *Labellmg* yang berfungsi untuk memberikan label pada citra salak berdasarkan kategori yang sudah ditentukan. *Output* yang dikeluarkan adalah citra salak yang sudah di-*label* akan tersimpan dalam bentuk *xml*. Informasi yang tersimpan dalam *file xml* tersebut berupa identifikasi dari objek khusus yang sudah ditandai beserta koordinat dari objek yang berguna pada proses selanjutnya.

#### 3.2.2.3 Convert TFRecord

Tahapan selanjutnya adalah *Convert TFRecord*. Pada tahap ini, data pelabelan berupa *xml* selanjutnya akan diubah ke dalam bentuk *File Tensorflow Record (TFRecord)*. Hal ini dilakukan akibat data yang ada cukup banyak dan mengakibatkan ukuran yang cukup besar. Melalui *TFRecord*, data yang ada akan disimpan ke dalam bentuk *string biner*. Sehingga akan berdampak terhadap kecepatan proses *training* yang akan dilakukan selanjutnya. Data *xml* dan citra berupa salak akan di-*convert* menjadi *file TFRecord*. Data yang sudah di-*convert* menjadi *file TFRecord* ini akan menggunakan lebih sedikit ruang pada *disk* sehingga proses yang akan dilakukan sistem akan membutuhkan waktu yang lebih sedikit pula.

filename	width	height	class	xmin	ymin
100_jpg.rf.684f188fz	640	640	Salak Pondoh	222	251
100_jpg.rf.684f188fz	640	640	Salak Pondoh	378	169
100_jpg.rf.684f188fz	640	640	Salak Pondoh	149	328
100_jpg.rf.684f188fz	640	640	Salak Pondoh	1	146
102_jpg.rf.7c652632	640	640	Salak Pondoh	88	36
103_jpg.rf.199b66fb	640	640	Salak Pondoh	19	346
103_jpg.rf.874009d4	640	640	Salak Pondoh	346	327
103_jpg.rf.8a339c42	640	640	Salak Pondoh	327	171
104_jpg.rf.d1bb5e5f	640	640	Salak Bali	109	115
104_jpg.rf.d86dff77c	640	640	Salak Bali	124	104
104_jpg.rf.f64b3d7e	640	640	Salak Bali	104	109
105_jpg.rf.06e5f5db	640	640	Salak Bali	266	239
105_jpg.rf.d3a511cf	640	640	Salak Bali	239	266
105_jpg.rf.f1b60e02	640	640	Salak Bali	269	239
106_jpg.rf.f3ca35d4	640	640	Salak Pondoh	127	144

**Gambar 3. 3** Data .xml setelah diconvert ke dalam tabel .csv

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

```

Mulai
  Definisikan path folder untuk data asli
  Definisikan path folder untuk menyimpan file TFRecord

  Fungsi create_tfrecord(data_path, output_path):
    Baca data dari data_path
    Buat file TFRecord baru di output_path
    Untuk setiap item data:
      Konversi item data menjadi Example Protocol Buffer (protobuf)
      Tulis Example protobuf ke file TFRecord

  Untuk setiap file data dalam folder data asli:
    Panggil fungsi create_tfrecord dengan parameter (path file data asli, path
    folder output)

Selesai

```

### 3.2.3 Image Identification

Selanjutnya pada tahap ini, data *training* yang sudah ada akan melalui tahap identifikasi jenis salak dengan menggunakan *SSD-MobileNet*. *SSD-MobileNet* terdiri dari *SSD* yang berperan sebagai *base model*, dan *MobileNet* sebagai *network*

*model*. *SSD* akan mengatur pendeteksian objek dengan *bounding box* sementara *MobileNet* akan bekerja untuk mengekstrak fitur yang nantinya akan diproses. Dua fitur utama dalam *SSD-MobileNet* adalah sebagai berikut:

### 3.2.3.1 Extract Feature Maps

*SSD* menggunakan *VGG16* untuk mengekstrak *feature maps* yang kemudian mendeteksi objek menggunakan *Conv4\_3 layer*. *SSD* akan memprediksi *anchor box offset* yang telah ditentukan sebelumnya untuk setiap lokasi *feature maps*. Setiap kotak akan memiliki ukuran dan posisi yang tetap pada sel. *Feature Maps* akan diskalakan ulang sehingga hanya bertanggung jawab untuk objek pada skala tertentu.

