

**PENGHITUNGAN PANJANG DAN BERAT IKAN
MENGGUNAKAN HARRIS-CORNERS DETECTORS**

Skripsi

**Disusun untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer**



**Oleh:
Prabowo Darmawi
1313619001**

**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA**

2025

LEMBAR PENGESAHAN

Dengan ini saya mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta

Nama : Prabowo Darmawi
No. Registrasi : 1313619001
Program Studi : Ilmu Komputer
Judul : Penghitungan Panjang Dan Lebar Ikan
Menggunakan Harris-Corners Detection

Menyatakan bahwa proposal ini telah siap diajukan untuk seminar pra skripsi.

Menyetujui,

Dosen Pembimbing I

Muhammad Eka Suryana, M.Kom

NIP. 19770615 200312 1 001

Dosen Pembimbing II

Med Irzal, M.Kom

NIP. 19851223 201212 1 002

Mengetahui,

Koordinator Program Studi Ilmu Komputer

Dr. Ria Arafiah, M.Si

NIP. 19751121 200501 2 004

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan proposal skripsi yang berjudul *Penghitungan Panjang dan Berat Ikan dengan Menggunakan Harris-Corner Detection*.

Keberhasilan dalam penyusunan proposal skripsi ini tidak lepas dari bantuan berbagai pihak yang telah tulus dan ikhlas memberikan masukan guna sempurnanya proposal skripsi ini. Dalam kesempatan ini, dengan kerendahan hati penulis mengucapkan banyak terima kasih kepada:

1. Yth. Ibu Dr.Ria Arafiyah, M.Si selaku Koordinator Program Studi Ilmu Komputer.
2. Yth. Bapak Muhammad Eka Suryana, M.Kom selaku Dosen Pembimbing I yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap proposal skripsi ini.
3. Yth. Bapak Med Irzal, M.Kom selaku Dosen Pembimbing II yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap proposal skripsi ini.
4. Kedua orang tua dan adik penulis yang telah mendukung dan memberikan semangat serta doa untuk penulis.
5. Teman-teman Program Studi Ilmu Komputer 2019 yang telah memberikan dukungan dan memiliki andil dalam penulisan proposal skripsi ini.

Penulis menyadari bahwa penyusunan proposal skripsi ini masih jauh dari sempurna karena keterbatasan ilmu dan pengalaman yang dimiliki. Oleh karenanya, kritik dan saran yang bersifat membangun akan penulis terima dengan senang hati.

Akhir kata, penulis berharap tugas akhir ini bisa bermanfaat bagi semua pihak baik itu bagi FMIPA Universitas Negeri Jakarta, teman-teman dari program studi Ilmu Komputer Universitas Negeri Jakarta dan para pembaca sekalian, khususnya penulis sendiri.

Semoga Allah SWT senantiasa membala kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan proposal skripsi ini.

Jakarta, 18 Juli 2025

Prabowo Darmawi

DAFTAR ISI

Lembar Pengesahan	iii
DAFTAR ISI	vi
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
II KAJIAN PUSTAKA	4
2.1 Pengertian Citra Digital	4
2.2 Pengolahan Citra Digital	4
2.3 Citra Grayscale	4
2.4 Multi-Scale Feature Detection	5
2.4.1 Gaussian scale-space	5
2.5 Gradient	7
2.5.1 Operator Sobel	8
2.6 Second moment matrix	9
2.7 Harris-Corner Detection	9
2.8 Euclidean dan Interpolasi	10
III METODE PENELITIAN	12
3.1 Deskripsi Sistem	12
3.2 Tahap Pengambilan Gambar	13
3.3 Pra-Processing Citra	14
3.4 Smoothing Gaussian	15
3.5 Perhitungan Gradien	15
3.6 Menghitung Second Moment Matriks	16
3.7 Harris Respon	17
3.8 Thresholding dan <i>Non-Maximum suppression</i>	17
3.9 Pemilihan Piksel	18
3.9.1 Menghitung Panjang dan Berat	19
3.10 Evaluasi Akurasi	20
IV UJI COBA DAN HASIL UJI COBA	21

DAFTAR PUSTAKA

22

DAFTAR GAMBAR

Gambar 2.1	Representasi gambar digital	5
Gambar 2.2	Grayscale image	6
Gambar 2.3	Uniform gaussian Kernel	7
Gambar 3.1	Alur Penelitian	13
Gambar 3.2	Contoh Gambar dan Jenis Ikan yang Digunakan	14
Gambar 3.3	Contoh hasil gambar yang telah dihilangkan backgroundnya .	14

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Indonesia merupakan negara kepulauan dengan kekayaan hayati yang melimpah, salah satunya adalah ikan yang menjadi sumber pangan utama masyarakat. Budidaya ikan di Indonesia terus berkembang untuk memenuhi kebutuhan konsumsi dan permintaan pasar, baik sebagai bahan pangan maupun ikan hias. Salah satu aspek penting dalam budidaya ikan adalah pengukuran panjang dan berat ikan, yang berperan dalam pemantauan pertumbuhan, penentuan dosis pakan, serta penentuan harga jual.

Namun, proses pengukuran panjang dan berat ikan di lapangan umumnya masih dilakukan secara manual, yaitu dengan menimbang dan mengukur satu per satu menggunakan alat ukur konvensional (Amri 2020). Cara ini tidak efisien, memakan waktu lama, dan dapat mempengaruhi kualitas ikan.

