

SKRIPSI

**ANALISIS PERBANDINGAN KINERJA YOLO DAN FASTER
R-CNN PADA DETEKSI JENIS DAN ESTIMASI BERAT IKAN
KERAPU DAN IKAN KAKAP MENGGUNAKAN CITRA
DIGITAL**

Disusun dan diajukan oleh:

**SABDA ANSARI BAKE
D121 19 1037**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**

LEMBAR PENGESAHAN SKRIPSI

ANALISIS PERBANDINGAN KINERJA YOLO DAN FASTER R-CNN PADA DETEKSI JENIS DAN ESTIMASI BERAT IKAN KERAPU DAN IKAN KAKAP MENGGUNAKAN CITRA DIGITAL

Disusun dan diajukan oleh

SABDA ANSARI BAKE
D121191037

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 14 Juni 2024 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,



Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.
NIP. 19750716 200212 1 004

Pembimbing Pendamping,



Elly Warni, S.T., M.T.
NIP. 19820216 200812 2 001

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.
NIP. 19750716 200212 1 004



PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Sabda Ansari Bake
NIM : D121191037
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Analisis Perbandingan Kinerja YOLO dan Faster R-CNN Pada Deteksi Jenis dan Estimasi Berat Ikan Kerapu dan Ikan Kakap Menggunakan Citra Digital

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 15 Maret 2024

Yang Menyatakan



Sabda Ansari Bake



ABSTRAK

SABDA ANSARI BAKE. *Analisis Perbandingan Kinerja YOLO dan Faster R-CNN Pada Deteksi Jenis dan Estimasi Berat Ikan Kerapu dan Ikan Kakap Menggunakan Citra Digital* (dibimbing oleh Indrabayu dan Elly Warni)

Ikan kerapu dan ikan kakap merupakan jenis ikan yang banyak dikonsumsi dan memiliki nilai ekonomi tinggi di pasar global. Untuk menentukan nilai ekonomi dari ikan tersebut, identifikasi jenis dan estimasi berat menjadi aspek penting dalam penentuan harga dan kualitas ikan yang diperdagangkan. Metode manual yang umum digunakan cenderung memakan waktu yang lama serta memerlukan tenaga kerja yang tidak sedikit. Berdasarkan hal tersebut, diperlukan metode berbasis komputer yang lebih efektif dalam melakukan kegiatan repetitif tersebut.

Tujuan dari penelitian ini yaitu menganalisis performa dari kedua model *deep learning* yang sering digunakan yaitu YOLO dan Faster R-CNN, pada kasus pendeteksian jenis dan estimasi berat pada jenis-jenis ikan kerapu dan ikan kakap tertentu.

Data yang digunakan berjumlah 2991 data yang terbagi 18 kelas. Kemudian data tersebut diaugmentasi menggunakan fitur *rotate* dan *flip* menjadi 6843 data citra. Pada proses deteksi digunakan *detection threshold* sebesar 0,8 yang berarti objek yang terdeteksi dengan *confidence* dibawah 0,8 akan diabaikan. Kedua model tersebut setelah dilatih akan diuji performanya menggunakan parameter *precision*, *recall*, dan *accuracy* untuk menguji seberapa akurat model dalam memprediksi jenis ikan pada data masukan serta *Mean Absolute Percentage Error* (MAPE) untuk mengevaluasi hasil estimasi oleh model.

Dalam perbandingan antara model YOLO dan Faster R-CNN, terdapat perbedaan dalam hasil evaluasi kuantitatif. Model YOLO menunjukkan tingkat *precision*, *recall* dan *accuracy* secara berturut sebesar 0,98, 0,98, dan 0,96, sementara model Faster R-CNN memiliki nilai *precision*, *recall*, dan *accuracy* sebesar 0,97, 0,98, dan 0,95. Selain itu, MAPE pada estimasi berat untuk model YOLO adalah 2,42% untuk data citra dan 3,66% untuk data video, sementara untuk Faster R-CNN, nilainya yaitu 14,62% untuk data citra dan 13,59% untuk data video. Dengan demikian, dapat disimpulkan bahwa model YOLO memberikan hasil yang lebih baik dalam evaluasi kuantitatif dibandingkan dengan Faster R-CNN.

Kata Kunci: kerapu, kakap, YOLO, Faster R-CNN, deteksi jenis, estimasi berat



ABSTRACT

SABDA ANSARI BAKE. *Comparative Analysis of YOLO and Faster R-CNN Performance in Detecting Varieties and Estimating Weight of Grouper and Snapper Fish Using Digital Images* (supervised by Mr. Indrabayu and Mrs. Elly Warni)

Grouper and snapper are economically valuable fish species widely consumed globally. Accurate identification of fish type and estimation of weight are crucial for determining their market value. The manual methods currently used for these tasks are time-consuming and labor-intensive, underscoring the need for a more efficient, computer-based approach.

This study assesses the performance of two popular deep learning models, YOLO and Faster R-CNN, in the identification of types and weight estimation of selected grouper and snapper species. The dataset includes 2991 data across 18 classes, which were augmented to 6843 images using rotation and flipping. Detection threshold of 0.8 were applied, which meaning if detected objects with confidence level below 0.8 were disregarded. These models were then trained to predict fish species from the input data.

The evaluation of model performance was based on precision, recall, and accuracy metrics. Additionally, the Mean Absolute Percentage Error (MAPE) was used to evaluate the accuracy of weight estimations provided by each model. This dual approach allowed for a comprehensive assessment of each model's capability in both identification and quantification tasks.

Quantitative comparisons showed distinct performance differences between the two models. The YOLO model demonstrated precision, recall, and accuracy rates of 0.98, 0.98, and 0.96, respectively, compared to Faster R-CNN, which scored 0.97, 0.98, and 0.95. In weight estimation, YOLO achieved a MAPE of 2.42% for images and 3.66% for video data, meanwhile Faster R-CNN's 14.62% for images and 13.59% for video data. These results suggest that YOLO is more effective than Faster R-CNN.

