

**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**

2024

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
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	6
2.5.1 Operator Sobel	7
2.6 Second moment matrix	8
2.7 Harris-Corner Detection	8
2.8 Euclidean dan Interpolasi	9
III METODE PENELITIAN	11
3.1 Deskripsi Sistem	11
3.2 Perancangan Sistem	11
3.3 Pemrosesan Citra	13
3.3.1 Smoothing Gaussian	13
3.3.2 Perhitungan Gradien	14
3.3.3 Menghitung Second Moment Matriks	15
3.3.4 Harris Respon	15
3.3.5 Thresholding dan <i>Non-Maximum suppression</i>	16
3.3.6 Memilih Piksel	16
3.3.7 Menghitung Panjang dan Berat	17
3.4 Perancangan Eksperimen	18
3.4.1 pra-processing	18
3.4.2 Deteksi Sudut	19
3.4.3 Pengukuran Panjang dan Berat	20
3.4.4 Evaluasi Akurasi	20

DAFTAR PUSTAKA

21

DAFTAR GAMBAR

Gambar 2.1	7
Gambar 3.1 Alur Penelitian	12
Gambar 3.2 Jenis Ikan yang Digunakan	19

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 berpotensi menimbulkan stres pada ikan sehingga dapat mempengaruhi kualitas dan pertumbuhan ikan. Selain itu, metode manual hanya memberikan estimasi jumlah atau berat secara keseluruhan, tanpa informasi detail mengenai distribusi ukuran ikan dalam satu populasi.

Seiring perkembangan teknologi, pengolahan citra digital menawarkan solusi otomatis untuk mengukur panjang dan berat ikan secara cepat dan akurat. Salah satu metode yang dapat digunakan adalah deteksi titik sudut (*corner detection*) pada citra ikan. 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 mendekripsi 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. Sistem hanya menghitung panjang dan berat ikan dengan menggunakan *Harris-Corners Detection*.
2. Jenis Ikan yang digunakan adalah ikan lele, ikan mas, dan ikan nila.
3. Sumber gambar berupa dataset yang diambil langsung dari lapangan.
4. Dataset telah dihilangkan latar belakangnya dan digantikan dengan warna solid hitam.

5. Citra yang digunakan hanya citra tampak samping.
6. Bahasa Pemrograman menggunakan Python 3 atau lebih baru.

1.4 Tujuan Penelitian

Tujuan dari Penelitian adalah Membangun sistem berbasis citra digital untuk mengestimasi panjang dan berat ikan menggunakan deteksi titik sudut dengan menggunakan metode *Harris-corner detection*.

1.5 Manfaat Penelitian

1. Bagi penulis Memperoleh gelar sarjana dalam bidang Ilmu Komputer, dan menambah pengalaman dalam pembangunan sebuah sistem komputer untuk aplikasi dunia nyata, serta pengetahuan tentang pendekslan sudut atau *corner* dari *Harris-corner Detection*.
2. Bagi Program Studi Ilmu Komputer Penelitian "Penghitungan Panjang dan Berat Ikan Menggunakan Harris-Corners Detection" dapat dijadikan sebagai referensi dan menambah wawasan mahasiswa dan sivitas akademika Ilmu Komputer Universitas Negeri Jakarta.

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. 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. Proses ini sering dilakukan untuk menyederhanakan analisis citra, karena mengurangi kompleksitas data yang harus diproses (Gonzalez dkk., 2018, hlm. 27–32). 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 menghasilkan representasi skala-ruang. Keunikan kernel Gaussian ini telah dikonfirmasi melalui berbagai formulasi dalam sejumlah penelitian lain (ibid., 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:

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

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*. 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 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).

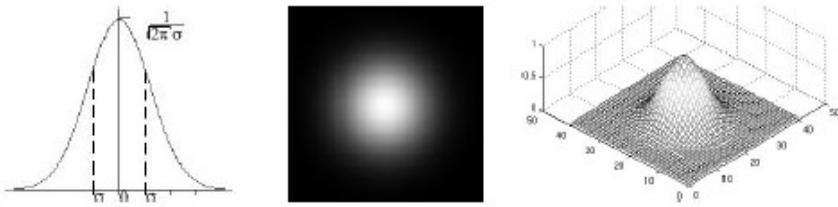
Uniform scale-space. Perbedaan tingkat pada representasi ruang-skala secara umum, terbuat dari konvolusi dengan *gaussian kernel* :

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

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



Gambar 2.1

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 dan vertikal. 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 (Sánchez dkk., 2018, hlm. 306–307).

2.6 Second moment matrix

Matriks momen kedua (*second moment matrix*) adalah matriks simetris 2x2 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 (ibid., 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). 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. Dua metode dasar yang sering digunakan adalah rumus Euclidean dan interpolasi.

Euclidean merupakan cara paling dasar 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).