Seiring perkembangan teknologi, mesin digunakan untuk membantu dalam budidaya ikan. Salah satunya dalam proses grading ikan, conveyor belt sering digunakan untuk memindahkan ikan secara otomatis. Namun, pengukuran panjang dan berat ikan masih dilakukan secara manual, yang dapat menyebabkan ketidakakuratan, ketidakefisienan, dan rusaknya ikan dalam proses grading. Oleh karena itu, diperlukan metode yang dapat mengotomatisasi proses pengukuran panjang dan berat ikan secara cepat, akurat, dan minim kontak langsung dengan ikan. Pengolahan citra digital menawarkan solusi otomatis untuk mengukur panjang dan berat ikan secara cepat, akurat dan minim kontak dengan ikan, salah satu metode yang dapat digunakan adalah deteksi titik sudut (*corner detection*). Titik sudut pada tubuh ikan, seperti ujung kepala dan ekor, dapat digunakan sebagai acuan untuk mengukur panjang ikan secara otomatis dari citra (Harris dkk., 2013). Dengan mengetahui panjang ikan, berat ikan juga dapat diestimasi menggunakan rumus atau model regresi yang sesuai (Diansari dkk., 2013).

Metode *Harris-Corner Detection* merupakan salah satu algoritma deteksi sudut yang banyak digunakan karena kestabilannya terhadap rotasi, noise, dan efisiensi komputasi (Harris dkk., 2013). Dengan menerapkan metode ini pada citra ikan, sistem dapat secara otomatis mendeteksi titik-titik sudut penting, sehingga

proses pengukuran panjang dan estimasi berat ikan dapat dilakukan secara digital, cepat, dan minim kontak langsung dengan ikan.

Berbagai penelitian sebelumnya telah mengembangkan metode untuk mendeteksi atau melacak ikan, seperti penggunaan metode GMM dan Kalman Filter untuk pelacakan gerakan (Alim, 2021), serta GrabCut untuk pemisahan objek dari latar belakang (Nugraha, 2022). Namun, metode-metode tersebut umumnya hanya fokus pada deteksi keberadaan ikan atau pelacakan pergerakan, dan belum secara langsung mengekstraksi fitur geometris seperti panjang dan berat ikan.

Di sisi lain, metode seperti SIFT (*Scale Invariant Feature Transform*) (Lowe 2004) memang kuat terhadap perubahan skala dan rotasi, namun proses perhitungannya relatif kompleks dan memerlukan waktu komputasi lebih tinggi. Untuk kasus estimasi bentuk linear seperti panjang dan lebar, pendekatan berbasis deteksi sudut seperti *Harris-Corner Detection* terbukti lebih efisien dan akurat dalam mendeteksi titik-titik sudut penting pada citra (Harris dkk., 2013). *Harris-Corner Detection* memberikan kestabilan terhadap rotasi dan noise lokal serta memiliki struktur komputasi yang lebih ringan dibanding SIFT, sehingga cocok untuk diterapkan dalam sistem real-time atau perangkat keras terbatas.

Oleh karena itu, penelitian ini bertujuan untuk membangun sistem pengukuran panjang dan estimasi berat ikan berbasis citra digital menggunakan metode *Harris-Corner Detection*. Diharapkan sistem ini dapat membantu pembudidaya ikan dalam melakukan monitoring pertumbuhan ikan secara efisien dan akurat, serta mendukung pengambilan keputusan dalam manajemen budidaya.

1.2 Rumusan Masalah

Berdasarkan pemaparan masalah di atas, perumusan masalah dalam penelitian ini adalah **“Bagaimana cara mengukur panjang serta menghitung berat ikan menggunakan metode *Harris-Corner Detection*? ”**

1.3 Batasan Masalah

1. Implementasi algoritma *Harris-Corners Detection* untuk menghitung panjang dan berat ikan.
2. Jenis Ikan yang digunakan adalah ikan lele, ikan mas, dan ikan nila.

3. Dataset telah dihilangkan latar belakangnya dan digantikan dengan warna solid putih.
4. Citra diambil Top down pada jarak tertentu.
5. Bahasa Pemrograman menggunakan Python 3 atau lebih.
6. Panjang, keliling, dan berat sudah diketahui sebagai data awal.
7. Algoritma tidak dijalakan secara real-time.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk implemntasi algoritma *Harris-Corner* untuk menghitung panjang dan berat Ikan.

1.5 Manfaat Penelitian

1. Bagi penulis

Sebagai salah satu pengabdian terhadap masyarakat, dan menambah pengalaman dan pengetahuan dalam pembangunan sebuah sistem komputer untuk aplikasi dunia nyata, dan pengetahuan tentang pendektsian sudut (*corner*) dari *Harris-Corner Detection*.

2. Bagi Program Studi Ilmu Komputer

Penelitian dapat dijadikan sebagai referensi dan menambah wawasan bagi mahasiswa dan sivitas akademika Ilmu Komputer Universitas Negeri Jakarta.

3. Bagi Industri

Sistem pengukuran panjang dan berat ikan berbasis citra digital dapat meningkatkan efisiensi proses monitoring dan seleksi ikan pada sektor perikanan. Teknologi ini membantu industri dalam mengoptimalkan manajemen budidaya, mempercepat proses grading, serta mengurangi risiko kerusakan fisik pada ikan akibat pengukuran manual.

BAB II

KAJIAN PUSTAKA

2.1 Pengertian Citra Digital

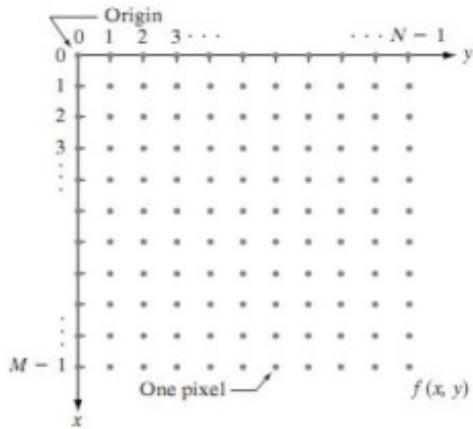
Citra digital adalah representasi visual dari objek atau pemandangan yang diwakili dalam bentuk data numerik. Sebuah citra dapat didefinisikan sebagai fungsi dua dimensi, $f(x, y)$, di mana x dan y adalah koordinat spasial (bidang), dan amplitudo f pada setiap pasangan koordinat (x, y) disebut intensitas atau tingkat keabuan citra pada titik tersebut. Ketika x , y , dan nilai intensitas f semuanya merupakan besaran yang terbatas dan diskret, maka citra tersebut disebut citra digital, seperti pada gambar 2.1. Bidang pengolahan citra digital mengacu pada pemrosesan citra digital dengan menggunakan komputer digital (Gonzalez dkk., 2018, hlm. 1–6).