Keywords: grouper, snapper, YOLO, Faster R-CNN, variety detection, weight estimation



DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	vii
DAFTAR SINGKATAN DAN ARTI SIMBOL	viii
DAFTAR LAMPIRAN.....	ix
KATA PENGANTAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Ruang Lingkup.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1. Ikan Kerapu.....	3
2.2. Ikan Kakap	4
2.3. Pengolahan Citra	5
2.4. <i>Object Detection</i>	6
2.5. <i>Deep Learning</i>	7
2.6. <i>Faster Region-proposed Convolutional Neural Network (Faster R-CNN)</i>	9
2.7. <i>You Only Look Once (YOLO)</i>	10
2.8. Estimasi Berat	11
2.9. <i>Confusion Matrix</i>	13
2.10. <i>Mean Absolute Percentage Error (MAPE)</i>	15
2.11. <i>Cosine Similarity</i>	16
BAB III METODE PENELITIAN/PERANCANGAN	17
3.1. Lokasi dan Waktu Penelitian	17
3.2. Instrumen Penelitian	17
3.3. Tahapan Penelitian	18
3.4. Teknik Pengambilan Data	19
3.5. Rancangan Sistem	20
BAB IV HASIL DAN PEMBAHASAN	34
4.1. Analisis Hasil <i>Training</i> dan <i>Testing</i> Model.....	34
4.2. Analisis Hasil Estimasi Berat oleh Model	45
BAB V KESIMPULAN DAN SARAN.....	52
5.1. Kesimpulan	52
5.2. Saran.....	52
DAFTAR PUSTAKA	53



DAFTAR GAMBAR

Gambar 1 Ikan Kerapu	3
Gambar 2 Ikan Kakap	4
Gambar 3 Representasi Citra Digital	6
Gambar 4 Arsitektur <i>Deep Learning</i>	8
Gambar 5 Arsitektur Faster R-CNN	9
Gambar 6 Arsitektur RPN.....	10
Gambar 7 Arsitektur YOLO.....	11
Gambar 8 Struktur dari <i>Confusion Matrix</i>	14
Gambar 9 Lokasi penelitian	17
Gambar 10 Alur tahapan penelitian	18
Gambar 11 Skenario pengambilan data untuk <i>testing</i> dan estimasi berat.....	20
Gambar 12 Alur proses sistem	21
Gambar 13 Contoh <i>dataset</i> citra	22
Gambar 14 Proses <i>preprocessing</i> berupa <i>resize</i>	23
Gambar 15 Proses <i>labelling</i>	23
Gambar 16 Contoh hasil augmentasi dari citra yang sama	24
Gambar 17 Alur proses deteksi	26
Gambar 18 Contoh hasil prediksi oleh model.....	28
Gambar 19 Rancangan <i>output</i> estimasi berat ikan.....	29
Gambar 20 Grafik <i>loss</i> pada YOLO.....	34
Gambar 21 Hasil deteksi YOLO pada sampel	35
Gambar 22 <i>Confusion matrix</i> pada model YOLO	36
Gambar 23 Grafik <i>loss</i> pada Faster R-CNN	39
Gambar 24 Hasil deteksi Faster R-CNN pada sampel	40
Gambar 25 <i>Confusion matrix</i> pada model Faster R-CNN	41
Gambar 26 <i>Confusion matrix</i> untuk jenis ikan yang mirip secara visual pada model YOLO.....	44
Gambar 27 <i>Confusion matrix</i> untuk jenis ikan yang mirip secara visual pada Faster R-CNN.....	44



DAFTAR TABEL

Tabel 1 <i>Range</i> nilai MAPE	15
Tabel 2 Pengaturan parameter pada proses <i>training</i> model	25
Tabel 3 Kategori ikan.....	30
Tabel 4 Kelompok jenis ikan dengan kemiripan visual	32
Tabel 5 Hasil evaluasi model YOLO	37
Tabel 6 Hasil evaluasi model Faster R-CNN.....	42
Tabel 7 Hasil pengujian <i>cosine similarity</i>	43
Tabel 8 Hasil deteksi pada data citra.....	46
Tabel 9 Hasil estimasi berat pada citra menggunakan model YOLO.....	46
Tabel 10 Evaluasi hasil estimasi berat pada citra menggunakan model YOLO ...	47
Tabel 11 Hasil estimasi berat pada citra menggunakan model Faster R-CNN.....	47
Tabel 12 Evaluasi hasil estimasi berat pada citra menggunakan model Faster R-CNN	48
Tabel 13 Hasil deteksi pada data video.....	49
Tabel 14 Hasil estimasi berat pada video menggunakan model YOLO	49
Tabel 15 Evaluasi hasil estimasi berat pada video menggunakan model YOLO	50
Tabel 16 Hasil estimasi berat pada data video menggunakan model Faster R- CNN	50
Tabel 17 Evaluasi hasil estimasi berat pada data video menggunakan model Faster R-CNN.....	51



DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
YOLO	<i>You Only Look Once</i>
R-CNN	<i>Region-based Convolutional Neural Network</i>
RPN	<i>Region Proposal Network</i>
IoU	<i>Intersection over Union</i>
mAP	<i>mean Average Precision</i>
MAPE	<i>Mean Absolute Percentage Error</i>
W	Lambang berat
a	Koefisien jenis
b	Koefisien pertumbuhan
L	Satuan panjang
N, n	Banyaknya kelas atau sampel
TP	<i>True Positive</i>
FP	<i>False Positive</i>
TN	<i>True Negative</i>
FN	<i>False Negative</i>



DAFTAR LAMPIRAN

Lampiran 1. Tabel jenis ikan pada <i>dataset</i>	57
Lampiran 2. Nilai pendekatan konstanta jenis dan pertumbuhan pada ikan.....	63
Lampiran 3. <i>Source code</i>	64
Lampiran 4. Tabel <i>Box Loss</i> dan <i>Classification Loss</i> pada Model YOLO dan Faster R-CNN.....	83
Lampiran 5. Lembar perbaikan skripsi	85



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan tugas akhir atau skripsi yang bertajuk “Analisis Perbandingan Kinerja YOLO dan Faster R-CNN pada Deteksi Jenis dan Estimasi Berat Ikan Kerapu dan Ikan Kakap Menggunakan Citra Digital” sebagai salah satu syarat utama dalam penyelesaian studi sarjana pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa untuk mencapai titik ini tentu saja tidak dapat dilakukan tanpa dukungan dari banyak pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih yang tulus kepada:

1. Allah SWT atas berkat dan rahmat-Nya sehingga penulis dimudahkan dalam proses penyelesaian tugas akhir ini.
2. Kedua orang tua penulis, ayahanda La Koansa, SP (Alm.) dan ibunda Ni Nyoman Seriasih, S.Pd. yang selalu memberikan dukungan, do'a dan restu yang mengalir tiada henti kepada penulis.
3. Keluarga dekat penulis, kakanda Yani Ansari Wuna, S.Pd., kakanda Ahmad Luthfi Hadi, S.Pd. dan ananda Lubna Nafisha Shareen Panguriseng yang menjadi penyemangat bagi penulis untuk berjuang menyelesaikan studi ini.
4. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng selaku Pembimbing I dan Ibu Elly Warni, S.T., M.T. selaku Pembimbing II atas waktu, tenaga, ilmu serta perhatian yang dicurahkan selama membimbing penulis dalam penyusunan tugas akhir.
5. Segenap dosen dan tendik Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu kelancaran selama proses perkuliahan.
6. Ibu Dr. Ir. Nadiarti, M.Sc. serta rekan-rekan dari Manajemen Sumberdaya Perairan, Fakultas Ilmu Kelautan dan Perikanan, Universitas Hasanuddin yang



membuka wawasan penulis mengenai ilmu perikanan, khususnya tentang kerapu dan ikan kakap yang sangat bermanfaat bagi penyusunan tugas ini maupun bagi pribadi penulis ke depannya.

