

BAB II

TINJAUAN LITERATUR

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai acuan dan pembanding dalam penelitian yang akan dilakukan. Adapun beberapa penelitian yang relevan dengan penelitian ini adalah sebagai berikut.

Penelitian yang dilakukan oleh **Mohammad Syahrul Ramadhan, Dedy Hartama, dan Solikhun (2021)** dengan judul "Deteksi Objek Menggunakan Metode Contour Detection Berbasis OpenCV" membahas penerapan metode contour detection untuk mendeteksi objek pada citra digital. Penelitian ini bertujuan untuk mengidentifikasi objek berdasarkan bentuk dan batas objek pada citra. Hasil penelitian menunjukkan bahwa metode contour detection mampu mendeteksi objek dengan baik pada citra yang memiliki kontras yang jelas antara objek dan latar belakang.

Penelitian yang dilakukan oleh **Rizky Pratama, Eko Prasetyo, dan Agus Harjoko. (2020)** dengan judul "Implementasi Pengolahan Citra Digital untuk Identifikasi Objek Menggunakan OpenCV" membahas proses identifikasi objek menggunakan teknik pengolahan citra digital berbasis OpenCV. Tahapan yang dilakukan meliputi grayscale, thresholding, dan contour detection. Hasil penelitian menunjukkan bahwa kombinasi tahapan tersebut dapat meningkatkan keberhasilan sistem dalam mendeteksi objek pada citra digital.

Penelitian yang dilakukan oleh **Dwi Cahyo Nugroho dan Siti Nurjanah (2022)** dengan judul "Penerapan Metode Contour Detection pada Sistem Deteksi Bentuk Objek" menjelaskan bahwa contour detection dapat digunakan untuk mendeteksi bentuk objek berdasarkan batas tepi objek. Hasil penelitian menunjukkan bahwa metode contour detection mampu memberikan hasil deteksi yang cukup baik pada objek yang memiliki bentuk yang jelas.

Penelitian yang dilakukan oleh **Ahmad Fauzi, Muhammad Iqbal, dan Sri Wahyuni. (2023)** dengan judul "Deteksi Objek pada Citra Digital Menggunakan OpenCV dan Python" membahas pengembangan sistem deteksi objek berbasis bahasa pemrograman Python dan pustaka OpenCV. Penelitian tersebut menunjukkan bahwa penggunaan OpenCV dapat mempermudah proses pengolahan citra dan menghasilkan deteksi objek yang cukup akurat.

Penelitian yang dilakukan oleh **Siska Wulandari, Fajar Nugraha, dan Budi Santoso (2024)** yang berjudul "Implementasi Metode Contour Detection untuk Identifikasi Objek Ikan pada Citra Digital" membahas penerapan metode contour detection untuk mendeteksi objek ikan pada citra digital. Penelitian ini menggunakan tahapan preprocessing berupa grayscale dan thresholding sebelum dilakukan proses pencarian kontur. Hasil penelitian menunjukkan bahwa metode contour detection mampu mendeteksi objek ikan dengan tingkat akurasi yang cukup baik pada kondisi pencahayaan yang memadai (**Wulandari, Nugraha, & Santoso,**

2024).

Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa metode contour detection memiliki kemampuan yang baik dalam mendeteksi objek pada citra digital. Oleh karena itu, penelitian ini menggunakan metode contour detection untuk mendeteksi objek ikan pada citra digital.

2.2 Pengolahan Citra Digital

Pengolahan citra digital (Digital Image Processing) merupakan salah satu cabang ilmu komputer yang mempelajari teknik untuk memperoleh, memproses, memanipulasi, dan menganalisis citra digital sehingga menghasilkan informasi yang lebih bermakna. Teknologi ini memungkinkan komputer melakukan interpretasi terhadap suatu citra secara otomatis sehingga banyak dimanfaatkan dalam berbagai bidang, seperti kesehatan, industri, keamanan, pertanian, hingga perikanan. (Dijaya & Setiawan, 2023).

Pada dasarnya, pengolahan citra digital dilakukan dengan memanipulasi nilai intensitas pada setiap piksel penyusun citra. Perubahan nilai piksel tersebut bertujuan untuk meningkatkan kualitas citra, mengurangi gangguan (noise), mempertajam objek, serta mempermudah proses analisis oleh sistem komputer. Berbagai teknik yang umum digunakan dalam pengolahan citra digital meliputi image enhancement, konversi grayscale, segmentasi citra, deteksi tepi, dan deteksi objek. (Ashillah et al., 2025).

