BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Banyak peneliti yang telah mencoba melakukan penelitian melakukan pemantauan kondisi udang dengan metode tanpa penangkapan secara langsung. Chen dalam penelitiannya mengusulkan pengukuran berat udang menggunakan metode hubungan berat panjang, dan menghubungkan berat bola mata udang dengan berat tubuh udang (Chen dkk., 2019), namun penelitian ini tidak dilakukan pada udang di dalam air. Lin dalam penelitiannya mengusulkan pengukuran panjang tubuh udang menggunakan pembelajaran mendalam, kelemahan dari penelitian ini sulit memisahkan objek citra udang dengan latar belakangnya (Lin dkk., 2019).

Penelitian untuk mengetahui ukuran berat dilakukan juga pada ikan. Pada penelitian ini dilakukan perhitungan ukuran berat ikan tuna menggunakan metode akustik dan pengolahan citra. Model pengambilan data dilakukan menggunakan kamera pada saat ikan tuna berenang dari keramba laut ke keramba penampungan. Untuk perhitungan jumlah ikan tuna dihitung menggunakan sinyal suara, sedangkan untuk ukuran ikan tuna dihitung dari melalui siluet perut ikan menggunakan tahapan visi komputer yaitu segmentasi citra, filter gumpalan, pencocokan, korespondensi stereo, pengukuran panjang mulut. Hasil penelitian menunjukkan bahwa sistem penghitungan tuna otomatis mencapai tingkat kesalahan di bawah 20% (Puig-Pons dkk., 2019).

Penelitian untuk mengetahui estimasi ukuran berat ikan kembali dilakukan. Pada penelitian tersebut dilakukan ekstraksi ukuran tubuh ikan, prediksi bobot ikan menggunakan data ikan sebanyak 1653 ekor ikan nila. Metode pengolahan citra yang dirancang mampu memisahkan tubuh ikan dengan benar dari latar belakang dan siripnya, dan area tubuh ikan yang diekstraksi dapat berhasil digunakan untuk prediksi bobot tubuh ikan, namun dalam pelaksanaan penelitian visi komputer dijalankan khusus pada ikan nila dan diletakan pada latar belakang berwarna hijau homogen (Fernandes dkk., 2020).

Tabel 2.1 Penelitian yang telah dilakukan sebelumnya

Peneliti dan Tahun	Metode	Hasil	Kelemahan
(Chen dkk., 2019).	Menghubungkan berat bola mata dengan ukuran berat udang dengan metode LWR.	Perhitungan ukuran berat dengan nilai toleransi kesalahan di bawah 30%.	Tidak untuk mengolah citra digital dalam air.
(Lin dkk., 2019).	Memperkirakan panjang tubuh udang menggunakan CNN Yolo v3.	Mendeteksi panjang udang mencapai presisi rata-rata 85%.	Sulit untuk memisahkan citra udang dengan latar belakangnya.
(Puig-Pons dkk., 2019).	Menghitung estimasi ukuran berat ikan tuna dengan teknik akustik dan visi komputer.	Penghitungan tuna otomatis dengan tingkat kesalahan di bawah 20%.	Estimasi berat diukur berdasar lebar perut ikan.
(Fernandes dkk., 2020).	Sistem visi komputer sistem untuk ekstraksi ukuran tubuh ikan, prediksi bobot ikan mati.	Sistem visi komputer mampu memisahkan tubuh ikan dari latar belakang dan siripnya.	Sistem digunakan pada ikan nila mati dan latar belakang ikan berwarna hijau homogen.
(Thai dkk., 2021).	Memperkirakan panjang udang dengan metode visi komputer segmentasi U-Net.	Memperkirakan panjang udang ketika mereka ditempatkan secara terpisah.	Kinerja algoritma menurun ketika udang tumpang tindih
(Poonnoy dan Asavasanti, 2021).	Estimasi berat udang tanpa cangkang menggunakan pengembangan algoritma JST.	Secara akurat memperkirakan berat udang dengan postur acak.	Implementasi pada udang mati, tidak pada udang hidup

Penelitian tentang udang yang lain dilakukan oleh Thai, dalam penelitiannya mengusulkan pengukuran panjang udang dengan segmentasi citra dan komputer visi. Penelitian ini belum melakukan pengukuran berat tubuh udang hidup di dalam air (Thai dkk., 2021). Pengenalan pola dan regresi digunakan untuk menghitung estimasi berat udang tanpa kulit dengan postur acak, penelitian ini menggunakan 103 citra udang dengan tujuh postur yang berbeda. Kesamaan citra udang dengan postur udang yaitu, kaki diperpanjang, kaki terpotong, tubuh melingkar, dan punggung ditentukan oleh model jaringan syaraf tiruan. Data ini digunakan sebagai masukan tambahan untuk mengatasi kekhususan postur, regresi

jaringan syaraf tiruan digunakan untuk estimasi berat udang tanpa cangkang (Poonnoy dan Asavasanti, 2021).