7. Keluarga Besar *AIMP Research Group* yang telah memberikan masukan, saran serta dukungan yang sangat berarti bagi penulis dalam penyusunan tugas akhir ini.
8. Pahrul, Dea, Arif, Reza, Giga dan Ijlal yang telah membantu serta memberikan saran dan perbaikan yang membangun bagi penulis dalam penyusunan tugas akhir ini.
9. Juan, Khalik, Yoga, Sayyid, Hijir, Albab, Abib, Rayyan, Wira, Zaki dan Fariz yang senantiasa menemani penulis dalam proses penyusunan tugas akhir ini.
10. Teman-teman S19NIFIER secara umum atas kenangan serta dukungan yang diberikan selama penulis menjalani proses perkuliahan dari awal hingga akhir.
11. Ai, Ridzuan, Karno, Elvira, Angel, Rauf, Haekal, Sila, Agil, Arfani, Icel, Thoriq, Mely, Kak Ness, Kak Yuu serta teman-teman dan kakak-kakak dari UKM WELCOME09 SMFT-UH yang telah mewadahi penulis untuk aktif di kegiatan ekstra selama menjalani perkuliahan di kampus ini.
12. Pihak lainnya yang tidak dapat penulis sebutkan satu per satu yang telah terlibat secara langsung maupun tidak langsung dalam proses penyusunan tugas akhir ini.

Penulis berharap semoga Tuhan Yang Maha Esa berkenan untuk membalas segala kebaikan serta jasa dari semua pihak yang terlibat dalam penyusunan tugas akhir ini. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran sangat diharapkan serta masukan yang membangun dari banyak pihak.

Gowa, Maret 2024

Penulis,
Sabda Ansari Bake



BAB I PENDAHULUAN

1.1 Latar Belakang

Ikan selalu menjadi sumber protein yang penting dalam makanan manusia dan dalam skala global, ikan dan produk ikan merupakan sumber protein yang paling penting dan diperkirakan lebih dari 30% ikan untuk konsumsi manusia. Selama tiga dekade terakhir, budidaya ikan telah berkembang menjadi sektor penghasil ikan terbesar di Indonesia. Sebagian besar produk ikan berasal dari produsen skala kecil di negara-negara berkembang (Bashir dkk., 2019).

Ikan kerapu dan ikan kakap merupakan dua jenis ikan yang banyak dikonsumsi dan memiliki nilai ekonomi tinggi di pasar global. Identifikasi jenis dan estimasi berat ikan merupakan salah satu aspek penting dalam industri perikanan untuk menentukan harga dan kualitas ikan. Tradisionalnya, identifikasi dan estimasi berat ikan dilakukan secara manual oleh para ahli, namun metode ini memerlukan waktu yang lama.

Metode yang umum digunakan untuk mendeteksi jenis dan estimasi berat dari ikan kerapu dan ikan kakap adalah metode manual. Metode manual ini memiliki beberapa kelemahan, yaitu membutuhkan waktu yang lama serta memerlukan tenaga kerja yang banyak. Oleh karena itu, diperlukan metode yang lebih efektif dan efisien, yaitu metode berbasis komputer. Salah satu metode berbasis komputer yang dapat digunakan untuk mendeteksi jenis dan estimasi berat dari ikan kerapu dan ikan kakap adalah metode pengolahan citra. Metode pengolahan citra ini memiliki beberapa keunggulan, yaitu cepat, akurat dan dapat digunakan untuk mendeteksi ikan. Penelitian yang mengangkat topik tersebut yaitu penelitian yang dilakukan oleh Fernandes dkk. (2020) dimana pada penelitian tersebut menggunakan ikan nila sebagai objek penelitian dan menggunakan metrik *Intersection over Union* (IoU) pada data *test* sebagai evaluasi model. Hasil IoU dari penelitian tersebut sebesar 99%, 90% dan 64% berturut untuk *background*, area badan dan sirip ikan (Fernandes dkk., 2020).



Oh karena itulah penulis memutuskan untuk mengangkat penelitian ini yang “Analisis Perbandingan Kinerja YOLO dan Faster R-CNN pada Deteksi Estimasi Berat Ikan Kerapu dan Ikan Kakap” sebagai tugas akhir dengan untuk mengeksplorasi potensi penggunaan teknologi kecerdasan buatan

dalam industri perikanan, khususnya dalam identifikasi jenis dan estimasi berat ikan. Penelitian ini membandingkan kinerja dari dua algoritma *deep learning*, yaitu *You Only Look Once* (YOLO) dan *Faster Region-Based Convolutional Neural Network* (Faster R-CNN). Hasil dari penelitian ini dapat memberikan wawasan baru dalam pengembangan teknologi perikanan. Dengan demikian, penelitian akan memberikan kontribusi pada peningkatan produktivitas dan keberlanjutan industri perikanan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah yang timbul adalah berikut :

- Bagaimana kinerja model *deep learning* seperti YOLO dan Faster R-CNN dalam mendeteksi jenis ikan kerapu/kakap dari citra?
- Bagaimana model dapat memperkirakan berat ikan kerapu/kakap dari citra?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menganalisis performa algoritma YOLO dan Faster R-CNN dalam mendeteksi spesies ikan kerapu dan ikan kakap pada citra. Selain itu, penelitian ini juga mengevaluasi tingkat akurasi estimasi berat ikan berdasarkan citra, menggunakan rumus dan konstanta yang telah ditentukan.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini yaitu dapat memberikan kontribusi bagi perguruan tinggi dalam pengembangan ilmu pengetahuan serta membantu penerapan teknologi dalam berbagai aspek kehidupan yang relevan.