Secara umum, proses pengolahan citra digital terdiri atas beberapa tahapan, yaitu akuisisi citra, preprocessing, segmentasi, ekstraksi ciri (feature extraction), dan interpretasi hasil. Setiap tahapan memiliki fungsi yang saling berkaitan untuk menghasilkan informasi yang akurat dari citra yang diproses. Tahap preprocessing bertujuan meningkatkan kualitas citra sebelum dilakukan analisis, sedangkan tahap segmentasi digunakan untuk memisahkan objek dari latar belakang sehingga proses deteksi dapat dilakukan dengan lebih optimal. (Sari et al., 2023).

Dalam penelitian ini, pengolahan citra digital digunakan untuk mendeteksi objek ikan pada citra digital menggunakan metode Contour Detection. Sebelum proses deteksi dilakukan, citra terlebih dahulu melalui tahap preprocessing yang meliputi konversi citra menjadi grayscale dan proses thresholding. Tahapan tersebut bertujuan untuk menyederhanakan informasi citra serta memperjelas batas objek sehingga metode Contour Detection dapat mendeteksi objek ikan dengan lebih baik. (Wardana, Maskuri, & Zaim, 2023).

2.3 Citra Digital

Citra digital merupakan representasi suatu objek dalam bentuk data digital yang tersusun atas sekumpulan piksel (picture element). Setiap piksel memiliki nilai intensitas tertentu yang menunjukkan tingkat kecerahan atau warna pada posisi tertentu dalam citra. Citra digital diperoleh melalui proses akuisisi menggunakan perangkat seperti kamera digital, scanner, atau sensor lainnya. Dengan berkembangnya teknologi

komputer, citra digital menjadi salah satu sumber informasi yang banyak dimanfaatkan dalam berbagai bidang, seperti pengolahan citra, visi komputer (computer vision), kecerdasan buatan, dan sistem deteksi objek. (Dijaya & Setiawan, 2023).

Citra digital memiliki karakteristik yang dipengaruhi oleh jumlah piksel (resolusi), kedalaman warna (color depth), serta format penyimpanan citra. Semakin tinggi resolusi citra, maka semakin banyak informasi visual yang dapat direpresentasikan sehingga kualitas citra menjadi lebih baik. Namun, peningkatan resolusi juga akan menyebabkan ukuran berkas (file size) menjadi lebih besar dan membutuhkan waktu pemrosesan yang lebih lama. Oleh karena itu, pemilihan resolusi citra harus disesuaikan dengan kebutuhan sistem yang akan dikembangkan. (Ardiansyah & Sela, 2023).

Berdasarkan jumlah kanal warnanya, citra digital dapat dibedakan menjadi tiga jenis, yaitu citra biner, citra grayscale, dan citra berwarna (RGB). Citra biner hanya memiliki dua nilai intensitas, yaitu hitam dan putih. Citra grayscale memiliki satu kanal warna dengan rentang intensitas 0–255, sedangkan citra RGB terdiri atas tiga kanal warna, yaitu merah (Red), hijau (Green), dan biru (Blue). Masing-masing jenis citra memiliki karakteristik yang berbeda dan digunakan sesuai dengan kebutuhan proses pengolahan citra. (Dewi & Munir, 2023).

Dalam penelitian ini, citra digital yang digunakan berupa citra ikan berwarna (RGB) yang diperoleh melalui proses pengambilan gambar.

Sebelum dilakukan proses deteksi objek, citra terlebih dahulu melalui tahap preprocessing, yaitu konversi menjadi citra grayscale dan proses thresholding. Tahapan tersebut bertujuan untuk meningkatkan kualitas citra dan mempermudah proses pendeteksian kontur sehingga objek ikan dapat dikenali dengan lebih akurat menggunakan metode Contour Detection. (Wardana, Maskuri, & Zaim, 2023).