*SSD* menandai *area bounding box* yang diprediksi kepada *default bounding box* melalui berbagai skala dan rasio untuk setiap lokasi *feature map*. *SSD* membandingkan objek dengan *default bounding box* dengan berbagai rasio selama masa *training*. Setiap *default box* dengan  $\text{IoU} > 0.5$  dikategorikan cocok. Setiap sel atau lokasi yang ada, akan terbuat 4 objek identifikasi. Setiap identifikasi terdiri dari kotak batas sebagai kelas untuk objek yang dibatasi.

### 3.2.3.2 Convolution Filters

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

### 3.2.4 Learned Model

Setelah proses *training*, kita akan mendapatkan sebuah *file learned model* yang bernama *savedmodel* dengan ekstensi *.pb* dikarenakan menggunakan Tensorflow versi 2. Informasi yang ada di dalam file tersebut berupa sebuah grafik serta *weights* (bobot) model hasil *training*, dimana objek salak mempunyai 3 jenis yaitu Salak Pondoh, Salak Sidempuan dan Salak Bali. Hasil identifikasi data *training* yang ada akan menjadi *learned model*. *Learned model* tersebut merupakan sebuah data yang sudah dilatih untuk selanjutnya akan di *convert* ke dalam bentuk *Tensorflow Lite* agar bisa digunakan pada aplikasi *mobile* berbasis *android*.

### 3.2.5 TF Lite Model

Pada tahap sebelumnya, hasil *training* mengeluarkan *output* berupa *savedmodel* yang berekstensi *.pb*. Pada tahap ini, file tersebut akan diconvert ke dalam bentuk *tensorflow lite*. File diconvert ke dalam bentuk *tflite* dikarenakan sistem yang akan dibangun akan berbasis *android*. *Tensorflow Lite* menyediakan sebuah *framework* untuk model *tensorflow* yang sudah selesai dilatih untuk selanjutnya akan dikompresi lalu diterapkan pada perangkat *mobile*, hal ini menyebabkan file yang telah diconvert ke dalam bentuk *tflite* akan berubah menjadi ukuran *biner* yang lebih kecil untuk bisa masuk ke dalam perangkat *mobile*. Proses *convert* ke dalam bentuk *TFLite* dapat dilihat pada *code* dibawah ini:

```
Mulai
    Definisikan path file model TensorFlow asli
    Definisikan path untuk menyimpan file model TF Lite

    Fungsi convert_to_tflite(model_path, output_path):
        Load model TensorFlow dari model_path
        Buat converter TF Lite dari model TensorFlow
        Konversi model menggunakan converter
        Simpan model yang telah di-convert ke output_path

    Panggil fungsi convert_to_tflite dengan parameter (path file model TensorFlow
    asli, path untuk menyimpan file model TF Lite)

Selesai
```



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

### 3.2.6 Proses Training

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

Setelah semua data untuk proses *training* telah ada, maka selanjutnya adalah menentukan nilai seperti nilai *input* citra, bobot, *batch size*, dan jumlah *steps* maksimal yang akan diatur dalam konfigurasi pipeline. Sebelum melakukan konfigurasi, diperlukan sebuah *labelmap* sebagai data identifikasi objek yang akan digunakan oleh sistem. File tersebut bernama *labelmap.txt* yang berisikan banyaknya kategori dan nama dari kategori tersebut. Proses *training* pada penelitian ini menggunakan *SSD-MobileNet* serta *tensorflow-gpu* dapat dilihat pada *pseudocode* dibawah ini :

```

Mulai
  Definisikan path folder untuk data citra asli
  Definisikan path folder untuk menyimpan file XML
  Definisikan path folder untuk menyimpan file train.record
  Definisikan file labelmap.txt
  Definisikan file pipeline.config