1.5 Ruang Lingkup

Adapun ruang lingkup pada penelitian ini dibatasi pada beberapa poin sebagai berikut :

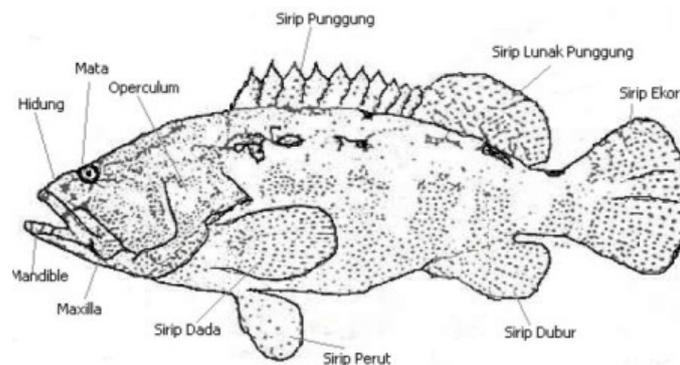


yang akan dijadikan *dataset* berupa gabungan 18 jenis yang terdiri dari 10 jenis ikan kerapu dan 7 jenis ikan kakap. Pengambilan data dilakukan dengan kondisi terang dan statis.

BAB II TINJAUAN PUSTAKA

2.1. Ikan Kerapu

Ikan kerapu, atau yang dikenal sebagai “*grouper*” dalam istilah internasional merupakan jenis ikan yang tergolong dalam *serranidae*. *Serranidae* adalah sekelompok ikan laut karnivora yang menghuni perairan tropis dan subtropis di seluruh dunia (Craig & Hastings, 2007). Sub-famili dari *Serranidae* tersebut diketahui terdiri dari 203 spesies yang tersebar di seluruh dunia (Chatla dkk., 2019). Karena sifat dari ikan kerapu yang karnivora tersebut, maka ikan ini cenderung memakan berbagai macam hewan, termasuk ikan-ikan kecil, plankton hewani, udang-udangan, invertebrata, dan hewan-hewan kecil lainnya. Ikan kerapu juga memiliki habitat yang cukup beragam, mulai dari terumbu karang, daerah berbatu hingga rumput laut (Darwin dkk., 2020). Gambar dari ikan kerapu dapat dilihat pada Gambar 1.



Gambar 1 Ikan Kerapu

Sumber : Ningsih, (2015)

Pada Gambar 1 dapat dilihat bahwa ikan kerapu memiliki tubuh yang kuat dan kokoh, dengan kepala yang besar dan mulut yang lebar. Seluruh tubuhnya ditutupi oleh sisik-sisik kecil yang berbentuk bulat dan rapat. Pada pinggiran operculum, terdapat duri-duri yang tajam. Dua sirip punggungnya yang pertama

entuk duri-duri (Ningsih, 2015).

n kerapu adalah ikan laut yang berukuran besar, mulai dari 30 cm hingga
Ikan ini memiliki ciri khas berupa bentuk, corak, dan warna tubuh yang
i. Namun, secara morfologi, kerapu sangat sulit dibedakan antar



spesiesnya. Hal ini dikarenakan kerapu memiliki banyak spesies yang mirip satu sama lain, sehingga beberapa masyarakat terkadang memiliki sebutan yang sama untuk dua hingga tiga ekor kerapu yang berbeda (Ding dkk., 2006).

2.2. Ikan Kakap

Ikan kakap yang memiliki marga *Lutjanus*, merupakan salah satu jenis ikan yang secara umum dimanfaatkan sebagai ikan konsumsi dan menjadi salah satu sumber daya yang penting secara komersial maupun rekresional (Oktaviyani, 2018).



Gambar 2 Ikan Kakap

Sumber : Saha dkk., (2018)

Ikan yang juga dikenal sebagai “*snapper*” ini termasuk ke dalam famili *Lutjanidae* ini terdapat di seluruh dunia di daerah tropis dan subtropis dan ditemukan dari daerah pesisir yang dangkal hingga dalam. Sebagian besar spesies hidup di atau dekat dasar dan sebagian besar terbatas pada landas kontinen dan lereng dan kedalaman yang sesuai di sekitar pulau; tetapi beberapa memasuki muara dan bahkan air tawar (Anderson, 1967).



Ikan kakap yang tersebar di kawasan tropis dan subtropis memegang peran dalam ekosistem laut. Di Asia Tenggara, kawasan yang kaya akan keanekaragaman hayati laut, ikan kakap tidak hanya berperan sebagai predator, tetapi juga sebagai spesies kunci dalam menjaga keseimbangan ekologis. Mereka berperan dalam mengontrol populasi ikan kecil dan invertebrata laut, yang jika

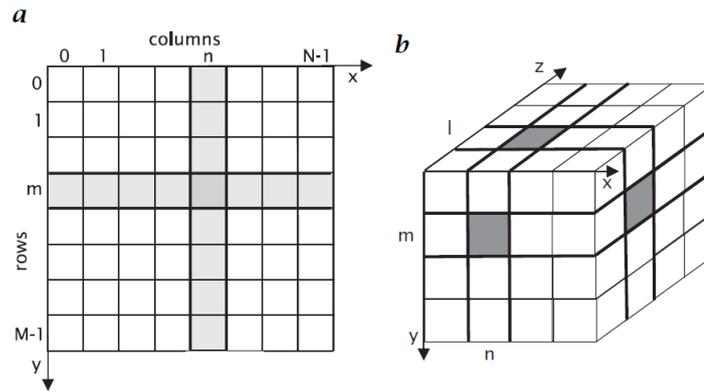
tidak dikendalikan, dapat menyebabkan ketidakseimbangan ekologis. Selain itu, ikan kakap juga penting secara ekonomi, terutama bagi komunitas pesisir yang menggantungkan hidupnya pada penangkapan ikan.

2.3. Pengolahan Citra

Pengolahan citra adalah suatu metode yang digunakan untuk menjalankan suatu operasi pada suatu citra dengan tujuan untuk meningkatkan kualitasnya atau untuk memperoleh informasi yang dianggap berguna dari citra tersebut. Pengolahan citra juga merupakan pemrosesan sinyal dengan menggunakan citra sebagai *input* dan *output* yang dihasilkan dapat berupa citra atau karakteristik maupun fitur yang terhubung dengan citra *input* tadi. Terdapat dua metode yang umum pada pengolahan citra, yaitu pengolahan citra analog dan pengolahan citra digital. Pengolahan citra analog dapat digunakan untuk penggunaan *hard copy* seperti percetakan dokumen maupun foto. Sedangkan pengolahan citra digital yaitu metode manipulasi citra menggunakan komputer.