2.4 Grayscale

Grayscale atau citra keabuan merupakan citra digital yang hanya memiliki satu kanal intensitas dengan rentang nilai 0 sampai 255. Nilai intensitas 0 menunjukkan warna hitam, sedangkan nilai 255 menunjukkan warna putih. Berbeda dengan citra berwarna (RGB) yang memiliki tiga kanal warna, citra grayscale hanya menyimpan informasi tingkat keabuan sehingga proses pengolahan citra menjadi lebih sederhana dan efisien. Konversi ke grayscale sering digunakan sebagai tahap awal (preprocessing) dalam berbagai penelitian pengolahan citra digital karena mampu mengurangi kompleksitas data yang akan diproses. (Kusnadi et., 2022)

Proses konversi citra RGB menjadi grayscale dilakukan dengan menggabungkan nilai komponen merah (Red), hijau (Green), dan biru (Blue) menjadi satu nilai intensitas. Konversi ini bertujuan untuk mempertahankan informasi penting berupa bentuk, tekstur, dan tingkat kecerahan objek, tanpa dipengaruhi oleh informasi warna. Dengan demikian, proses analisis citra menjadi lebih cepat dan kebutuhan

komputasi menjadi lebih rendah. (Kusnadi et., 2022)

Secara matematis, konversi citra RGB menjadi grayscale dapat dihitung menggunakan persamaan berikut:

$$Gray = (0,299 \times R) + (0,587 \times G) + (0,114 \times B)$$

Keterangan:

Gray = nilai intensitas citra grayscale.

R = nilai komponen warna merah (*Red*).

G = nilai komponen warna hijau (*Green*).

B = nilai komponen warna biru (*Blue*).

Persamaan tersebut merupakan metode yang umum digunakan karena mempertimbangkan sensitivitas mata manusia terhadap masing-masing komponen warna, sehingga menghasilkan representasi citra keabuan yang lebih mendekati kondisi visual sebenarnya. (Kusnadi et., 2022)

Dalam berbagai penelitian pengolahan citra digital, tahap grayscale merupakan bagian dari proses preprocessing sebelum dilakukan segmentasi maupun deteksi objek. Pengubahan citra menjadi grayscale memudahkan proses selanjutnya, seperti thresholding, deteksi tepi, maupun contour detection, karena sistem hanya mengolah satu kanal intensitas. Hal ini dapat meningkatkan efisiensi proses komputasi serta menghasilkan proses segmentasi yang lebih optimal. (Radillah et al., 2023)

Pada penelitian ini, proses grayscale diterapkan setelah citra ikan berhasil diakuisisi. Hasil konversi grayscale kemudian digunakan sebagai masukan pada proses thresholding untuk memisahkan objek ikan dari latar belakang. Selanjutnya, citra biner hasil thresholding digunakan dalam proses Contour Detection untuk memperoleh batas atau kontur objek ikan sehingga sistem mampu mendeteksi objek dengan lebih akurat. (Radillah et al., 2023)

2.5 Thresholding

Thresholding merupakan salah satu teknik segmentasi citra yang digunakan untuk memisahkan objek dari latar belakang berdasarkan nilai ambang (threshold) tertentu. Metode ini bekerja dengan mengubah citra grayscale menjadi citra biner sehingga setiap piksel hanya memiliki dua nilai, yaitu hitam (0) dan putih (255). Proses tersebut bertujuan untuk mempermudah sistem dalam mengenali objek yang akan dianalisis pada tahapan berikutnya. Teknik thresholding banyak digunakan dalam pengolahan citra karena memiliki proses komputasi yang sederhana, cepat, dan efektif dalam memisahkan objek dengan latar belakang yang memiliki perbedaan intensitas yang jelas (Damayanti & Wibawa, 2023).

Pada proses thresholding, setiap piksel pada citra grayscale dibandingkan dengan nilai ambang (threshold) yang telah ditentukan. Apabila nilai intensitas piksel lebih besar atau sama dengan nilai ambang, maka piksel tersebut akan diubah menjadi warna putih yang merepresentasikan objek. Sebaliknya, apabila nilai intensitas piksel lebih

kecil dari nilai ambang, maka piksel akan diubah menjadi warna hitam yang merepresentasikan latar belakang. Hasil proses ini berupa citra biner yang lebih mudah diproses pada tahap segmentasi maupun deteksi objek (Radillah, Ameliza, & Fitriyanto, 2023).