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

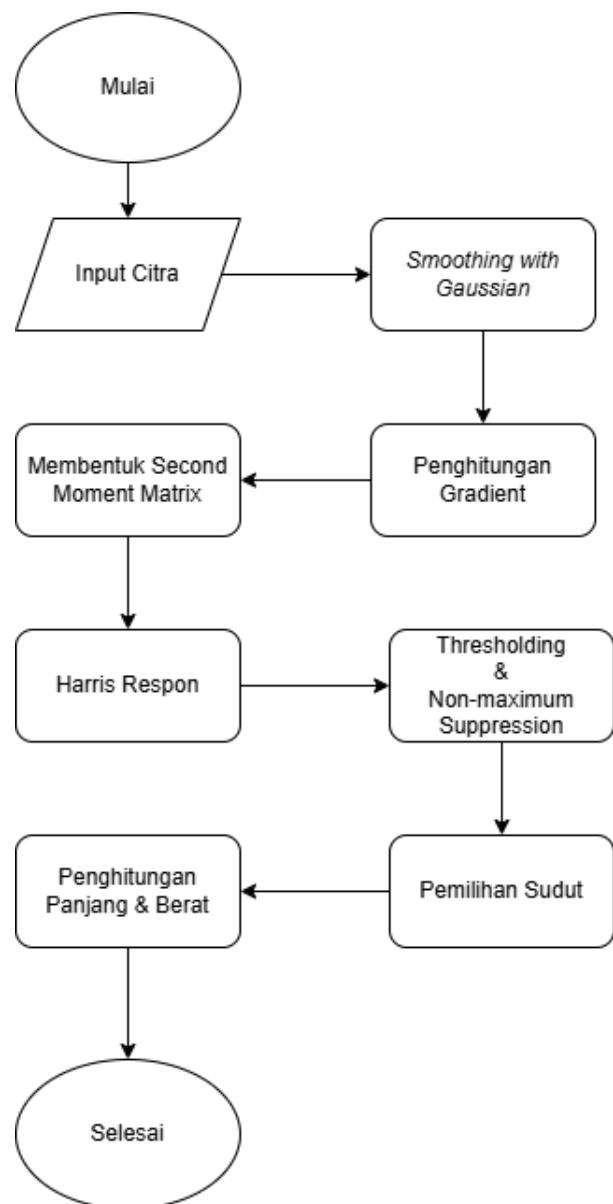
3.1 Deskripsi Sistem

Sistem bertujuan untuk melakukan penghitungan panjang serta berat objek ikan yang menggunakan metode *Harris-Corner Detection* dan *Gaussian Laplace*. Penelitian berfokus dalam menghasilkan berupa data hasil panjang dan berat objek ikan. Dataset citra ikan diambil secara langsung menggunakan *Smartphone* pada sebuah penangkaran ikan. Kemudian bahasa yang akan digunakan dalam membangun sistem adalah Python versi 3.

Tahapan dalam penghitungan panjang dan berat ikan menggunakan *Harris-Corner Detection* adalah Pemrosesan Citra, *Smoothing Gaussian*, 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 Perancangan Sistem

Pada bagian ini akan membahas tentang proses yang akan dilakukan untuk mengetahui tahapan yang ditempuh dalam membangun sistem penghitungan berat dan panjang ikan. Tahapan pertama yang dilakukan adalah melakukan *input* gambar yang akan dibaca sistem, setelah itu citra akan diproses menggunakan *Smoothing Gaussian* untuk mengurangi *noise*. Kemudian perhitungan gradien dilakukan bertujuan untuk menghitung setiap perubahan intensitas yang berada di sekitar titik, hasil tersebut akan diolah kembali menggunakan autokorelasi matriks gradien, dan dilanjutkan dengan menjalankan *Harris Respons*. *Harris Respons* bertujuan untuk mengetahui bahwa titik tersebut adalah sebuah sudut, atau sisi berdasarkan pada hasil respon Harris. Hasil dari Harris akan memasuki tahapan thresholding dan *non-maximum suppression* untuk memilih sudut maksimum pada area tertentu. Setelah mendapatkan titik yang merupakan sebuah sisi, titik-titik tersebut akan dipilih. Hasil dari pemilihan tersebut akan digunakan untuk menghitung panjang, lebar, dan berat ikan. Gambaran perancangan sistem yang sederhana terdapat pada gambar 3.1



Gambar 3.1: Alur Penelitian

3.3 Pemrosesan Citra

Gambar yang akan digunakan berformat .jpg atau .png, dengan latar belakang yang telah dihilangkan dan digantikan dengan warna solid hitam. Jenis dataset dapat dilihat pada Gambar 3.2. 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. Setelah citra berhasil diinput, langkah selanjutnya adalah konversi ke *grayscale* untuk mempermudah proses pendekripsi sudut agar lebih akurat.

3.3.1 Smoothing Gaussian

Metode *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) .

3.3.2 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 menujukan 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.

Berikut adalah kernel yang akan digunakan dalam operator sobel :

1. Kernel Sobel Horizontal

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2. Kernel Sobel Vertical

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

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 :

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

3.3.3 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 :

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

3.3.4 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.

3.3.5 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.

