

**IMPLEMENTASI METODE *CAPSULE NETWORK*
(CAPSNET) UNTUK IDENTIFIKASI
KESEGERAN CUMI-CUMI**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA
FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN
UNIVERSITAS MARITIM RAJA ALI HAJI
TANJUNGPINANG
2025**

**IMPLEMENTASI METODE *CAPSULE NETWORK*
(CAPSNET) UNTUK IDENTIFIKASI
KESEGERAN CUMI-CUMI**



Skripsi

Untuk memenuhi syarat memperoleh Derajat
Sarjana Teknik (S.T.)

Oleh:

Vinandra Adam Saputra
2101020106

Telah mengetahui dan disetujui oleh:

Pembimbing I,

Pembimbing II,

Nola Ritha, S.T.,M.Cs.
NIP. 199011142019032016

Marisha Pertiwi, S.Tr.Kom, M.Kom
NIP. 199703272024062001

HALAMAN PENGESAHAN

Judul Skripsi : Implementasi Metode *Capsule Network* (CapsNet) Untuk
Identifikasi Kesegaran Cumi-Cumi
Nama Mahasiswa : Vinandra Adam Saputra
NIM : 2101020106
Jurusan : Teknik Elektro dan Informatika
Program Studi : Teknik Informatika

Telah dipertahankan di depan Dewan Penguji dan dinyatakan lulus
pada tanggal 10 Juli 2025

Susunan Tim Pembimbing dan Penguji

Jabatan	Nama Dosen	Tanda Tangan	Tanggal
Pembimbing I	: Nola Ritha, S.T.,M.Cs ₂
Pembimbing II	: Marisha Pertiwi, S.Tr.Kom, M.Kom
Ketua Penguji	: Tekad Matulatan, S.Kom., M.Info.Tech
Anggota Penguji I	: Nolan Efranda, M.Kom
Anggota Penguji II	: Berta Erwin SLAM, S.T., M.Kom

Tanjungpinang, Juli 2025
Universitas Maritim Raja Ali Haji
Fakultas Teknik dan Teknologi Kemaritiman
Dekan Fakultas Teknik dan Teknologi Kemaritiman

Martaleli Bettiza, S.Si., M.Sc.
NI PPPK 197508282021212006

PERNYATAAN ORISINALITAS

Dengan ini menyatakan bahwa skripsi saya yang berjudul Implementasi Metode *Capsule Network* (CapsNet) Untuk Identifikasi Kesegaran Cumi-Cumi adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Jika kemudian hari ternyata terbukti pernyataan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Maritim Raja Ali Haji.

Tanjungpinang, 26 Juni 2025

(Vinandra Adam Saputra)

HALAMAN PERSEMBAHAN

Dengan segala puji dan syukur kepada Allah SWT, atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi ini dengan lancar. Serta dengan penghargaan dan kasih sayang yang tak ternilai, skripsi ini dengan hormat penulis persembahkan kepada:

1. Kedua orang tua penulis, Bapak Yudhi Setiawan dan Ibu Rokhayati yang senantiasa tanpa henti memberikan doa dan dukungan sepanjang proses penyusunan skripsi ini.
2. Ibu Nola Ritha, S.T., M.Cs dan Ibu Marisha Pertiwi S.Tr.Kom, M.Kom yang telah membantu dan membimbing penulis dengan memberikan arahan dan dukungan selama proses penyelesaian skripsi ini.
3. Dosen penguji yang telah memberikan saran serta masukan berharga untuk penelitian ini.
4. Bapak Aidil Fadli Ilhamdy, S.Pi, M.Si selaku dosen pakar di bidang Teknologi Hasil Perikanan yang sudah membantu penulis terkait metodologi pengumpulan data dan pelabelan kelas kesegaran.
5. Muhamad Nur Syami, Sahra Fatimah Yasir, dan Silvia Indah Maharani, rekan satu bimbingan yang telah memberikan dukungan, rangkulan, dan kebersamaan selama proses penyusunan skripsi ini.
6. Amanda Dwi Mulyani, Muhamad Dhaoed Arrazaq, Muhamad Rinaldi, Muhammad Rizki Fachriza Bailey, Nur Hidayad, Rekha Clarhista Amelia, dan Yoga Syahputra, sebagai rekan-rekan dan sahabat seperjalanan yang telah berbagi tawa, cerita, dan solidaritas selama masa studi dan penyelesaian skripsi.
7. Teman-teman seperjuangan Teknik Informatika 2021 yang telah banyak membantu dan memotivasi penulis dalam menyelesaikan skripsi ini.

HALAMAN MOTO

“Pray For The Best and Let God Do The Rest”



KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala limpahan Rahmat, karunia, serta hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Implementasi Metode *Capsule Network* (CapsNet) Untuk Identifikasi Kesegaran Cumi-Cumi” dengan baik dan tepat waktu. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T.) pada Program Studi Teknik Informatika.

Penyusunan skripsi ini tidak lepas dari bimbingan, arahan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang tulus kepada semua pihak yang terlibat baik secara langsung maupun tidak langsung. Karena tanpa dorongan dan motivasi dari mereka, penulis mungkin tidak akan mampu menyelesaikan skripsi ini dengan baik.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, baik dari segi isi maupun teknik penulisan, mengingat keterbatasan pengetahuan dan pengalaman yang dimiliki. Oleh karena itu, kritik dan saran yang membangun dari semua pihak sangat penulis harapkan demi perbaikan di masa mendatang.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat dan kontribusi positif, baik secara teoritis maupun praktis, serta dapat menjadi referensi bagi penelitian selanjutnya.

Tanjungpinang, 26 Juni 2025

(Vinandra Adam Saputra)

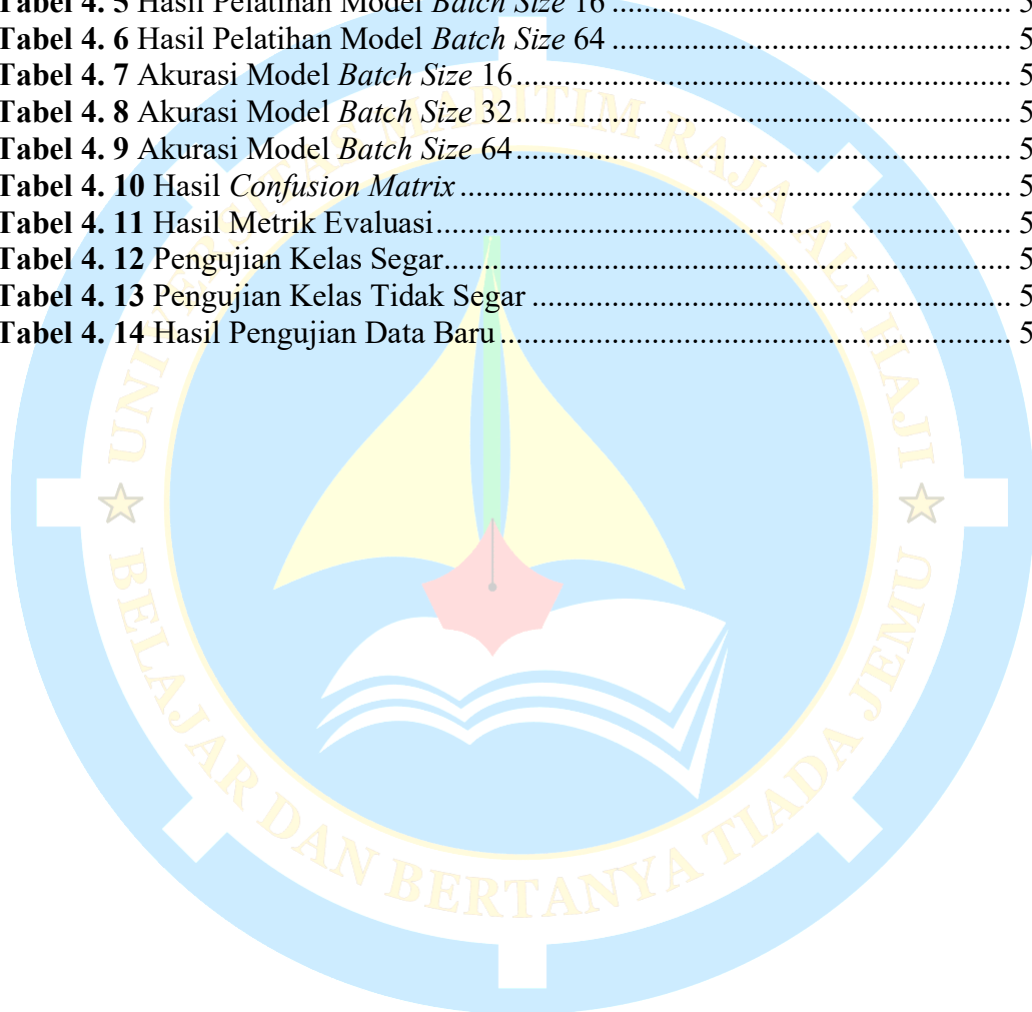
DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTO	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
GLOSARIUM.....	xi
ABSTRAK.....	xiii
ABSTRACT.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
BAB II KAJIAN LITERATUR	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	11
2.2.1 Cumi-Cumi	11
2.2.2 Kesegaran Cumi-Cumi.....	11
2.2.3 Computer Vision.....	13
2.2.4 Deep Learning.....	13
2.2.5 Capsule Network (CapsNet)	14
2.2.5.1 Convolution Layer	15
2.2.5.2 Primary Capsule.....	17
2.2.5.3 Dynamic Routing.....	18
2.2.5.4 Activation Function	20
2.2.5.5 Digit Capsule	21

2.2.6 Python	21
2.2.7 Google Colaboratory.....	22
2.2.8 Confusion Matrix	22
2.2.9 Parameter yang Digunakan	23
BAB III METODE PENELITIAN	25
3.1 Jenis Penelitian.....	25
3.2 Waktu dan Tempat Penelitian.....	25
3.3 Tahapan Penelitian.....	25
3.4 Studi Pustaka.....	27
3.5 Alat dan Instrumen Penelitian.....	27
3.6 Pengumpulan Data	28
3.7 Pembagian Dataset.....	30
3.8 Analisis dan Perancangan	30
3.8.1 Analisis Data	30
3.8.2 Perancangan Model <i>Capsule Network</i> (CapsNet).....	33
3.8.3 Perhitungan Manual <i>Capsule Network</i> (CapsNet).....	35
3.9 Perancangan Sistem	47
3.9.1 <i>Use Case Diagram</i>	47
3.9.2 Perancangan <i>User Interface</i>	48
BAB IV HASIL DAN PEMBAHASAN	49
4.1 Hasil Training Model.....	49
4.1.1 Analisis Pengaruh <i>Learning Rate</i> dan Jumlah <i>Epoch</i>	49
4.1.2 Pengaruh <i>Batch Size</i>	52
4.2 Pengujian Model Menggunakan <i>Confusion Matrix</i>	55
4.3 Pengujian Model Menggunakan Data Baru	58
4.4 Pengujian Kelas Non Cumi.....	59
4.5 Pembahasan.....	60
4.6 Hasil Perancangan Aplikasi Web.....	61
BAB V PENUTUP	63
5.1 Kesimpulan	63
5.2 Saran	63
DAFTAR PUSTAKA	64
LAMPIRAN.....	70

DAFTAR TABEL

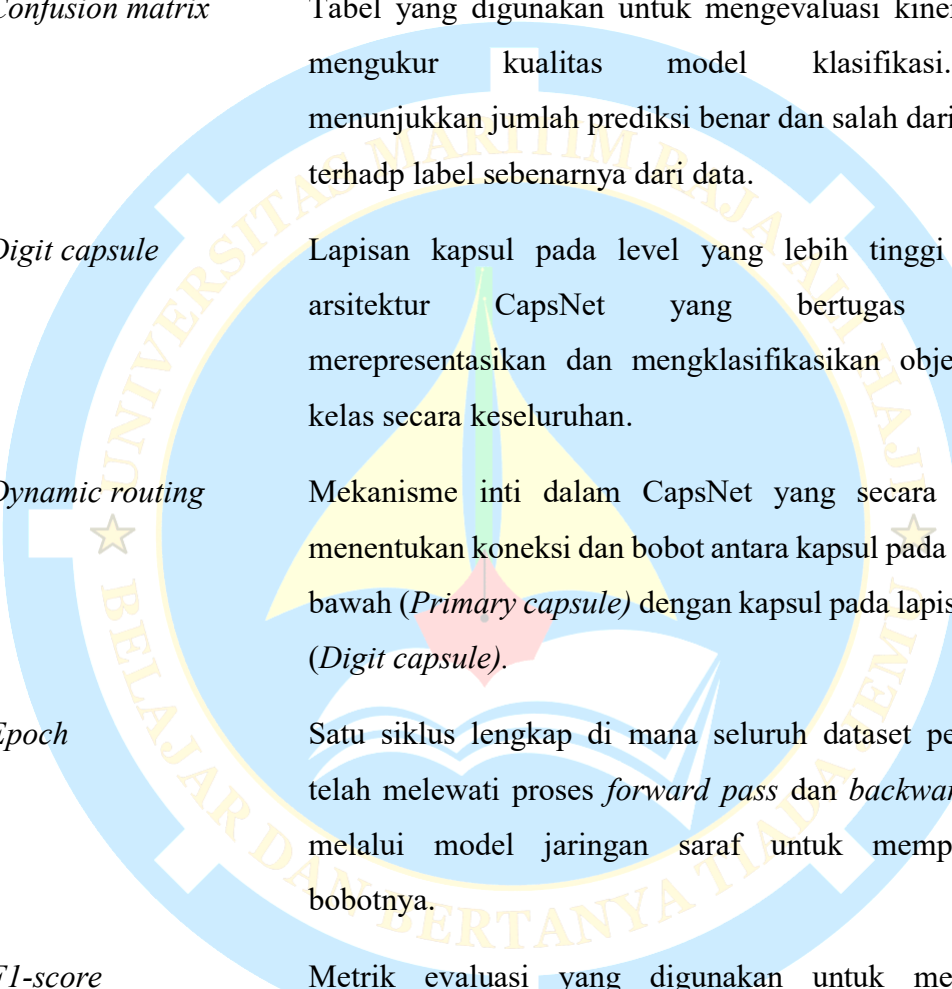
Tabel 2. 1 Indikator Kesegaran Cumi-Cumi	12
Tabel 3. 1 Pembagian Dataset	30
Tabel 3. 2 Rincian Arsitektur Model	34
Tabel 4. 1 Percobaan <i>Learning Rate</i> dan <i>Epoch</i>	49
Tabel 4. 2 Hasil Pelatihan Model <i>Epoch</i> 50	52
Tabel 4. 3 Hasil Pelatihan Model <i>Epoch</i> 100	52
Tabel 4. 4 Percobaan Batch Size	53
Tabel 4. 5 Hasil Pelatihan Model <i>Batch Size</i> 16	55
Tabel 4. 6 Hasil Pelatihan Model <i>Batch Size</i> 64	55
Tabel 4. 7 Akurasi Model <i>Batch Size</i> 16	57
Tabel 4. 8 Akurasi Model <i>Batch Size</i> 32	57
Tabel 4. 9 Akurasi Model <i>Batch Size</i> 64	57
Tabel 4. 10 Hasil <i>Confusion Matrix</i>	57
Tabel 4. 11 Hasil Metrik Evaluasi	58
Tabel 4. 12 Pengujian Kelas Segar	58
Tabel 4. 13 Pengujian Kelas Tidak Segar	58
Tabel 4. 14 Hasil Pengujian Data Baru	59



DAFTAR GAMBAR

Gambar 2. 1	Arsitektur CapsNet	15
Gambar 2. 2	Proses Konvolusi	16
Gambar 3. 1	Tahapan Penelitian	26
Gambar 3. 2	Pengambilan Gambar Dataset	28
Gambar 3. 3	Pelabelan Cumi Uji Organoleptik	29
Gambar 3. 4	Citra Cumi Tidak Segar	30
Gambar 3. 5	Citra Cumi Segar	30
Gambar 3. 6	Tahapan <i>Preprocessing</i> Data	32
Gambar 3. 7	Citra Cumi Sebelum <i>Cropping</i>	32
Gambar 3. 8	Citra Cumi Setelah <i>Cropping</i>	33
Gambar 3. 9	Citra Cumi Sebelum <i>Resize</i>	33
Gambar 3. 10	Citra Cumi Setelah <i>Resize</i>	33
Gambar 3. 11	<i>Flip</i> Horizontal	34
Gambar 3. 12	<i>Random Zoom</i>	34
Gambar 3. 13	Rotasi 90°	34
Gambar 3. 14	Rotasi 180°	34
Gambar 3. 15	Rotasi 270°	34
Gambar 3. 16	Rancangan Arsitektur Model CapsNet	35
Gambar 3. 17	<i>Flowchart</i> Metode CapsNet	36
Gambar 3. 18	<i>Input Layer</i>	38
Gambar 3. 19	<i>Input Channel Red</i>	38
Gambar 3. 20	Contoh Filter	39
Gambar 3. 21	Hasil Konvolusi Posisi Ke-1	39
Gambar 3. 22	<i>Feature Map Channel Red</i>	41
Gambar 3. 23	<i>Feature Map</i> Citra Input	41
Gambar 3. 24	<i>Use Case Sistem</i>	48
Gambar 3. 25	Tampilan Halaman Utama	48
Gambar 3. 26	Tampilan Setelah Mengunggah Gambar	48
Gambar 4. 1	<i>Learning Rate 1e-04 50 Epoch</i>	50
Gambar 4. 2	<i>Learning Rate 1e-04 100 Epoch</i>	50
Gambar 4. 3	<i>Learning Rate 1e-05 50 Epoch</i>	51
Gambar 4. 4	<i>Learning Rate 1e-05 100 Epoch</i>	51
Gambar 4. 5	<i>Learning Rate 1e-04 Batch Size 16</i>	53
Gambar 4. 6	<i>Learning Rate 1e-05 Batch Size 16</i>	54
Gambar 4. 7	<i>Learning Rate 1e-04 Batch Size 64</i>	54
Gambar 4. 8	<i>Learning Rate 1e-05 Batch Size 64</i>	55
Gambar 4. 9	<i>Confusion Matrix Batch Size 16</i>	56
Gambar 4. 10	<i>Confusion Matrix Batch Size 32</i>	56
Gambar 4. 11	<i>Confusion Matrix Batch Size 64</i>	56
Gambar 4. 12	Hasil Pengujian Non Cumi	59
Gambar 4. 13	Halaman Utama Aplikasi	61
Gambar 4. 14	Tampilan Identifikasi Kesegaran	62

GLOSARIUM



<i>Batch size</i>	Jumlah sampel pelatihan yang dipropagasi melalui jaringan saraf dalam satu iterasi sebelum bobot model diperbarui.
<i>Confusion matrix</i>	Tabel yang digunakan untuk mengevaluasi kinerja dan mengukur kualitas model klasifikasi. dengan menunjukkan jumlah prediksi benar dan salah dari model terhadap label sebenarnya dari data.
<i>Digit capsule</i>	Lapisan kapsul pada level yang lebih tinggi dalam arsitektur CapsNet yang bertugas untuk merepresentasikan dan mengklasifikasikan objek atau kelas secara keseluruhan.
<i>Dynamic routing</i>	Mekanisme inti dalam CapsNet yang secara iteratif menentukan koneksi dan bobot antara kapsul pada lapisan bawah (<i>Primary capsule</i>) dengan kapsul pada lapisan atas (<i>Digit capsule</i>).
<i>Epoch</i>	Satu siklus lengkap di mana seluruh dataset pelatihan telah melewati proses <i>forward pass</i> dan <i>backward pass</i> melalui model jaringan saraf untuk memperbarui bobotnya.
<i>F1-score</i>	Metrik evaluasi yang digunakan untuk mengukur keseimbangan antara presisi dan <i>recall</i> .
<i>Learning rate</i>	Parameter dalam pelatihan model <i>deep learning</i> yang mengontrol seberapa besar penyesuaian bobot model dilakukan pada setiap iterasi berdasarkan gradien dan fungsi <i>loss</i> .

Precision

Metrik evaluasi yang mengukur proporsi prediksi positif yang benar dari total kasus yang diprediksi sebagai positif oleh model.

Primary Capsule

Lapisan kapsul bagian bawah dalam arsitektur CapsNet yang menerima input dari lapisan konvolusi dan berfungsi untuk mendeteksi fitur-fitur dasar dari gambar, kemudian mengeluarkan output berupa vektor yang mengkodekan properti fitur tersebut.

Recall

Metrik evaluasi yang mengukur proporsi kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model.

Squash function

Fungsi aktivasi non-linear yang digunakan dalam CapsNet untuk menormalisasi panjang vektor *output* dari sebuah kapsul, sehingga panjangnya berada dalam rentang 0-1, yang merepresentasikan probabilitas keberadaan kelas.

ABSTRAK

Saputra, Vinandra Adam. 2025. *Implementasi Metode Capsule Network (CapsNet) Untuk Identifikasi Kesegaran Cumi-Cumi* Skripsi. Tanjungpinang: Jurusan Teknik Elektro dan Informatika, Fakultas Teknik dan Teknologi Kemaritiman, Universitas Maritim Raja Ali Haji. Pembimbing I: Nola Ritha S.T., M.Cs. Pembimbing II: Marisha Pertiwi S.Tr.Kom, M.Kom

Cumi-cumi merupakan salah satu komoditas perikanan penting di Indonesia, namun penentuan kesegarannya seringkali masih dilakukan secara manual dan subjektif, yang dapat menyebabkan kesalahan penilaian dan kerugian ekonomi. Penelitian ini bertujuan untuk mengimplementasikan dan mengetahui tingkat akurasi metode *Capsule Network* (CapsNet) untuk mengidentifikasi kesegaran cumi-cumi secara otomatis berdasarkan citra digital. Penelitian ini menggunakan pendekatan kuantitatif dengan dataset yang dikumpulkan secara mandiri, yang terdiri dari 648 citra cumi-cumi setelah melalui proses augmentasi. Model dilatih untuk mengklasifikasikan dua kelas kesegaran, yaitu “Segar” dan “Tidak Segar”, dengan tambahan kelas “Non Cumi” yang berisi berbagai objek selain cumi-cumi. Tahapan pra-pemrosesan data yang diterapkan seperti *labelling*, *cropping*, dan *resize* menjadi 128x128 piksel. Hasil pengujian menunjukkan performa terbaik dicapai dengan *learning rate* 1e-05 dan *batch size* 16, menghasilkan akurasi, presisi, *recall*, dan F1-score sebesar 91% pada data uji. Lebih lanjut, pada pengujian menggunakan 20 data baru di luar dataset, model mampu mencapai akurasi 80%. Hasil ini membuktikan bahwa model CapsNet memiliki kemampuan generalisasi yang baik dan berpotensi untuk diimplementasikan sebagai alat bantu yang objektif dan akurat bagi industri perikanan dalam menjaga kualitas produk cumi-cumi.