Metode thresholding dapat dibedakan menjadi dua jenis, yaitu global thresholding dan adaptive thresholding. Global thresholding menggunakan satu nilai ambang untuk seluruh citra sehingga cocok digunakan pada citra dengan pencahayaan yang relatif merata. Sementara itu, adaptive thresholding menentukan nilai ambang yang berbeda pada setiap bagian citra sehingga lebih sesuai digunakan pada citra dengan pencahayaan yang tidak seragam atau memiliki variasi intensitas yang tinggi (Pranata & Wibawa, 2023)

Dalam penelitian ini, proses thresholding digunakan sebagai tahapan preprocessing setelah konversi citra menjadi grayscale. Tujuan dari proses ini adalah untuk memisahkan objek ikan dari latar belakang sehingga batas objek menjadi lebih jelas. Hasil citra biner yang diperoleh selanjutnya digunakan sebagai masukan pada proses Contour Detection untuk mendeteksi kontur atau batas objek ikan secara lebih akurat (Damayanti & Wibawa, 2023).

Persamaan Thresholding

Secara matematis, proses thresholding dapat dinyatakan sebagai berikut:

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases}$$

Keterangan:

$g(x,y)$ = citra hasil thresholding.

$f(x,y)$ = nilai intensitas piksel pada citra grayscale.

T = nilai ambang (threshold).

2.6 Contour Detection

Contour Detection merupakan salah satu teknik dalam pengolahan citra digital yang digunakan untuk mendeteksi batas atau tepi suatu objek pada citra. Kontur merupakan sekumpulan titik yang membentuk garis batas suatu objek sehingga dapat digunakan untuk membedakan objek dengan latar belakang. Teknik ini banyak dimanfaatkan dalam berbagai bidang computer vision, seperti deteksi objek, segmentasi citra, pengenalan pola, dan analisis bentuk objek karena mampu memberikan informasi mengenai karakteristik geometris suatu objek (Manuaba & Indah, 2022).

Metode Contour Detection umumnya diterapkan setelah citra melalui tahap preprocessing, seperti konversi citra RGB menjadi grayscale dan proses thresholding. Tahapan tersebut bertujuan untuk mengurangi kompleksitas data serta memisahkan objek dari latar belakang sehingga kontur objek dapat dikenali dengan lebih jelas. Setelah proses segmentasi selesai, sistem akan mencari kumpulan piksel yang saling terhubung dan membentuk batas objek untuk selanjutnya dianalisis (Radillah & Ameliza, 2023).

Dalam implementasinya menggunakan OpenCV, proses pendeteksian kontur dilakukan dengan fungsi `findContours()`. Fungsi ini digunakan untuk mencari seluruh kontur yang terdapat pada citra biner, sedangkan fungsi `drawContours()` digunakan untuk menggambarkan kembali kontur yang telah ditemukan pada citra asli. Kombinasi kedua fungsi tersebut memungkinkan sistem menampilkan hasil deteksi objek secara visual sehingga mempermudah proses identifikasi objek (Manuaba & Indah, 2022).

Metode Contour Detection memiliki beberapa keunggulan, antara lain proses komputasi yang relatif cepat, mudah diimplementasikan menggunakan pustaka OpenCV, serta mampu memberikan informasi mengenai bentuk, luas (area), keliling (perimeter), dan posisi objek. Namun demikian, tingkat keberhasilan metode ini dipengaruhi oleh kualitas citra, kondisi pencahayaan, serta hasil proses segmentasi. Apabila citra masih mengandung noise atau objek tidak terpisah dengan baik dari latar belakang, maka hasil deteksi kontur menjadi kurang optimal (Manuaba & Indah, 2022).

Dalam penelitian ini, metode Contour Detection digunakan sebagai metode utama untuk mendeteksi objek ikan pada citra digital. Setelah citra melalui proses grayscale dan thresholding, sistem akan mencari kontur objek ikan menggunakan fungsi `findContours()` pada OpenCV. Kontur yang berhasil diperoleh kemudian ditampilkan menggunakan fungsi `drawContours()`, sehingga bentuk dan batas objek ikan dapat dikenali

dengan lebih jelas. Dengan penerapan metode ini diharapkan sistem mampu mendeteksi objek ikan secara otomatis dengan tingkat akurasi yang baik (Manuaba & Indah, 2022).

2.6.1. Cara Kerja Contour Detection

Contour Detection bekerja dengan mencari batas atau garis luar dari objek yang terdapat pada citra. Pada penelitian ini, proses pencarian kontur tidak dilakukan langsung terhadap citra RGB, tetapi menggunakan citra yang telah melalui beberapa tahap preprocessing. Tahapan tersebut terdiri dari konversi citra menjadi grayscale, pengurangan noise menggunakan Gaussian Blur, dan thresholding untuk menghasilkan citra biner. Citra biner tersebut kemudian digunakan sebagai masukan dalam proses pencarian kontur.