2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah proses manipulasi citra digital untuk meningkatkan kualitas, mengubah representasi, atau mengekstrak informasi yang berguna. Proses ini melibatkan berbagai teknik dan algoritma yang dapat digunakan untuk mengolah citra, seperti penghalusan, peningkatan kontras, deteksi tepi, dan segmentasi. Pengolahan citra digital memiliki aplikasi luas dalam berbagai bidang, termasuk pengenalan pola, visi komputer, dan analisis citra medis (*ibid.*, hlm. 27–32). Salah satu tahapan penting dalam pengolahan citra adalah konversi citra ke dalam bentuk grayscale sebelum dilakukan ekstraksi fitur atau deteksi tepi (Sánchez dkk., 2018, hlm. 307–308).

2.3 Citra Grayscale

Citra grayscale adalah representasi citra yang hanya menggunakan satu saluran warna, yaitu tingkat keabuan. Citra ini dihasilkan dengan mengubah citra berwarna menjadi skala abu-abu, di mana setiap piksel dalam citra hanya memiliki nilai intensitas yang mewakili tingkat keabuan, seperti pada gambar 2.2. Proses ini sering dilakukan untuk menyederhanakan analisis citra, karena mengurangi kompleksitas data yang harus diproses (Gonzalez dkk., 2018, hlm. 27–32).



Gambar 2.1: Representasi gambar digital

Penggunaan citra grayscale sangat umum dalam proses deteksi fitur multi-skala dan algoritma deteksi sudut seperti Harris-Corner Detection, karena memudahkan perhitungan gradien dan matriks momen kedua (Sánchez dkk., 2018, hlm. 307–308).

2.4 Multi-Scale Feature Detection

Deteksi fitur multi-skala adalah pendekatan yang digunakan untuk mendeteksi fitur pada berbagai skala dalam citra. Pendekatan ini penting karena objek dalam citra dapat muncul pada berbagai ukuran, dan deteksi fitur yang efektif harus mampu mengenali objek tersebut terlepas dari skala yang digunakan. Salah satu metode yang umum digunakan dalam deteksi fitur multi-skala adalah *Gaussian Pyramid*, di mana citra asli dihaluskan dan diperkecil untuk membuat serangkaian citra dengan resolusi yang berbeda. Dengan cara ini, fitur dapat diekstraksi dari setiap tingkat piramida, memungkinkan deteksi yang lebih robust terhadap perubahan skala (Lowe 2004, hlm. 91–92).

2.4.1 Gaussian scale-space

Dalam domain diskrit citra digital, parameter skala juga direpresentasikan secara diskrit. Dengan demikian, representasi ruang-skala merupakan kumpulan gambar yang direpresentasikan oleh perbedaan tingkat resolusi diskrit. Sebuah penelitian menunjukkan bahwa ruang-skala harus memenuhi persamaan yang dapat dicapai melalui konvolusi dengan kernel Gaussian. Selanjutnya, penelitian tersebut memperlihatkan bahwa kernel Gaussian adalah satu-satunya kernel unik yang dapat



(a) Input Image



(b) Grayscale image

Gambar 2.2: Grayscale image

menghasilkan representasi skala-ruang. Keunikan kernel Gaussian ini telah dikonfirmasi melalui berbagai formulasi dalam sejumlah penelitian lain (Lowe 2004, hlm. 92–93; Mikolajczyk 2002, hlm. 13–15). Temuan ini menyimpulkan bahwa konvolusi dengan kernel Gaussian merupakan solusi optimal untuk masalah konstruksi representasi multi-skala. Fungsi Gaussian dua dimensi didefinisikan sebagai berikut:

$$g(\sigma) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{(x^2 + \mu_x) + (y^2 + \mu_y)}{2\sigma_x\sigma_y}\right)$$

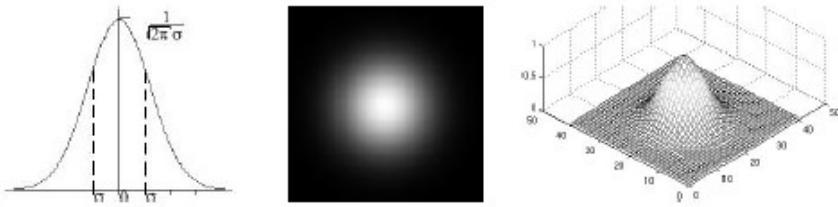
Dimana μ_x dan μ_y adalah *mean* atau rata-rata dari sumbu-x dan sumbu-y, dan σ_x dan σ_y adalah *standard deviation* atau deviasi standar dari sumbu-x dan sumbu-y. Jika $\mu_x = \mu_y = 0$ dan $\sigma_x = \sigma_y$ maka dapat disederhanakan menjadi:

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.1)$$

Keunikan dari kernel Gaussian dapat diketahui dari beberapa sifat berikut: *linearity*, *separability*, *causality*, and *semi group property*. Pembahasan lebih lanjut mengenai properti ini dapat ditemukan dalam literatur *Gaussian Scale-Space Theory* oleh Sporring dkk., 1997. Keterpisahan atau *separability* memungkinkan *multi-dimensional Gaussian kernel* diperoleh sebagai hasil perkalian kernel satu dimensi:

$$g(x, y) = g(x)g(y)$$

Sifat ini sangat berguna dalam praktik karena penghalusan sinyal dua dimensi



Gambar 2.3: Uniform gaussian Kernel

dapat digantikan oleh dua penghalusan satu dimensi, satu untuk setiap dimensi (Lowe 2004, hlm. 92). Filter Gaussian satu dimensi dapat diimplementasikan sebagai filter rekursif, yang secara signifikan mempercepat proses komputasi pada kasus kernel Gaussian yang lebih besar (yaitu $> \sqrt{2}$). Kondisi kausalitas menyatakan bahwa tidak ada struktur tambahan atau buatan yang dibuat saat menghitung citra pada skala kasar, yaitu citra pada skala yang lebih kasar merupakan representasi yang disederhanakan dari citra pada skala yang lebih halus (Mikolajczyk 2002, hlm. 13–14). Sifat semi-grup komutatif menyatakan bahwa n kali penghalusan berturut-turut pada sebuah citra akan memberikan hasil yang sama dengan satu kali penghalusan menggunakan ukuran kernel yang sama dengan jumlah seluruh n kernel tersebut. Selain itu, operasi n tersebut dapat dilakukan dalam urutan apa pun:

$$g(\sigma_1) * \dots * g(\sigma_n) * I(x) = g(\sigma_1 + \dots + \sigma_n) * I(x)$$