Kata kunci: *Capsule Network*, CapsNet, Identifikasi Kesegaran

ABSTRACT

Saputra, Vinandra Adam. 2025. *Implementation of Capsule Network (CapsNet) Method for Freshness Identification of Squid* Thesis. Tanjungpinang: Departmen of Electrical Engineering and Informatics, Faculty of Engineering and Maritime Technology, University of Maritime Raja Ali Haji. Advisor: Nola Ritha S.T., M.Cs. Co-Advisor: Marisha Pertiwi S.Tr.Kom, M.Kom

Squid is one of the most important fishery commodities in Indonesia, but the determination of freshness is often still done manually and subjectively, which can lead to misjudgment and economic loss. This study aims to implement and determine the accuracy of the Capsule Network (CapsNet) method to automatically identify squid freshness based on digital images. This study uses a quantitative approach with a self-collected dataset consisting of 648 squid images after augmentation. The model was trained to classify two freshness classes, namely “Fresh” and “Not Fresh”, with the addition of the “Non Squid” class which contains various objects other than squid. Data pre-processing stages applied such as labelling, cropping, and resizing to 128x128 pixels. The test results showed that the best performance was achieved with a learning rate of 1e-05 and batch size 16, resulting accuracy, precision, recall, and F1-score of 91% on the test data. Furthermore, in tests using 20 new data outside the dataset, the model was able to achieve 80% accuracy. These results prove that CapsNet model has good generalization ability and has the potential to be implemented as an objective and accurate tool for the fisheries industry in maintaining the quality of squid products.

Keywords: *Capsule Network, CapsNet, Freshness Identification*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Cumi-cumi (*Loligo spp.*) merupakan salah satu komoditas penting dalam industri perikanan Indonesia, dalam tingkat nasional maupun regional. Menurut data dari Badan Pusat Statistik (BPS), cumi-cumi menempati urutan ketiga hasil tangkapan laut yang dijual di tempat pelelangan ikan (TPI) seluruh Indonesia pada tahun 2022 dengan volume penjualan sebesar 52,15 ribu ton, dibawah cakalang (*skipjack tuna*), yaitu sebanyak 141,19 ribu ton dan ikan jenis layang (*scad*) dengan volume produksi mencapai 105,1 ribu ton. Sementara di Kepulauan Riau sendiri, menurut data dari Kementerian Kelautan dan Perikanan (KKP) per Januari 2025, cumi-cumi menempati urutan kelima produksi perikanan tangkap pada tahun 2023 dengan volume 15.327 ton. Dengan penghasil terbesar adalah Kabupaten Karimun dengan volume 7.742 ton. Sebagai salah satu komoditas laut yang penting, baik untuk konsumsi lokal maupun ekspor, kualitas dan kesegaran cumi sangat mempengaruhi nilai jualnya di pasar, di mana kesegaran yang baik dapat meningkatkan daya tarik konsumen dan mengurangi risiko kerugian ekonomi akibat pembusukan (Kasmianti *et al.*, 2022). Oleh karena itu, menjaga kesegaran cumi sangat penting untuk memastikan keamanan pangan dan keberlanjutan industri perikanan.

Identifikasi kesegaran cumi-cumi saat ini masih menghadapi berbagai tantangan. Warna tubuh cumi-cumi segar cenderung cerah dengan tekstur kenyal, sedangkan cumi-cumi yang tidak segar menunjukkan perubahan warna seperti kemerahan, kecoklatan, atau kehitaman tergantung dari jenisnya dan memiliki tekstur yang lebih lunak. Penentuan kesegaran cumi-cumi umumnya dilakukan secara manual oleh tenaga ahli atau pedagang berdasarkan parameter *organoleptic*, yaitu penilaian yang dilakukan oleh manusia dengan melihat aroma, tekstur, dan warna untuk mengevaluasi kualitas produk. Metode ini sangat subjektif dan bergantung pada pengalaman serta keahlian individu. Keterbatasan ini dapat menyebabkan kesalahan dalam penentuan kesegaran, yang berdampak pada

kualitas produk yang sampai ke konsumen (Kılıçarslan *et al.*, 2024). Selain itu, metode manual sering kali memakan waktu dan tidak efisien, sehingga diperlukan pendekatan yang lebih objektif dan cepat untuk mengidentifikasi kesegaran cumi (Amin *et al.*, 2023). Ketidakakuratan dalam penentuan kesegaran dapat mengakibatkan kerugian ekonomi yang signifikan bagi nelayan dan pedagang, serta berpotensi membahayakan kesehatan masyarakat.

Dalam perkembangannya, pemanfaatan teknologi *machine learning* dan *deep learning* telah berkembang pesat dan mulai diterapkan dalam klasifikasi kesegaran produk perikanan. Untuk klasifikasi kesegaran menggunakan cumi-cumi sebagai objek penelitian, sejauh ini telah dilakukan oleh Tsany *et al.*, (2021) dengan menggunakan metode *Support Vector Machine* (SVM) berdasarkan fitur tekstur *Gray Level Co-occurrence Matrix* (GLCM) dan fitur warna HSI. Penelitian ini memperoleh akurasi sebesar 58,5% untuk klasifikasi kesegaran dua kelas yang dicapai dengan fitur tekstur GLCM. Selain itu, Hu *et al.*, (2020) menggunakan pendekatan *improved Faster R-CNN* untuk mengklasifikasikan spesies cumi-cumi dan mengevaluasi kesegaran berdasarkan informasi warna pada citra digital, dengan akurasi rata-rata yang dicapai sebesar 85,7%. Hal ini menunjukkan adanya peluang inovasi dalam metode klasifikasi untuk meningkatkan akurasi dan efisiensi dalam identifikasi kesegaran cumi-cumi.

Penelitian ini mengusulkan penggunaan *Capsule Network* (CapsNet) sebagai metode baru untuk klasifikasi kesegaran cumi. CapsNet memiliki keunggulan dalam memahami struktur spasial serta hubungan antar fitur dalam citra, memungkinkan model mengenali pola yang lebih kompleks dibandingkan dengan metode konvensional seperti *Convolutional Neural Network* (CNN) (Riti *et al.*, 2023). Beberapa penelitian terdahulu telah menunjukkan keberhasilan CapsNet dalam klasifikasi citra objek alami, termasuk klasifikasi penyakit tanaman, karena kemampuannya dalam mempertahankan informasi spasial dan menangani permasalahan rotasi serta distorsi objek (Khodadadzadeh *et al.*, 2021). Dalam klasifikasi penyakit tanaman, penggunaan CapsNet bersama dengan lapisan perhatian (*attention layers*) dan klasifikasi *feedforward* berhasil mencapai akurasi sebesar 99,8% pada dataset *Plant Village*, yang terdiri dari lebih dari 50.000 gambar

tanaman sehat dan terinfeksi. Metode ini lebih unggul dibandingkan pendekatan *deep learning* tradisional (Kalpana & Anandan, 2023).

Tujuan dari penelitian ini adalah untuk mengembangkan model *Capsule Network* (CapsNet) yang dapat mengidentifikasi kesegaran cumi-cumi berdasarkan parameter visual seperti warna dan tekstur tubuh. Dataset yang digunakan terdiri dari foto cumi-cumi yang diambil dari beberapa pasar tradisional dan nelayan di daerah sekitar Tanjungpinang dan Bintan. Penelitian ini akan menganalisis tingkat keakuratan metode CapsNet dalam mengklasifikasikan kesegaran cumi-cumi dengan variasi kondisi kesegaran yang mencakup dua kelas, yaitu segar dan tidak segar. Diharapkan, hasil penelitian ini dapat memberikan kontribusi signifikan bagi industri perikanan dengan menyediakan metode yang lebih cepat, akurat, dan efisien untuk penentuan kesegaran produk perikanan, khususnya cumi-cumi.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian yang telah diuraikan di atas, maka rumusan masalah penelitian ini adalah bagaimana tingkat akurasi metode *Capsule Network* (CapsNet) dalam mengidentifikasi kesegaran cumi-cumi?

1.3 Batasan Masalah

Batasan masalah yang penulis tentukan dalam penelitian ini adalah sebagai berikut:

1. Objek penelitian yang digunakan pada penelitian ini adalah spesies cumi-cumi (*Loligo spp.*).
2. Data yang digunakan diambil langsung dari tempat penjualan cumi di daerah Teluk Bakau, Kecamatan Gunung Kijang, Bintan.
3. Penelitian ini akan berfokus pada parameter warna tubuh cumi-cumi untuk klasifikasi kesegaran.
4. Citra *input* yang diharapkan sistem adalah gambar yang menampilkan satu objek cumi-cumi dengan fokus pada bagian tubuh dan berlatar belakang putih.

5. Kamera yang digunakan untuk mengambil gambar cumi-cumi adalah kamera HP dengan aspek rasio 1:1, dimensi 3024 x 3024 piksel.
6. *Output* yang dihasilkan dalam penelitian ini terdiri dari 3 kelas, yaitu segar, tidak segar, dan non cumi.
7. Jumlah dataset yang digunakan dalam penelitian ini sebanyak 648 citra cumi-cumi dengan perbandingan 70% data latih, 20% data validasi, dan 10% data uji

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disusun, tujuan penulis melaksanakan penelitian ini adalah untuk mengetahui tingkat akurasi metode *Capsule Network* (CapsNet) dalam mengidentifikasi kesegaran cumi-cumi.

1.5 Manfaat Penelitian

Manfaat penelitian dari penelitian yang penulis lakukan adalah sebagai berikut : ★

1. Bagi penulis, penelitian dan penulisan karya ilmiah ini memberikan wawasan dan pengalaman yang berharga, khususnya dalam memahami dan menerapkan metode *Capsule Network* (CapsNet) dalam klasifikasi kesegaran hewan laut.
2. Penelitian ini memberikan kontribusi dalam penggunaan *Capsule Network* (CapsNet) dalam bidang identifikasi kesegaran hewan laut.
3. Bagi industri perikanan, penelitian ini menawarkan solusi praktis bagi para nelayan dan pedagang untuk meningkatkan kualitas produk dengan memungkinkan pengotomatisan pengukuran kesegaran cumi-cumi.
4. Bagi masyarakat, penelitian ini bermanfaat untuk memastikan kualitas cumi-cumi segar yang aman untuk dikonsumsi.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri dari beberapa tahapan yang dibagi menjadi beberapa bab yaitu sebagai berikut.

BAB I PENDAHULUAN

Pada bagian ini berisi penjelasan terkait latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN LITERATUR

Pada bagian ini menjelaskan penelitian-penelitian terdahulu yang relevan dengan judul skripsi yang diambil, dan teori yang digunakan dalam topik penelitian dengan metode yang sama.

BAB III METODOLOGI PENELITIAN

Pada bagian ini berisi penjelasan tahapan penelitian yang dilakukan, waktu dan tempat penelitian, teknik pengumpulan data, instrumen penelitian yang digunakan, serta analisa dan perancangan aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini menyajikan hasil penelitian dan pembahasan mengenai hasil yang diperoleh dari perancangan sistem aplikasi yang dibuat.

BAB V PENUTUP

Pada bagian ini berisi kesimpulan dari hasil penelitian yang diperoleh, serta saran yang dapat dilakukan untuk mengembangkan penelitian dengan topik serupa.

DAFTAR PUSTAKA

Bagian ini berisi sumber-sumber yang digunakan sebagai referensi dan pendukung dalam penelitian terdahulu.

LAMPIRAN

Bagian ini berisi lampiran-lampiran yang digunakan sebagai pendukung dalam penelitian.

BAB II

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Dalam penelitian ini, beberapa penelitian terdahulu dengan topik yang relevan akan disertakan sebagai bahan acuan penelitian yang akan dilakukan. Tsany *et al.*, (2021) melakukan penelitian untuk klasifikasi kesegaran cumi-cumi menggunakan metode *Support Vector Machine* (SVM) dengan ekstraksi fitur tekstur *Gray Level Co-occurrence* (GLCM) dan ekstraksi fitur warna HSI. Dataset gambar cumi-cumi yang digunakan dalam penelitian ini mencakup 360 data untuk klasifikasi jenis perlakuan, meliputi gambar cumi-cumi dengan kondisi tanpa pendinginan, direndam dalam air es, dan disimpan dalam kulkas. Lalu sebanyak 495 data untuk 3 kelas kesegaran, yaitu segar, tidak segar, dan busuk. Dan yang terakhir untuk 2 kelas kesegaran, yaitu segar dan tidak segar sebanyak 330 data. Proses *preprocessing* yang dilakukan meliputi *cropping*, *resize*, augmentasi data, dan konversi warna. Hasil pengujian yang didapatkan untuk klasifikasi jenis perlakuan dengan akurasi terbaik 67,75% menggunakan fitur warna HSI. Sedangkan untuk klasifikasi kesegaran 2 kelas, akurasi tertinggi sebesar 58,5% dicapai menggunakan fitur tekstur GLCM. Selain itu, pada penelitian ini juga menguji beberapa faktor yang dapat mempengaruhi akurasi model, seperti resolusi gambar, kualitas dataset, dan rotasi gambar.

Fatoni & Ernastuti (2023) melakukan penelitian untuk mengembangkan model klasifikasi citra tanaman hias yang dapat meningkatkan akurasi dan efisiensi komputasi dalam sistem rekomendasi konten pada media sosial tanaman hias. Metode *Integration of Convolution with Capsule Network* (ICCN) digunakan dalam penelitian ini. Metode ini menggabungkan beberapa lapisan *strided convolution* dengan *Capsule Network* (CapsNet) untuk mengatasi keterbatasan CapsNet dalam komputasi serta masalah translasi invarian yang sering terjadi pada *Convolutional Neural Network* (CNN). Penelitian ini menggunakan berbagai langkah, seperti pengumpulan dan pemrosesan dataset, perancangan arsitektur model, pelatihan, dan pengujian model. Data yang digunakan terdiri dari 4000 foto tanaman hias yang

diperoleh dari *Google* dan *Kaggle*. Kemudian dimasukkan ke dalam empat kategori: kalamansi, bonsai, aglonema, dan bunga matahari. Model ICCN memiliki akurasi 95%, melampaui CNN dengan akurasi 93%.

Riti *et al.*, (2023) melakukan penelitian untuk membandingkan performa algoritma *Convolutional Neural Network* (CNN) dengan arsitektur *ResNet50V2* dan *Capsule Network* (CapsNet) dalam klasifikasi jenis rumah adat di Nusa Tenggara Timur (NTT). Penelitian ini berfokus untuk mengatasi tantangan yang ada dalam klasifikasi rumah adat yang memiliki kemiripan bentuk, bahan bangunan dan struktur. Tahapan yang dilakukan pada penelitian meliputi pengumpulan dataset, *preprocessing* data, pelatihan model, dan evaluasi performa algoritma. Sebanyak 500 gambar rumah adat yang digunakan sebagai dataset dalam penelitian ini. Data tersebut dikumpulkan dari berbagai sumber termasuk pengambilan secara langsung maupun melalui internet. Selanjutnya data diklasifikasikan ke dalam lima kategori rumah adat yaitu Sao Ngada (Bajawa), Sao Ria (Ende), Ume Kbubu (Pulau Timor), Sao Ata Mosa Lakitana (Sumba), dan Mbaru Niang (Wae Rebo). Setelah melalui tahap *preprocessing* data, seperti normalisasi, *resizing*, dan penghapusan *noise*, selanjutnya dataset dibagi dengan perbandingan 70:20:10. Hasil penelitian menunjukkan bahwa CNN memiliki akurasi 98% dengan nilai loss 0,72, sedangkan CapsNet memiliki akurasi 72% dengan nilai loss 1,37. CNN terbukti lebih unggul dalam mengenali dan mengklasifikasikan rumah adat dibandingkan CapsNet, yang membutuhkan waktu pelatihan lebih lama dan memiliki kompleksitas lebih tinggi.

Abouelmagd *et al.*, (2024) melakukan penelitian untuk meningkatkan akurasi klasifikasi penyakit pada daun tomat dengan menggunakan *Capsule Network* (CapsNet) yang dioptimalkan. Dataset yang digunakan diambil dari dataset *PlantVillage* yang terdiri dari 10 jenis penyakit daun tomat. *Preprocessing* yang dilakukan seperti melakukan augmentasi data dengan tujuan untuk mengurangi *overfitting* dan meningkatkan ketahanan model terhadap variasi data. Hasil penelitian menunjukkan CapsNet mencapai akurasi klasifikasi sebesar 96,39%, mengungguli berbagai model CNN konvensional. Model ini mampu secara efektif mengidentifikasi 10 jenis penyakit daun tomat berdasarkan karakteristik visualnya.

Jiang *et al.*, (2020) melakukan penelitian untuk meningkatkan akurasi klasifikasi citra hiperspektral (HSI) dengan mengusulkan metode Conv-Caps yang menggabungkan *Convolutional Neural Network* (CNN) dengan *Capsule Network* (CapsNet) dan menggunakan *Markov Random Field* (MRF) sebagai metode pemrosesan pasca-klasifikasi. Dengan mempertimbangkan kekurangan CNN yang sering mengabaikan informasi spasial dalam citra hiperspektral, penelitian ini mengembangkan model yang mempertahankan informasi spektral dan spasial untuk meningkatkan akurasi klasifikasi. Metodologi penelitian ini dimulai dengan pemrosesan awal data HSI menggunakan CNN tanpa lapisan *fully connected* untuk mengekstrak fitur awal. Kemudian, fitur ini diproses dengan CapsNet, yang menggantikan neuron tradisional dengan kapsul yang menyimpan informasi vektor, memungkinkan model mempertahankan hubungan spasial antara fitur. Setelah itu, hasil klasifikasi dari CapsNet diproses lebih lanjut menggunakan MRF, yang menerapkan metode *graph cut expansion* untuk menyempurnakan segmentasi dan meningkatkan konsistensi label antar piksel dalam citra hiperspektral. Hasil penelitian menunjukkan bahwa metode Caps-MRF yang dikembangkan dalam studi ini memiliki akurasi klasifikasi yang lebih tinggi dibandingkan metode konvensional. Selain itu, penggunaan MRF membantu mengurangi kesalahan klasifikasi dengan memperbaiki segmentasi citra melalui konsistensi spasial antar piksel. Pengujian lebih lanjut juga menunjukkan bahwa metode ini lebih tahan terhadap *noise*, dengan penurunan akurasi yang lebih kecil dibandingkan metode lain saat tingkat *noise* meningkat.

Chaudhari *et al.*, (2023) melakukan penelitian untuk mengembangkan metode klasifikasi otomatis untuk bakteri berdasarkan citra mikroskopis dengan menggunakan *Capsule Network* yang dioptimalkan dengan fitur histogram. Penggunaan fitur histogram untuk mengurangi jumlah parameter yang harus dilatih, sehingga akan mengoptimalkan komputasi. Dataset yang digunakan diperoleh dari dataset DiBAS, yang mencakup 33 kategori spesies bakteri. Pada penelitian ini dilakukan perbandingan dengan algoritma *machine learning* lainnya, seperti KNN, *Decision Tree*, SVM, dan *Naïve Bayes*. Hasil penelitian menunjukkan bahwa akurasi yang dicapai *Capsule Network* adalah sebesar 95,08% dan mengungguli

berbagai metode *machine learning* lainnya. Model ini juga memiliki jumlah parameter yang lebih sedikit, sehingga lebih efisien dibandingkan dengan model konvensional.

Vasudevan & Karthick (2023) melakukan penelitian untuk meningkatkan akurasi deteksi penyakit pada daun anggur menggunakan pendekatan hibrida yang menggabungkan *Efficient Generative Adversarial Network* (E-GAN) untuk augmentasi data dan *Capsule Network* (CapsNet) untuk klasifikasi penyakit serta tahapannya. Metode yang digunakan dalam penelitian ini terdiri dari tiga tahap utama yaitu segmentasi, augmentasi, dan klasifikasi penyakit serta tahapannya. Segmentasi dilakukan dengan metode *graph-based segmentation* dan *Circular Hough Transform* (CHT) untuk mengekstrak area daun dari gambar tanaman. Untuk mengatasi keterbatasan dataset, E-GAN digunakan untuk menghasilkan citra sintetis dengan menggabungkan fitur penyakit dari daun yang terinfeksi dengan gambar daun sehat. Setelah augmentasi, CapsNet digunakan untuk mengidentifikasi jenis penyakit dan tahap perkembangannya dengan menganalisis fitur spasial dan frekuensi dari gambar yang telah diproses. Eksperimen dilakukan menggunakan dataset *Plant Village grape leaf* serta gambar *real-time* yang diambil dengan kamera SD1000. Hasil penelitian menunjukkan bahwa model CapsNet yang dikombinasikan dengan E-GAN mencapai akurasi 97,63%, lebih tinggi dibandingkan model CNN lainnya seperti AlexNet (96,25%), VGG16 (95,12%), dan Xception (95,32%). Selain itu, metode augmentasi E-GAN terbukti lebih efektif dibandingkan DCGAN dan LeafGAN, dengan hasil klasifikasi yang lebih akurat dan stabil.

Tan *et al.*, (2021) melakukan penelitian untuk meningkatkan akurasi dan efisiensi identifikasi spesies *cephalopoda* berdasarkan karakteristik paruh, yang sulit dilakukan secara manual dikarenakan sifat tubuh lunak yang dimiliki oleh *cephalopoda*. Data yang digunakan dalam penelitian ini sebanyak 174 sampel dari 7 spesies *cephalopoda* berbeda dari 2 lokasi pendaratan perikanan di Pantai barat Semenanjung Malaysia antara November 2017 dan Januari 2018. Sampel tersebut difoto dengan mengambil gambar paruh atas dan bawah, yang kemudian masuk ke tahap *preprocessing* seperti *cropping*, *rescalling*, dan segmentasi. Fitur tradisional

seperti *Histogram of Oriented Gradients* (HOG) dan *Morphological Shape Descriptors* (MSD) diekstraksi, sementara fitur mendalam diperoleh menggunakan model *Convolutional Neural Network* (CNN) seperti VGG19, InceptionV3, dan ResNet50. Delapan algoritma machine learning, termasuk *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), dan *Random Forest* (RF), digunakan untuk mengklasifikasikan spesies. Hasil penelitian menunjukkan bahwa model ANN dengan fitur mendalam yang diekstraksi oleh VGG19 dari gambar paruh bawah mencapai akurasi pengujian tertinggi sebesar 91,14%, jauh lebih baik dibandingkan fitur tradisional yang hanya mencapai akurasi maksimum 73,09% dengan deskriptor hibrida (colour HOG + MSD).