Sebuah citra dapat didefinisikan sebagai fungsi dua dimensi, $f(x, y)$, di mana x dan y adalah koordinat spasial (bidang), dan amplitudo f pada pasangan koordinat mana pun (x, y) disebut intensitas atau tingkat keabuan gambar pada titik tersebut. Ketika x , y , dan nilai intensitas f merupakan besaran diskrit yang terbatas, kita menyebut gambar tersebut sebagai gambar digital. Bidang pengolahan citra digital mengacu pada pemrosesan pada citra dengan menggunakan komputer digital. Perlu diperhatikan bahwa gambar digital terdiri atas sejumlah elemen yang terbatas, yang masing-masing memiliki lokasi dan nilai tertentu. Elemen-elemen ini disebut elemen gambar dan piksel. Piksel adalah istilah yang paling banyak digunakan untuk menunjukkan elemen-elemen gambar digital (Gonzalez & Woods, 2018). Representasi elemen citra dapat dilihat pada Gambar 3.





Gambar 3 Representasi Citra Digital

Sumber : Jähne (2005)

Berdasarkan Gambar 3, dapat dilihat representasi letak dari suatu piksel pada citra dapat dinyatakan dalam bentuk matriks. Indeks pertama yang biasa dituliskan sebagai m merepresentasikan letak suatu titik pada baris dan indeks kedua yang dituliskan sebagai n merepresentasikan letak suatu titik pada kolom. Jika suatu citra digital berisi $M \times N$ piksel, maka citra tersebut direpresentasikan oleh matriks $M \times N$ dimana M berisi jumlah baris dan N berisi jumlah kolom. Pada matriks citra tersebut yang direpresentasikan sebagai $M \times N$, nilai n berada pada rentang 0 hingga $N - 1$ dan nilai m berada pada rentang 0 hingga $M - 1$. Selain itu, sumbu y pada matriks tersebut dimulai dari atas hingga ke bawah. Sedangkan sumbu x dimulai dari kiri ke kanan (Jähne, 2005).

2.4. Object Detection

Object detection merupakan metode deteksi yang sering digunakan dalam penerapan visi komputer. Pendeteksian yang biasa dilakukan seperti deteksi pejalan kaki, deteksi wajah atau pengenalan wajah dan lain-lain. Saat ini, terdapat dua kategori yang lazim digunakan pada *object detection* ini, yaitu *traditional machine learning* dan *deep learning* (Zou, 2019).

Object detection juga merupakan proses kompleks yang melibatkan ekstraksi fitur, klasifikasi, regresi *bounding box*, *non-maximum suppression* dan *post-*

processing. Proses kompleks tersebut dapat dibagi menjadi beberapa poin berikut. *Feature Extraction* atau Ekstraksi Fitur merupakan tahapan pertama dalam proses deteksi objek yang diperlukan untuk mengambil fitur-fitur penting pada citra maupun video. Fitur-fitur tersebut dapat berupa gambar, tekstur,



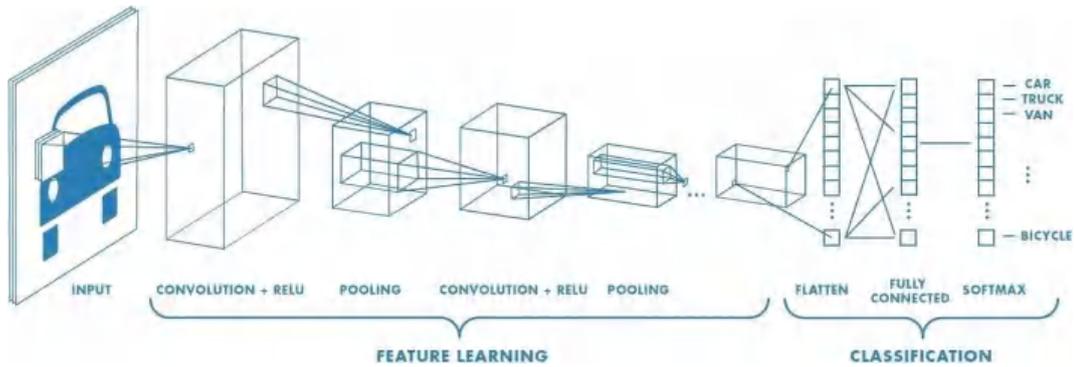
bentuk maupun fitur visual lainnya yang membedakan objek tersebut dengan keadaan sekitarnya (Lin dkk., 2019).

2. *Classification* : Setelah fitur pada data diekstrak, dijalankan proses klasifikasi untuk membedakan objek tersebut berdasarkan fitur sebelumnya. Proses klasifikasi ini membandingkan fitur yang sudah dikenali dengan fitur yang ada pada data yang akan diklasifikasi (Park & Berg, 2016).
3. *Bounding-box regression*: Setelah klasifikasi, langkah selanjutnya adalah menentukan lokasi objek pada gambar. Ini biasanya dilakukan dengan menggunakan algoritma regresi kotak pembatas, yang memprediksi koordinat kotak pembatas di sekitar objek (Karim dkk., 2021).
4. *Non-maximum suppression* atau NMS: Algoritma deteksi objek sering mengembalikan beberapa kotak pembatas untuk objek yang sama. NMS adalah teknik yang digunakan untuk menghapus kotak pembatas duplikat dan hanya menyimpan satu dengan skor kepercayaan tertinggi (Lin dkk., 2019).
5. *Post-processing*: Akhirnya, output dari algoritma deteksi objek adalah *post-process* untuk menghasilkan hasil akhir. Ini dapat mencakup memfilter positif palsu, menggabungkan kotak pembatas yang tumpang tindih, dan menampilkan hasilnya dalam format yang mudah digunakan (Lin dkk., 2019; Park & Berg, 2016).

2.5. Deep Learning

Deep learning merupakan cabang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk melakukan pemodelan data yang kompleks. Metode ini telah menjadi kunci dalam berbagai terobosan di bidang kecerdasan buatan, terutama dalam pengolahan bahasa alami, visi komputer, dan pemahaman suara. Berbeda dengan pendekatan *traditional machine learning*, *deep learning* memungkinkan model untuk belajar fitur-fitur dari data secara otomatis tanpa perlu intervensi manual melalui proses yang disebut dengan *feature learning* (LeCun dkk., 2015). Arsitektur model *deep learning*, khususnya model untuk *computer vision* dapat dilihat pada Gambar 4.





Gambar 4 Arsitektur *Deep Learning*

Sumber : Saha (2018)

Berdasarkan Gambar 4, dapat dilihat bahwa *deep learning* memiliki beberapa *layer* atau lapisan yang memiliki fungsi-fungsi tertentu. Lapisan-lapisan yang ada pada *deep learning* secara umum adalah sebagai berikut.