Penelitian sebelumnya yang membahas tentang berat udang dan ikan ditampilkan pada Tabel 2.1. Dari penelitian yang telah dilakukan sebelumnya, estimasi berat dilakukan pada udang mati dan ikan mati, dengan media pengambilan data bukan di dalam air, ciri morfometrik yang digunakan juga hanya panjang atau lebar udang, metode yang digunakan juga belum ada yang menggunakan pengembangan metode kalibrasi dan prediksi menggunakan pembelajaran mesin. Dari sini didapatkan gap riset **belum ada metode pengukuran estimasi berat tubuh udang hidup di dalam air menggunakan ciri morfometrik berbasis analisis citra digital dan pembelajaran mesin.**

2.2 Dasar Teori

2.2.1 Udang

Udang merupakan spesies yang berkembang biak di lingkungan perairan. Budidaya udang merupakan salah satu proses budidaya yang menghasilkan protein berkualitas tinggi bagi manusia (Karim dkk., 2021). Udang telah menjadi salah satu makanan laut paling favorit dalam beberapa tahun terakhir. Seiring pertumbuhan produksi udang di seluruh dunia, pengontrolan kualitas udang menjadi tugas penting bagi industri makanan dari hasil laut (Do dan Quoc, 2022). Pemantauan kesehatan udang sangat penting untuk meningkatkan kualitas udang dan efisiensi produksi. Sebelum udang dikirim ke pasar, mereka harus dinilai dan diklasifikasikan menurut standar yang ditetapkan (Ray dkk., 2021).

Perkembangan teknologi telah mempengaruhi perkembangan budidaya perairan di seluruh dunia (Puig-Pons dkk., 2019). Budidaya perairan membutuhkan banyak tenaga kerja dalam jumlah yang besar (Yeh dan Chen, 2019). Budidaya perairan menggunakan teknologi modern berkomitmen untuk peningkatan produktivitas ikan dan udang (Zhao dkk., 2021). Kualitas udang ditentukan oleh tiga faktor, yaitu spesies, pakan dan manajemen (Zihao Liu, 2020). Ada beberapa faktor yang kurang maksimalnya pengelolaan budidaya udang : (1) Air kolam yang keruh menyebabkan visibilitas rendah dan sulit untuk mengamati budidaya, (2)

Kurangnya alat analisis dan kontrol untuk menghadapi kondisi budidaya yang berubah dengan cepat, (3) Sulit untuk memahami secara akurat kondisi pertumbuhan dan kuantitas udang, yang mengakibatkan kurang maksimalnya evaluasi biaya dan produksi, (4) Kesulitan untuk memperkirakan secara akurat jumlah pakan, dan pakan yang berlebihan akan mengakibatkan pemborosan, peningkatan biaya dan pencemaran air, dan akhirnya menyebabkan penyakit atau bahkan kematian udang, (5) Generasi muda enggan untuk terlibat dalam budidaya tradisional, yang mengakibatkan kurangnya pengalaman beternak (Huang dkk., 2019).

Teknik pengolahan citra telah diterapkan pada bidang budidaya udang, sistem pencitraan dalam air dikembangkan untuk mendapatkan citra berkualitas tinggi di kolam budidaya air yang keruh dan kurang cahaya (Valenti dkk., 2021). Sistem visi komputer dapat digunakan untuk memantau pemberian pakan udang, mengevaluasi ukuran udang, kesehatan udang, dan kelebihan bahan organik. Model sistem pemantauan dalam air yang cerdas dikembangkan untuk memantau pemberian makan dan untuk mengelola kualitas air berdasarkan pengamatan dan analisis dalam air (Yang dkk., 2021).

Memperkirakan estimasi ukuran berat udang hidup merupakan salah satu metode yang paling umum dan penting dalam budidaya. Perolehan informasi ukuran berat secara teratur telah diidentifikasi sebagai kebutuhan penting bagi pembudidaya untuk mengoptimalkan pemberian pakan harian (Li dkk., 2020). Mengontrol padat tebar dan menentukan waktu optimal untuk panen. Sampai saat ini, mengetahui estimasi ukuran berat sebagian besar didasarkan pada pengambilan sampel secara langsung, memakan waktu dan melelahkan. Oleh karena itu, sangat dibutuhkan pengembangan metode dengan tanpa penangkapan secara langsung, cepat dan hemat biaya (Thai dkk., 2021).