Pattanayak *et al.*, (2024) melakukan penelitian untuk meningkatkan akurasi dalam klasifikasi penyakit *Alzheimer* melalui pendekatan *Capsule Network* (CapsNet) pada citra *Magnetic Resonance Imaging* (MRI). Penelitian ini mengembangkan model berbasis *deep learning* untuk mengklasifikasikan tahap penyakit *Alzheimer* secara lebih akurat dibandingkan metode konvensional. Metodologi yang digunakan dalam penelitian ini melibatkan analisis citra MRI yang diperoleh dari *Alzheimer's Disease Neuroimaging Initiative* (ADNI), yang mencakup tiga kategori, yaitu *Alzheimer's Disease* (AD), *Mild Cognitive Impairment* (MCI), dan *Cognitive Normal* (CN). Data diolah menggunakan pendekatan *deep learning*, di mana model CapsNet dibandingkan dengan *Convolutional Neural Networks* (CNN) dalam tugas klasifikasi. Hasil penelitian menunjukkan bahwa CapsNet memiliki akurasi 98.40%, lebih tinggi dibandingkan CNN yang mencapai 96.82%. Selain itu, CapsNet memiliki *recall* 98.60% dan AUC 99.42%, sementara CNN mencatat *recall* 96.90% dan AUC 98.60%. Evaluasi menggunakan *confusion matrix* menunjukkan bahwa CapsNet berhasil mengklasifikasikan AD dengan akurasi 98.6%, MCI dengan 98.2%, dan CN dengan 98.4%, menunjukkan keunggulannya dalam mempertahankan informasi spasial dan menangkap hubungan hirarkis antar fitur citra.

2.2 Landasan Teori

2.2.1 Cumi-Cumi

Cumi-cumi adalah jenis hewan laut yang termasuk dalam kelompok *cephalopoda*, dan menjadi salah satu jenis hewan laut yang paling berharga karena memiliki nilai gizi yang tinggi. Cumi-cumi biasanya disebut dengan nama ilmiah *Loligo sp.* dan *Uroteuthis chinensis*, yang merupakan spesies yang sering ditemukan di perairan Indonesia. Ciri khas cumi-cumi adalah bentuk tubuh dan tentakel yang panjang, serta kemampuan mereka untuk mengubah warna dan tekstur kulit mereka untuk menyesuaikan dan mempertahankan diri (Mudirta *et al.*, 2024).

Uroteuthis chinensis adalah salah satu dari banyak jenis cumi-cumi yang dikenal secara luas, yang banyak ditangkap di perairan Indonesia dan memiliki nilai ekonomis yang signifikan (Sitompul *et al.*, 2021). Selain itu, *Uroteuthis chinensis* juga sering diteliti karena biologi reproduksinya dan kesuksesan penetasan telurnya yang dipengaruhi oleh kadar garam di lingkungannya (Pirmansa *et al.*, 2020). Selain berfungsi sebagai predator untuk berbagai spesies ikan, cumi-cumi memainkan peran penting dalam ekosistem laut (Sitompul *et al.*, 2021).

2.2.2 Kesegaran Cumi-Cumi

Cumi-cumi segar tidak hanya memiliki rasa yang lebih baik, tetapi juga lebih aman untuk dimakan. Kesegaran cumi-cumi sangat penting untuk kualitas dan nilai jual produk perikanan ini. Penanganan cumi-cumi setelah ditangkap adalah salah satu faktor yang mempengaruhi kesegarannya. Untuk mempertahankan kesegaran cumi-cumi dan memperpanjang umur simpannya dapat dicapai dengan menyimpannya pada suhu yang didinginkan. Penanganan yang tepat setelah penangkapan, seperti pendinginan cepat, sangat penting untuk melindungi kualitas daging cumi-cumi dan menghentikan penyebaran bakteri (Asandri *et al.*, 2022).

Dalam konteks standarisasi mutu, Standar Nasional Indonesia (SNI) mendefinisikan cumi-cumi segar sebagai produk yang telah melalui proses penanganan higienis, aman untuk dikonsumsi, serta memenuhi persyaratan mutu untuk dapat dipasarkan di pasar domestik maupun luar negeri. Kelayakan ini

ditentukan melalui serangkaian pengujian, seperti uji *organoleptic* atau uji sensori, uji mikrobiologi, uji kimia, dan uji fisika. Kesegaran cumi-cumi dapat identifikasi dengan menggunakan parameter seperti warna, tekstur, dan bau. Untuk warna dapat dinilai berdasarkan warna kulit, warna daging dan mata cumi-cumi (Hien *et al.*, 2020).

Menurut SNI 2731:2010 penilaian untuk spesifikasi cumi segar dapat dilihat pada Tabel 2.1 berikut, dengan ambang batas nilai 7 sebagai indikator untuk menentukan cumi-cumi segar.

Tabel 2. 1 Indikator Kesegaran Cumi-Cumi

Spesifikasi	Nilai
1 Kenampakan	
• Warna spesifik jenis cumi-cumi, sangat cemerlang	9
• Warna spesifik jenis cumi-cumi, cemerlang	8
• Warna spesifik jenis cumi-cumi, putih sedikit krem, kurang cemerlang	7
• Warna spesifik jenis cumi-cumi, putih krem, agak pucat, kurang cemerlang	★ 6
• Warna spesifik jenis cumi-cumi, krem, pucat, kurang cemerlang	5
• Krem, agak kecoklatan, pucat	3
• Krem, kecoklatan, pucat	1
2 Bau	
• Spesifik cumi-cumi, sangat segar	9
• Segar	8
• Netral	7
• Mulai amis, sedikit busuk	5
• Amis, sedikit busuk	3
• Busuk	1
3 Tekstur	
• Kompak, padat, elastis	9
• Kompak, elastis	8
• Kompak, kurang elastis	7
• Tidak elastis, agak lunak	5
• Lunak	3
• Sangat lunak	1

2.2.3 Computer Vision

Computer vision telah mengalami kemajuan besar sejak awal perkembangannya pada tahun 1960-an dan sekarang merupakan salah satu bidang aplikasi kecerdasan buatan (AI) yang paling populer. Bidang ini berkaitan dengan pengembangan algoritma dan teknik yang memungkinkan komputer untuk memahami dan memproses informasi visual dari dunia nyata, seperti gambar dan video (Anderson *et al.*, 2022). Untuk keamanan publik, teknologi ini dapat digunakan untuk berbagai tujuan, termasuk pengenalan objek, pengolahan gambar medis, dan analisis video (Zhang *et al.*, 2022).

Pengenalan jaringan saraf konvolusional (CNN) adalah kemajuan besar dalam kecerdasan buatan yang telah mengubah cara komputer mengidentifikasi dan mengklasifikasikan gambar. CNN memungkinkan komputer untuk belajar dari data visual dalam skala besar, seperti dataset ImageNet, yang terdiri dari ribuan gambar dengan berbagai kelas (Liu *et al.*, 2020). Dengan menggunakan CNN, komputer dapat secara otomatis mengekstrak fitur dari gambar, yang sebelumnya memerlukan pemrograman manual (Li *et al.*, 2021). Hal ini memungkinkan pengembangan aplikasi yang lebih kompleks seperti pengenalan wajah, deteksi objek, dan segmentasi gambar (F. Cao *et al.*, 2022).

2.2.4 Deep Learning

Subbidang kecerdasan buatan (AI) yang dikenal sebagai *deep learning* berkonsentrasi pada pembuatan algoritma dan model yang memiliki kemampuan untuk belajar dari data yang sangat besar melalui struktur jaringan saraf yang dimodelkan. *Deep learning* telah mengubah banyak hal sejak diperkenalkan secara luas pada tahun 2010-an, seperti pengolahan bahasa alami, *computer vision*, dan analisis data (Dewi, 2024). *Deep learning* sangat bermanfaat untuk berbagai aplikasi, seperti pengenalan gambar dan diagnosis medis (Skourt & Nikolov, 2022).

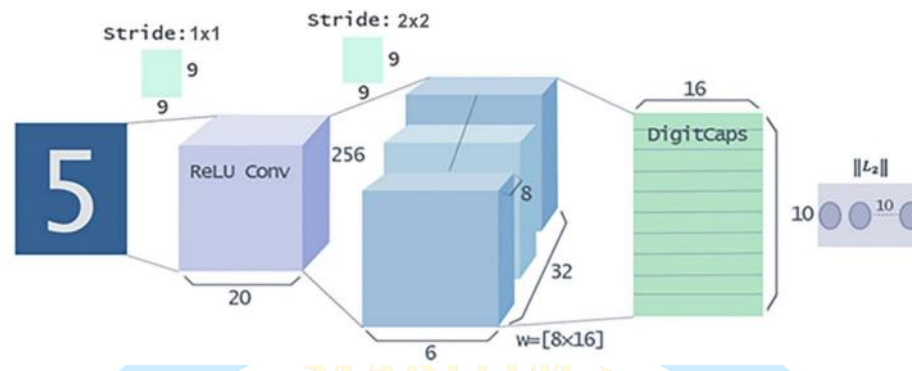
Pengenalan objek dan wajah adalah salah satu aplikasi *deep learning* paling terkenal. Jaringan saraf konvolusional (CNN) telah menjadi pilihan arsitektural utama dalam situasi ini. CNN memiliki kemampuan untuk secara otomatis mengekstrak fitur dari gambar, yang memungkinkan sistem untuk mengenali pola

dan objek dengan tingkat akurasi yang tinggi. Studi menunjukkan bahwa CNN dapat melakukan hal-hal yang lebih baik daripada metode konvensional, seperti klasifikasi gambar dan deteksi objek. Misalnya, teknologi ini digunakan oleh *Facebook* untuk mengembangkan sistem pengenalan wajah yang sangat akurat yang dapat mengidentifikasi orang dalam foto dengan sangat akurat (Pernando *et al.*, 2022).

2.2.5 Capsule Network (CapsNet)

Geoffrey Hinton dan timnya memperkenalkan arsitektur jaringan saraf yang disebut CapsNet pada tahun 2017. CapsNet dibuat dengan tujuan untuk mengatasi berbagai keterbatasan jaringan saraf konvolusional (CNN), terutama yang berkaitan dengan pengenalan objek yang lebih kompleks serta variasi dalam posisi dan orientasi objek dalam gambar. CapsNet menggunakan "kapsul" dalam konteks ini, yang merupakan kelompok neuron yang bekerja sama untuk mendeteksi fitur tertentu dari objek dan mengkodekan informasi tentang pose, orientasi, dan deformasi objek tersebut (Wang *et al.*, 2020).

CapsNet meningkatkan kemampuan pengenalan objek dibandingkan dengan CNN karena informasi tentang karakteristik objek biasanya dikumpulkan melalui lapisan konvolusi yang bertindak secara hierarkis. Namun, jika objek mengalami rotasi, translasi, atau perubahan skala, CNN seringkali tidak dapat mengenali objek. Sebaliknya, CapsNet menggunakan struktur kapsul, yang memungkinkannya untuk mengidentifikasi hubungan spasial antara fitur-fitur yang berbeda yang terdapat dalam objek. Menurut Altan (2020), setiap kapsul dapat mengeluarkan *output* yang menunjukkan kemungkinan keberadaan objek dan pose-nya. Ini membuat model lebih tahan terhadap variasi data.



Gambar 2. 1 Arsitektur CapsNet

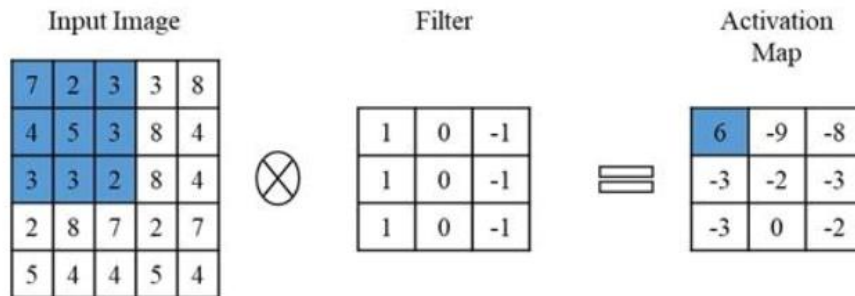
Sumber : (Huang & Zhou, 2020)

Arsitektur CapsNet yang ditampilkan pada Gambar 1 dimulai dengan lapisan konvolusi pertama (Conv1), yang menerima citra *input*. Lapisan ini menggunakan 256 kernel konvolusi berukuran 9x9 dengan *stride* 1 dan aktivasi ReLU, menghasilkan 256 *feature map* berukuran 20x20. *Feature map* ini kemudian menjadi *input* untuk lapisan *Primary Capsules*, yang merupakan lapisan kapsul konvolusi dengan 32 kanal, masing-masing menghasilkan vektor 8 dimensi (8D). *Primary Capsules* menerapkan kernel 9x9 dengan *stride* 2, sehingga menghasilkan grid 6x6 per kanal, atau total $32 \times 6 \times 6 = 1152$ kapsul, masing-masing dengan vektor 8D. *Output* dari *Primary Capsules* (u_i), dihubungkan ke lapisan DigitCaps melalui mekanisme *dynamic routing*. Lapisan DigitCaps terdiri dari 10 kapsul (satu untuk setiap kelas digit 0-9), masing-masing menghasilkan vektor 16D (v_j), dengan panjang vektor menunjukkan probabilitas keberadaan kelas tersebut.

2.2.5.1 Convolution Layer

Dalam CapsNet, lapisan konvolusi berfungsi untuk mengekstrak fitur dari gambar yang dimasukkan. Lapisan konvolusi ini biasanya terdiri dari beberapa filter yang bergerak di atas gambar untuk menghasilkan peta fitur, tetapi dalam konteks CapsNet, lapisan konvolusi ini berfungsi sebagai tahap awal untuk mengidentifikasi fitur dasar gambar, yang kemudian dilanjutkan ke lapisan kapsul

(Ali & Ali, 2021). Penelitian yang dilakukan oleh Celebi *et al.*, (2023) telah menunjukkan bahwa penggunaan dua lapisan konvolusi dapat cukup efektif dalam situasi ini.



Gambar 2. 2 Proses Konvolusi

Sumber : (Mostafa & Wu, 2021)

Dalam operasi konvolusi, *filter* atau kernel, adalah matriks kecil yang digunakan untuk mendeteksi pola tertentu dalam gambar. Setiap filter memiliki ukuran tertentu, seperti 3x3 atau 5x5, dan memiliki bobot yang dapat dilatih selama proses pelatihan. Ketika *filter* diterapkan pada gambar, ia melakukan operasi konvolusi dengan mengalikan elemen *filter* dengan elemen gambar yang sesuai dan menjumlahkan hasilnya untuk menciptakan nilai baru di peta fitur (Wang *et al.*, 2023). Berikut persamaan (2.1) untuk menghitung proses konvolusi:

$$Z(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i + m, j + n) \cdot K(m, n) \quad (2.1)$$

Dimana:

$Z(i, j)$ = Hasil konvolusi di posisi (i, j)

$X(i + m, j + n)$ = Nilai piksel dari input citra pada posisi $(i + m, j + n)$

$K(m, n)$ = Kernel (*filter*) konvolusi dengan ukuran $M \times N$

M dan N = Dimensi dari kernel konvolusi

Sedangkan untuk *output* dari operasi konvolusi atau biasa disebut dengan *feature map* dapat dihitung menggunakan persamaan (2.2) berikut :

$$Output_{size} = \frac{W - N + 2P}{S} + 1 \quad (2.2)$$

Dimana:

W = Ukuran *input*

P = Jumlah *padding*

N = Ukuran kernel *filter*

S = Jumlah *stride* atau pergeseran

2.2.5.2 Primary Capsule

Primary Capsule merupakan lapisan yang terdiri dari sejumlah kapsul yang digunakan untuk mengkodekan informasi yang telah diekstraksi. Data yang diterima merupakan lanjutan dari lapisan konvolusi. Dalam lapisan ini, setiap kapsul mengeluarkan vektor yang menunjukkan kemungkinan keberadaan fitur dan pose-nya. Selanjutnya, *output* dari kapsul primer dihubungkan ke kapsul yang lebih tinggi yaitu digit kapsul dengan menggunakan mekanisme *routing*. Ini memungkinkan model untuk memahami hubungan spasial antara bagian-bagian objek (Hollósi et al., 2023).

Primary Capsules dibentuk untuk menerapkan operasi konvolusi tambahan pada *feature map* dari *convolution layer*. Setiap vektor kapsul (u_i) dihasilkan melalui transformasi linier dari fitur *input* menggunakan matriks bobot, yang kemudian menjadi masukan untuk proses *routing*. Untuk mencari nilai vektor

Primary Capsules dengan transformasi linier dapat menggunakan persamaan (2.3) berikut:

$$u_i = W \cdot input_i \quad (2.3)$$

Dimana:

u_i : Vektor *output Primary Capsule* ke- i

W : Matriks bobot transformasi

$input_i$: Vektor fitur dari *feature map* pada posisi i

2.2.5.3 Dynamic Routing

Dynamic Routing merupakan mekanisme inti dalam CapsNet yang berfungsi untuk menghubungkan *Primary Capsules* ke *Digit Capsules*. Mekanisme ini menggantikan Teknik *pooling* yang ada pada jaringan saraf konvensional dengan tujuan utama adalah memastikan bahwa informasi dari *child capsule* dapat diteruskan ke *parent capsule* berdasarkan keselarasan (Sun et al., 2023).

Proses *routing* dilakukan secara iteratif, dengan jumlah iterasi tiga yang dianggap cukup untuk mencapai konvergensi, ditandai dengan stabilitas logit dan koefisien coupling setelah iterasi ketiga. Sebagaimana dibuktikan melalui analisis eksperimental pada dataset MNIST dan CIFAR10 (Sabour et al., 2017). Dengan langkah pertama adalah menghasilkan vektor prediksi (\hat{u}_{ji}) melalui setiap *Primary Capsules* untuk *Digit Capsules* menggunakan matriks bobot. Untuk menghitung vektor prediksi dapat menggunakan persamaan (2.4) berikut:

$$\hat{u}_{ji} = W_{ij} u_i \quad (2.4)$$

Dimana:

\hat{u}_{ji} : Vektor prediksi dari *Primary Capsule* i ke *Digit Capsule* j

W_{ij} : Matriks bobot transformasi

u_i : Nilai dari *primary capsule* ke- i

Selanjutnya, koefisien *coupling* (c_{ij}) diinisialisasikan dengan logit $b_{ij} = 0$ yang menunjukkan probabilitas prior untuk setiap koneksi. Nilai koefisien *coupling* (c_{ij}) dihitung menggunakan fungsi aktivasi *softmax* untuk memastikan jumlah

koefisien untuk setiap *Digit Capsule* sama dengan 1. Untuk menghitung koefisien *coupling* dapat menggunakan persamaan (2.5) berikut:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (2.5)$$

Dimana:

c_{ij} : Koefisien *coupling*

b_{ij} : Logit untuk koneksi i ke j

Kemudian, total input untuk setiap *Digit Capsule* (s_j) diperoleh dengan menjumlahkan vektor prediksi yang dikalikan dengan koefisien *coupling* masing-masing. Persamaan untuk menghitung total *input* menggunakan persamaan (2.6) berikut:

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \quad (2.6)$$

Dimana:

s_j : Total *input* *Digit Capsule* j

c_{ij} : Koefisien *coupling*

$\hat{u}_{j|i}$: Vektor prediksi dari *Primary Capsule* i ke *Digit Capsule* j

Total *input* ini lalu dinormalisasi menggunakan *squash function* untuk menghasilkan vektor *output* *Digit Capsule* (v_j), yang panjangnya mencerminkan probabilitas keberadaan kelas.

Terakhir, logit (b_{ij}) diperbarui berdasarkan keselarasan (*agreement*) antara vektor prediksi dan *output* *Digit Capsule*, yang diukur dengan *dot product*. Persamaan untuk pembaruan logit adalah:

$$b_{ij} = b_{ij} + \hat{u}_{j|i} \cdot v_j \quad (2.7)$$

Di mana $\hat{u}_{j|i} \cdot v_j$ adalah produk skalar yang menunjukkan tingkat keselarasan. Proses ini diulangi untuk setiap iterasi hingga logit dan koefisien *coupling* konvergen, memastikan alokasi fitur yang optimal antar kapsul.

2.2.5.4 Activation Function

Activation function atau fungsi aktivasi adalah bagian penting dari jaringan saraf, termasuk *Capsule Network* (CapsNet), yang menentukan *output* neuron berdasarkan *input* yang diterima. Fungsi aktivasi membantu jaringan belajar dan menangkap pola data yang kompleks. Fungsi aktivasi CapsNet mengubah sinyal *input* menjadi *output* yang memungkinkan pengambilan keputusan. Fungsi ini membantu jaringan menangkap non-linearitas dalam data, ini sangat penting untuk tugas-tugas pengenalan pola dan klasifikasi. Dalam berbagai aplikasi, seperti pengenalan objek dan analisis citra, pemilihan fungsi aktivasi yang tepat dapat berdampak pada kinerja dan efektivitas model CapsNet. Beberapa fungsi aktivasi yang umum digunakan dalam CapsNet adalah *Rectified Linear Unit* (ReLU), *Softmax*, dan *Squash*.

1. Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

ReLU adalah salah satu fungsi aktivasi yang paling populer dan banyak digunakan dalam berbagai arsitektur jaringan saraf, termasuk CapsNet (Yuen *et al.*, 2021). Fungsi ini didefinisikan dalam persamaan (2.8) berikut :

$$f(x) = \max(0, x) \quad (2.8)$$

ReLU mengubah semua nilai negatif menjadi nol, dan nilai positif tetap sama. ReLU memungkinkan konvergensi yang lebih cepat selama pelatihan dan mengurangi masalah *vanishing gradient*, yang sering terjadi pada fungsi aktivasi lain seperti sigmoid atau tanh (Bodyanskiy & Kostiuk, 2022).