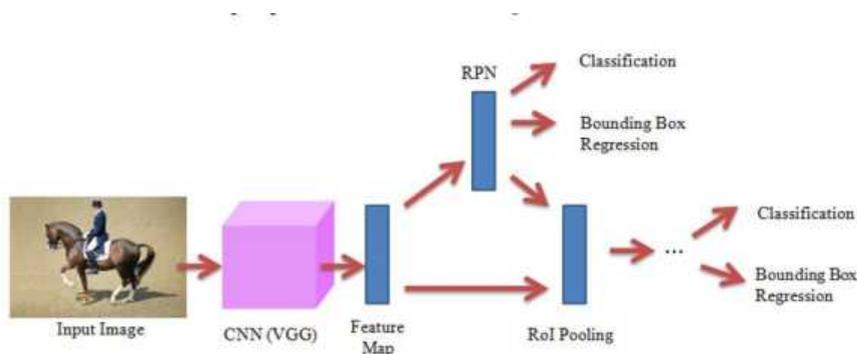
1. *Input layer*: sebagai titik awal memasukkan data yang telah dipra-proses sebelumnya agar kompatibel dengan struktur jaringan tersebut.
2. *Convolutional layer*: lapisan yang sangat penting untuk pemrosesan citra, dimana lapisan-lapisan ini mengaplikasikan proses konvolusi untuk memperoleh fitur-fitur penting seperti tepian objek, kontur dan *corner* pada citra.
3. *Pooling layer*: lapisan ini berfungsi untuk mengurangi dimensi pada data (lebar dan tinggi) dari volume *input* untuk lapisan konvolusional berikutnya. Lapisan tersebut melakukan operasi *down-sampling* di sepanjang dimensi spasial *input*, secara efektif mengurangi jumlah parameter dan perhitungan yang diperlukan dalam jaringan.
4. *ReLU layer*: lapisan yang berperan dalam fungsi aktivasi ReLU (*Rectified Linear Unit*) memiliki fungsi yang penting dalam *neural network*, khususnya CNN. Lapisan tersebut menerapkan non-linieritas pada suatu model yang berperan penting dalam mempelajari pola-pola kompleks pada data.
5. *Fully Connected Layer*: lapisan ini mengintegrasikan fungsi-fungsi yang ada dari lapisan sebelumnya.



2.6. *Faster Region-proposed Convolutional Neural Network (Faster R-CNN)*

Faster R-CNN merupakan algoritma yang ditingkatkan dari R-CNN dan mulai diperkenalkan pada tahun 2015. Faster R-CNN sendiri membaurkan Fast R-CNN dan *Region Proposal Network* (RPN) sebagai arsitekturnya. Dengan kata lain, Fast R-CNN dan Faster R-CNN dapat dikatakan hampir serupa, hanya saja ada fitur pada Fast R-CNN yang diganti untuk menciptakan Faster R-CNN (Shianto dkk., 2019).

RPN atau *Region Proposal Network* merupakan salah satu komponen penting dalam arsitektur Faster R-CNN. Fungsi utama dari RPN sendiri adalah untuk mengidentifikasi lokasi suatu objek pada suatu citra. RPN bekerja dengan mengambil gambar sebagai *input* dan memberikan *output* berupa bentuk kotak prediksi (Ren dkk., 2017). Proses ini dimulai dengan memasukkan gambar ke dalam *Convolutional Neural Network* (CNN), seperti VGG atau ResNet, yang bertindak sebagai pengolah fitur. CNN ini mengubah gambar menjadi peta fitur berdimensi tinggi yang menangkap berbagai aspek dari citra masukan. Kemudian, RPN beroperasi pada peta fitur ini menggunakan pendekatan *sliding window*. Di setiap posisi dari *sliding window* ini, RPN mengevaluasi sejumlah kotak jangkar (*anchor boxes*) berbentuk persegi panjang dengan skala dan rasio aspek yang beragam. Kotak jangkar ini dirancang untuk mencakup berbagai ukuran dan bentuk objek yang mungkin terdapat dalam citra. Gambaran arsitektur pada Faster R-CNN dapat dilihat pada Gambar 5.



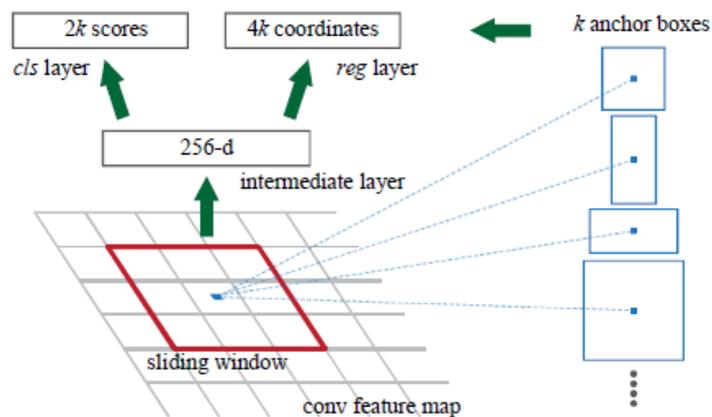
Gambar 5 Arsitektur Faster R-CNN

Sumber : Sultana dkk. (2020)

Berdasarkan Gambar 5, terdapat pembagian fitur antara RPN dan Fast R-CNN. Fitur konvolusional yang diekstrak dari citra tidak hanya digunakan untuk



menghasilkan *proposed region* oleh RPN, tetapi juga dapat dimanfaatkan oleh Fast R-CNN untuk proses klasifikasi dan regresi *bounding box* sehingga pendekatan ini memungkinkan adanya peningkatan efisiensi dengan mengurangi ketergantungan pada proses perhitungan berulang (Ren dkk., 2017).



Gambar 6 Arsitektur RPN

Sumber : Ren dkk., (2017)

Dalam proses *training* pada RPN, terdapat perhitungan *loss function* dimana setiap label kelas diberikan penanda pada setiap anchornya. Label positif berlaku pada dua kondisi :

1. *Anchor* dengan *Intersection over Units* (IoU) memiliki kecocokan yang tinggi pada label *ground-truth*, atau;
2. *Anchor* yang memiliki IoU lebih dari 0.7 terhadap label *ground-truth*.