```

```

Definisikan path untuk menyimpan model hasil training

Fungsi label_data(data_path, output_xml_path):
    Untuk setiap file citra dalam folder data_path:
        Beri label pada citra
        Simpan label sebagai file XML di output_xml_path

Fungsi convert_to_tfrecord(xml_folder, output_record_path):
    Baca semua file XML dari xml_folder
    Konversi file XML menjadi file train.record
    Simpan file train.record ke output_record_path

Panggil fungsi label_data dengan parameter (path folder data citra asli, path
folder untuk menyimpan file XML)
Panggil fungsi convert_to_tfrecord dengan parameter (path folder untuk menyimpan
file XML, path folder untuk menyimpan file train.record)

Fungsi buat_labelmap(labelmap_path):
    Tentukan banyaknya kategori dan nama dari setiap kategori
    Simpan data identifikasi objek ke labelmap_path sebagai labelmap.txt

Panggil fungsi buat_labelmap dengan parameter (file labelmap.txt)

Definisikan konfigurasi pipeline:
    Tentukan path untuk file train.record
    Tentukan path untuk file labelmap.txt
    Tentukan path untuk file model pralatih SSD MobileNet
    Tentukan nilai input citra
    Tentukan bobot awal
    Tentukan batch size
    Tentukan jumlah steps maksimal
    Simpan konfigurasi ke file pipeline.config

Fungsi train_model(pipeline_config_path, output_model_path):
    Load pipeline configuration dari pipeline_config_path
    Load dataset dan labelmap
    Load model pralatih SSD MobileNet
    Mulai proses training
    Simpan model hasil training ke output_model_path

Panggil fungsi train_model dengan parameter (file pipeline.config, path untuk
menyimpan model hasil training)

Selesai

```

Setelah proses *training* selesai, akan menghasilkan sebuah model dengan

ekstensi .pb yang kemudian akan dikonversi ke dalam bentuk *tflite* agar dapat digunakan pada perangkat mobile

### 3.2.7 Proses Testing

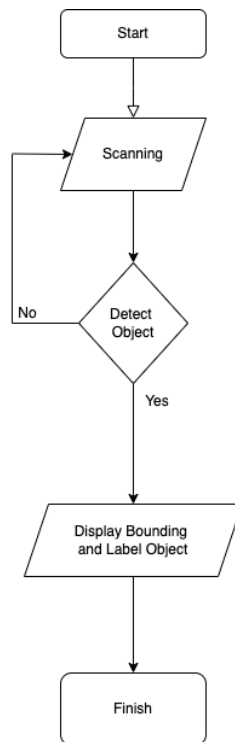
Selanjutnya adalah proses *testing* yang merupakan tahapan pengujian dari model yang sudah terbentuk setelah proses *training* di tahap sebelumnya. Proses ini dilakukan untuk dapat mengetahui apakah model yang sudah ada dapat dibaca oleh sistem, apakah objek yang sudah dibuat berupa jenis salak dapat dikenali oleh sistem, dan apakah sistem dapat melakukan identifikasi objek salak menjadi tiga kategori yaitu Salak Pondoh, Salak Sidempuan dan Salak Bali, serta untuk menghitung tingkat keakuratan metode *SSD-MobileNet* yang telah diterapkan pada sistem identifikasi jenis salak di penelitian ini.

### 3.2.8 Output

Hasil akhir dari sistem yang telah dibuat yaitu berupa informasi akurasi dan *bounding box* yang menunjukkan hasil identifikasi jenis salak dengan 3 kategori yaitu salak pondoh, salak bali dan salak sidempuan yang dilakukan pada saat proses *testing*.

## 3.3 Flowchart Sistem

Pada sub-bab ini dijelaskan alur kerja sistem, serta aspek-aspek teknik yang merupakan solusi perancangan sistem. Adapun cara kerja sistem dapat dilihat pada Gambar 3.4.