2. Fungsi Aktivasi *Softmax*

Fungsi aktivasi *softmax* pada CapsNet sering digunakan pada lapisan akhir untuk mengklasifikasikan objek berdasarkan output kapsul yang lebih tinggi, seperti DigitCaps. Dengan menggunakan *softmax*, CapsNet dapat memberikan probabilitas untuk setiap kelas, yang memudahkan pengambilan keputusan klasifikasi (Marchisio *et al.*, 2022). Fungsi aktivasi softmax dapat dilihat pada persamaan (5).

3. Fungsi Aktivasi *Squash*

Fungsi aktivasi *squash* adalah bagian penting dari arsitektur Capsule Networks, yang dirancang untuk mengatasi beberapa keterbatasan pada fungsi aktivasi tradisional. Dalam arsitektur *Capsule Network*, fungsi ini membatasi panjang vektor aktivasi kapsul agar tetap dalam rentang tertentu, sehingga informasi tentang objek yang diidentifikasi dapat direpresentasikan dengan lebih baik dan membantu menjaga informasi spasial dan hubungan antar bagian dari jaringan (Adu et al., 2021). Fungsi ini didefinisikan pada persamaan (2.9) berikut:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (2.9)$$

Dimana :

v_j : Vektor *output* dari kapsul j

s_j : Total *input* ke kapsul

$\|s_j\|$: panjang dari vektor s_j

2.2.5.5 Digit Capsule

Digit *Capsule* (DigitCaps) merupakan lapisan kapsul yang lebih tinggi dalam arsitektur CapsNet yang berfungsi untuk mengklasifikasikan objek berdasarkan fitur yang diekstrak dari lapisan sebelumnya. DigitCaps menerima *input* dari kapsul primer dan mengeluarkan *output* yang menunjukkan kemungkinan keberadaan objek dalam kelas tertentu.

2.2.6 Python

Python adalah bahasa pemrograman tingkat tinggi yang terkenal karena sintaksisnya yang mudah dipahami dan kemudahan penggunaannya. Python telah menjadi salah satu bahasa pemrograman paling populer di dunia, terutama dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan, dan pengembangan web (Cao et al., 2023). Python juga mendukung berbagai

paradigma pemrograman, seperti pemrograman imperatif, pemrograman berorientasi objek, dan pemrograman fungsional (Wang *et al.*, 2020).

Python dibuat pertama kali oleh Guido van Rossum dan diluncurkan pada tahun 1991. Python telah berkembang pesat sejak saat itu dan memiliki ekosistem yang kaya dengan banyak pustaka dan *framework* yang mendukung berbagai aplikasi. Python memiliki komunitas yang besar dan aktif, yang berkontribusi pada pengembangan pustaka dan alat baru secara konsisten (Roy *et al.*, 2023). salah satu alasan utama popularitasnya adalah kemudahan penggunaan dan pembelajarannya (Wang *et al.*, 2020).

2.2.7 Google Colaboratory

Google Colab adalah platform berbasis *cloud* yang dikembangkan oleh Google yang memungkinkan pengguna menjalankan kode Python dalam *notebook Jupyter*. Sebagai alat yang dimaksudkan untuk analisis data dan pembelajaran mesin, Google Colab memungkinkan pengguna memanfaatkan kapasitas komputasi tinggi seperti CPU dan GPU secara gratis, memfasilitasi akses yang lebih baik ke sumber daya komputasi yang sebelumnya hanya dapat diakses oleh institusi besar atau organisasi kecil (Abdillah, 2023).

Kemampuan untuk menjalankan kode Python tanpa memerlukan instalasi perangkat lunak di komputer lokal adalah salah satu fitur utama Google Colab. Platform ini sangat mudah digunakan dengan hanya menggunakan akun Google (Bravenec & Ward, 2023). Selain itu, NumPy, Pandas, TensorFlow, dan PyTorch adalah beberapa pustaka Python populer untuk pembelajaran mesin dan analisis data yang disediakan oleh Google Colab (Nagi *et al.*, 2023).

2.2.8 Confusion Matrix

Confusion matrix adalah alat evaluasi yang digunakan dalam *machine learning*. khususnya untuk klasifikasi data yang diawasi (*supervised classification*) atau untuk menentukan perilaku model klasifikasi. Direpresentasikan dalam baris dan kolom, dengan baris mewakili kelas aktual, sedangkan kolom mewakili kelas yang diprediksi oleh model (Hasnain *et al.*, 2020).

Menurut Hasnain *et al.*, (2020), *Confusion matrix* untuk masalah klasifikasi biner direpresentasikan sebagai matriks 2x2 dengan komponen utama sebagai berikut:

1. True Positive (TP) : Jumlah kasus di mana model dengan benar memprediksi kelas positif.
2. True Negative (TN) : Jumlah kasus di mana model dengan benar memprediksi kelas negatif.
3. False Positive (FP) : Jumlah kasus di mana model salah memprediksi kelas positif.
4. False Negative (FN) : Jumlah kasus di mana model salah memprediksi kelas negatif.

Pengukuran akurasi, presisi, *recall*, dan skor F1 dapat menggunakan persamaan berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.10)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2.11)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.12)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (2.13)$$

2.2.9 Parameter yang Digunakan

Dalam proses pelatihan model, terdapat beberapa parameter yang berperan penting dan memiliki pengaruh signifikan terhadap kinerja model, konvergensi, stabilitas, dan generalisasi, seperti di antaranya adalah *optimizer*, *learning rate*, *epoch*, dan ukuran *batch*.

Optimizer berfungsi untuk memperbarui bobot model untuk meminimalkan fungsi kerugian atau *loss*. Adam (*Adaptive Moment Estimation*) merupakan salah satu *optimizer* populer yang memungkinkan konvergen lebih cepat dibandingkan dengan *optimizer* lainnya. Hal ini dikarenakan Adam menggabungkan konsep adaptasi laju pembaruan dan momentum dengan nilai rata-rata dan variansi dari gradien yang dihitung. Tujuan Adam adalah untuk mengurangi efek fluktuasi gradien, yang berperan penting untuk menyelesaikan pelatihan model dalam waktu

yang lebih singkat (Wen et al., 2023). Berdasarkan penelitian yang dilakukan oleh Anggara *et al.*, (2023), *optimizer* Adam lebih sering mencapai konvergensi lebih cepat dan menghasilkan akurasi yang lebih tinggi dibandingkan dengan *optimizer* lain seperti SGD (*Stochastic Gradient Descent*) dan RMSprop.

Learning rate merupakan parameter yang menunjukkan seberapa besar perubahan bobot model berdasarkan *error* yang dihitung selama setiap iterasi pelatihan. Pemilihan nilai *learning rate* yang tepat perlu dilakukan karna dapat mempengaruhi performa model. Nilai yang terlalu kecil dapat menyebabkan model memerlukan waktu yang lama untuk konvergensi dan berisiko terjebak minimum lokal, sementara nilai yang terlalu tinggi dapat menyebabkan divergensi dan melampaui solusi ideal (Martínez-Fernandez et al., 2023).

Epoch adalah istilah yang mengacu pada satu siklus pelatihan penuh di mana semua data pada dataset digunakan sekali untuk memperbarui bobot model. Untuk mencapai konvergensi yang optimal, pemilihan jumlah *epoch* perlu diperhatikan dengan cermat. Terlalu sedikit *epoch* dapat menyebabkan model *underfitting*, sementara terlalu banyak *epoch* dapat menyebabkan *overfitting*, di mana model belajar terlalu banyak pada data latih tetapi tidak dapat generalisasi pada data baru (Shen et al., 2021).

Batch size adalah parameter yang menunjukkan jumlah sampel pelatihan yang dipropagasi melalui jaringan saraf dalam satu iterasi. Penggunaan ukuran *batch* yang besar, menghasilkan estimasi gradien yang lebih akurat dan stabil yang mengarah pada jalur konvergensi lebih mulus dan cepat dalam waktu komputasi per *epoch*, hal ini dikarenakan *optimizer* menghitung gradien berdasarkan lebih banyak sampel pada setiap langkah. Sementara penggunaan ukuran *batch* yang lebih kecil dapat memberikan informasi gradien yang lebih sering, namun dapat menghasilkan fluktuasi dalam pembaruan bobot yang dapat mengganggu konvergensi.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan dataset yang digunakan terdiri dari 3 kelas, yakni ‘Segar’, ‘Tidak Segar’, dan ‘Non Cumi’. Data untuk kelas ‘Segar’ dan ‘Tidak Segar’ dikumpulkan secara mandiri di mana sampel cumi-cumi ditempatkan pada suhu ruang dan diobservasi secara berkala. Penentuan label untuk setiap sampel dilakukan dengan menggunakan pengujian organoleptik atau uji sensori. Kualitas setiap sampel dinilai menggunakan *scoresheet* yang mengacu pada kriteria dari Standar Nasional Indonesia (SNI), di mana sampel dengan skor mutu 7 ke atas diklasifikasikan sebagai “Segar” dan di bawah 7 sebagai “Tidak Segar”. Sementara, data kelas “Non Cumi” dikumpulkan dari sumber internet. Hasil dari model yang telah dirancang dan dilatih akan dievaluasi menggunakan *confusion matrix* sehingga menghasilkan nilai *accuracy*, *recall*, *precision* dan *f1-score*.

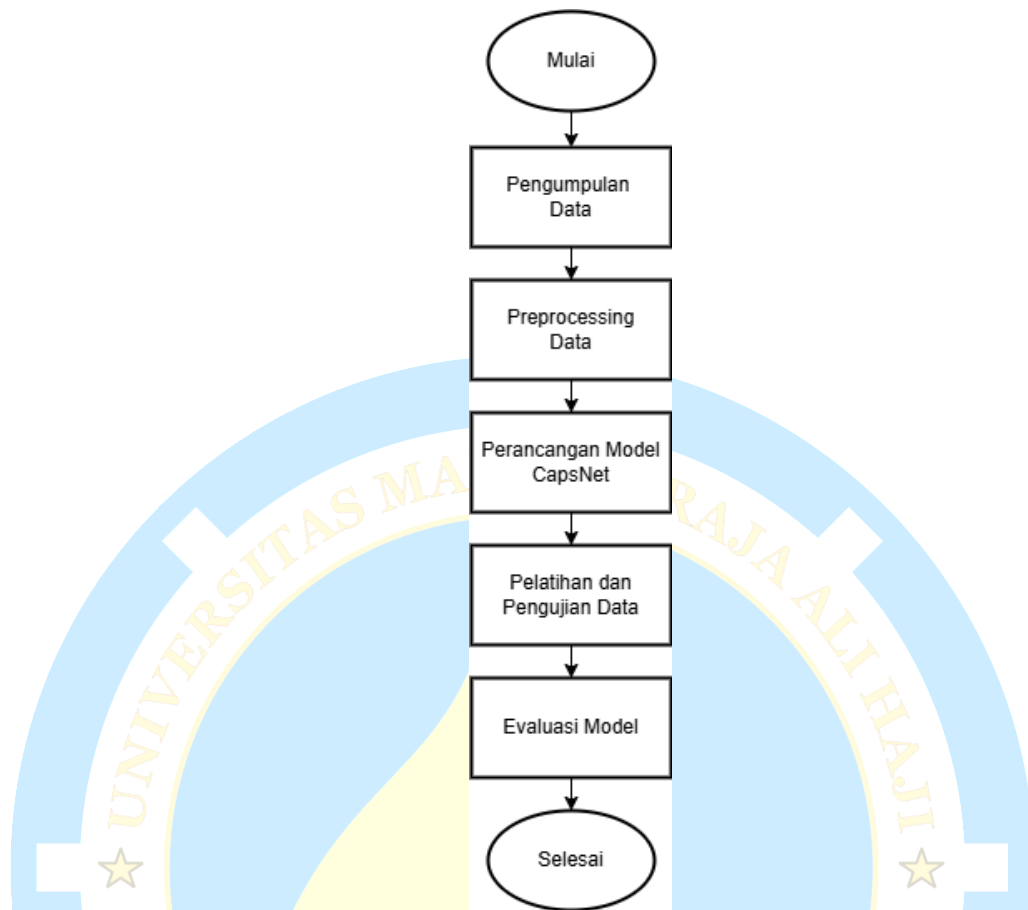
3.2 Waktu dan Tempat Penelitian

Penelitian terkait identifikasi kesegaran cumi-cumi menggunakan *Capsule Network* (CapsNet) dilakukan selama 5 bulan, terhitung sejak Februari 2025 sampai dengan Juni 2025. Proses pengumpulan data dilakukan di daerah tempat tinggal penulis dengan membeli objek dari warung atau tempat penjualan cumi-cumi terdekat, lalu dilakukan pengambilan gambar selama interval 1 jam dengan kondisi cumi-cumi dibiarkan di suhu ruang.

3.3 Tahapan Penelitian

Penelitian ini diawali dengan proses pengumpulan data. Selanjutnya data-data yang telah dikumpulkan tersebut masuk ke tahapan *preprocessing*. Data yang telah melalui tahapan *preprocessing* selanjutnya digunakan dalam tahap pengembangan model, termasuk pelatihan dan pengujian. Hingga pada akhirnya hasil dari model yang dibangun akan dianalisis dan dilakukan evaluasi.

Langkah-langkah atau tahapan pada penelitian ini diilustrasikan pada *flowchart* berikut:



Gambar 3. 1 Tahapan Penelitian

Berdasarkan *flowchart* di atas, penelitian yang dilakukan dijabarkan melalui tahapan-tahapan berikut.

1. **Pengumpulan Data**

Tahapan awal yang dilakukan dalam penelitian ini adalah pengumpulan dataset cumi-cumi yang diambil melalui pembelian di tempat penjualan cumi-cumi di daerah Teluk Bakau, Kecamatan Gunung Kijang, Bintan, dengan data awal yang dibeli berjumlah 30 ekor cumi-cumi.

2. **Preprocessing Data**

Tahapan ini melibatkan proses-proses untuk persiapan data mentah yang telah dikumpulkan, seperti *labelling*, *cropping*, *resize*, dan augmentasi data agar siap digunakan untuk proses selanjutnya.

3. Perancangan Model CapsNet

Tahap ini merupakan proses perancangan arsitektur model *Capsule Network* (CapsNet) yang akan digunakan untuk melatih dan menguji data citra cumi-cumi yang telah disiapkan.

4. Pelatihan dan Pengujian Data

Data yang telah melalui *preprocessing* digunakan untuk melatih (*training*) dan menguji (*testing*) model CapsNet yang telah dirancang untuk mengetahui performanya.

5. Evaluasi Model

Setelah model selesai dilatih dan diuji, tahap ini dilakukan untuk menganalisis dan mengevaluasi hasil performa model secara keseluruhan menggunakan *confusion matrix*.

3.4 Studi Pustaka

Tahap ini dilakukan dengan tujuan untuk memperoleh informasi-informasi berupa landasan teoritis terkait penelitian terdahulu dari para ahli ataupun penulis mengenai cumi-cumi, kesegaran cumi-cumi, kesegaran ikan, klasifikasi kesegaran, *deep learning*, CNN dan *Capsule Network* (CapsNet). Hal ini dilakukan agar mendapat gambaran terkait apa yang akan diteliti nantinya. Studi pustaka dilakukan dengan memanfaatkan berbagai sumber ilmiah, termasuk jurnal penelitian nasional, jurnal penelitian internasional, dan sumber-sumber lain yang dijadikan referensi.

3.5 Alat dan Instrumen Penelitian

Berikut alat yang digunakan dalam mendukung berjalannya penelitian ini:

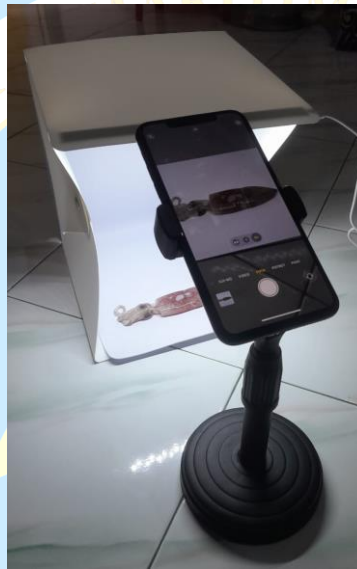
<i>Operating System</i>	: Windows 10 Home Single Language
<i>Processor</i>	: AMD Ryzen 3 5300U with Radeon Graphics 2.60 GHz
<i>Memory</i>	: RAM 8GB
<i>Smartphone</i>	: iPhone 11 Pro Max

Adapun perangkat lunak yang digunakan dalam mendukung implementasi sistem yang akan dibuat sebagai berikut:

1. *Python*
2. *Google Colab*
3. *Visual Studio Code*

3.6 Pengumpulan Data

Pengambilan gambar dilakukan menggunakan kamera HP dengan rasio 1:1 dalam *photobox* studio mini dengan latar belakang putih dan dilengkapi dengan LED sehingga pencahayaannya konsisten. Jarak pengambilan gambar adalah 15-20 cm menggunakan bantuan *tripod* untuk menjaga konsistensi jarak tiap citra dataset.



Gambar 3. 2 Pengambilan Gambar Dataset

Prosedur pengumpulan data pada penelitian ini telah disempurnakan berdasarkan hasil diskusi dan konsultasi dengan dosen pakar di bidang Teknologi Hasil Perikanan, Bapak Aidil Fadli Ilhamdy, S.Pi, M.Si untuk memastikan validitas metodologis. Pengumpulan data untuk kelas “Segar” dan “Tidak Segar” dilakukan secara mandiri melalui pembelian di tempat penjualan cumi di daerah tempat tinggal penulis. Selanjutnya cumi-cumi ditempatkan pada suhu ruang dan diobservasi secara berkala dengan interval setiap 1 jam. Pada setiap sesi observasi, digunakan 3 sampel cumi-cumi yang diambil secara acak dari stok dan diberi label, diuji 1 kali dan tidak digunakan lagi untuk observasi selanjutnya. Hal ini dilakukan untuk memastikan bahwa penurunan mutu kesegaran disebabkan oleh degradasi alami, bukan akibat kerusakan fisik dari penanganan berulang.



Gambar 3. 3 Pelabelan Cumi Uji Organoleptik

Penentuan label untuk setiap sampel didasarkan pada uji organoleptik atau uji sensori dengan mengacu pada lembar skor (*scoresheet*) dari Standar Nasional Indonesia (SNI). Penilaian untuk setiap sampel dilakukan oleh beberapa orang panelis untuk meningkatkan objektivitas dan mengurangi potensi bias subjektif, di mana skor akhir yang digunakan adalah nilai rata-rata dari seluruh panelis. Parameter yang dinilai mencakup kenampakan, bau, dan tekstur seperti yang ditampilkan pada Tabel 2.2. Sampel diklasifikasikan sebagai “Segar” jika memperoleh skor mutu rata-rata 7 atau lebih tinggi, dan sebagai “Tidak Segar” jika skornya di bawah 7. Sedangkan data untuk kelas “Non Cumi” dikumpulkan dari sumber-sumber di internet dan *Kaggle*, yang berisi berbagai objek acak selain cumi, seperti bangunan, buah-buahan, pemandangan gunung, laut, sampah plastik, kaca, dan sebagainya. Dalam penelitian ini parameter kesegaran yang digunakan adalah fokus kepada tekstur dan warna tubuh cumi-cumi seperti pada Gambar 3.4 dan Gambar 3.5 berikut.



Gambar 3. 4 Citra Cumi Segar



Gambar 3. 5 Citra Cumi Tidak Segar

3.7 Pembagian Dataset

Dataset awal yang dikumpulkan melalui pembelian berjumlah 30 ekor cumi-cumi, lalu setelah dilakukan pengujian organoleptik, jumlah dataset yang terkumpul adalah 12 cumi segar dan 18 cumi tidak segar. Selanjutnya setiap sampel cumi-cumi diambil gambar dengan 3 sudut berbeda, sehingga menghasilkan dataset awal yang berjumlah 36 data segar dan 54 data tidak segar. Dengan penerapan teknik augmentasi seperti *flip horizontal*, *random zoom*, dan rotasi sudut 180° menghasilkan 216 citra untuk kelas “Tidak Segar”. Sedangkan pada kelas “Segar” dilakukan *oversampling* dengan augmentasi tambahan seperti rotasi sudut 90° dan 270° . Sehingga menghasilkan total data yang seimbang untuk masing-masing kelas, termasuk kelas “Non Cumi” yaitu 216 data. Total data keseluruhan yang digunakan pada penelitian ini berjumlah 648 data, dibagi menjadi 3 bagian, yaitu 70% data latih, 20% data validasi dan 10% untuk data uji. Pembagian data dapat dilihat pada tabel berikut.

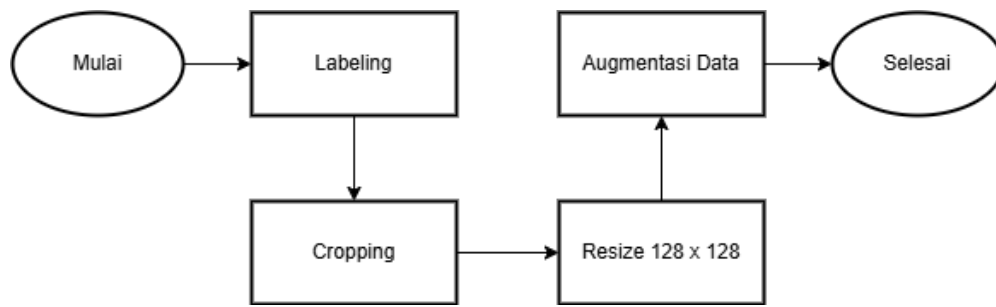
Tabel 3. 1 Pembagian Dataset

Pembagian		Segar	Tidak Segar	Non Cumi
Data latih	449	150	148	151
Data validasi	125	42	40	43
Data uji	74	24	28	22

3.8 Analisis dan Perancangan

3.8.1 Analisis Data

Penelitian ini berfokus pada identifikasi kesegaran cumi-cumi berdasarkan warna dan tekstur tubuh. Sebelum dataset digunakan dalam pelatihan dan pengujian, terlebih dahulu melalui tahapan *preprocessing*. Berikut adalah tahapan yang dilakukan dalam *preprocessing* data.



Gambar 3. 6 Tahapan *Preprocessing* Data

1. *Labeling*

Sebagai langkah awal, proses *labeling* digunakan untuk mengubah dataset berdasarkan parameter kesegaran, yaitu kelas “Segar” dan “Tidak Segar”. Tujuan pelabelan adalah mempermudah pengorganisasian data dan identifikasi kelas. Aplikasi *Advanced Renamer Tools* akan digunakan sebagai alat bantu dalam proses pelabelan setiap citra pada dataset, yang akan menghasilkan nama file yang konsisten seperti *segar1.jpg*, *segar2.jpg* dan seterusnya.