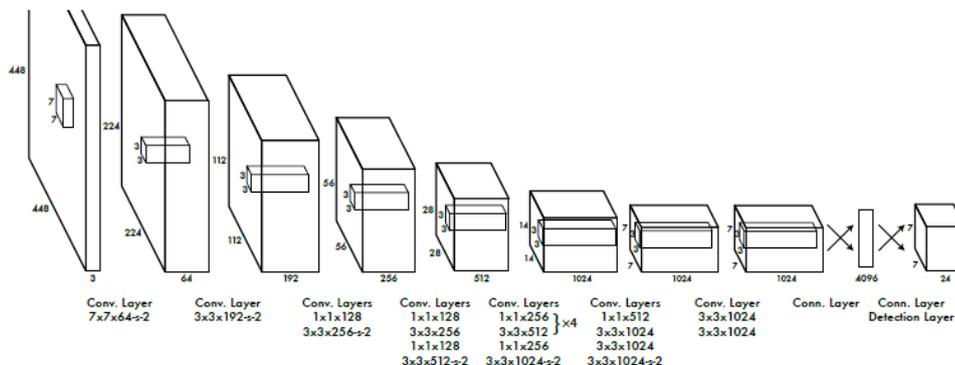
2.7. You Only Look Once (YOLO)

You Only Look Once atau yang biasa disingkat YOLO adalah salah satu algoritma yang banyak dipakai untuk kasus *object detection*. Pada tahun 2015, Redmon dkk. memberikan pengenalan versi YOLO pertama. Inti dari algoritma deteksi target YOLO terletak pada ukuran model yang kecil dan kecepatan kalkulasi yang cepat. Struktur YOLO sangat mudah. Sehingga YOLO dapat langsung menampilkan prediksi dari kelas dan posisi dengan *bounding box* melalui jaringan



cepatan YOLO tinggi karena YOLO hanya perlu memasukkan gambar ke jaringan untuk mendapatkan hasil deteksi akhir, sehingga YOLO juga dapat melakukan deteksi waktu video. Secara langsung, YOLO menggunakan gambar untuk deteksi, yang dapat menyandikan informasi global dan mengurangi

kesalahan mendeteksi latar belakang sebagai objek. YOLO memiliki kemampuan generalisasi yang kuat karena YOLO dapat mempelajari fitur yang sangat umum untuk ditransfer ke bidang lain (Jiang dkk., 2022).



Gambar 7 Arsitektur YOLO

Sumber : Redmon dkk., (2015)

YOLO membagi citra masukan menjadi kotak $S \times S$. Pada setiap kotak tersebut memprediksi *bounding box* dan probabilitas dari suatu kelas. Prediksi *bounding box* tersebut sudah termasuk titik koordinat, ukuran dan skor keyakinan dari prediksi model tersebut yang menandakan seberapa yakin model tersebut dalam memprediksi dan akurasi dari *bounding box* tersebut. Arsitektur ini berdasarkan pada *convolutional neural network* (CNN) dimana model tersebut memproses seluruh bagian dari citra masukan, sehingga model tersebut dapat menangkap informasi dan kontekstual dengan lebih efektif (Ahmed dkk., 2024).

Untuk mengevaluasi kinerja dari model YOLO yang telah dibuat, model diuji dengan menggunakan metrik *mean Average Precision* (mAP) yang berfungsi untuk mengukur akurasi model dalam mendeteksi objek pada berbagai kelas dan skenario yang telah ditentukan. Selain mAP, metrik lain seperti *Precision* dan *Recall* juga sering digunakan. Kombinasi dari metrik tersebut dapat membantu dalam memahami keseimbangan antara akurasi dan kemampuan model untuk mendeteksi semua objek yang relevan (Zhang dkk., 2024).

2.8. Estimasi Berat



Estimasi berat ikan dalam ilmu perikanan merupakan komponen yang sangat penting untuk mengevaluasi dan mengelola ikan secara kontinu. Sebagai salah satu parameter utama, analisis hubungan antara panjang dan berat ikan sangat penting. Panjang dan berat tersebut secara umum tidak hanya dapat digunakan

untuk estimasi berat, tetapi juga dapat digunakan untuk membedakan kondisi ikan, membandingkan tingkat pertumbuhan ikan pada daerah tertentu serta sebagai pelengkap untuk studi reproduksi maupun pemberian pakan terhadap spesies tertentu (Froese dkk., 2011). Dengan korelasi antara panjang dan berat tersebut, Le Cren (1951) merumuskan persamaan panjang-berat ikan yang menggunakan hubungan antara panjang ikan (L) dengan berat ikan (W) yang dapat dinyatakan dengan Persamaan 1.

$$W = a \times L^b \quad (1)$$

dimana,

W = berat ikan yang diestimasi (gr)

a = Konstanta ikan

L = Panjang tubuh ikan (cm)

b = Konstanta pertumbuhan ikan

Parameter 'a' dan 'b' ini ditentukan berdasarkan penelitian empiris yang spesifik pada setiap spesies (Froese & Pauly, 2019). Parameter 'a' adalah konstanta yang merepresentasikan aspek khusus pada ikan, dalam hal ini menggambarkan kondisi fisik ikan yang mencerminkan aspek seperti kepadatan dan komposisi tubuh ikan. Adapun parameter 'b' adalah eksponen yang menggambarkan pola pertumbuhan ikan. Untuk memperoleh nilai parameter 'a' dan 'b', dilakukan langkah-langkah berikut.

1. **Transformasi Data**, dengan melakukan perhitungan logaritma dari panjang (W) dan berat (L) ikan. Transformasi ini mengubah Persamaan 1 menjadi bentuk linear :

$$\log(W) = \log(a) + b \log(L) \quad (2)$$

dengan $\log(a)$ sebagai *intercept* (A).

2. **Regresi Linear**, digunakan pada data yang telah ditransformasi sebelumnya untuk menentukan *slope* (b) dan *intercept* (A). Rata-rata log panjang ($\overline{\log(L)}$) dan log berat ($\overline{\log(W)}$) dapat dihitung dengan Persamaan 3 dan Persamaan 4.

$$\overline{\log(L)} = \frac{\sum_{i=1}^n \log(L)_i}{n} \quad (3)$$

$$\overline{\log(W)} = \frac{\sum_{i=1}^n \log(W)_i}{n} \quad (4)$$



setelah menghitung rata-rata dari log panjang dan berat, nilai *slope* (b) dapat

hitung dengan Persamaan 5.

$$b = \frac{\sum(\log(L)_i - \overline{\log(L)})(\log(W)_i - \overline{\log(W)})}{\sum(\log(L)_i - \overline{\log(L)})^2} \quad (5)$$

Terakhir, nilai *intercept* (A) dihitung dengan Persamaan 6.

$$A = \overline{\log(W)} - b\overline{\log(L)} \quad (6)$$

3. **Back-Transformation**, untuk memperoleh nilai dari parameter 'a', nilai dari *intercept* (A) perlu di *back-transform* menjadi 'a'. Untuk mendapatkan kembali nilai 'a' tersebut, dapat dilakukan proses eksponensiasi pada A dengan Persamaan 7.

$$a = e^{-A} \quad (7)$$

Parameter 'a' ini kemudian dapat digunakan dalam model eksponensial untuk mendeskripsikan hubungan antara panjang dan berat ikan (Herath dkk., 2019)

Pendekatan parameter ini tentunya memberikan keuntungan dalam estimasi cepat dan efisien tanpa perlu menimbang setiap ikan, yang juga membantu dalam proses survei ikan skala besar. Selain itu, dengan memahami metode ini juga berkontribusi dalam pemahaman mengenai aspek ekologi seperti pertumbuhan dan kondisi ikan (Wootton, 1990).