Biasanya, ruang-skala yang seragam digunakan, tetapi representasi ruang-skala yang lebih umum dihitung menggunakan filter afine (*ibid.*, hlm. 14–15).

Uniform scale-space. Perbedaan tingkat pada representasi ruang-skala secara umum, terbuat dari konvolusi dengan *gaussian kernel*:

$$G(x, \sigma) = g(\sigma) * I(x) \quad (2.2)$$

Dengan I adalah gambar dan $x = (x, y)$ adalah lokasi poin. *Kernel* adalah simetris sirkular dan diparameterisasi dari satu skala faktor σ .

2.5 Gradient

Gradien dalam matematika dan pemrosesan citra adalah sebuah konsep yang menunjukkan arah perubahan sebuah nilai intensitas pada suatu titik pada gambar 2-Dimensi atau 3-Dimensi. Dalam gambar atau citra digital, Gradien memainkan

peran utama dalam membentuk matriks auto-korelasi, yang nilai eigennya menggambarkan struktur lokal gambar. Wilayah dengan nilai eigen tinggi di semua arah menunjukkan sudut, sedangkan wilayah dengan satu nilai eigen tinggi menggambarkan tepi (Sánchez dkk., 2018, hlm. 305–306).

Berbagai operator gradien, seperti Sobel, Prewitt, dan Scharr, digunakan untuk menghitung gradien dalam gambar dengan mendeteksi perubahan intensitas piksel di sepanjang arah tertentu. Operator Sobel, sebagai salah satu metode yang paling umum, menggunakan kernel konvolusi untuk menghitung gradien dalam arah horizontal (x) dan vertikal (y), menghasilkan peta gradien yang mengidentifikasi tepi dan perubahan penting dalam struktur gambar.

2.5.1 Operator Sobel

Operator Sobel adalah salah satu metode yang paling umum digunakan untuk menghitung gradien dalam citra digital. Metode ini menggunakan dua kernel konvolusi, satu untuk mendeteksi perubahan horizontal dan satu lagi untuk perubahan vertikal. Kernel ini dirancang untuk memberikan bobot lebih pada piksel yang berada di tengah, sehingga lebih sensitif terhadap perubahan intensitas yang tajam.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.3)$$

Dengan menggunakan kernel ini, gradien dalam arah x dan y dapat dihitung dengan mengkonvolusi citra dengan masing-masing kernel. Hasilnya adalah dua peta gradien yang menunjukkan perubahan intensitas dalam arah horizontal (S_x) dan vertikal (S_y). Magnitudo gradien dapat dihitung dengan rumus berikut:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

Di mana G_x dan G_y adalah hasil konvolusi citra dengan kernel Sobel untuk arah x dan y. Peta gradien ini sangat berguna dalam deteksi tepi, segmentasi citra, dan ekstraksi fitur, karena membantu mengidentifikasi area dengan perubahan intensitas yang signifikan (ibid., hlm. 306–307).

2.6 Second moment matrix

Matriks momen kedua (*second moment matrix*) adalah matriks simetris 2×2 yang mencerminkan variasi intensitas di sekitar piksel dalam sebuah citra. Matriks ini seringkali digunakan untuk mendeteksi fitur, atau mendeskripsikan struktur dari *local image*. Matriks ini juga panggil sebagai matriks auto-korelasi, seperti pada rumus 2.5

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = g(\sigma_I) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.5)$$

Hasil dari turunan dihitung pada setiap posisi dan koefisien dari matriks dikonvolusi dengan fungsi gaussian. Langkah ini merupakan langkah paling lama karena menggunakan tiga konvolusi Gaussian (Sánchez dkk., 2018, hlm. 307).

Matriks ini memiliki dua nilai eigen, yang memungkinkan identifikasi suatu area sebagai berikut:

- Jika kedua nilai eigen kecil, maka area tersebut adalah *flat area*.
- Jika satu nilai besar, dan satu kecil, maka area tersebut adalah tepi.
- Jika kedua nilai eigen besar, maka kemungkinan besar area tersebut adalah sudut (*corner*).

Dengan menggunakan sifat tersebut, respon sudut dapat dihitung menggunakan analisis nilai eigen atau metode lain, salah satunya fungsi kekuatan sudut Harris R seperti yang dijelaskan pada rumus 2.6.