2. *Cropping*

Proses ini bertujuan untuk menghilangkan area pada gambar yang tidak relevan dan hanya berfokus pada bagian tubuh cumi-cumi sesuai pada batasan masalah yang diterapkan. Untuk mempertahankan keseragaman rasio tiap citra dataset, proses *cropping* dilakukan dengan menggunakan rasio persegi atau 1:1.



Gambar 3. 7 Citra Cumi Sebelum *Cropping*



Gambar 3. 8 Citra Cumi Setelah *Cropping*

3. *Resize* 128 x 128

Ukuran gambar diubah menjadi 128x128 piksel dengan tujuan mencapai keseragaman resolusi yang optimal. Penyesuaian atau perubahan ukuran ini bertujuan untuk memastikan bahwa setiap gambar memiliki dimensi yang konsisten. Dengan ukuran yang seragam, model pembelajaran mesin akan lebih mudah untuk menganalisis karena variabilitas dimensi yang tidak penting dihilangkan dari input data serta mengurangi kompleksitas komputasi. ★



Gambar 3. 9 Citra Cumi Sebelum *Resize*



Gambar 3. 10 Citra Cumi Setelah *Resize*

4. Augmentasi Data

Langkah terakhir yaitu augmentasi data, yang diterapkan dengan tujuan memperluas variasi dan meningkatkan jumlah dataset serta mengurangi resiko *overfitting*. Teknik augmentasi data yang diterapkan pada dataset penelitian ini seperti *flip horizontal*, *random zoom*, rotasi 90°, 180°, dan 270° menggunakan *library* dari python yaitu *Albumentations*.



Gambar 3.11
Flip Horizontal



Gambar 3.12
Random Zoom



Gambar 3.13
Rotasi 90°



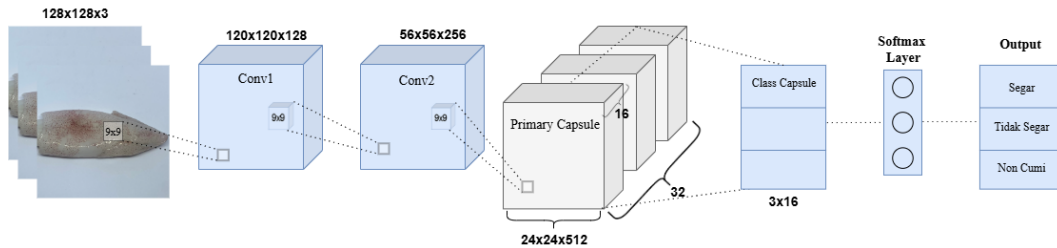
Gambar 3.14
Rotasi 180°



Gambar 3.15
Rotasi 270°

3.8.2 Perancangan Model *Capsule Network* (CapsNet)

Arsitektur model *Capsule Network* (CapsNet) yang dirancang disesuaikan untuk klasifikasi biner, sesuai dengan masalah dalam penelitian ini, yaitu mengidentifikasi kesegaran cumi-cumi berdasarkan citra RGB berukuran 128x128 piksel.



Gambar 3. 16 Rancangan Arsitektur Model CapsNet

Gambar 3.16 di atas menggambarkan arsitektur model *Capsule Network* (CapsNet) yang diimplementasikan untuk klasifikasi citra cumi-cumi ke dalam tiga kategori kelas, yaitu “Segar”, “Tidak Segar”, dan “Non Cumi”. Arsitektur ini diawali dengan dua buah lapisan konvolusional yang berfungsi untuk mengekstraksi fitur-fitur awal dari citra *input*. Kemudian, hasil ekstraksi ini diteruskan ke lapisan *Primary Capsule* yang terdiri dari sejumlah kapsul vektor berdimensi 16, di mana setiap kapsul merepresentasikan fitur spasial pada citra. Selanjutnya, keluaran dari *Primary Capsule* akan diproses oleh *Class Capsule*, yang merupakan lapisan kapsul yang merepresentasikan masing-masing kelas. Sebagai tahap akhir, *output* dari *Class Capsule* kemudian dimasukkan ke dalam *Softmax Layer* untuk menghasilkan probabilitas prediksi untuk setiap kelas. Rincian arsitektur model dapat dilihat pada tabel berikut.

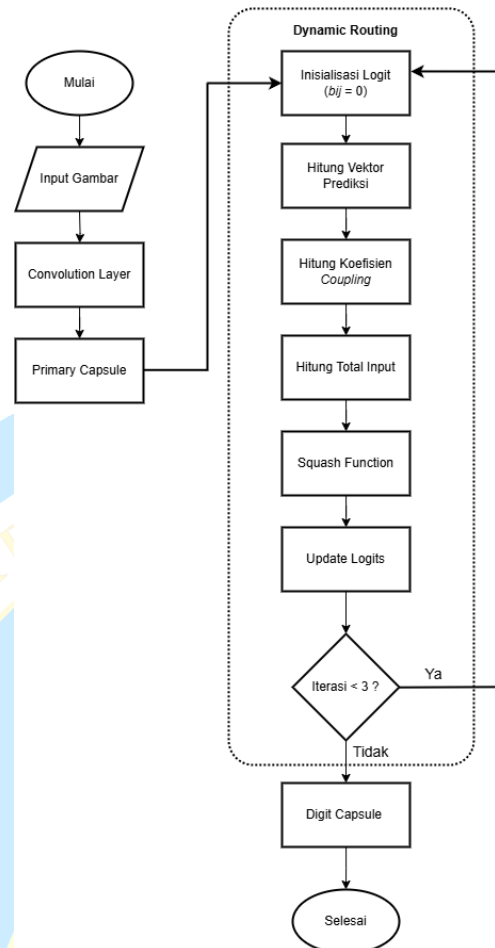
Tabel 3. 2 Rincian Arsitektur Model

Layer	Input	Output
Input Layer	(128, 128, 3)	(128, 128, 3)
Conv1	(128, 128, 3)	(120, 120, 128)
Conv2	(120, 120, 128)	(56, 56, 256)
Primary Capsule	(56, 56, 256)	(24, 24, 512)
Reshape	(24, 24, 512)	(18432, 16)
Class Capsule	(18432, 16)	(3, 16)
Flatten	(3, 16)	(48)
Dense	(48)	(3)

Berdasarkan rincian arsitektur model pada Tabel 3.2, model menerima *input* citra RGB berukuran 128x128 piksel. Lapisan konvolusional pertama dengan 64 filter berukuran 9x9, *stride* 1, dan aktivasi ReLU, menghasilkan *output feature map* sebesar 120x120x128. Selanjutnya, pada lapisan konvolusional kedua menggunakan 128 filter 9x9 dengan *stride* 2, menghasilkan *output* sebesar 56x56x256. *Output* dari lapisan konvolusional 2 dilanjutkan ke *Primary Capsule* dengan 512 filter, hasil dari perkalian antara jumlah kapsul yaitu 32 dengan dimensi kapsul yaitu 16, dengan kernel 9x9 dan *stride* 2, menghasilkan *output* sebesar 24x24x512. Selanjutnya dilakukan *reshaping* menjadi 576 kapsul (24x24), yang berdimensi masing-masing 16, sehingga menghasilkan bentuk $(576 \times 32, 16) = (18432, 16)$. Vektor-vektor ini kemudian dilewatkan ke *Class Capsule*, yang terdiri dari 3 kapsul, sesuai dengan jumlah kelas yang ada dengan dimensi vektor masing-masing 16. Untuk menghasilkan probabilitas kelas akhir, *output* dari *Class Capsule Layer* diratakan menjadi 1 vektor panjang. Vektor ini kemudian dimasukkan ke lapisan *dense* terakhir dengan 3 unit dan fungsi aktivasi *softmax* untuk menghasilkan prediksi probabilitas dari ketiga kelas.

3.8.3 Perhitungan Manual *Capsule Network* (CapsNet)

Adapun *flowchart* metode *Capsule Network* (CapsNet) yang menggambarkan bagaimana metode ini bekerja dan menjadi dasar perhitungan manual adalah sebagai berikut:

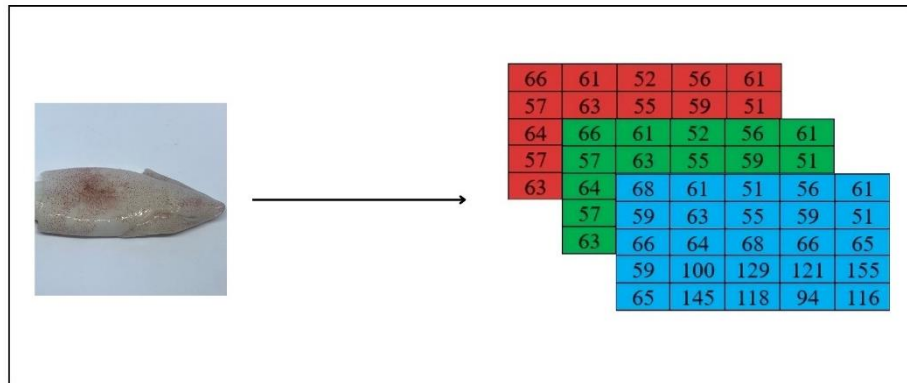


Gambar 3. 17 Flowchart Metode CapsNet

Langkah-langkah perhitungan manual pada metode *Capsule Network* (CapsNet) berdasarkan *flowchart* pada Gambar 3.17 dijelaskan sebagai berikut.

Langkah 1 : Convolution Layer

Tahap pertama yang dilakukan sesuai dengan arsitektur CapsNet adalah proses *input layer*. Citra yang digunakan merupakan citra yang telah di *resize* sebelumnya sehingga ukurannya menjadi 128x128x3. Pada tahap ini, gambar akan dikonversi menjadi array 3 dimensi, Di mana setiap *channel* merepresentasikan intensitas warna RGB.



Gambar 3.18 *Input Layer*

Data nilai RGB (merah, hijau, dan biru) untuk setiap piksel akan diproses melalui konvolusi. Lapisan konvolusi akan menggunakan filter untuk mengekstraksi fitur *feature map* yang ada pada gambar. Sebagai contoh, Gambar 3.19 menunjukkan *input* gambar pada *channel* warna merah. Selanjutnya, data gambar akan di konvolusi dengan *filter* yang telah dipilih sebelumnya. *Filter* ini terdiri dari blok atau kubus yang ukurannya lebih kecil dibandingkan dengan input gambar, tetapi kedalamannya sesuai dengan kedalaman input gambar yang diproses.

66	61	52	56	61
57	63	55	59	51
64	63	67	65	64
57	98	127	119	153
63	142	114	90	112

Gambar 3. 19 *Input Channel Red*

Selanjutnya citra yang sudah diinputkan masuk pada tahap *feature extractor* untuk mengekstraksi fitur dari citra. Konvolusi pertama menggunakan input citra berukuran 128x128,3 dengan jumlah filter 64 serta kernel yang digunakan berukuran 3x3. Kernel yang digunakan adalah sebagai berikut:

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

Gambar 3. 20 Contoh Filter

Citra *input* 5x5 digunakan untuk menyederhanakan perhitungan manual dalam proses konvolusi dan hanya melampirkan perhitungan pada *channel Red* (R) sebagai gambaran bagaimana proses konvolusi bekerja. Dalam proses ini, filter akan melintasi citra *input* dengan menghitung perkalian elemen antara nilai-nilai filter dan bagian citra yang sesuai, lalu menjumlahkannya untuk menghasilkan *output* dari konvolusi tersebut. Adapun pada proses konvolusi layer pertama menggunakan persamaan (2.1) yang menggambarkan bagaimana operasi konvolusi ini berjalan.

$$\begin{bmatrix} 66 & 61 & 52 & 56 & 61 \\ 57 & 63 & 55 & 59 & 51 \\ 64 & 63 & 67 & 65 & 64 \\ 57 & 98 & 127 & 119 & 153 \\ 63 & 142 & 114 & 90 & 112 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 76 & & \\ & & \\ & & \end{bmatrix}$$

5x5 3x3 3x3

Gambar 3. 21 Hasil Konvolusi Posisi Ke-1

Pada Gambar 3.21 menunjukkan proses konvolusi *channel Red* (R) pada posisi ke-1 dengan *stride* atau jumlah pergeseran bernilai 1. Hasil konvolusi nya adalah 76. Nilai tersebut didapatkan atas perkalian antara piksel citra *input* dengan nilai kernel, yaitu:

$$(66 \times 1) + (61 \times -1) + (52 \times 1) + (57 \times -1) + (63 \times 1) + (55 \times -1) + (64 \times 1) + (63 \times -1) + (67 \times 1) = 76$$

Selanjutnya proses ini diulang untuk setiap bagian citra yang dilalui oleh *filter*. Proses perhitungan lebih lengkap pada konvolusi *channel Red (R)* dapat dilihat sebagai berikut.

- Posisi 1 : $(66 \times 1) + (61 \times -1) + (52 \times 1) + (57 \times -1) + (63 \times 1) + (55 \times -1) + (64 \times 1) + (63 \times -1) + (67 \times 1) = 76$
- Posisi 2 : $(61 \times 1) + (52 \times -1) + (56 \times 1) + (63 \times -1) + (55 \times 1) + (59 \times -1) + (63 \times 1) + (67 \times -1) + (65 \times 1) = 59$
- Posisi 3 : $(52 \times 1) + (56 \times -1) + (61 \times 1) + (55 \times -1) + (59 \times 1) + (51 \times -1) + (67 \times 1) + (65 \times -1) + (64 \times 1) = 76$
- Posisi 4 : $(57 \times 1) + (63 \times -1) + (55 \times 1) + (64 \times -1) + (63 \times 1) + (67 \times -1) + (57 \times 1) + (98 \times -1) + (127 \times 1) = 67$
- Posisi 5 : $(63 \times 1) + (55 \times -1) + (59 \times 1) + (63 \times -1) + (67 \times 1) + (65 \times -1) + (98 \times 1) + (127 \times -1) + (119 \times 1) = 96$
- Posisi 6 : $(55 \times 1) + (59 \times -1) + (51 \times 1) + (67 \times -1) + (65 \times 1) + (64 \times -1) + (127 \times 1) + (119 \times -1) + (153 \times 1) = 142$
- Posisi 7 : $(64 \times 1) + (63 \times -1) + (67 \times 1) + (57 \times -1) + (98 \times 1) + (127 \times -1) + (63 \times 1) + (142 \times -1) + (114 \times 1) = 17$
- Posisi 8 : $(63 \times 1) + (67 \times -1) + (65 \times 1) + (98 \times -1) + (127 \times 1) + (119 \times -1) + (142 \times 1) + (114 \times -1) + (90 \times 1) = 89$
- Posisi 9 : $(67 \times 1) + (65 \times -1) + (64 \times 1) + (127 \times -1) + (119 \times 1) + (153 \times -1) + (114 \times 1) + (90 \times -1) + (112 \times 1) = 41$

Hasil dari *output* konvolusi atau *feature map channel Red (R)* adalah sebagai berikut.

66	61	52	56	61
57	63	55	59	51
64	63	67	65	64
57	98	127	119	153
63	142	114	90	112

5x5

×

1	-1	1
-1	1	-1
1	-1	1

3x3

=

76	59	76
67	96	142
17	89	41

3x3

Gambar 3. 22 Feature Map Channel Red

Proses operasi konvolusi ini juga diterapkan pada *channel* warna lain, yaitu *channel Green* (G) dan *channel Blue* (B). Ukuran *output* konvolusi atau *feature map* dapat dihitung menggunakan persamaan (2.2).

$$Output_{size} = \frac{5 - 3 + 2(0)}{1} + 1 = 3$$

Berikut *output* hasil konvolusi atau *feature map* ketiga *channel* warna pada citra data input.

76	59	76
67	96	142
17	89	41

Feature Map Red

76	59	76
67	96	142
16	89	40

Feature Map Green

77	61	76
69	97	143
20	91	44

Feature Map Blue

Gambar 3. 23 Feature Map Citra Input

Setelah konvolusi selesai, fungsi ReLU (*Rectified Linear Unit*) akan diaktifkan. Fungsi aktivasi ReLU meningkatkan kualitas dengan mengurangi

kesalahan dengan mengubah nilai piksel setiap gambar yang bernilai kurang dari nol menjadi nol. Operasi ReLU diaktifkan dengan menggunakan persamaan (2.8).

Langkah 2 : *Primary Capsule*

Setelah mendapat *feature map* untuk masing-masing channel warna, Langkah selanjutnya adalah *reshape feature map* menjadi *primary capsule*. Caranya adalah dengan mengambil nilai dari ketiga channel warna pada setiap posisi grid 3x3, sehingga akan menghasilkan 9 vektor awal yang berdimensi 3. Misalnya pada posisi (1,1) vektor awal yang terbentuk adalah [76, 76, 77]. Proses ini dilakukan ulang pada setiap posisi lainnya yang akan menghasilkan 9 vektor yang merepresentasikan fitur lokal pada grid 3x3. Sehingga hasil akhirnya adalah sebagai berikut:

- Posisi (1,1): [76, 76, 77]
- Posisi (1,2): [59, 59, 61]
- Posisi (1,3): [76, 76, 76]
- Posisi (2,1): [67, 67, 69]
- Posisi (2,2): [96, 96, 97]
- Posisi (2,3): [142, 142, 143]
- Posisi (3,1): [17, 16, 20]
- Posisi (3,2): [89, 89, 91]
- Posisi (3,3): [41, 40, 44]

Selanjutnya, setiap vektor 3D ini akan ditransformasikan menjadi vektor 2D menggunakan matriks bobot yang diinisialisasikan secara acak. Operasi transformasi ini menggunakan persamaan (2.3). Pemilihan vektor 2D agar perhitungan lebih sederhana dan mempermudah menggambarkan cara kerja metode CapsNet.

Berikut matriks bobot yang digunakan:

$$W = \begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 0.3 \\ 0.4 & 0.2 \end{bmatrix}$$

Matriks ini mengubah vektor 3D menjadi 2D dengan operasi perkalian matriks. Sebagai contoh, untuk vektor posisi (1,1), yaitu [76, 76, 77], yang

menghasilkan *primary capsule* pertama (u_1) = [84.0, 45.8]. Perhitungannya adalah sebagai berikut sebagai berikut:

- Elemen pertama: $0.5 \times 76 + 0.2 \times 76 + 0.4 \times 77 = 38 + 15.2 + 30.8 = 84.0$
- Elemen kedua: $0.1 \times 76 + 0.3 \times 76 + 0.2 \times 77 = 7.6 + 22.8 + 15.4 = 45.8$

Proses ini diulangi untuk setiap posisi *grid* lainnya, sehingga diperoleh 9 *primary capsules* yang masing-masing berupa vektor 2D. Hasil akhirnya dapat dilihat berikut:

$$\begin{array}{ll} u_1 = [84.0, 45.8] & u_6 = [156.6, 85.4] \\ u_2 = [65.7, 35.8] & u_7 = [19.7, 10.5] \\ u_3 = [83.6, 45.6] & u_8 = [98.7, 53.8] \\ u_4 = [74.5, 40.6] & u_9 = [46.1, 24.9] \\ u_5 = [106.0, 57.8] & \end{array}$$

Dalam penelitian ini, vektor-vektor tersebut dapat dianggap mencerminkan properti seperti intensitas warna atau tekstur pada bagian tertentu dari citra, yang nantinya akan digunakan untuk membedakan cumi-cumi segar dan tidak segar.

Langkah 3 : *Dynamic Routing*

Setelah 9 *primary capsules* terbentuk, tahap berikutnya adalah *dynamic routing*, di mana proses ini bertujuan untuk menghubungkan *primary capsules* (u_i) ke 2 digit *capsules* yang mewakili kelas dalam penelitian ini, yaitu “Segar” (DigitCaps1) dan “Tidak Segar” (DigitCaps2), melalui mekanisme *routing-by-agreement*. Mekanisme ini memungkinkan setiap *primary capsules* memilih digit *capsules* yang paling sesuai berdasarkan keselarasan (*agreement*) antara prediksi yang dihasilkan dengan *output* digit *capsules*. Proses ini melibatkan beberapa langkah sebagai berikut.

A. Inisialisasi Logit ($b_{ij} = 0$)

Langkah pertama dalam mekanisme *dynamic routing* adalah menginisialisasi logit (b_{ij}), yaitu log prior *probabilities* yang menentukan seberapa kuat setiap *primary capsule* terhubung ke digit *capsule*. Untuk nilai logit awal diatur menjadi 0.