2.2.2 Berat Udang Hidup

Ukuran berat individu udang hidup didefinisikan sebagai berat organisme udang hidup individu di area tertentu dalam waktu tertentu. Kuantitas ukuran berat disajikan baik secara spasial dan temporal (Houghton dan Hole, 2008). Definisi

ukuran berat juga dijelaskan oleh Rosillo dalam penelitiannya, yang didefinisikan sebagai berat individu hidup di suatu area atau volume tertentu. Organisme individu dipilih melalui prosedur seleksi yang tepat untuk menentukan beratnya (Rosillo-Calle, 2006).

Cara mengetahui estimasi ukuran berat yang paling umum menggunakan berat rata-rata udang di kolam atau keramba diperoleh dengan model sampling periodik (Rodríguez-Sánchez dkk., 2018). Namun pengambilan sampel secara manual dapat menyebabkan kerusakan fisik atau stres pada udang yang mempengaruhi perkembangan pertumbuhannya. Pengambilan sampel manual juga biasanya memakan waktu, tenaga dan kurang akurat (Ihsanario dan Ridwan, 2021). Menggunakan metode tanpa penangkapan secara langsung, cepat dan ekonomis untuk estimasi ukuran berat udang diperlukan untuk memenuhi kebutuhan budidaya perairan (Li dkk., 2020).

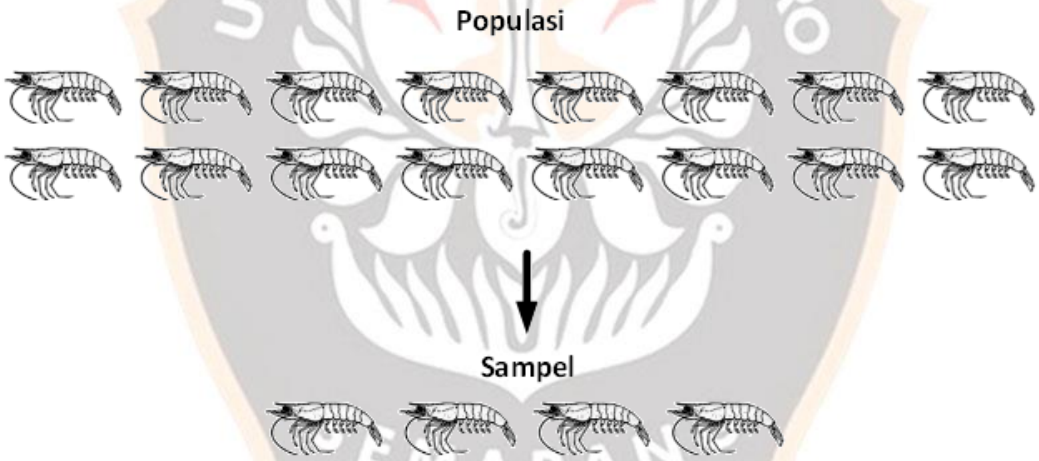
Naufal dalam penelitiannya menjelaskan bahwa pengukuran ukuran makhluk hidup dapat dilakukan tanpa menggunakan metode kontak langsung. Metode yang digunakan dengan pengukuran jarak jauh menggunakan kamera. Dalam penelitian ini dapat mengetahui ukuran makhluk hidup secara otomatis, kamera memancarkan inframerah dan menerima pantulannya untuk mendapatkan citra suatu objek. Citra digital secara otomatis tersegmentasi dan dinormalisasi menjadi citra biner. Perangkat lunak menghitung ukuran objek dari citra tersegmentasi, objek merupakan citra 2 dimensi yang bisa diketahui panjang dan lebarnya (Naufal dkk., 2021).

Dalam sistem budidaya estimasi ukuran berat organisme biasanya dengan mengumpulkan sampel dari area tertentu dan mengukur rangkaian indikator seperti panjang dan berat badan, untuk menentukan kebutuhan jumlah pakan yang dibutuhkan. *Length Weight Relationship* (LWR) merupakan metode dasar untuk mengevaluasi ukuran berat. Menggunakan panjang untuk mengetahui berat badan, sehingga bisa diperkirakan ukuran berat organisme dalam sistem budidaya. Metode LWR spesies laut telah berlangsung sejak tahun 1980, menggunakan model regresi untuk menganalisis LWR dan membandingkan indeks pertumbuhan udang pada kepadatan budidaya kolam yang berbeda. Secara umum, model LWR masih

menjadi dasar utama untuk menghitung ukuran berat di udang pada sistem budidaya (Chen dkk., 2019).