**Gambar 3. 4** Flowchart Sistem

### 3.4 Perancangan Antarmuka

Perancangan antarmuka merupakan tahapan yang menjelaskan alur kerja sistem, tahapan ini diharapkan dapat membuat pengguna lebih tertarik dan mendapatkan kemudahan dalam pengaplikasian sistem. Rancangan tampilan yang ditunjukkan pada Gambar 3.5 dan 3.6 merupakan tampilan yang akan digunakan *user* untuk mendeteksi jenis salak. Pada tampilan *dashboard* terlihat ada nama pengembang dan juga ada tombol identifikasi yang mana nanti akan mengarahkan ke tampilan deteksi. Pada tampilan deteksi ini aplikasi akan langsung mengidentifikasi objek saat diarahkan ke objek yang dituju beserta munculnya *bounding box* yang berisi informasi jenis salak yang di sorot.

IDENTIFIKASI JENIS SALAK
TALITHA ASHVI RAYHAN 171402083
<div>Deteksi</div>

**Gambar 3. 5** Rancangan Halaman Dashboard

IDENTIFIKASI JENIS SALAK
<div>Tampilan identifikasi salak Beserta bounding box</div>

**Gambar 3. 6** Rancangan Halaman Identifikasi

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini akan membahas implementasi *SSD-MobileNet* dalam identifikasi jenis salak dari bab sebelumnya serta membahas hasil pengujian terhadap sistem yang telah dirancang.

#### **4.1 Implementasi Sistem**

Pada tahap ini, implementasi *SSD-MobileNet* dalam identifikasi jenis salak secara *real time* menggunakan perangkat *android*. Oleh karena itu sistem ini memerlukan perangkat keras dan perangkat lunak pendukung, yaitu:

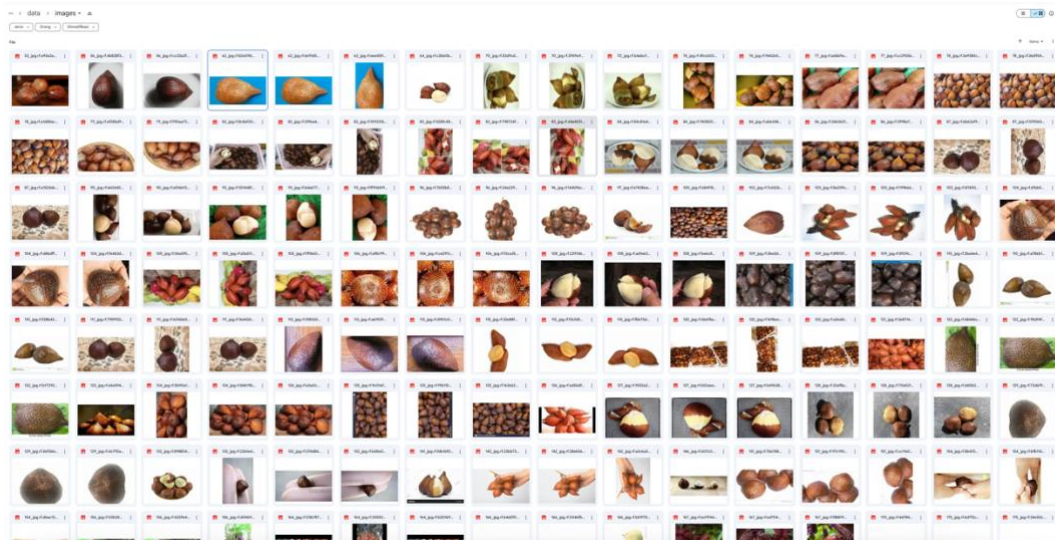
##### **4.1.1 Perangkat Keras dan Perangkat Lunak**

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem identifikasi jenis buah salak ini adalah:

1. Perangkat laptop yang digunakan adalah MacBook Pro M1 2020
2. Processor 3,2 GHz Apple M1
3. Kapasitas RAM sebesar 8 GB
4. Kapasitas Hardisk sebesar 256 GB
5. Sistem operasi yang digunakan macOS Ventura

##### **4.1.2 Implementasi Data**

Data yang digunakan dalam penelitian ini menggunakan dua data yaitu *data train* dan *data test*. *Dataset* yang ada berdasarkan data yang diambil dari internet yaitu *Kaggle* dan secara langsung dengan tiga (3) jenis salak yaitu salak pondoh, salak sidempuan dan salak bali sebanyak 600 citra dengan 200 salak pondoh, 200 salak sidempuan dan 200 salak bali. Contoh data jenis salak yang sudah dilabeling dapat dilihat secara pada Gambar 4.1.