B. Hitung Vektor Prediksi

Kemudian, langkah selanjutnya adalah mentransformasikan setiap *primary capsule* menjadi vektor prediksi ($\hat{u}_{j|i}$) untuk masing-masing digit *capsule* menggunakan matriks bobot yang berbeda. Berikut matriks bobot yang diinisialisasi secara acak untuk DigitCaps1 ($W_{i|1}$) dan DigitCaps2 ($W_{i|2}$).

$$W_{i|1} = [[0.3, 0.1], [0.2, 0.4]]$$

$$W_{i|2} = [[0.1, 0.5], [0.3, 0.2]]$$

Untuk menghitung vektor prediksi ($\hat{u}_{j|i}$) dapat menggunakan persamaan (2.4). Sebagai contoh, untuk *primary capsule* 1 (u_1), perhitungannya sebagai berikut:

$$\hat{u}_{1|1} = [0.3 \times 84.0 + 0.2 \times 45.8, 0.1 \times 84.0 + 0.4 \times 45.8] = [34.36, 26.72]$$

$$\hat{u}_{2|1} = [0.1 \times 84.0 + 0.3 \times 45.8, 0.5 \times 84.0 + 0.2 \times 45.8] = [22.14, 51.16]$$

Proses ini diulang untuk setiap *primary capsule* ($u_2 - u_9$), yang akan menghasilkan vektor prediksi berjumlah 18, dengan 9 vektor prediksi untuk masing-masing DigitCaps. Hasil akhir vektor prediksi dapat dilihat sebagai berikut:

$$\hat{u}_{1|1} = [34.36, 26.72], \hat{u}_{1|2} = [26.82, 20.86], \dots, \hat{u}_{1|9} = [18.81, 14.58]$$

$$\hat{u}_{2|1} = [22.14, 51.16], \hat{u}_{2|2} = [17.32, 39.94], \dots, \hat{u}_{2|9} = [11.97, 27.74]$$

C. Hitung Koefisien *Coupling*

Lalu dilakukan perhitungan koefisien *coupling* (c_{ij}) yang dihitung menggunakan fungsi aktivasi *softmax* pada persamaan (2.5). Sehingga didapat nilai c_{i1} dan c_{i2} untuk setiap *primary capsule* (u_i) adalah 0.5, dengan perhitungan sebagai berikut:

$$c_{ij} = \frac{\exp(0)}{\sum_k \exp(0)}$$

$$c_{ij} = \frac{1}{1 + 1} = 0.5$$

D. Hitung Total *Input* (s_j)

Langkah selanjutnya adalah menghitung total *input* untuk setiap digit *capsule* (s_j) dengan menjumlahkan vektor prediksi ($\hat{u}_{j|i}$) yang telah didapatkan sebelumnya dengan koefisien *coupling* (c_{ij}). Perhitungannya menggunakan persamaan (2.6). Berikut perhitungan total input untuk DigitCaps1 (s_1) dan DigitCaps2 (s_2).

$$s_1 = 0.5 \times ([34.36, 26.72] + [26.82, 20.86] + \dots + [18.81, 14.58]) = [150.195, 116.74]$$

$$s_2 = 0.5 \times ([22.14, 51.16] + [17.32, 39.94] + \dots + [193.67, 447.54]) = [96.835, 223.77]$$

E. Aktivasi *Squash Function*

Setelah nilai *output* DigitCaps (s_j) diperoleh, vektor *output* DigitCaps (v_j) dihitung menggunakan fungsi aktivasi *squash*, yang merupakan salah satu fungsi aktivasi penting dalam CapsNet yang berfungsi untuk menormalkan panjang vektor agar berada dalam rentang $[0, 1]$, yang merepresentasikan probabilitas kelas. Rumus *squash function* dapat dilihat pada persamaan (2.9). Operasi pada tahap ini melibatkan perhitungan norma atau panjang, perhitungan skalar, dan normalisasi arah vektor. Perhitungan untuk *output* DigitCaps 1 (s_1) sebagai berikut:

$$\|s_1\| = \sqrt{150.195^2 + 116.74^2} \approx 190.23$$

$$\text{Skalar} = \frac{\|s_1\|^2}{1 + \|s_1\|^2} = \frac{190.23^2}{1 + 190.23^2} \approx 0.99997$$

$$v_1 = 0.99997 \cdot \frac{s_1}{190.23}$$

$$v_1 = 0.99997 \cdot \left[\frac{150.195}{190.23}, \frac{116.74}{190.23} \right]$$

$$v_1 = 0.99997 \times [0.7896, 0.6138] \approx [0.7895, 0.6137]$$

Selanjutnya lakukan operasi perhitungan serupa untuk DigitCaps 2 (s_2), hasilnya sebagai berikut:

$$||s_2|| \approx 243.82$$

$$\text{Skalar} = \frac{||s_2||^2}{1 + ||s_2||^2} = \frac{243.82^2}{1 + 243.82^2} \approx 0.99998$$

$$v_2 = 0.99998 \times [0.3972, 0.9177] \approx [0.3972, 0.9176]$$

F. Update Logit (b_{ij})

Iterasi pertama diakhiri dengan pembaruan logit (b_{ij}) berdasarkan keselarasan (*agreement*) antara vektor prediksi dan *output* digit *capsule*. Perhitungan untuk menghitung *agreement* dilakukan dengan menggunakan persamaan (2.7) seperti berikut:

$$b_{ij} = b_{ij} + \hat{u}_{ji} \cdot v_j$$

$$b_{1|1} = 0 + [34.36, 26.72] \cdot [0.7895, 0.6137] = 43.53$$

$$b_{1|2} = 0 + [22.14, 51.16] \cdot [0.3972, 0.9176] = 55.71$$

Proses ini dilakukan untuk semua *primary capsule* yang akan menghasilkan logit baru untuk digunakan pada iterasi berikutnya. Hasil akhir *agreement* untuk setiap *primary capsule* adalah sebagai berikut:

$$b_{1|1} = 43.53, b_{1|2} = 55.71 \quad b_{6|1} = 81.06, b_{6|2} = 103.86$$

$$b_{2|1} = 33.98, b_{2|2} = 45.43 \quad b_{7|1} = 10.11, b_{7|2} = 12.95$$

$$b_{3|1} = 43.33, b_{3|2} = 55.42 \quad b_{8|1} = 51.06, b_{8|2} = 65.55$$

$$b_{4|1} = 38.58, b_{4|2} = 49.52 \quad b_{9|1} = 23.80, b_{9|2} = 30.22$$

$$b_{5|1} = 54.95, b_{5|2} = 70.76$$

Selanjutnya, dilakukan *update* koefisien *coupling* (c_{ij}) berdasarkan nilai logit baru yang didapatkan sebelumnya. Sebagai contoh perhitungan pada *primary capsule* 1:

$$c_{11} = \frac{\exp(43.53)}{\exp(43.53) + \exp(55.71)} \approx 0.00000028$$

$$c_{12} = 1 - 0.00000028 \approx 0.99999972$$

Pada iterasi kedua, koefisien *coupling* (c_{ij}) diperbarui dengan fungsi aktivasi *softmax* dari logit baru. Dikarenakan nilai logit baru pada setiap DigitCaps2 (Tidak Segar) lebih besar, sehingga c_{i2} mendekati 1 dan c_{i1} mendekati 0 untuk semua i . Hal ini mengakibatkan total *input* DigitCaps1 (s_1) $\approx [0, 0]$ dan DigitCaps2 (s_2) $\approx [193.67, 447.54]$, dengan v_2 tetap stabil di $[0.3972, 0.9176]$ dan $v_1 \approx [0, 0]$. Selanjutnya pada iterasi ketiga menunjukkan hasil serupa, yang menandakan konvergensi *routing*.

Langkah 4 : Digit Capsule

Setelah mekanisme *Dynamic Routing* menyelesaikan iterasinya, hasil akhir yang diperoleh adalah vektor output dari setiap Digit Capsule. Vektor-vektor ini merepresentasikan probabilitas keberadaan masing-masing kelas yang diprediksi oleh model. Panjang atau norma L2 dari vektor Digit Capsule adalah indikator probabilitas tersebut, dengan nilai yang mendekati 1 menunjukkan probabilitas tinggi dan nilai mendekati 0 menunjukkan probabilitas rendah. Sehingga, berdasarkan hasil iterasi pada *Dynamic Routing* sebelumnya, *output* dari Digit Capsule adalah sebagai berikut:

$v_1 = [0, 0]$, dengan panjang vektor 0 (probabilitas sangat rendah)

$v_2 = [0.3972, 0.9176]$, dengan panjang vektor $\sqrt{0.3972^2 + 0.9176^2} \approx 0.999$ (probabilitas sangat tinggi)

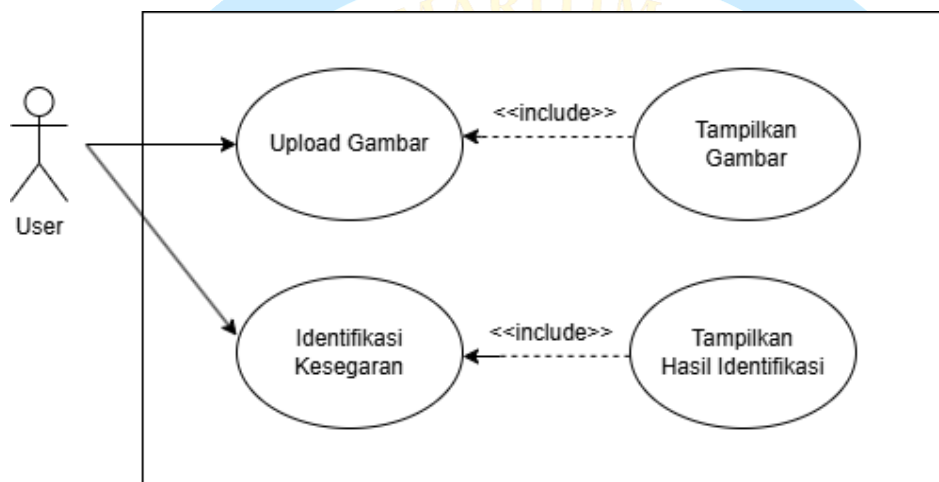
Dengan demikian, hasil akhir dari proses ini menunjukkan bahwa citra *input* lebih mungkin diklasifikasikan kedalam kelas “Tidak Segar”.

3.9 Perancangan Sistem

Proses perancangan sistem dilakukan untuk memahami kebutuhan sistem dari sudut pandang pengguna maupun sistem itu sendiri dan difokuskan bagaimana sistem dibangun dan diimplementasikan.

3.9.1 Use Case Diagram

Use Case Diagram menyajikan deskripsi visual mengenai interaksi yang terjadi antara aktor (pengguna) dan sistem yang dibangun. *Use Case* sistem identifikasi kesegaran cumi-cumi dapat dilihat pada Gambar 3.24 berikut.



Gambar 3. 24 *Use Case Sistem*

Berdasarkan gambar di atas, sistem identifikasi kesegaran cumi-cumi berbasis web ini, terdapat satu aktor yaitu *User* atau pengguna, di mana aktor ini berinteraksi dengan sistem melalui 2 *case* utama yaitu *Upload Gambar*, di mana pada *use case* ini, pengguna dapat mengunggah gambar cumi-cumi ke dalam sistem. Setelah gambar berhasil diunggah, maka sistem secara otomatis akan menampilkan gambar tersebut. Selanjutnya pada *use case* Identifikasi Kesegaran, pengguna dapat menekan tombol identifikasi untuk memulai proses identifikasi kesegaran. Setelah proses selesai, hasil identifikasi akan ditampilkan kepada pengguna.

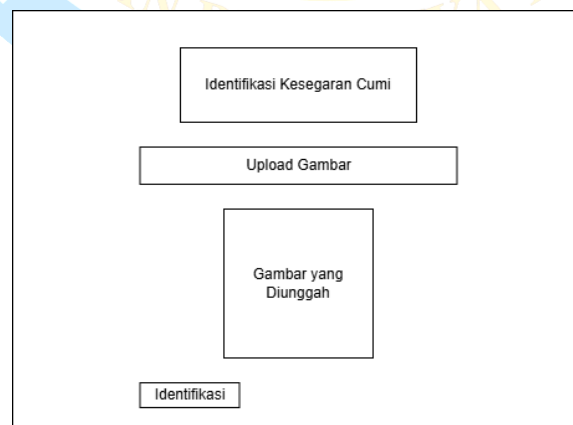
3.9.2 Perancangan *User Interface*

Desain antarmuka pengguna (*user interface*) dibuat untuk memberi pengguna visualisasi awal mengenai bagaimana sistem akan terlihat saat digunakan. Rancangan ini diupayakan sederhana dan intuitif agar pengguna tidak kesulitan dalam mengoperasikan sistem.

Ketika aplikasi dijalankan, akan tampil halaman utama yang menampilkan judul aplikasi dan tempat untuk mengunggah gambar, seperti yang dapat dilihat pada Gambar 3.25. Selanjutnya setelah mengunggah gambar, tampilan akan seperti pada gambar 3.26 di mana menampilkan hasil gambar yang diunggah dan terdapat tombol identifikasi yang jika ditekan akan memulai proses identifikasi. Saat proses identifikasi selesai, maka hasil identifikasi kesegaran akan muncul disertai nilai *confidence* atau keyakinan model dengan hasil identifikasinya.



Gambar 3. 25 Tampilan Halaman Utama



Gambar 3. 26 Tampilan Setelah Mengunggah Gambar

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Training Model

Subbab ini akan menampilkan hasil pelatihan model *Capsule Network* (CapsNet) yang akan digunakan dalam identifikasi kesegaran cumi-cumi berdasarkan citra RGB berukuran 128x128 piksel. Proses pelatihan dilakukan secara bertahap dengan melakukan beberapa penyesuaian pada parameter yang berpengaruh terhadap performa model. Evaluasi performa model dilakukan dengan memperhatikan metrik akurasi dan *loss* yang ada pada data latih dan data validasi.

4.1.1 Analisis Pengaruh *Learning Rate* dan Jumlah *Epoch*

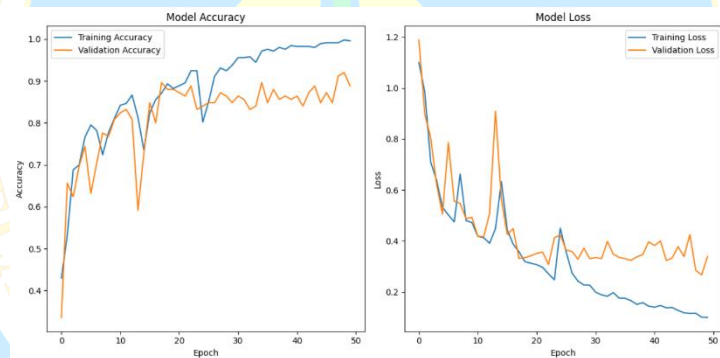
Tahap ini, dilakukan pengujian dengan menggunakan beberapa nilai *learning rate* dan jumlah *epoch* untuk melihat hasilnya terhadap performa model. *Learning rate* merupakan salah satu *hyperparameter* yang cukup penting dalam proses *training* atau pelatihan suatu model karena menentukan seberapa besar langkah pembaruan bobot dilakukan dalam setiap iterasi. Pemilihan nilai *learning rate* yang tepat dapat mempercepat konvergensi model, sementara jumlah *epoch* menentukan durasi pelatihan yang dilakukan dan dapat mempengaruhi sejauh mana model mampu mempelajari pola dari data.

Tabel 4. 1 Percobaan *Learning Rate* dan *Epoch*

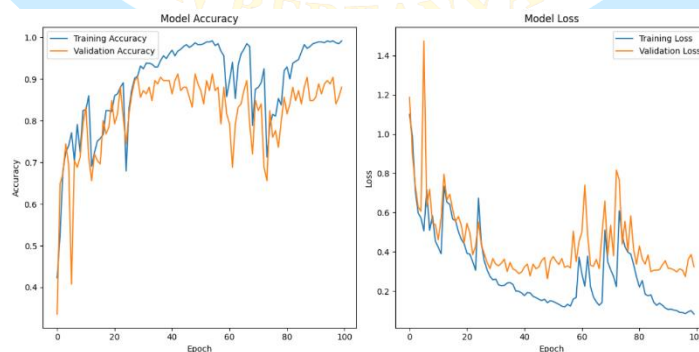
Percobaan Ke-	<i>Learning Rate</i>	<i>Epoch</i>
1	1e-04	50
2	1e-04	100
3	1e-05	50
4	1e-05	100

Percobaan pertama yang dilakukan adalah menggunakan *learning rate* 1e-04 atau 0.0001 dengan 50 *epoch*. Model menunjukkan kemampuan belajar yang baik pada tahap awal, seperti yang dapat dilihat pada Gambar 4.1 di mana kurva akurasi

pelatihan dan akurasi validasi pada awalnya menunjukkan kenaikan yang konsisten dan saling beriringan hingga sekitar *epoch* ke-20. Namun setelah melewati titik tersebut, mulai terlihat adanya *gap* yang melebar antara akurasi pelatihan yang terus meningkat mendekati 100% dan akurasi validasi yang cenderung stagnan serta berfluktuasi di rentang 85-90%. Indikasi *overfitting* ini juga diperkuat oleh grafik *loss*, di mana nilai *loss* pada pelatihan terus menurun sementara *loss* validasi berhenti membaik dan bergerak horizontal. Ketika jumlah *epoch* ditingkatkan menjadi 100 *epoch*, fenomena *overfitting* yang teramati sebelumnya menjadi semakin jelas, seperti yang ditampilkan pada Gambar 4.2, di mana kurva akurasi pelatihan berhasil mencapai puncaknya, namun kurva akurasi validasi menjadi tidak stabil dan fluktuatif. Grafik *loss* juga menunjukkan hal serupa, di mana nilai *loss* validasi sangat berfluktuatif sementara *loss* pada pelatihan terus menurun hingga mencapai nilai yang sangat rendah.

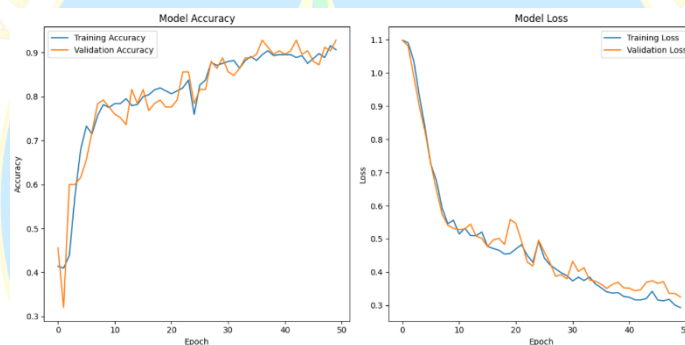


Gambar 4. 1 Learning Rate $1e-04$ 50 Epoch

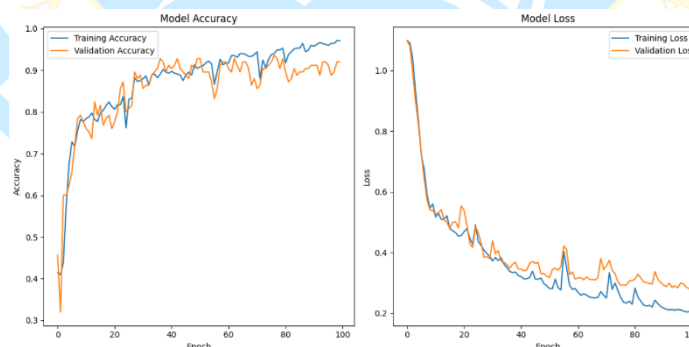


Gambar 4. 2 Learning Rate $1e-04$ 100 Epoch

Selanjutnya dilakukan pengujian dengan mengurangi *learning rate* menjadi $1e-05$ atau 0.00001 untuk melihat pengaruhnya terhadap stabilitas dan konvergensi model. Pada percobaan ke-3 dengan 50 *epoch*, grafik pada Gambar 4.3 menunjukkan kurva akurasi dan *loss* pelatihan dan validasi memiliki tren yang konsisten, dengan tidak ada *gap* yang signifikan di antara keduanya, yang mengindikasikan tidak adanya tanda-tanda *overfitting*. Pada percobaan ke-4, Ketika pelatihan dilanjutkan hingga 100 *epoch*, perilaku pelatihan cenderung sama. Namun, seperti yang ditampilkan pada Gambar 4.4, setelah melewati sekitar *epoch* ke-60, mulai muncul *overfitting* ringan. Hal ini ditandai dengan mulai terbentuknya celah kecil antara kurva pelatihan dan validasi. Grafik *loss* juga menunjukkan hal yang sama, di mana *loss* validasi berhenti menurun secara signifikan sementara *loss* pelatihan berlanjut turun.



Gambar 4. 3 *Learning Rate 1e-05 50 Epoch*



Gambar 4. 4 *Learning Rate 1e-05 100 Epoch*

Hasil pelatihan model dengan percobaan variasi penggunaan *learning rate* dan jumlah *epoch* dapat dilihat pada tabel berikut.

Tabel 4. 2 Hasil Pelatihan Model *Epoch* 50

<i>Epoch</i> 50	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi Pelatihan	99%	89%
Akurasi Validasi	88%	93%

Tabel 4. 3 Hasil Pelatihan Model *Epoch* 100

<i>Epoch</i> 100	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi Pelatihan	99%	98%
Akurasi Validasi	88%	92%

4.1.2 Pengaruh *Batch Size*

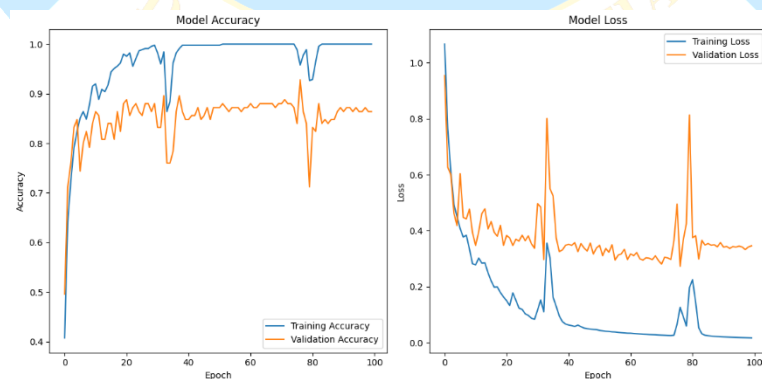
Tahap ini melibatkan pengujian lanjutan dengan menggunakan berbagai nilai *batch size* untuk melihat sejauh mana parameter ini dapat mempengaruhi performa dan hasil pelatihan model. *Batch size* merupakan parameter yang menentukan jumlah data yang diproses sebelum melakukan pembaruan bobot. Pemilihan *batch size* yang tepat dapat berdampak pada stabilitas konvergensi, kecepatan pelatihan, dan kemampuan generalisasi model.

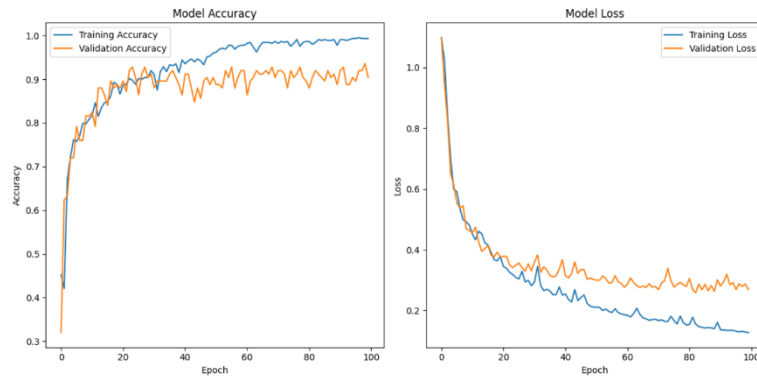
Berdasarkan hasil pengujian sebelumnya pada *learning rate* 1e-04 dan 1e-05 dengan variasi *epoch* 50, dan 100, di mana semua percobaan tersebut menggunakan *batch size* 32. Dapat disimpulkan bahwa model dengan jumlah *epoch* yang lebih besar memiliki performa yang lebih optimal dan konvergensi yang lebih stabil. Oleh karena itu, pada pengujian *batch size* ini hanya akan menggunakan 100 *epoch* untuk masing-masing *learning rate* agar lebih efisien dan fokus pada kondisi pelatihan yang paling menjanjikan. Percobaan yang dilakukan pada tahap ini dapat dilihat pada Tabel 4.5 berikut.