2.7 Harris-Corner Detection

Harris-Corner Detection merupakan salah satu dari sekian banyak algoritma untuk mendeteksi fitur (*feature*), algoritma ini pertama kali diperkenalkan oleh Chris Harris dan Mike Stephens pada tahun 1988 (Harris dkk., 2013). Ide dibalik metode Harris adalah untuk mendeteksi titik sudut berdasarkan variasi intensitas di area sekitar: wilayah yang kecil di sekitar fitur menunjukkan perubahan intensitas yang besar dibandingkan dengan pergeseran jendela ke segala arah (Sánchez dkk., 2018, hlm. 305–308). Sudut adalah salah satu fitur di mana terdapat perubahan arah yang tajam pada tepi. Keunggulan penting dari metode Harris adalah sifatnya yang invariant terhadap rotasi, Rotasi adalah proses transformasi geometris yang memutar

posisi piksel citra terhadap titik pusat dengan sudut tertentu tanpa mengubah bentuk asli objek. Dengan kata lain titik sudut tetap dapat terdeteksi dengan akurat meskipun citra mengalami perputaran orientasi. Hal ini menjadikan algoritma ini andal untuk aplikasi seperti pelacakan objek, pengenalan pola, dan pencocokan citra yang memerlukan konsistensi fitur meskipun terjadi perubahan sudut pandang. Dengan melibatkan matriks momen kedua (*second moment matrix*) untuk menghitung variasi perubahan intensitas dalam dua arah utama.

$$R = \det(M) - k * \text{trace}(M)^2 \quad (2.6)$$

Nilai R akan menunjukkan keberadaan sudut pada area tersebut. Jika nilai R yang besar akan menunjukkan keberadaan sudut di area tersebut, sebaliknya nilai R yang kecil menunjukkan daerah tersebut merupakan tepi.

2.8 Euclidean dan Interpolasi

Pada pengolahan citra digital, perhitungan jarak atau panjang antara dua titik sangat penting untuk berbagai aplikasi, seperti pengukuran objek, deteksi fitur, dan transformasi citra. Beberapa contoh metode yang digunakan adalah Euclidean dan atau Manhattan sebagai metode untuk menghitung gambar , dan Interpolasi sebagai metode untuk menghitung berat

Euclidean merupakan salah satu cara untuk mengukur jarak lurus antara dua titik dalam ruang dua dimensi. Rumus ini didasarkan pada teorema Pythagoras, di mana jarak antara dua titik $P_1(x_1, y_1)$ dan $P_2(x_2, y_2)$ dihitung sebagai:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.7)$$

Dengan d adalah jarak Euclidean antara dua titik. Rumus ini juga dapat diperluas ke dimensi yang lebih tinggi dan digunakan secara luas untuk mengukur kedekatan atau kemiripan antara dua piksel atau fitur pada citra (Gonzalez dkk., 2018, hlm. 34).

Manhattan

Interpolasi adalah metode untuk memperkirakan nilai di antara dua atau lebih titik data yang diketahui. Dalam konteks citra digital, interpolasi digunakan untuk menentukan nilai intensitas piksel pada posisi yang tidak tepat berada pada grid piksel, misalnya saat melakukan rotasi, penskalaan, atau transformasi citra. Selain itu, interpolasi juga dapat digunakan untuk menghitung nilai berat (weight)

pada titik-titik tertentu, seperti dalam operasi filter atau penyesuaian nilai piksel hasil transformasi. Salah satu contoh rumus interpolasi linier untuk dua titik adalah:

$$f(x) = f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} [f(x_1) - f(x_0)] \quad (2.8)$$

Untuk interpolasi bilinear pada citra dua dimensi, nilai pada titik (x, y) di antara empat piksel tetangga dapat dihitung dengan:

$$f(x, y) = a_0 + a_1x + a_2y + a_3xy$$

di mana koefisien a_0, a_1, a_2, a_3 ditentukan berdasarkan nilai intensitas keempat piksel tetangga. Metode interpolasi ini sangat penting dalam proses resampling, rotasi, transformasi citra, maupun dalam perhitungan bobot piksel pada berbagai algoritma pengolahan citra (Gonzalez dkk., 2018, hlm. 89–91).

BAB III

METODE PENELITIAN

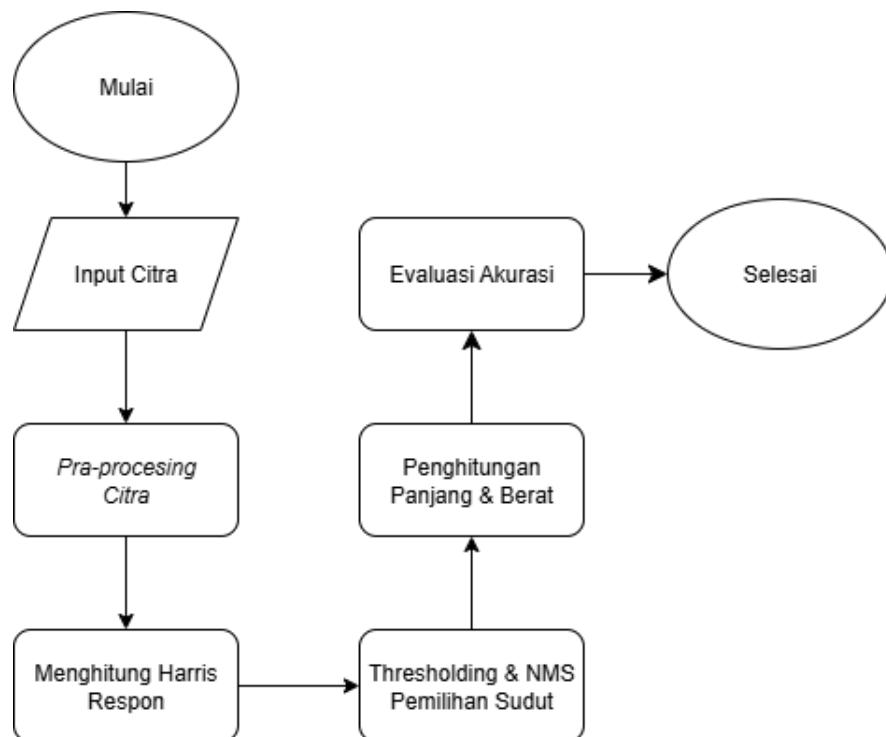
3.1 Deskripsi Sistem

Implementasi algoritma bertujuan untuk melakukan penghitungan panjang serta berat objek ikan yang menggunakan metode *Harris-Corner Detection*. Penelitian berfokus dalam menghasilkan berupa data hasil panjang dan berat objek ikan. Penggunaan conveyor belt dalam penangkaran ikan bertujuan untuk memudahkan dalam pengambilan citra ikan. Citra ikan diambil menggunakan *Smartphone* pada sebuah penangkaran ikan. Setelah itu citra akan melalui tahapan *pra-processing* untuk mengoptimalkan dalam pendekripsi sudut.