**Gambar 4. 1** Data Citra Salak

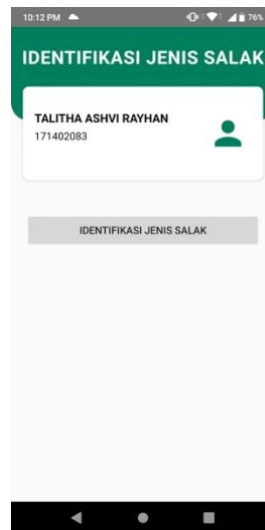
#### 4.1.3 Implementasi Antarmuka

Implementasi antarmuka dibuat berdasarkan perancangan antarmuka sistem yang telah dibahas sebelumnya pada bab 3. Tampilan sistem terdiri dari satu (2), yaitu tampilan dashboard dan tampilan identifikasi.

##### 1. Tampilan Dashboard

Tampilan dashboard merupakan halaman awal saat pengguna pertama kali membuka atau menjalankan aplikasi. Pada tampilan ini terdapat nama, nim dan tombol identifikasi yang mana tombol tersebut mengarah ke halaman identifikasi. Tampilan dashboard ditujukan pada gambar 4.2.

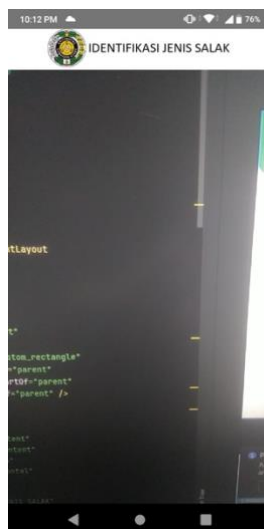




**Gambar 4. 2** Tampilan Halaman Dashboard

## 2. Tampilan Identifikasi

Pada tampilan ini pengguna langsung bisa menggunakan aplikasi dengan cara mengarahkan ke objek yang ingin diidentifikasi. Tampilan Identifikasi ditunjukkan pada gambar 4.3.



**Gambar 4. 3** Tampilan Halaman Identifikasi

## 4.2 Prosedur Operasional

Tampilan aplikasi yang dibangun terdiri dari halaman identifikasi dan halaman *dashboard*. Halaman identifikasi yang dibangun dapat dilihat pada Gambar 4.4 dimana pengguna dapat mengarahkan kamera pada objek jenis salak yang ingin diketahui jenisnya yaitu salak pondoh, salak sidempuan atau salak bali.



**Gambar 4. 4** Prosedur Operasional

Ketika objek di deteksi menggunakan *smartphone* maka *bounding box* akan terbentuk seperti Gambar 4.4. Pada Gambar 4.4 dapat dilihat bahwasannya objek yang berada di depan kamera merupakan Salak Bali yang berada di dalam *bounding box* berwarna orans.





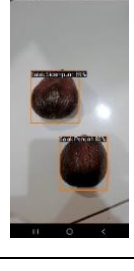
*Bounding box* akan bergerak mengikuti arah kamera *smartphone* yang akan mengidentifikasi salak yang ada pada *dataset* yang telah dibuat. Sistem yang dibuat juga mampu mengidentifikasi jenis salak lainnya yaitu salak pondoh dan salak sidempuan.






## 4.3 Pengujian Sistem


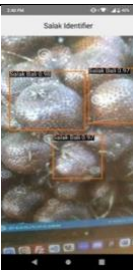
Pada tahap ini dilakukan pengujian sistem untuk melihat hasil menggunakan *SSD-MobileNet* dalam mengidentifikasi jenis salak yaitu salak pondoh, salak bali dan







salak sidempuan. Tabel pengujian sistem dapat dilihat pada Tabel. 4.1.