Tabel 4. 4 Percobaan Batch Size

Percobaan Ke-	<i>Learning Rate</i>	<i>Batch Size</i>
1	1e-04	16
2	1e-05	16
3	1e-04	64
4	1e-05	64

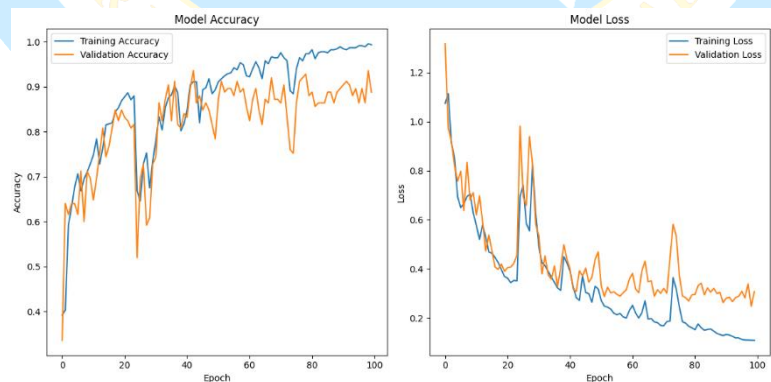
Percobaan pertama yang dilakukan adalah dengan menggunakan *batch size* 16 dan *learning rate* 1e-04, terlihat pada Gambar 4.5 model menunjukkan performa yang tidak stabil selama proses pelatihan, ditandai dengan kurva akurasi pelatihan yang mampu mencapai nilai yang sangat tinggi, sedangkan kurva akurasi validasi yang terlihat cenderung stagnan dan fluktuatif di beberapa *epoch*. Adanya *gap* yang besar dan tidak konsisten antara kedua kurva yang juga terlihat pada grafik *loss* yang mengindikasikan terjadinya *overfitting* yang parah dan proses optimisasi yang tidak stabil. Sebaliknya, pada percobaan ke-2 di mana menggunakan *learning rate* yang lebih kecil yaitu 1e-05, model menunjukkan hasil yang jauh lebih baik dan stabil. Pada Gambar 4.6, kurva pelatihan dan validasi untuk akurasi maupun *loss* terlihat bergerak beriringan dan konsisten sepanjang pelatihan. Celah di antara keduanya sangat kecil, yang menandakan kemampuan generalisasi model yang sangat baik dan *overfitting* yang minimal.

**Gambar 4. 5** *Learning Rate* 1e-04 *Batch Size* 16

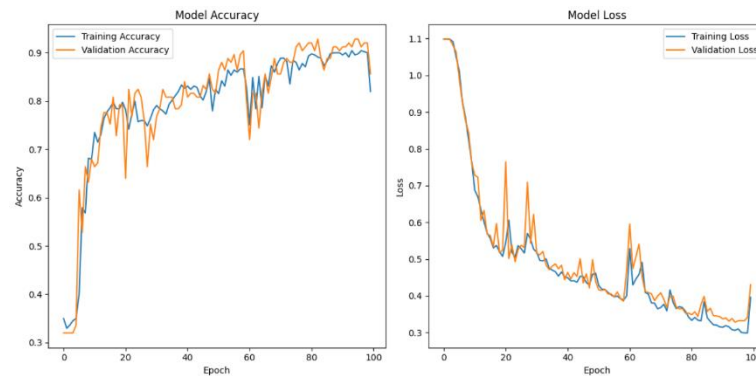


Gambar 4. 6 *Learning Rate 1e-05 Batch Size 16*

Percobaan selanjutnya yaitu percobaan ke-3 dengan menggunakan *batch size* yang lebih besar dari sebelumnya, yakni 64. Dengan *learning rate* 1e-04, seperti terlihat pada Gambar 4.7, model menunjukkan tanda-tanda *overfitting* yang jelas disertai dengan ketidakstabilan pada performa validasi. Setelah sekitar *epoch* ke-25, mulai terlihat celah yang signifikan antara kurva akurasi pelatihan yang terus meningkat dan kurva akurasi validasi yang sangat fluktuatif dan tidak konsisten. Hal ini juga terlihat pada grafik *loss*, di mana nilai *loss* validasi tidak menunjukkan tren penurunan yang konsisten, berbeda dengan nilai *loss* pelatihan yang terus menurun. Hasil yang jauh lebih baik ditunjukkan pada percobaan ke-4 ketika *learning rate* diturunkan menjadi 1e-05, seperti yang ditampilkan pada Gambar 4.8, model menunjukkan proses pelatihan yang sangat stabil dengan kurva akurasi dan *loss* untuk pelatihan dan validasi yang bergerak beriringan dengan *gap* yang minim.



Gambar 4. 7 *Learning Rate 1e-04 Batch Size 64*



Gambar 4. 8 *Learning Rate 1e-05 Batch Size 64*

Hasil pelatihan model dengan percobaan variasi *batch size* ditampilkan pada tabel dibawah ini.

Tabel 4. 5 Hasil Pelatihan Model *Batch Size 16*

<i>Batch Size 16</i>	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi Pelatihan	100%	100%
Akurasi Validasi	86%	90%

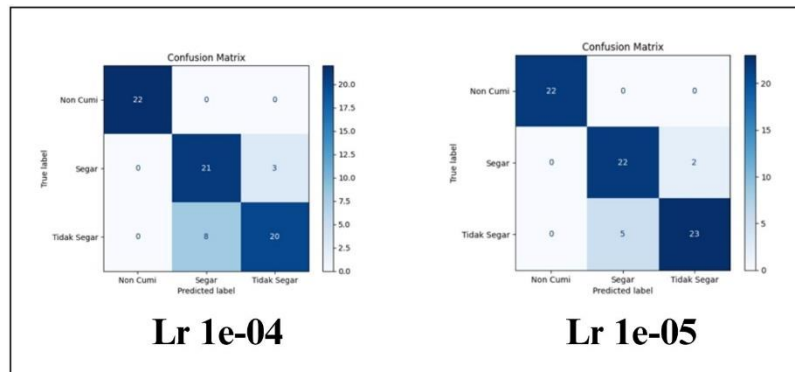
Tabel 4. 6 Hasil Pelatihan Model *Batch Size 64*

<i>Batch Size 64</i>	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi Pelatihan	99%	82%
Akurasi Validasi	89%	86%

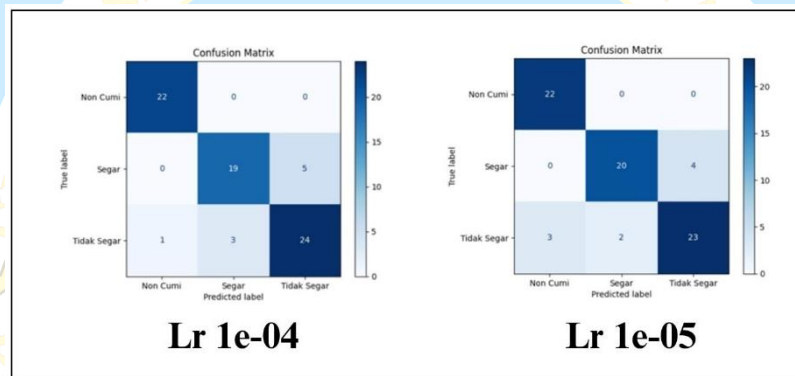
4.2 Pengujian Model Menggunakan *Confusion Matrix*

Tahap pengujian model ini dilakukan untuk menguji model yang telah dilatih sebelumnya, dengan menggunakan data uji sebanyak 74 gambar, sesuai dengan jumlah dataset yang telah dibagi yang ditunjukkan pada Tabel 3.1. Pengujian

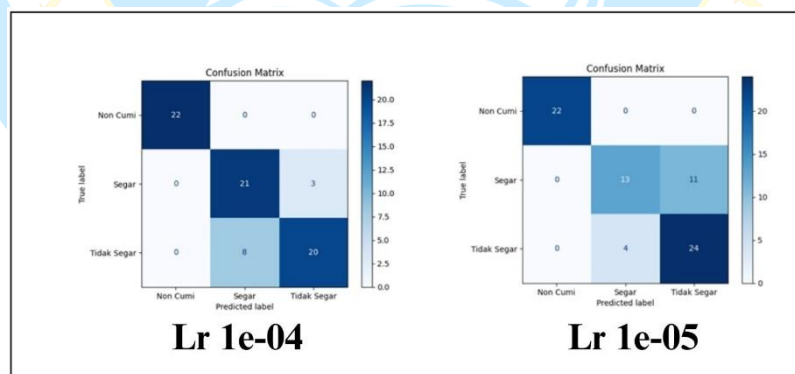
dilakukan dengan tujuan untuk mengevaluasi performa model yang telah dilatih berdasarkan metrik akurasi, presisi, *recall*, dan skor f1.



Gambar 4. 9 *Confusion Matrix Batch Size 16*



Gambar 4. 10 *Confusion Matrix Batch Size 32*



Gambar 4. 11 *Confusion Matrix Batch Size 64*

Akurasi yang dicapai model berdasarkan *confusion matrix* pada gambar di atas ditampilkan pada tabel berikut.

Tabel 4. 7 Akurasi Model *Batch Size* 16

<i>Batch Size</i> 16	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi	85%	91%

Tabel 4. 8 Akurasi Model *Batch Size* 32

<i>Batch Size</i> 32	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi	88%	88%

Tabel 4. 9 Akurasi Model *Batch Size* 64

<i>Batch Size</i> 64	<i>Learning Rate</i>	
	1e-04	1e-05
Akurasi	85%	80%

Sehingga berdasarkan tabel akurasi tersebut, dapat disimpulkan bahwa model dengan *batch size* 16 dan *learning rate* 1e-05 menghasilkan performa terbaik dengan akurasi tertinggi sebesar 90%, dengan rincian hasilnya ditampilkan pada Tabel 4.10 berikut.

Tabel 4. 10 Hasil *Confusion Matrix*

Kategori	Prediksi Segar	Prediksi Tidak Segar	Prediksi Non Cumi	Total
Segar	22	2	0	24
Tidak Segar	5	23	0	28
Non Cumi	0	0	22	22
Total	27	25	22	74

Berdasarkan hasil dari *confusion matrix* pada Tabel 4.10, dilakukan perhitungan untuk mendapatkan nilai metrik evaluasi seperti akurasi, presisi, *recall*, dan *f1-score*. Hasil perhitungan metrik evaluasi dapat dilihat pada Tabel 4.11 berikut.

Tabel 4. 11 Hasil Metrik Evaluasi

Metrik	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
	91%	91%	91%	91%

4.3 Pengujian Model Menggunakan Data Baru

Pengujian dengan data baru dilakukan dengan tujuan untuk melihat dan menilai sejauh mana model mampu beradaptasi dengan data yang benar-benar diluar dataset yang digunakan dan belum pernah dijumpai sebelumnya. Data baru yang digunakan dalam pengujian berjumlah 20 data, yang terdiri masing-masing 10 data untuk kelas “Segar” dan “Tidak Segar”. Hasil dari pengujian ini akan memberikan gambaran tentang seberapa baik kemampuan generalisasi dan efektivitas model dalam implementasi sebenarnya.

Tabel 4. 12 Pengujian Kelas Segar

Pengujian Ke-	Hasil Prediksi
1	Benar
2	Benar
3	Benar
4	Benar
5	Salah
6	Salah
7	Salah
8	Salah
9	Benar
10	Benar

Tabel 4. 13 Pengujian Kelas Tidak Segar

Pengujian Ke-	Hasil Prediksi
1	Benar
2	Benar
3	Benar
4	Benar

5	Benar
6	Benar
7	Benar
8	Benar
9	Benar
10	Benar

Berdasarkan tabel pengujian masing-masing kelas diatas, maka hasilnya ditampilkan pada tabel berikut.

Tabel 4. 14 Hasil Pengujian Data Baru

Kategori	Prediksi Segar	Prediksi Tidak Segar	Total
Segar	6	4	10
Tidak Segar	0	10	10
Total	6	14	20

Sehingga berdasarkan Tabel 4.14 di atas, dapat diperoleh akurasi yang dicapai model dengan pengujian menggunakan data baru sebesar 80%.

4.4 Pengujian Kelas Non Cumi

Pengujian ini bertujuan untuk memastikan model mampu mendeteksi citra atau objek selain cumi-cumi dan mengidentifikasikannya sebagai non cumi. Data yang digunakan pada pengujian ini berjumlah 10 data yang dikumpulkan dari sumber internet.



Gambar 4. 12 Hasil Pengujian Non Cumi

Berdasarkan Gambar 4.12 dapat dilihat bahwa model berhasil mengidentifikasi semua data sebagai objek non cumi, sehingga akurasi yang dicapai sebesar 100%. Dengan hasil ini menunjukkan bahwa model berhasil mengklasifikasikan citra selain objek cumi-cumi yang dipelajari selama pelatihan.

4.5 Pembahasan

Berdasarkan hasil pelatihan model dengan melakukan beberapa percobaan dan pengujian, dapat disimpulkan bahwa pemilihan parameter yang tepat seperti *learning rate*, jumlah *epoch*, dan *batch size* dapat memberikan pengaruh yang signifikan terhadap performa model. Di mana kombinasi penggunaan nilai *learning rate* yang lebih kecil dan jumlah *epoch* yang lebih banyak dapat menghasilkan pelatihan yang lebih stabil dengan *gap* minim antara akurasi pelatihan dan validasi, dengan penerapan *early stopping* yang terbukti efektif dalam mencegah terjadinya *overfitting* dan mendapatkan titik optimal performa model. Selain itu, pemilihan ukuran *batch* yang lebih kecil juga berdampak signifikan terhadap performa dan akurasi model.

Berdasarkan hasil pelatihan, model dengan *learning rate* $1e-05$ dan *batch size* 16 menunjukkan performa paling optimal, dengan akurasi pelatihan sebesar 100% dan akurasi validasi sebesar 90%, yang dicapai menggunakan 100 *epoch*. Dengan grafik kurva akurasi dan *loss* menunjukkan tren yang konsisten dan stabil.

Sedangkan pada tahap pengujian, model *Capsule Network* (CapsNet) yang telah dilatih sebelumnya diuji menggunakan data uji sebanyak 74 gambar. Hasil pengujian menunjukkan bahwa model mampu mengidentifikasi 67 data benar data uji, sehingga akurasi yang dicapai model adalah sebesar 91%.

Sementara, pada pengujian menggunakan data baru diluar dari dataset yang digunakan baik selama pelatihan maupun pengujian, model mampu mengidentifikasi 16 data benar dari total 20 data, dengan akurasi yang dicapai sebesar 80%. Hasil ini menunjukkan bahwa model *Capsule Network* (CapsNet) memiliki kemampuan generalisasi yang sudah cukup baik dan akurat dalam mengidentifikasi kesegaran citra cumi-cumi.

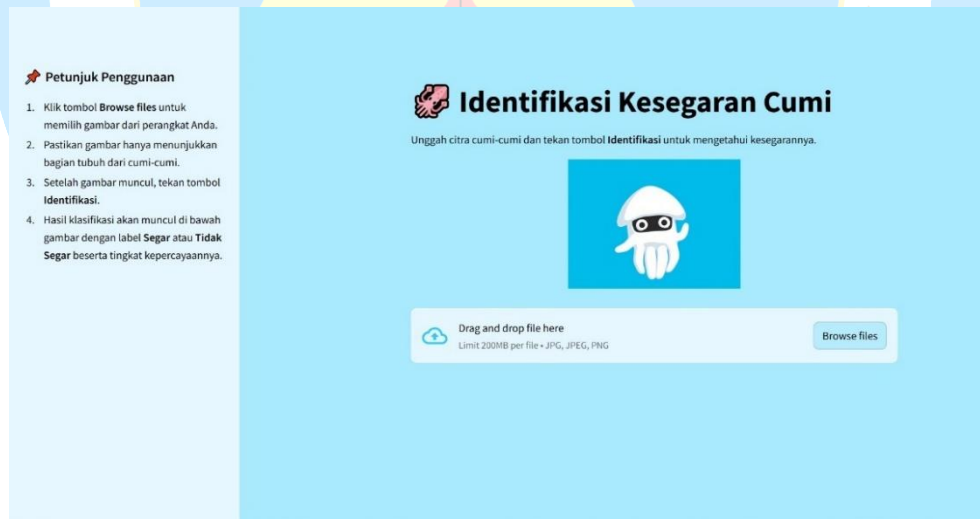
Pada tahap pengujian non cumi dengan menggunakan 20 data yang terdiri dari berbagai objek selain cumi-cumi dan berada di luar dari dataset, model berhasil

mencapai performa yang sangat signifikan dengan sempurna mengidentifikasi seluruh data dengan benar. Hasil ini membuktikan bahwa model memiliki kemampuan yang sangat baik dalam mengidentifikasi berbagai objek acak selain cumi-cumi yang dipelajari selama pelatihan.

Secara keseluruhan, rangkaian percobaan ini menunjukkan bahwa pemilihan *hyperparameter* yang tepat, seperti nilai *learning rate*, jumlah *epoch*, dan ukuran *batch*, sangat penting untuk memaksimalkan performa dan kemampuan model. Kemampuan model *Capsule Network* (CapsNet) dalam mempertahankan performa yang sangat baik pada data baru dan mencapai tingkat akurasi yang tinggi pada data uji menunjukkan bahwa model ini memiliki potensi besar dalam implementasi sebenarnya untuk menjaga kualitas produk perikanan di lapangan, khususnya cumi-cumi.

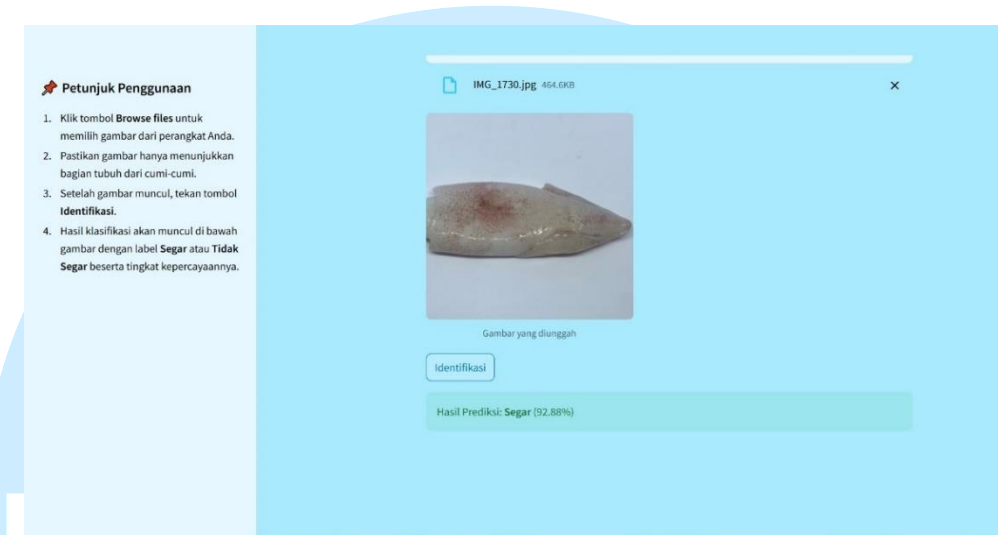
4.6 Hasil Perancangan Aplikasi Web

Aplikasi web yang diimplementasikan dengan model yang telah dilatih sebelumnya dirancang menggunakan *framework streamlit*, dengan tampilan sebagai berikut.



Gambar 4. 13 Halaman Utama Aplikasi

Pada gambar yang ditampilkan diatas, merupakan halaman utama aplikasi web yang didalamnya terdapat judul aplikasi, tombol untuk mengunggah gambar cumi-cumi yang akan diidentifikasi kesegarannya, serta terdapat panduan atau petunjuk penggunaan aplikasi pada bagian *sidebar*. Setelah gambar berhasil diunggah, maka gambar dan tombol identifikasi akan tampil di halaman utama. Lalu ketika *user* menekan tombol identifikasi, maka hasil identifikasi akan ditampilkan.



Gambar 4. 14 Tampilan Identifikasi Kesegaran

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis pelatihan dan pengujian yang telah dilakukan, metode *Capsule Network* (CapsNet) terbukti efektif untuk mengidentifikasi kesegaran cumi-cumi berdasarkan citra digital. Performa model terbaik diperoleh dengan kombinasi *learning rate* $1e-05$, *batch size* 16 serta 100 *epoch*, dengan tingkat akurasi 91%, *precision* 91%, *recall* 91%, dan *f1-score* 91% pada data uji. Dengan adanya *noise* pada penggunaan *batch size* yang lebih kecil sehingga menghasilkan performa yang lebih baik. Kemampuan generalisasi model juga terbukti sangat baik, di mana pada pengujian menggunakan 20 data baru yang berada di luar dataset, model mampu mencapai akurasi sebesar 80%.

Hasil ini menunjukkan bahwa model CapsNet tidak hanya mampu mempelajari pola dari data latih, tetapi juga mampu mengklasifikasikan data yang belum pernah dilihat sebelumnya dengan akurat. Dengan demikian, metode CapsNet memiliki potensi besar untuk diimplementasikan sebagai solusi teknologi yang efisien bagi industri perikanan untuk memastikan standar kualitas produk cumi-cumi secara lebih konsisten.

5.2 Saran

Untuk pengembangan lebih lanjut, disarankan untuk menambah jumlah dan kondisi dataset yang lebih beragam agar dapat meningkatkan performa, akurasi, serta kemampuan generalisasi model secara keseluruhan. Selain itu, penambahan fitur lain yang relevan dengan indikator kesegaran, seperti warna mata cumi-cumi agar memperkaya informasi yang dapat dipelajari model. Dapat pula dilakukan eksplorasi dan optimasi *hyperparameter* lebih lanjut dapat dilakukan untuk menemukan kombinasi parameter yang paling optimal sehingga dapat memaksimalkan potensi performa model. Terakhir, penggunaan metode atau algoritma klasifikasi lain juga perlu dieksplorasi untuk melihat potensi adanya pendekatan yang lebih efektif atau lebih sesuai dengan karakteristik data.