2.2.3 Metode Pengambilan Sampel

Metode pengambilan sampel menggunakan probabilitas sampel, metode ini merupakan pengambilan contoh data secara acak dari suatu populasi. Semua contoh data mempunyai kemungkinan yang sama untuk dipilih. Metode ini mewajibkan sampel data harus berada di dalam populasi, dan peluang pengambilan data tidak boleh nol. Teknik pengambilan data dilakukan secara acak (Li dkk., 2020). Sebelum pengambilan data, dalam probabilitas pemilihan harus diketahui terlebih dahulu elemen pemilihan (Rodríguez-Sánchez dkk., 2018).



Gambar 2.1 Probabilitas sampel

Langkah-langkah yang dilakukan untuk menentukan sampel adalah: (1) menentukan populasi, (2) mencari data akurat unit populasi, (3) memilih sampel yang representatif, (4) menentukan jumlah sampel yang memadai. Pada penelitian ini metode pengambilan sampel yang digunakan adalah pengambilan sampel acak sederhana, setiap anggota populasi dimasukkan ke dalam daftar, dan dipilih secara acak dari daftar tersebut, seperti dijelaskan pada Gambar 2.1.

Metode pengambilan sampel udang dilakukan dengan dua cara, yaitu penangkapan udang secara langsung dan tidak langsung. Penangkapan udang secara langsung dilakukan menggunakan jaring dan ditangkap tangan, sedangkan metode penangkapan secara tidak langsung dilakukan dengan menggunakan alat pemantau udang berupa kamera (Chan dkk., 1998). Metode penangkapan udang secara langsung dilakukan untuk mengetahui ukuran pasti ciri morfometrik udang, hasil ukuran pasti ini kemudian dibandingkan dengan hasil estimasi perhitungan ukuran ciri morfometrik udang dengan hasil penangkapan secara tidak langsung (Li dkk., 2020).

2.2.4 Citra Digital

Citra digital didefinisikan sebagai fungsi dua dimensi *x* dan *y,* koordinat spasial *f(x, y)* disebut intensitas atau tingkat keabuan citra pada titik tertentu. Bidang pemrosesan citra digital mengacu pada pemrosesan citra digital melalui komputer digital.

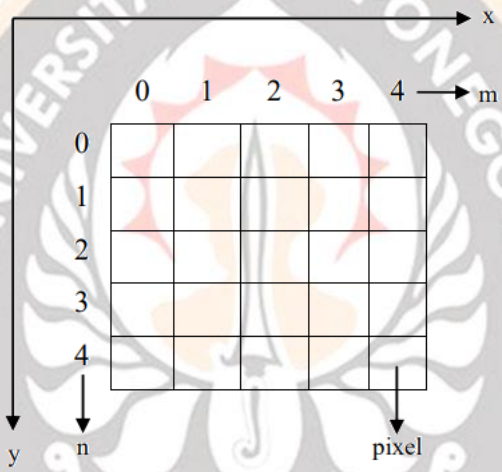
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,m-1) \\ f(1,0) & f(1,1) & \dots & f(1,m-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(n-1) & f(n-1,1) & \dots & f(n-1,m-1) \end{bmatrix}$$

Gambar 2.2 Matriks citra digital *f(x,y)*

Citra digital terdiri dari sejumlah elemen yang terbatas, yang masing-masing memiliki area tertentu (Gonzalez dan Woods, 2008). Dimensi citra dijelaskan dalam bentuk matriks, seperti dijelaskan pada Gambar 2.2. Dengan X_{mxn} adalah matriks citra digital, m adalah banyak baris pada matriks, n adalah banyak kolom pada matriks (Shen dkk., 2021).