**Tabel 4. 1** Hasil Pengujian


No	Data Salak	Actual Output	Desired Output	Keterangan
1		Salak Bali	Salak Bali	Berhasil
2		Salak Bali	Salak Bali	Berhasil
3		Salak Bali	Salak Pondoh	Gagal
4		Salak Bali	Salak Pondoh	Gagal
5		Salak Sidempuan Salak Pondoh	Salak Sidempuan Salak Pondoh	Berhasil

6		Salak Pondoh	Salak Pondoh	Berhasil
7		Salak Pondoh Salak Sidempuan	Salak Pondoh Salak Sidempuan	Berhasil
8		Salak Pondoh	Salak Pondoh	Berhasil
9		Salak Sidempuan Salak Pondoh	Salak Pondoh Salak Sidempuan	Gagal
10		Salak Pondoh	Salak Pondoh	Berhasil
11		Salak Pondoh	Salak Pondoh	Berhasil

12		Salak Pondoh	Salak Bali	Gagal
13		Salak Bali	Salak Bali	Berhasil
14		Salak Bali	Salak Bali	Berhasil
15		Salak Pondoh	Salak Pondoh	Berhasil
16		Salak Pondoh	Salak Pondoh	Berhasil
17		Salak Bali	Salak Bali	Berhasil

18		Salak Bali	Salak Bali	Berhasil
19		Salak Bali	Salak Bali	Berhasil
20		Salak Sidempuan	Salak Bali	Gagal
21		Salak Pondoh	Salak Pondoh	Berhasil
22		Salak Sidempuan	Salak Bali	Gagal
23		Salak Bali	Salak Bali	Berhasil

24		Salak Pondoh	Salak Pondoh	Berhasil
25		Salak Pondoh	Salak Pondoh	Berhasil
26		Salak Pondoh	Salak Pondoh	Berhasil
27		Salak Sidempuan	Salak Sidempuan	Berhasil
28		Salak Sidempuan	Salak Sidempuan	Berhasil
29		Salak Sidempuan	Salak Sidempuan	Berhasil

30		Salak Sidempuan	Salak Sidempuan	Berhasil
----	---	-----------------	-----------------	----------

Berdasarkan pengujian sistem yang telah dilakukan pada salak, diperoleh nilai akurasi mencapai 93%, terdapat beberapa kesalahan yang dilakukan oleh sistem dalam melakukan identifikasi jenis salak dari total 120 pengujian yang dilakukan.

Adapun penyebab kesalahan data yang ada dikarenakan banyaknya bentuk salak yang bentuknya hampir serupa satu sama lain dalam data yang ada setelah melalui tahap *resizing* sehingga sistem mendapatkan kesulitan yang lebih saat mendeteksi perbedaan antara ketiga salak yang ada serta faktor cahaya yang juga mempengaruhi dalam mendeteksi jenis salak.

**Tabel 4. 2** Confusion Matrix

	Salak Pondoh	Salak Bali	Salak Sidempuan	Total
Salak Pondoh	36	3	1	40
Salak Bali	3	36	1	40
Salak Sidempuan	1	1	38	40
Total	40	40	40	120

**Tabel 4. 3** Nilai TP, FP dan FN dari Salak

	TP	FP	FN
Salak Pondoh	36	4	4
Salak Bali	36	4	4
Salak Sidempuan	38	2	2
Total	110	10	10



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

	Precision	Recall	F1-Score
Salak Pondoh	0,90	0,90	0,90
Salak Bali	0,90	0,90	0,90
Salak Sidempuan	0,95	0,95	0,95

Penelitian dengan judul Identifikasi jenis buah salak menggunakan *SSD-MobileNet* secara *realtime* mampu mengidentifikasi jenis salak dengan mendapatkan tingkat akurasi sebesar 91,6%. Sistem yang ada mampu melakukan identifikasi jenis salak secara realtime menggunakan android.

Terdapat *error* sebesar 8,4% yang terjadi pada sistem dalam melakukan pendeteksian dikarenakan kemiripan yang terjadi di antara beberapa salak. Bentuk kedua salak beberapa kali memiliki kemiripan setelah dilakukan proses *resizing* sehingga membuat sistem memiliki kesulitan lebih dalam melakukan pendeteksian untuk mengidentifikasi jenis salak.

## **BAB 5**

### **PENUTUP**

Pada bab ini membahas mengenai kesimpulan dari identifikasi jenis salak secara *real time* menggunakan *SSD-MobileNet* dan saran untuk referensi pengembangan penelitian selanjutnya.