Sistem pendekripsi sudut Harris-Corner pada penelitian ini diawali dengan tahapan pemrosesan citra yang terstruktur. Citra ikan yang diambil melalui kamera smartphone terlebih dahulu dikonversi ke format grayscale dan diberi Gaussian smoothing guna mengurangi noise. Setelah itu, sistem melakukan perhitungan gradien menggunakan operator Sobel untuk memperoleh nilai gradien arah horizontal I_x dan vertikal I_y . Nilai gradien ini penting untuk mendekripsi perubahan intensitas piksel pada dua arah utama.

Hasil gradien kemudian digunakan untuk membentuk matriks momen kedua *second moment matrix* atau matriks autokorelasi, yang memuat informasi variasi intensitas di sekitar setiap piksel. Matriks ini merepresentasikan struktur lokal citra; daerah dengan kedua nilai eigen besar menandakan keberadaan sudut. Dengan matriks momen kedua ini, sistem menghitung respon Harris menggunakan persamaan $R = \det(M) - k * \text{trace}(M)^2$, Nilai R menjadi indikator utama untuk menentukan apakah suatu piksel merupakan sudut *corner*, tepi, atau area datar.

Tahapan selanjutnya adalah *thresholding* dan *non – maximum suppression* untuk menyaring piksel-piksel dengan respon Harris tertinggi sehingga hanya titik sudut yang paling signifikan yang dipertahankan. Dari titik-titik sudut inilah sistem menghitung jarak *Euclidean* antara dua titik terjauh sebagai panjang ikan, kemudian mengestimasi berat ikan berdasarkan data referensi panjang-berat menggunakan regresi interpolasi. Seluruh rangkaian proses ini dirancang agar dapat bekerja otomatis dan efisien dalam mengukur panjang serta berat ikan secara digital



Gambar 3.1: Alur Penelitian

Tahapan dalam penghitungan panjang dan berat ikan menggunakan *Harris-Corner Detection* adalah Pemrosesan Citra, Perhitungan gradien, autokorelasi matriks gradien, *Harris Respons*, melakukan threshold-ing dan *Non-Maximum Suppression*, menentukan koordinat sudut, perhitungan panjang dan lebar, Penghitungan berat, dan terakhir adalah mengasosiasikan data.

3.2 Tahap Pengambilan Gambar

Pengambilan gambar dilakukan pada sebuah penangkaran ikan menggunakan kamera *smartphone* dengan resolusi 12 MP. Gambar diambil pada siang hari dengan pencahayaan alami yang cukup untuk memastikan kualitas gambar yang baik. Ikan diletakkan di atas conveyor belt. Setiap gambar diambil dari sudut pandang yang konsisten, yaitu *TopDown* dengan jarak 45 centimeter dari Ikan, untuk memastikan bahwa seluruh bagian ikan terlihat jelas dalam gambar. Gambar-gambar ini kemudian disimpan dalam format JPEG atau PNG untuk digunakan sebagai input dalam sistem penghitungan panjang dan berat ikan.



(a) Ikan Mas

(b) Ikan lele

(c) Ikan Nila Merah

Gambar 3.2: Contoh Gambar dan Jenis Ikan yang Digunakan



(a) Ikan Mas



(b) Ikan lele



(c) Ikan Nila Merah

Gambar 3.3: Contoh hasil gambar yang telah dihilangkan backgroundnya

3.3 Pra-Processing Citra

Pada tahap ini, citra ikan yang akan digunakan sebagai input sistem akan melalui beberapa tahapan pra-processing. Pra-processing bertujuan untuk mempersiapkan citra agar siap untuk diproses lebih lanjut. Langkah-langkah pra-processing yang dilakukan adalah sebagai berikut:

1. Gambar yang akan digunakan berformat jpg atau png, dengan latar belakang yang telah dihilangkan dan digantikan dengan warna solid putih menggunakan library **rembg**. Jenis dataset dapat dilihat pada Gambar 3.2.
 2. Proses input citra ke dalam sistem dilakukan menggunakan library *scikit-image (skimage)* pada bahasa pemrograman Python. *Scikit-image* merupakan library open-source yang digunakan untuk pemrosesan citra digital, menyediakan berbagai fungsi seperti pembacaan, konversi, filtering, segmentasi, dan analisis citra secara efisien. Dengan *skimage.io.imread*, citra dapat dibaca dan diproses lebih lanjut secara efisien.
 3. Setelah citra berhasil diinput, langkah selanjutnya adalah konversi ke *grayscale* untuk mempermudah proses pendekripsi sudut agar lebih akurat.
 4. Setelah citra dikonversi ke *grayscale*, langkah selanjutnya adalah menerapkan *Gaussian smoothing* untuk mengurangi *noise* pada gambar.

3.4 Smoothing Gaussian

Smoothing Gaussian digunakan untuk mengurangi *noise* pada gambar sebelum melanjutkan ke perhitungan lebih lanjut. *Noise* adalah gangguan yang muncul dalam gambar digital yang dapat mengaburkan detail atau informasi penting. *Noise* dapat disebabkan oleh berbagai faktor, seperti kualitas sensor kamera yang rendah, pencahayaan yang buruk, atau kesalahan dalam proses pemindahan data. Dalam gambar, *noise* biasanya terlihat sebagai bintik-bintik acak atau intensitas piksel yang tidak diinginkan. Keberadaan *noise* ini dapat menjadi masalah dalam proses pendekripsi sudut, karena *noise* dapat menyebabkan munculnya sudut palsu yang tidak sesuai dengan gambar asli.