DAFTAR PUSTAKA

- Abdillah, T. R. (2023). Analisis Komparasi Cycles X Render Dan Cycles Render Menggunakan Google Colab. *Jurnal TIKA*, 8(1), 90–94. <https://doi.org/10.51179/tika.v8i1.1937>
- Abouelmagd, L. M., Shams, M. Y., Marie, H. S., & Hassanien, A. E. (2024). An optimized capsule neural networks for tomato leaf disease classification. *Eurasip Journal on Image and Video Processing*, 2024(1). <https://doi.org/10.1186/s13640-023-00618-9>
- Adu, K., Yu, Y., Cai, J., Twumasi, B., & Wang, X. (2021). DHS-CapsNet : Dual horizontal squash capsule networks for lung and colon cancer classification from whole slide histopathological images. *International Journal of Imaging Systems and Technology*, 31. <https://doi.org/10.1002/ima.22569>
- Ali, M., & Ali, R. (2021). Multi-input dual-stream capsule network for improved lung and colon cancer classification. *Diagnostics*, 11(8), 1–18. <https://doi.org/10.3390/diagnostics11081485>
- Altan, G. (2020). Performance Evaluation of Capsule Networks for Classification of Plant Leaf Diseases. *International Journal of Applied Mathematics Electronics and Computers*, 8(3), 57–63. <https://doi.org/10.18100/ijamec.797392>
- Amin, U., Shahzad, M. I., Shahzad, A., Shahzad, M., Khan, U., & Mahmood, Z. (2023). Automatic Fruits Freshness Classification Using CNN and Transfer Learning. *Applied Sciences (Switzerland)*, 13(14). <https://doi.org/10.3390/app13148087>
- Anggara, D., Suarna, N., & Arie Wijaya, Y. (2023). Performance Comparison Analysis Of Optimizer Adam, SGD, and RMSPROP on The H5 Model. *Jurnal Ilmiah NERO*, 8(1), 2023. <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset>
- Asandri, A. L., Liviawaty, E., Buwono, I. D., & Junianto, J. (2022). Analisis Pengaruh Yoghurt Terhadap Umur Simpan Cumi-Cumi (*Loligo sp.*) yang Disimpan Pada Suhu Chilling. *Akuatika Indonesia*, 7(1), 28. <https://doi.org/10.24198/jaki.v7i1.35826>
- Bodyanskiy, Y., & Kostiuk, S. (2022). Adaptive Hybrid Activation Function for Deep Neural Networks. *System Research and Information Technologies*, 2022(1), 87–96. <https://doi.org/10.20535/SRIT.2308-8893.2022.1.07>

- Bravenec, A. D., & Ward, K. D. (2023). Interactive Python Notebooks for Physical Chemistry. *Journal of Chemical Education*, 100(2), 933–940. <https://doi.org/10.1021/acs.jchemed.2c00665>
- Cao, F., Zheng, C., Huang, L., Wang, A., Zhang, J., Zhou, F., Ju, H., Guo, H., & Du, Y. (2022). Research of Self-Attention in Image Segmentation. *Journal of Information Technology Research*, 15(1), 1–12. <https://doi.org/10.4018/jitr.298619>
- Cao, T., Li, Q., Huang, Y., & Li, A. (2023). plotnineSeqSuite: a Python package for visualizing sequence data using ggplot2 style. *BMC Genomics*, 24(1), 1–8. <https://doi.org/10.1186/s12864-023-09677-8>
- Celebi, A. R. C., Bulut, E., & Sezer, A. (2023). Artificial intelligence based detection of age-related macular degeneration using optical coherence tomography with unique image preprocessing. *European Journal of Ophthalmology*, 33(1), 65–73. <https://doi.org/10.1177/11206721221096294>
- Chaudhari, J. P., Mewada, H., Patel, A. V., & Mahant, K. (2023). Automated bacteria genera classification using histogram-oriented optimized capsule network. *Engineering Science and Technology, an International Journal*, 46(May), 101500. <https://doi.org/10.1016/j.jestch.2023.101500>
- Dewi, C. (2024). Seminar and Workshop on Object Recognition using Deep Learning at Sam Ratulangi University Manado. *Jurnal Pengabdian Masyarakat*, 5(1), 151–156. <https://doi.org/10.32815/jpm.v5i1.1379>
- Fatoni, M., & Ernastuti, E. (2023). Ornamental Plants Classification Using Integration of Convolution With Capsule Network. *MIND (Multimedia Artificial Intelligent Networking Database)*, 8(2), 158–172. <https://ejurnal.itenas.ac.id/index.php/mindjournal/article/view/9157%0Ahttps://ejurnal.itenas.ac.id/index.php/mindjournal/article/download/9157/3486>
- Hasnain, M., Pasha, M. F., Ghani, I., Imran, M., Alzahrani, M. Y., & Budiarto, R. (2020). Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking. *IEEE Access*, 8, 90847–90861. <https://doi.org/10.1109/ACCESS.2020.2994222>
- Hollósi, J., Ballagi, Á., & Pozna, C. R. (2023). Simplified Routing Mechanism for Capsule Networks. *Algorithms*, 16(7). <https://doi.org/10.3390/a16070336>
- Hu, J., Zhou, C., Zhao, D., Zhang, L., Yang, G., & Chen, W. (2020). A rapid, low-cost deep learning system to classify squid species and evaluate freshness based on digital images. *Fisheries Research*, 221(June 2019), 105376. <https://doi.org/10.1016/j.fishres.2019.105376>

- Huang, W., & Zhou, F. (2020). DA-CapsNet: dual attention mechanism capsule network. *Scientific Reports*, 10(1), 1–13. <https://doi.org/10.1038/s41598-020-68453-w>
- Jiang, X., Zhang, Y., Liu, W., Gao, J., Liu, J., Zhang, Y., & Lin, J. (2020). Hyperspectral image classification with capsnet and markov random fields. *IEEE Access*, 8, 191956–191968. <https://doi.org/10.1109/ACCESS.2020.3029174>
- Joshua Anderson, Rebecca Casad, Charles Cathcart, Anh Nguyen, & Tauheed Khan Mohd. (2022). *Median American Sign Language Interpretation Software*. December. <https://doi.org/10.20944/preprints202212.0471.v1>
- Kalpana, P., & Anandan, R. (2023). A Capsule Attention Network for Plant Disease Classification. *Traitement Du Signal*, 40(5), 2051–2062. <https://doi.org/10.18280/ts.400523>
- Kasmiati, K., Syahrul, S., Fahrul, F., Amir, N., Karsidi, S., & Rahmi, M. H. (2022). Quality and safety of fresh squid (*Loligo forbesii*) sold in Daya Traditional Market, Makassar, Indonesia. *IOP Conference Series: Earth and Environmental Science*, 1119(1). <https://doi.org/10.1088/1755-1315/1119/1/012050>
- Khodadadzadeh, M., Ding, X., Chaurasia, P., & Coyle, D. (2021). A Hybrid Capsule Network for Hyperspectral Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 11824–11839. <https://doi.org/10.1109/JSTARS.2021.3126427>
- Kılıçarslan, S., Hız Çiçekliyurt, M. M., & Kılıçarslan, S. (2024). Fish Freshness Detection Through Artificial Intelligence Approaches: A Comprehensive Study. *Turkish Journal of Agriculture - Food Science and Technology*, 12(2), 290–295. <https://doi.org/10.24925/turjaf.v12i2.290-295.6670>
- Li, Y., He, Z., Wang, S., Wang, Z., & Huang, W. (2021). Multideep Feature Fusion Algorithm for Clothing Style Recognition. *Wireless Communications and Mobile Computing*, 2021. <https://doi.org/10.1155/2021/5577393>
- Liu, B., Kang, H., Li, H., Vasconcelos, N., & Hua, G. (2020). Few-shot open-set recognition using meta-learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8795–8804. <https://doi.org/10.1109/CVPR42600.2020.00882>
- Marchisio, A., Bussolino, B., Salvati, E., Martina, M., Masera, G., & Shafique, M. (2022). Enabling Capsule Networks at the Edge through Approximate Softmax and Squash Operations. *Proceedings of the International Symposium*

- Martínez-Fernandez, E., Rojas-Valenzuela, I., Valenzuela, O., & Rojas, I. (2023). Computer Aided Classifier of Colorectal Cancer on Histopatological Whole Slide Images Analyzing Deep Learning Architecture Parameters. *Applied Sciences (Switzerland)*, 13(7). <https://doi.org/10.3390/app13074594>
- Mostafa, S., & Wu, F. X. (2021). Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images. *Neural Engineering Techniques for Autism Spectrum Disorder: Volume 1: Imaging and Signal Analysis*, January, 23–38. <https://doi.org/10.1016/B978-0-12-822822-7.00003-X>
- Mudirta, M., Pamungkas, B. F., Sulistiawati, S., Rusdin, I., & Mismawati, A. (2024). Penerimaan Konsumen dan Sifat Fisikokimia Cookies dengan Penambahan Tinta Loligo sp. *Jurnal Pascapanen Dan Bioteknologi Kelautan Dan Perikanan*, 18(2), 107. <https://doi.org/10.15578/jpbkp.v18i2.953>
- Nagi, S. C., Miles, A., & Donnelly, M. J. (2023). *AgamPrimer: Primer Design in Anopheles gambiae informed by range-wide genomic variation*. <https://doi.org/https://doi.org/10.1101/2022.12.31.521737>
- Pernando, Y., Febrianti, E. L., Syafrinal, I., Roza, Y., & Afifah, U. F. (2022). Deep Learning for Faces on Orphanage Children Face Detection. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 9(1), 25–32. <https://doi.org/10.33330/jurteks.v9i1.1858>
- Pirmansa, J. S., Prasetyono, E., Sari, S. P., Febrianti, D., & Syarif, A. F. (2020). DAYA TETAS TELUR CUMI-CUMI (Uroteuthischinensis) PADA SALINITAS YANG BERBEDA. *Journal of Tropical Marine Science*, 3(1), 1–10. <https://doi.org/https://doi.org/10.33019/jour.trop.mar.sci.v3i1.1702>
- Riti, Y. F., Wahyuningsih, Y., Roosandriantini, J., & Siswanto, P. W. (2023). Perbandingan Algoritma Convolutional Neural Netwok dan Capsule Network Dalam Klasifikasi Jenis Rumah Adat. *Teknika*, 12(3), 243–251. <https://doi.org/10.34148/teknika.v12i3.702>
- Rodlotu Yula, H., Bahri, S., Ambarwati, Y., Idris, I., & Andhulang, G. (2021). ANALISIS KANDUNGAN LOGAM BERAT MERKURI (Hg) PADA CUMICUMI (Loligo sp.) DENGAN MENGGUNAKAN METODE ATOMIC ABSORPTION SPECTROSCOPY (AAS). *Analit:Analytical and Environmental Chemistry*, 6(01), 83–91. <https://doi.org/10.23960/aec.v6.i1.2021.p83-91>
- Roy, A., Pivarski, J., & Freer, C. W. (2023). An array-oriented Python interface for FastJet. *Journal of Physics: Conference Series*, 2438(1), 2–7.

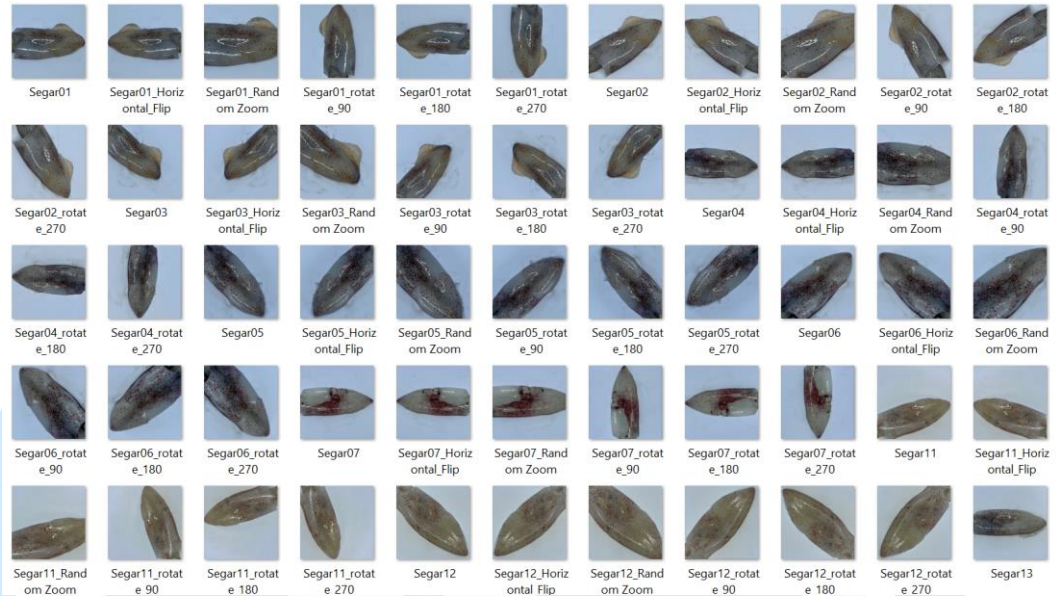
- <https://doi.org/10.1088/1742-6596/2438/1/012011>
- Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 3857–3867.
- Sitompul, N. M., Nuraini, R. A. T., & Munasik. (2021). Faktor pertumbuhan *Loligo chinensis* (Mollusca : Cephalopoda) Gray, 1849 yang Didaratkan di TPI Tambak Lorok, Semarang. *Journal of Marine Research*, 10(2), 190–199.
- Shen, M., Yang, J., Li, S., Zhang, A., & Bai, Q. (2021). Nonlinear hyperparameter optimization of a neural network in image processing for micromachines. *Micromachines*, 12(12), 1–25. <https://doi.org/10.3390/mi12121504>
- Skourt, B. A. I. T., & Nikolov, N. S. (2022). *Multi-Scale ConvLSTM Attention-Based Brain Tumor Segmentation*. 13(11).
- Sun, K., Zhang, J., Liu, J., Xu, S., Cao, X., & Fei, R. (2023). Modified Dynamic Routing Convolutional Neural Network for Pan-Sharpness. *Remote Sensing*, 15(11), 1–18. <https://doi.org/10.3390/rs15112869>
- Tan, H. Y., Goh, Z. Y., Loh, K. H., Then, A. Y. H., Omar, H., & Chang, S. W. (2021). Cephalopod species identification using integrated analysis of machine learning and deep learning approaches. *PeerJ*, 9. <https://doi.org/10.7717/peerj.11825>
- Thi, B., Hien, T., Pham, D. T., Tuyen, P. Van, & Tran, H. (2020). *A quality index method for squid Uroteuthis (Photololigo) chinensis (Gray, 1849) preserved on ice*. 8(2), 216–222.
- Tsany, F. Z., Bimantoro, F., & Nugraha, S. G. (2021). Klasifikasi tingkat kesegaran cumi-cumi berdasarkan fitur tekstur dan warna dengan menggunakan metode support vector machine. *Jtika*, 3(2), 220–231. <https://jtika.if.unram.ac.id/index.php/JTIKA/>
- Vasudevan, N., & Karthick, T. (2023). A Hybrid Approach for Plant Disease Detection Using E-GAN and CapsNet. *Computer Systems Science and Engineering*, 46(1), 337–356. <https://doi.org/10.32604/csse.2023.034242>
- Wang, J., Ran, R., & Fang, B. (2023). Global and Local Structure Network for Image Classification. *IEEE Access*, 11(January), 27963–27973. <https://doi.org/10.1109/ACCESS.2023.3258972>
- Wang, S., Liu, D., Wang, N., & Yuant, Y. (2020). Design and Implementation of an Online Python Teaching Case Library for the Training of Application Oriented Talents. *International Journal of Emerging Technologies in Learning*, 15(21), 217–230. <https://doi.org/10.3991/IJET.V15I21.18191>

- Wang, Y., Huang, L., Jiang, S., Wang, Y., Zou, J., Fu, H., & Yang, S. (2020). Capsule networks showed excellent performance in the classification of hERG blockers/nonblockers. *Frontiers in Pharmacology*, 10(January), 1–12. <https://doi.org/10.3389/fphar.2019.01631>
- Wen, X., Zeng, M., Chen, J., Maimaiti, M., & Liu, Q. (2023). Recognition of Wheat Leaf Diseases Using Lightweight Convolutional Neural Networks against Complex Backgrounds. *Life*, 13(11), 1–22. <https://doi.org/10.3390/life13112125>
- Yuen, B., Hoang, M. T., Dong, X., & Lu, T. (2021). Universal activation function for machine learning. *Scientific Reports*, 11(1), 18757. <https://doi.org/10.1038/s41598-021-96723-8>
- Zairion, Kamal, M. M., Oktaviani, D., Suryanto, Nugroho, D., Kembaren, D. D., Fauzi, M., Muhtadi, A., Idriss, R., & Lahengko, S. M. (2022). *Program Perbaikan Perikanan Cumi Medan, Perairan Selat Malaka WPPNRI 571 Tim A1 Dan A2: Riset Spesies, Habitat, Marine Ekosistem Dan Pengkajian Stok*.
- Zhang, X. Q., Wang, G. Y., & Zhao, S. G. (2022). CapsNet-COVID19: Lung CT image classification method based on CapsNet model. *Mathematical Biosciences and Engineering*, 19(5), 5055–5074. <https://doi.org/10.3934/mbe.2022236>

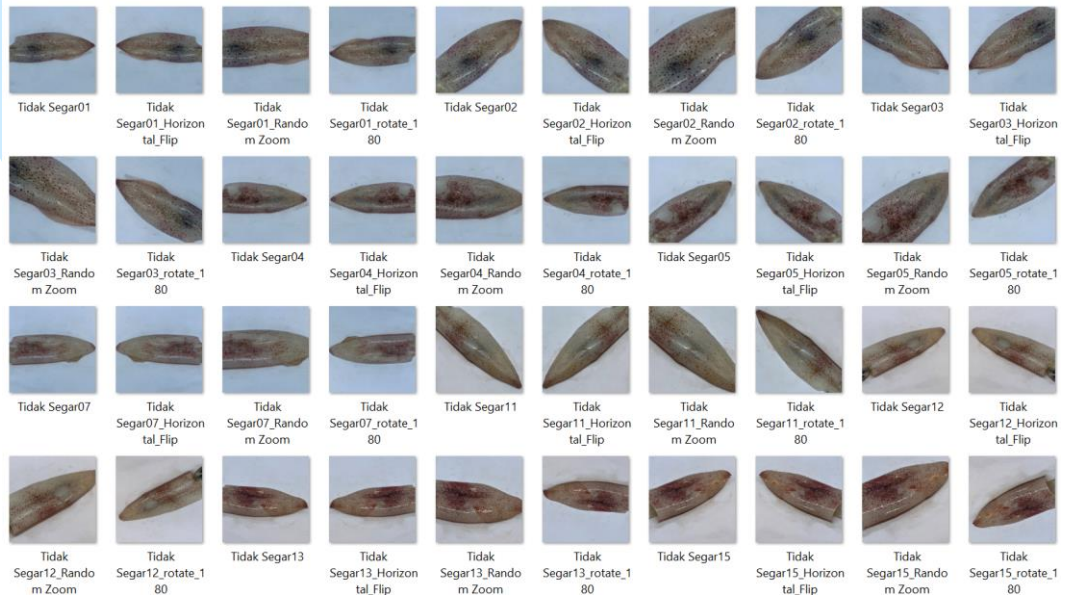
LAMPIRAN

Lampiran 1. Data yang digunakan

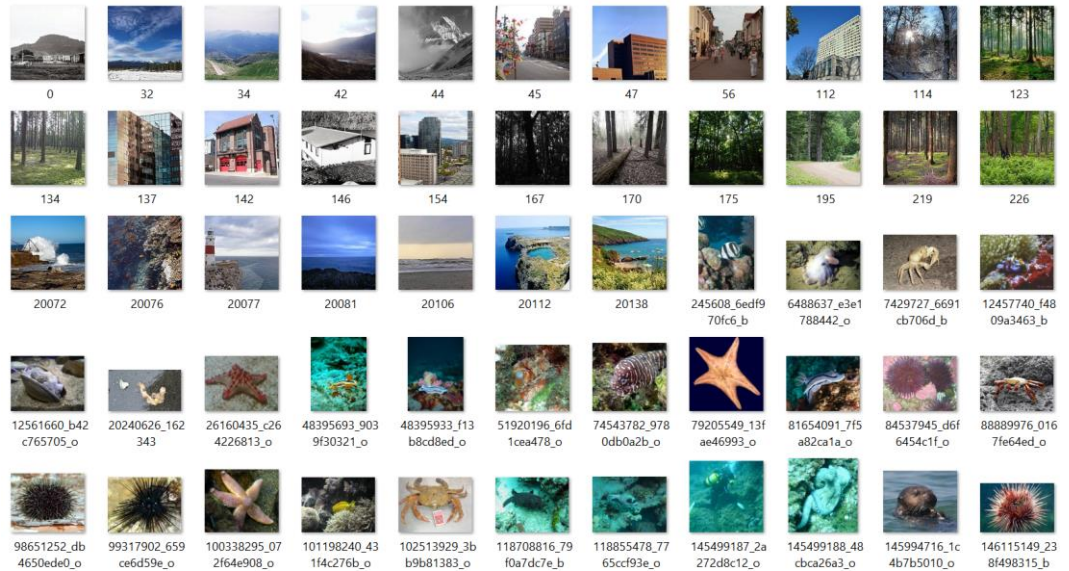
a. Segar



b. Tidak Segar



c. Non Cumi



Lampiran 2. Data uji yang digunakan

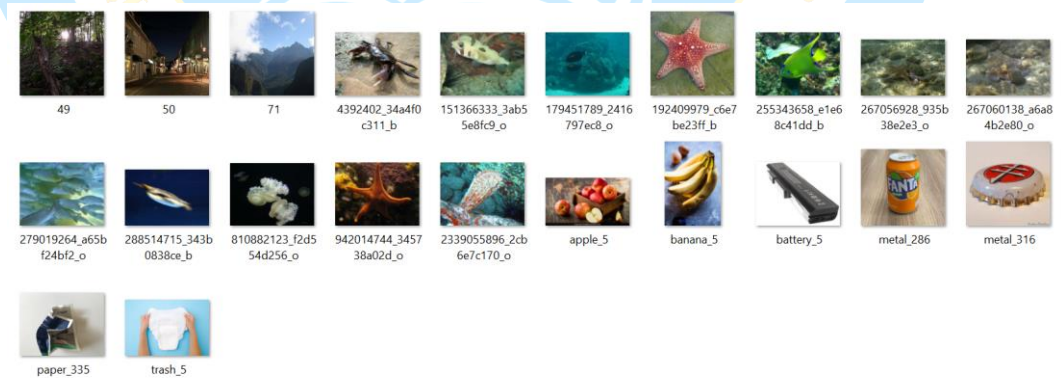
a. Segar



b. Tidak Segar



c. Non Cumi



Lampiran 3. Data Pengujian Organoleptik