Pendekatan metode estimasi berat dengan teknologi saat ini, terutama visi komputer, telah meningkatkan efisiensi dalam proses estimasi berat ikan. Sistem berbasis visi komputer yang menggunakan kamera dan algoritma pengolahan citra dapat mengukur ikan dengan presisi baik. Sehingga teknik ini dapat mengurangi potensi cedera pada ikan yang terjadi pada ikan jika dibandingkan dengan proses pengukuran manual (White dkk., 2006). Dengan kemajuan dalam kecerdasan buatan dan *machine learning* metode estimasi berat juga menjadi lebih optimal dalam mengidentifikasi dan mengukur ikan dari gambar, kemudian memberikan estimasi berat yang lebih akurat berdasarkan ukuran visual ikan (Zion, 2012).

2.9. Confusion Matrix

Confusion matrix merupakan suatu metode yang digunakan untuk mengevaluasi kinerja dari suatu model, khususnya model klasifikasi. Matriks

terdiri dari susunan tabel khusus yang memungkinkan untuk alisasikan performa dari suatu algoritma, yang biasanya pada *supervised*. Setiap baris pada matriks mewakili kelas yang diprediksi, sedangkan



kolom pada matriks mewakili kelas aktual (*ground-truth*). Matriks ini sangat berguna untuk mengukur *precision*, *recall*, akurasi dan skor F1.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 8 Struktur dari *Confusion Matrix*

Sumber : Nugroho, (2020)

Confusion matrix memberikan wawasan mendalam mengenai kinerja model klasifikasi. Dalam matriks tersebut terdapat empat komponen utama meliputi *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN) dan *False Negatives* (FN). TP dan TN masing-masing merupakan jumlah kasus yang benar diidentifikasi sebagai positif dan negatif oleh model, sedangkan FP dan FN adalah jumlah kasus salah identifikasi. Dari keempat komponen tersebut, kita dapat menghitung jumlah akurasi yang merupakan rasio dari jumlah prediksi yang benar (TP + TN) terhadap jumlah kasus (TP + TN + FP + FN). Akurasi ini menggambarkan seberapa banyak prediksi yang berhasil dibuat oleh model dengan tepat, tetapi tidak membedakan antara jenis kesalahan yang dibuat oleh model (He & Garcia, 2009). Untuk menghitung tingkat akurasi dapat menggunakan Persamaan 8.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Selanjutnya, *precision* merupakan ukuran yang menilai berapa persen prediksi positif yang benar-benar akurat, dihitung dengan Persamaan 9.

$$precision = \frac{TP}{TP + FP} \quad (9)$$

Selanjutnya, *recall* atau sensitivitas merupakan ukuran yang menilai seberapa baik model dalam mengidentifikasi kelas positif aktual (Chicco & Jurman, 2020). *Recall* dapat dihitung dengan Persamaan 10.

$$recall = \frac{TP}{TP + FN} \quad (10)$$



Confusion matrix juga dapat membantu dalam mengembangkan pemahaman yang lebih baik tentang kinerja model di berbagai ambang batas (*threshold*) klasifikasi yang berguna dalam pengaplikasian secara praktis (Ting, 2017).

2.10. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) adalah metode pengukuran untuk mengevaluasi ketepatan atau akurasi dalam memprediksi atau regresi pada model (de Myttenaere dkk., 2016). Perlu dipahami bahwa nilai MAPE merupakan hasil persen dari rata-rata selisih absolut yang ada diantara nilai aktual dan nilai prediksi dan penggunaannya pada evaluasi dari hasil prediksi adalah dapat melihat seberapa akurat akurasi tersebut terhadap nilai aktual dan nilai prediksi dari suatu model (Nabillah & Ranggadara, 2020). Adapun persamaan MAPE dapat dilihat pada Persamaan 11.

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (11)$$

Dimana,

A_t = nilai aktual

F_t = nilai prediksi

n = jumlah sampel yang diteliti

Semakin rendah nilai MAPE, maka kemampuan dari model dalam memprediksi sesuatu dapat dikatakan dengan baik. Maricar (2019) telah memetakan rentang atau *range* nilai MAPE menjadi empat kategori yang menandakan akurasi dari model tersebut yang lebih jelasnya dapat dilihat pada Tabel 1.

Tabel 1 *Range* nilai MAPE

Range MAPE	Keterangan
< 10%	Kemampuan prediksi sangat baik
10% – 20%	Kemampuan prediksi baik
0%	Kemampuan prediksi layak
	Kemampuan prediksi buruk

Maricar (2019)



2.11. Cosine Similarity

Cosine similarity adalah metrik yang sering digunakan dalam visi komputer untuk mengukur kesamaan antara vektor dalam suatu ruang. Dalam konteks ini, cosine similarity digunakan untuk membandingkan kesamaan fitur citra yang dihasilkan menggunakan teknik pembelajaran mendalam seperti *Convolutional Neural Networks* (CNNs). Perhitungan dasar *cosine similarity* melibatkan penentuan *cosinus* dari sudut antara dua vektor, yang setara dengan perkalian titik dari vektor-vektor tersebut dibagi dengan hasil kali dari magnitudonya. Hal ini sangat berguna dalam aplikasi seperti deteksi dan lokalisasi objek, di mana *cosine similarity* membantu mengidentifikasi dan menemukan objek secara akurat dengan membandingkan fitur-fitur objek yang terdeteksi dengan objek referensi (Zhong dkk., 2017). Rumus dari *cosine similarity* dapat dilihat pada Persamaan 12.

$$\text{Cosine similarity} = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (12)$$

Sumber : Miesle (2023)

dimana,

A, B = titik vektor