Langkah pertama dalam *smoothing Gaussian* adalah membentuk *Gaussian kernel*. Pemilihan ukuran kernel sangat berpengaruh terhadap hasil *smoothing*, khususnya pada tingkat kehalusan gambar dan jumlah *noise* yang berkurang. Semakin besar kernel yang digunakan, semakin banyak *noise* yang dihilangkan, tetapi hal ini dapat menyebabkan detail gambar ikut terhapus. Sebaliknya, jika kernel yang digunakan berukuran kecil, detail gambar akan lebih terjaga, tetapi *noise* mungkin masih tetap ada. Setelah menentukan ukuran kernel yang akan digunakan kernel dapat dihitung dengan menggunakan persamaan (2.1).

Langkah selanjutnya adalah mengaplikasikan kernel pada gambar. Proses ini dilakukan dengan menerapkan kernel secara berulang pada setiap piksel di seluruh gambar. Namun, sebelum langkah tersebut, gambar perlu di-*padding* terlebih dahulu. *Padding* adalah proses menambahkan piksel pada tepi gambar sebelum pemrosesan dengan kernel atau model dimulai. Tujuan dari *padding* adalah untuk mempertahankan ukuran gambar setelah diterapkannya operasi tertentu, seperti konvolusi. Setelah *padding* diterapkan, konvolusi akan dilakukan dengan menggunakan persamaan (2.2).

3.5 Perhitungan Gradien

Gambar yang telah diperhalus akan dilanjutkan dengan menghitung gradien pada setiap *pixel*-nya. Tujuan dari penghitungan gradien adalah untuk mengetahui perubahan intensitas pada setiap piksel, yang pada umumnya akan menujukkan tepi atau kontur dalam sebuah gambar. Penghitungan gradien juga memerlukan kernel untuk dapat bekerja, terdapat banyak kernel yang dapat digunakan untuk menghitung

gradien, contohnya kernel sobel.

Kernel Sobel digunakan untuk menghitung estimasi gradien dengan memberikan bobot yang lebih besar pada piksel tetangga terdekat di sekitar piksel pusat. Kernel sobel terdiri dari dua matriks 3×3 , yang bertujuan untuk menghitung gradien pada dua arah yaitu:

- **Gradien Horizontal (G_x):** Mengukur perubahan intensitas sepanjang sumbu x.
- **Gradien Vertical (G_y):** Mengukur perubahan intensitas sepanjang sumbu y.

Kernel operator sobel dapat dilihat pada persamaan (2.3). Perhitungan gradien juga akan menggunakan rumus (2.2) tetapi gaussian filter akan diganti dengan kernel sobel horizontal (G_x) dan kernel sobel vertical (G_y). Maka rumusannya akan seperti berikut:

$$\begin{aligned} G_x &= S_x * I(x) \\ G_y &= S_y * I(x) \end{aligned} \quad (3.1)$$

3.6 Menghitung Second Moment Matriks

Setelah hasil dari perhitungan gradien didapatkan, hasil tersebut akan dibuat menjadi matriks kembali dengan rumus (2.5). Nilai gradien dihitung untuk mendapatkan koefisien matriks autokorelasi. Setelah koefisien didapatkan, koefisien tersebut akan dikonvolusi dengan filter gaussian. Hasil dari konvolusi digunakan untuk membentuk matriks M yang akan digunakan untuk perhitungan selanjutnya. Bentuk dari matriks M sebagai berikut:

$$\begin{aligned} \Sigma I_x &= g(\sigma) * G_x^2 \\ \Sigma I_x I_y &= g(\sigma) * G_x * G_y \\ \Sigma I_y &= g(\sigma) * G_y^2 \\ M &= \begin{bmatrix} \Sigma I_x & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y \end{bmatrix} \end{aligned} \quad (3.2)$$

3.7 Harris Respon

Matriks M digunakan untuk menghitung respon Harris, menggunakan rumus Harris yaitu:

$$R(x, y) = \text{Det}(M) - k * \text{trace}(M)^2 \quad (3.3)$$

Dimana $\text{Det}(M)$:

$$\text{Det}(M) = \Sigma I_x * \Sigma I_y - \Sigma I_x I_y^2$$

Dan $\text{trace}(M)$

$$\text{trace}(M) = \Sigma I_x + \Sigma I_y$$

Dimana $k = 0.04$. Penggunaan *Harris-Corner Detection* bertujuan untuk menandai bahwa piksel tersebut adalah sudut atau tidak.

Nilai respon Harris (R) menunjukkan seberapa besar kemungkinan suatu piksel merupakan titik sudut pada citra. Jika nilai R besar dan positif, maka area tersebut memiliki perubahan intensitas yang signifikan ke segala arah dan kemungkinan merupakan sudut (*corner*). Jika nilai R kecil atau negatif, maka area tersebut cenderung merupakan tepi atau bagian datar. Dengan demikian, perhitungan respon Harris sangat penting untuk membedakan antara sudut, tepi, dan area datar pada citra, sehingga hanya titik-titik yang benar-benar merupakan sudut yang akan dipilih untuk proses selanjutnya.

3.8 Thresholding dan *Non-Maximum suppression*

Setelah nilai harris didapatkan tahapan langkah selanjutnya adalah melakukan Thresholding. Thresholding adalah cara untuk memilih hasil deteksi dengan cara menetapkan nilai ambang batas. Jika nilai harris melebihi dari ambang batas, maka piksel tersebut akan dianggap sebagai potensial sudut kuat, Sebaliknya jika nilai harris lebih kecil maka piksel tersebut akan diabaikan. Tujuannya supaya mengurangi piksel yang perlu diproses lebih lanjut.

$$\text{Corner}(x, y) = \begin{cases} 1 & \text{jika } R(x, y) > T \\ 0 & \text{jika } R(x, y) \leq T \end{cases} \quad (3.4)$$

Setelah melewati Thresholding, setiap piksel yang telah lolos akan diperiksa kembali dengan *Non-Maximum Suppression* atau NMS, NMS bertujuan untuk

mengelola kembali piksel yang saling berdekatan satu sama lain. NMS akan memilih piksel pada rentang tertentu, piksel dengan nilai R terbesar akan terpilih sebagai puncak lokal dan dipertahankan sebagai sudut, dan yang lainnya akan diabaikan.

$$NMS(x, y) = \begin{cases} 1 & \text{jika } R(x, y) = MaxLocalR(i, j) \\ 0 & \text{lainnya} \end{cases}$$

Dimana $MaxLocal R(i, j)$ adalah nilai maksimum yang berada di sekitar piksel (x, y) .