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

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

3.3.6 Memilih 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.

3.3.7 Menghitung Panjang dan Berat

Setelah titik-titik sudut utama diperoleh dari proses pemilihan piksel, langkah selanjutnya adalah menghitung panjang ikan berdasarkan jarak Euclidean antara dua titik terjauh tersebut. Rumus yang digunakan adalah:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.6)$$

di mana (x_1, y_1) dan (x_2, y_2) adalah koordinat dua titik sudut terjauh hasil deteksi. Nilai d ini merupakan panjang ikan dalam satuan piksel.

Untuk mengonversi panjang dari satuan piksel ke satuan nyata, diperlukan faktor skala. Skala dapat diperoleh dengan dua cara, yaitu menggunakan objek referensi yang diketahui ukurannya dalam gambar, atau berdasarkan resolusi gambar:

$$\begin{aligned} Skala &= \frac{\text{Ukuran nyata (cm)}}{\text{Ukuran pada gambar (piksel)}} \\ Panjang Nyata &= d \times Skala \end{aligned} \quad (3.7)$$

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:

$$W_n = W_x + \frac{(L_n - L_x)}{(L_y - L_x)} \times (W_y - W_x) \quad (3.8)$$

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

3.4 Perancangan Eksperimen

Pada bagian ini akan membahas tentang perancangan eksperimen yang dilakukan untuk menguji sistem yang telah dibangun. Eksperimen ini bertujuan untuk mengevaluasi akurasi pengukuran panjang dan berat ikan menggunakan metode *Harris-Corner Detection*. Dataset yang digunakan dalam eksperimen ini adalah citra ikan yang diambil secara langsung dari penangkaran ikan. Citra yang digunakan berjumlah 32 gambar ikan dengan berbagai jenis dan ukuran dapat dilihat pada Gambar 3.2. Setiap gambar akan melalui *pra-processing* untuk menghilangkan latar belakang, mengubah ke format *grayscale*, dan melakukan *smoothing Gaussian*. Setelah itu, sistem akan melakukan deteksi sudut menggunakan metode *Harris-Corner Detection* untuk mendeteksi sudut-sudut penting pada tubuh ikan. Setelah sudut-sudut terdeteksi, sistem akan memilih titik-titik sudut yang relevan untuk pengukuran panjang dan lebar ikan. Panjang ikan akan dihitung berdasarkan jarak Euclidean antara dua titik sudut terjauh yang terdeteksi.

3.4.1 pra-processing

Sebelum melakukan eksperimen, citra ikan yang digunakan akan melalui tahap *pra-processing*. Tahap ini bertujuan untuk mempersiapkan citra agar siap untuk diproses lebih lanjut. Langkah-langkah *pra-processing* yang dilakukan meliputi pengambilan citra ikan menggunakan kamera smartphone, kemudian latar



(a) Ikan Mas

(b) Ikan lele

(c) Ikan Nila Merah

Gambar 3.2: Jenis Ikan yang Digunakan

belakang citra dihilangkan secara manual untuk mendapatkan objek ikan secara lebih jelas. Selanjutnya, citra yang diambil akan diubah ke format *grayscale* untuk mempermudah proses deteksi sudut. Setelah itu, citra grayscale akan diproses menggunakan metode *smoothing Gaussian* untuk mengurangi *noise* yang ada pada citra sehingga hasil deteksi sudut menjadi lebih akurat.

3.4.2 Deteksi Sudut

Setelah tahap pra-processing selesai, sistem akan melakukan deteksi sudut menggunakan metode *Harris-Corner Detection*. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. **Perhitungan Gradien:** Sistem akan menghitung gradien pada setiap piksel citra yang telah dihaluskan menggunakan kernel Sobel.
 2. **Matriks Autokorelasi:** Hasil gradien akan digunakan untuk membentuk matriks autokorelasi M yang berisi informasi tentang perubahan intensitas pada citra.
 3. **Respon Harris:** Sistem akan menghitung nilai respon Harris R untuk setiap piksel menggunakan matriks M .
 4. **thresholding dan NMS:** Nilai respon Harris akan di thresholding untuk menghilangkan piksel-piksel yang tidak signifikan, kemudian dilakukan *Non-Maximum Suppression* (NMS) untuk mengekstrak sudut-sudut yang paling menonjol.
 5. **Pemilihan Piksel:** Sistem akan memilih piksel-piksel sudut yang relevan berdasarkan jarak maksimum antar titik, untuk menentukan titik-titik utama pada tubuh ikan.

3.4.3 Pengukuran Panjang dan Berat

Setelah titik-titik sudut utama diperoleh, sistem akan menghitung panjang ikan berdasarkan jarak Euclidean antara dua titik terjauh yang terdeteksi. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. **Penghitungan Panjang:** Sistem akan menghitung panjang ikan menggunakan rumus jarak Euclidean antara dua titik sudut terjauh.
2. **Penghitungan Skala:** Sistem akan menentukan faktor skala untuk mengonversi panjang dari satuan piksel ke satuan nyata, menggunakan objek referensi atau resolusi gambar.
3. **Estimasi Berat:** Sistem akan mengestimasi berat ikan menggunakan rumus interpolasi linier berdasarkan data panjang dan berat ikan yang telah diketahui sebelumnya.

3.4.4 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:

$$Akurasi = \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + |E_i|} \cdot 100\% \quad (3.9)$$

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.

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.