Setelah citra menjadi biner, fungsi `findContours()` pada OpenCV digunakan untuk mencari kumpulan titik yang saling terhubung dan membentuk batas suatu objek. Kumpulan titik tersebut disebut sebagai kontur. Setiap kontur merepresentasikan batas dari suatu area yang terdapat pada citra biner. Dengan demikian, Contour Detection bekerja berdasarkan bentuk dan batas objek yang terbentuk setelah proses thresholding, bukan berdasarkan pengenalan objek menggunakan pembelajaran mesin.

Kontur yang ditemukan kemudian dapat dianalisis berdasarkan karakteristik geometrisnya, seperti luas, keliling, dan

posisi. Pada penelitian ini, luas kontur digunakan sebagai salah satu kriteria untuk menentukan kontur yang dianggap sebagai objek ikan. Fungsi `contourArea()` digunakan untuk menghitung luas setiap kontur. Kontur yang memenuhi nilai luas minimum dipertahankan, sedangkan kontur yang memiliki luas terlalu kecil diabaikan karena dapat berasal dari noise atau objek kecil lainnya.

Setelah kontur yang memenuhi kriteria diperoleh, sistem menentukan posisi objek menggunakan bounding box. Bounding box digunakan untuk memberikan batas berupa kotak pada area objek yang terdeteksi sehingga posisi objek ikan dapat diketahui secara visual. Dengan demikian, proses Contour Detection pada penelitian ini dapat diringkas menjadi alur: citra preprocessing → citra biner → pencarian kontur → perhitungan luas kontur → seleksi kontur → bounding box → hasil deteksi.

2.7 OpenCV

OpenCV (Open Source Computer Vision Library) merupakan sebuah pustaka (library) bersifat open source yang dikembangkan untuk mendukung pengolahan citra digital (digital image processing) dan visi komputer (computer vision). OpenCV menyediakan berbagai fungsi yang dapat digunakan untuk membaca, memproses, memanipulasi, dan menganalisis citra maupun video digital. Dengan berbagai fitur yang dimilikinya, OpenCV telah menjadi salah satu library yang paling banyak digunakan dalam pengembangan aplikasi berbasis pengolahan citra dan

deteksi objek. (Julham et al., 2023).

OpenCV mendukung berbagai bahasa pemrograman, seperti Python, C++, Java, dan MATLAB. Di antara bahasa tersebut, Python menjadi salah satu yang paling banyak digunakan karena memiliki sintaks yang sederhana, mudah dipelajari, serta didukung oleh berbagai pustaka lain seperti NumPy dan Matplotlib. Kombinasi Python dan OpenCV mempermudah proses pengembangan aplikasi computer vision, mulai dari pengolahan citra sederhana hingga implementasi sistem deteksi objek secara real-time. (Supiyandi et al., 2024).

OpenCV menyediakan berbagai fungsi yang mendukung setiap tahapan pengolahan citra digital, seperti pembacaan citra (image acquisition), konversi ruang warna (color space conversion), penyaringan citra (image filtering), segmentasi, deteksi tepi (edge detection), ekstraksi fitur, hingga deteksi objek. Selain itu, OpenCV juga memiliki performa yang baik dalam memproses citra sehingga banyak dimanfaatkan dalam penelitian maupun pengembangan sistem berbasis kecerdasan buatan dan visi komputer. (Badri, 2023).

Dalam penelitian ini, OpenCV digunakan sebagai *library* utama untuk mengimplementasikan metode *Contour Detection* dalam mendeteksi objek ikan pada citra digital. OpenCV dimanfaatkan untuk mendukung proses pembacaan citra, konversi citra berwarna menjadi *grayscale*, proses *thresholding*, pendeteksian kontur objek, serta visualisasi hasil deteksi. Penggunaan OpenCV dipilih karena menyediakan berbagai fungsi

pengolahan citra yang lengkap, efisien, dan mudah diimplementasikan menggunakan bahasa pemrograman Python (Pahlevi & Setiaji, 2023).

2.8 Python

Python merupakan bahasa pemrograman tingkat tinggi (high-level programming language) yang dirancang dengan sintaks yang sederhana sehingga mudah dipelajari dan digunakan. Python mendukung berbagai paradigma pemrograman, seperti pemrograman prosedural, berorientasi objek (object-oriented programming), dan fungsional. Kemudahan tersebut menjadikan Python sebagai salah satu bahasa pemrograman yang banyak digunakan dalam pengembangan aplikasi, analisis data, kecerdasan buatan (Artificial Intelligence), machine learning, serta pengolahan citra digital. (Setiadi et al., 2025).