3.9 Pemilihan Piksel

Setelah sudut-sudut hasil Thresholding dan NMS diperoleh, langkah selanjutnya adalah memilih piksel-piksel yang akan digunakan sebagai acuan pengukuran panjang dan lebar ikan. Pemilihan titik-titik sudut ini didasarkan pada jarak maksimum antar titik, sehingga diperoleh titik-titik yang mewakili bagian-bagian utama tubuh ikan, seperti ujung kepala (mulut), ujung ekor, sirip atas, dan bagian bawah perut.

Proses pemilihan dilakukan dengan cara mencari pasangan titik sudut yang memiliki jarak Euclidean terjauh satu sama lain. Rumus jarak Euclidean antara dua titik (x_1, y_1) dan (x_2, y_2) adalah:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.5)$$

Langkah-langkah pemilihan titik adalah sebagai berikut:

1. Hitung jarak Euclidean untuk setiap pasangan titik sudut yang terdeteksi.
2. Pilih dua titik dengan jarak maksimum sebagai titik ujung (biasanya mewakili mulut dan ekor ikan).
3. Untuk menentukan titik sirip atas dan bawah, cari dua titik lain yang memiliki jarak maksimum secara vertikal terhadap garis yang menghubungkan dua titik utama (mulut dan ekor).

Dengan pendekatan ini, pemilihan titik menjadi lebih objektif dan akurat karena didasarkan pada distribusi spasial sudut-sudut hasil deteksi, bukan hanya pada posisi relatif secara manual. Jika jenis ikan memiliki morfologi khusus (misal

ikan lele), penyesuaian dapat dilakukan dengan memilih titik-titik yang relevan sesuai bentuk tubuh ikan.

3.9.1 Menghitung Panjang dan Berat

Setelah titik-titik sudut utama dan jarak antar titik diperoleh dari proses sebelumnya, langkah selanjutnya adalah menghitung panjang ikan dengan mengkonversi jarak pixel ke satuan nyata (cm), diperlukan data skala untuk mengkonversi jarak piksel ke satuan nyata. Skala dapat diperoleh dengan dua cara, yaitu menggunakan objek referensi yang diketahui ukurannya dalam gambar, atau berdasarkan resolusi gambar.

$$Skala = \frac{\text{Ukuran nyata (cm)}}{\text{Ukuran pada gambar (piksel)}}$$

$$Panjang Nyata = d \times Skala \quad (3.6)$$

Setelah panjang ikan dalam satuan nyata diketahui, estimasi berat ikan dapat dilakukan menggunakan rumus interpolasi linier berdasarkan data panjang dan berat ikan yang telah diketahui sebelumnya. Rumus interpolasi linier yang digunakan adalah:

$$W_n = W_x + \frac{(L_n - L_x)}{(L_y - L_x)} \times (W_y - W_x) \quad (3.7)$$

di mana:

- W_n = berat ikan yang dicari,
- L_n = panjang ikan hasil pengukuran,
- L_x dan L_y = panjang ikan pada data referensi sebelum dan sesudah,
- W_x dan W_y = berat ikan pada data referensi sebelum dan sesudah.

Dengan metode ini, sistem dapat menghitung panjang ikan secara otomatis dari citra digital dan mengestimasi berat ikan berdasarkan data referensi, sehingga proses pengukuran menjadi lebih efisien, akurat, dan minim kontak langsung dengan ikan.

3.10 Evaluasi Akurasi

Setelah sistem selesai melakukan pengukuran panjang dan berat ikan, langkah terakhir adalah mengevaluasi akurasi hasil pengukuran. Evaluasi dilakukan dengan membandingkan hasil pengukuran sistem dengan data referensi yang telah diketahui sebelumnya. Akurasi dihitung menggunakan rumus:

$$Akurasi = \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + |E_i|} \cdot 100\% \quad (3.8)$$

di mana E_i adalah selisih antara hasil pengukuran sistem dan data referensi, dan n adalah jumlah sampel yang diuji. Hasil evaluasi akan memberikan gambaran tentang seberapa baik sistem dalam mengukur panjang dan berat ikan.

BAB IV

UJI COBA DAN HASIL UJI COBA

DAFTAR PUSTAKA

- Amri, C. F. A. (2020). “Rancang Bangun Fish Counter Untuk Menghitung Bibit Ikan Lele”. In: *Skripsi Program Sarjana Teknik Elektro, Universitas Islam Indonesia*, pp. 1–37.
- Diansari, R. R., A. Endang, and E. Tita (2013). “PENGARUH KEPADATAN YANG BERBEDA TERHADAP KELULUSHIDUPAN DAN PERTUMBUHAN IKAN NILA (*Oreochromis niloticus*) PADA SISTEM RESIRKULASI DENGAN FILTER ZEOLIT”. In: *Jurnal Aquakultur Manajemen dan Teknologi* 2.3, pp. 37–45.
- Gonzalez, R. C. and R. E. Woods (2018). *Digital Image Processing*. 4th ed. Pearson.
- Harris, C. and M. Stephens (2013). “A Combined Corner and Edge Detector”. In: pp. 23.1–23.6.
- Lowe, D. G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2, pp. 91–110.
- Mikolajczyk, K. (2002). “Detection of local features invariant to affines transformations”. Theses. Institut National Polytechnique de Grenoble - INPG.
- Sánchez, J., N. Monzón, and A. Salgado (2018). “An analysis and implementation of the harris corner detector”. In: *Image Processing On Line* 8.March, pp. 305–328.
- Sporring, J. dkk., (1997). *Gaussian Scale-Space Theory*. Ed. by J. Sporring dkk., vol. 8. Springer Netherlands.