#### **5.1 Kesimpulan**

Hasil dari pengujian aplikasi untuk mengidentifikasi jenis salak secara *real time* menggunakan *SSD-MobileNet* didapatkan kesimpulan, yaitu:

1. Metode *SSD-MobileNet* dapat mengidentifikasi secara *real-time* berjenis salak pondoh, salak sidempuan dan salak bali dengan hasil akurasi baik sebesar 91,6%
2. Adapun faktor yang membuat tingkat akurasi tidak sempurna pada penelitian yang telah dilakukan ini adalah kesalahan yang terjadi pada proses identifikasi salak. Hal itu disebabkan karena terjadinya kemiripan di antara beberapa salak, dimana objek salak tersebut memiliki bentuk yang mirip setelah melalui proses *resizing* sehingga membuat sistem memiliki kesulitan lebih saat melakukan proses identifikasi.

#### **5.2 Saran**

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

1. Menggunakan metode yang lain sebagai perbandingan pada hasil identifikasi salak dari citra salak yang diperoleh menggunakan metode *SSD-MobileNet*.
2. Diharapkan agar dapat menyempurnakan nilai akurasi untuk mendeteksi jenis salak dengan berbagai kondisi.
3. Menambahkan identifikasi salak jenis yang lainnya secara *realtime*.

## DAFTAR PUSTAKA

- Al-Azzo, F & Taqi, M. A. (2018). Human Related-Health Actions Detection Using Android Camera Based On Tensorflow Object Detection Api. *International Journal Of Advanced Computer Science And Applications (Ijacs)*, 9(10).
- Arabi, S., Haghighat, A., & Sharma, A. (2019). A deep learning based solution for construction equipment detection: from development to deployment. *arXiv preprint arXiv:1904.09021*.
- Avif, B. A. (2019). Klasifikasi Tanaman Herbal Zingiber Berdasarkan Citra Mikroskopis Stomata Menggunakan Algoritma Probabilistic Neural Network (PNN). *Universitas Sumatera Utara*.
- Biswas, D., Su, H., Wang, C., Stevanovic, A., & Wang, W. (2019). An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD. *Physics and Chemistry of the Earth*, 110, 176–184.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861*.
- Ichsan, A. (2021) Implementasi relevance vector machine untuk mengidentifikasi penyakit mata glaukoma. *Institut Teknologi Nasional*.
- Kanimozhi, S., Gayathri, G., & Mala, T. (2019). Multiple object identification using single shot multi-box detection. *ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings*, 1–5.
- Khatun, M., Ali, F., Turzo, N. A., Nine, J., & Sarker, P. (2020). Fruits Classification using Convolutional Neural. *GRD Journals- Global Research and Development Journal for Engineering*, 5(8).
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Yang Fu, C. & Breg A. C. (2016). SSD: Single Shot MultiBox Detector. *arXiv:1512.02325*.
- Mirwansyah, D., & Wibowo, A. (2022). Fruit Image Classification Using Deep Learning Algorithm. *Multica Science And Technology*, 2(2), 38-41.
- Suhardjo, H. L., Deaton, B. J. & Driskel, J. A. (1985). Pangan, Gizi dan Pertanian. *UI-Pers Jakarta*.
- Soelarso, B. (1997). Budidaya Apel. *Yogyakarta: Penerbit Kanisius (Anggota IKAPI)*.
- Rahman, F. A. (2020) Klasifikasi Invasive Ductal Carcinoma Menggunakan Convolutional Neural Network. *Universitas Muhammadiyah Malang*.
- Wijaya, N., & Ridwan, A. (2019). Klasifikasi Jenis Buah Apel Dengan Metode KNearest Neighbors Dengan Ekstraksi Fitur HSV dan LBP. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 8(1), 74-78.

- Xiao, F., Wang, H., Xu, Y. & Zhang, R. (2023). Fruit Detection and Recognition Based on Deep Learning for. *Agronomy*, 13(1625).
- Zeng, G. (2017). Fruit and vegetables classification system using image saliency and convolutional neural network. *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*. 613-617.