Dalam bidang pengolahan citra digital, Python banyak dimanfaatkan karena memiliki berbagai pustaka (library) yang lengkap, seperti OpenCV, NumPy, Matplotlib, dan Pillow. Library tersebut menyediakan berbagai fungsi untuk membaca citra, melakukan manipulasi gambar, meningkatkan kualitas citra, segmentasi, hingga mendeteksi objek. Selain itu, sintaks Python yang sederhana membuat proses implementasi algoritma pengolahan citra menjadi lebih mudah dibandingkan dengan bahasa pemrograman lainnya. (Saluky & Marine, 2023).

Python memiliki kompatibilitas yang sangat baik dengan library OpenCV sehingga sering digunakan dalam pengembangan aplikasi

berbasis computer vision. Kombinasi Python dan OpenCV memungkinkan implementasi berbagai teknik pengolahan citra, seperti konversi grayscale, thresholding, deteksi tepi, deteksi kontur (contour detection), hingga klasifikasi objek. Oleh karena itu, Python menjadi salah satu bahasa pemrograman yang paling banyak digunakan dalam penelitian yang berkaitan dengan pengolahan citra digital. (Rivandi, 2025).

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama untuk membangun sistem deteksi objek ikan pada citra digital menggunakan metode Contour Detection. Seluruh proses, mulai dari pembacaan citra, tahap preprocessing, proses thresholding, pendeteksian kontur menggunakan OpenCV, hingga menampilkan hasil deteksi dilakukan menggunakan Python. Penggunaan Python diharapkan mampu menghasilkan sistem yang efisien, mudah dikembangkan, serta memiliki performa yang baik dalam mendeteksi objek ikan pada citra digital. (Setiadi et al., 2025; Saluky & Marine, 2023).

2.9 Deteksi Objek

Deteksi objek (object detection) merupakan salah satu teknik dalam bidang computer vision dan pengolahan citra digital yang bertujuan untuk mengidentifikasi keberadaan suatu objek sekaligus menentukan lokasi objek tersebut pada suatu citra atau video. Berbeda dengan klasifikasi citra yang hanya mengenali jenis objek, deteksi objek juga memberikan informasi posisi objek sehingga dapat dimanfaatkan dalam berbagai bidang, seperti industri, keamanan, pertanian, kesehatan, dan

perikanan. (Nasution, Harahap, & Syakbani, 2025).

Proses deteksi objek umumnya diawali dengan akuisisi citra, kemudian dilanjutkan dengan tahap preprocessing untuk meningkatkan kualitas citra, seperti resize, konversi ke grayscale, dan segmentasi. Setelah itu, sistem melakukan proses deteksi berdasarkan karakteristik objek, seperti bentuk, warna, tekstur, maupun tepi objek. Tahapan tersebut bertujuan agar sistem mampu membedakan objek dari latar belakang sehingga hasil deteksi menjadi lebih akurat. (Adhim, Prastyo, Akbar, Pardede, & Anggraeny, 2025).

Berbagai metode telah dikembangkan untuk melakukan deteksi objek, mulai dari metode berbasis pengolahan citra klasik hingga metode berbasis deep learning. Metode klasik, seperti deteksi tepi, segmentasi, dan analisis kontur, masih banyak digunakan karena memiliki kompleksitas komputasi yang relatif rendah dan mudah diimplementasikan. Sementara itu, metode berbasis deep learning menawarkan akurasi yang tinggi, tetapi membutuhkan dataset yang besar dan sumber daya komputasi yang lebih tinggi. (Harahap & Muliantara, 2024).

Dalam penelitian ini, proses deteksi objek dilakukan menggunakan metode Contour Detection. Metode ini dipilih karena mampu mendeteksi batas atau kontur objek ikan berdasarkan hasil proses grayscale dan thresholding. Kontur yang diperoleh kemudian digunakan untuk mengidentifikasi bentuk objek ikan sehingga sistem dapat menampilkan hasil deteksi secara otomatis. Penggunaan metode Contour Detection

diharapkan mampu menghasilkan proses deteksi yang sederhana, efisien, dan sesuai untuk citra dengan latar belakang yang relatif tidak kompleks. (Toyib, 2023).