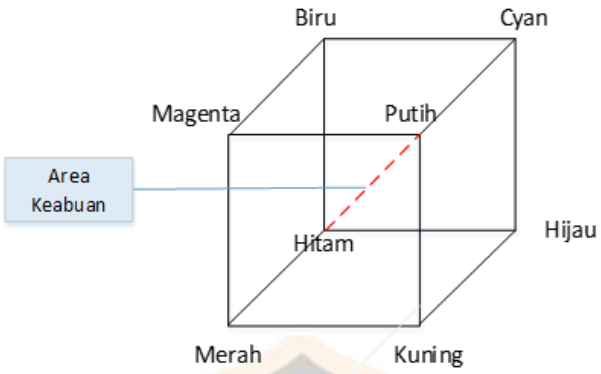
Citra digital merupakan data yang mempunyai dimensi tinggi (Belarbi dkk., 2017). Citra digital merupakan suatu vektor yang didalamnya terdiri dari nilai piksel 2 dimensi, setiap piksel pada citra digital terdiri dari nilai bit *Red Green Blue* (RGB) dengan ruang data berdimensi tinggi (Khaing dkk., 2020). Citra digital

merupakan fungsi dua dimensi *x* dan *y*, yang merupakan koordinat spasial. Citra digital terdiri dari sejumlah elemen yang terbatas, yang masing-masing memiliki area tertentu (Gonzalez dan Woods, 2008). Citra digital merepresentasikan karakteristik kuantitas masing-masing elemen citra yang disebut piksel, citra digital juga merepresentasikan pencahayaan objek yang diambil dari kamera. Citra digital merupakan model dua dimensi yang memberikan informasi deskriptif kuantitatif secara samar (Jain, 1989). Representasi koordinat *x* dan *y* pada citra digital ditampilkan pada Gambar 2.3.



Gambar 2.3 Representasi piksel pada citra digital

Citra digital merepresentasikan karakteristik kuantitas masing-masing elemen citra yang disebut piksel, citra digital juga merepresentasikan pencahayaan objek yang diambil dari kamera, radar pendeteksi dan inframerah. Citra digital merupakan model dua dimensi yang memberikan informasi deskriptif kuantitatif (Jain, 1989). Citra digital dikirimkan secara elektronik, berkas ditransfer ke media penyimpanan fisik, identifikasi objek berdasarkan foto dan video merupakan salah satu layanan yang bisa digunakan (Taoufik dkk., 2021). Sudut sumbu warna pada citra digital ditampilkan pada Gambar 2.4.



Gambar 2.4 Sudut sumbu warna pada citra digital

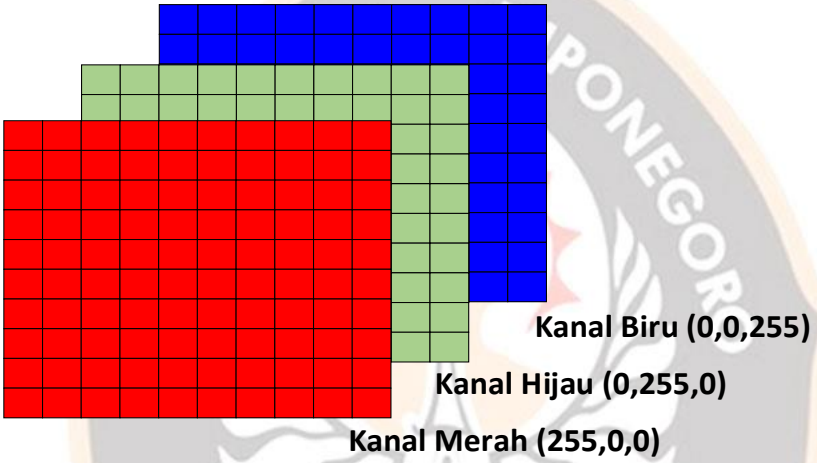
1. Citra Berwarna

Citra berwarna merupakan jenis citra yang mempunyai komponen merah, hijau, biru (Red Green Blue - RGB). Setiap komponen warna mempunyai rentang nilai antara 0 sampai dengan 255 (Note, 2011). RGB merupakan model warna yang digabungkan membentuk kombinasi warna yang luas. Skala rentang 256 didasarkan pada representasi 8-digit bilangan biner, sehingga kombinasi warna yang dihasilkan adalah 256 x 256 x 256 = 1.677.726 jenis warna. Unit terkecil dari data citra digital adalah bit angka biner 0 dan 1, kumpulan data 8-bit merupakan satuan data yang dinamakan byte antara 0 – 255 (Pratt, 2013).

Satuan citra digital menggunakan piksel, dengan resolusi menggunakan satuan *dots per inch* (dpi), piksel merupakan titik elemen paling kecil dari citra digital. Resolusi merupakan karakteristik secara detail dari sebuah citra digital, resolusi didefinisikan sebagai area yang diwakili oleh sekumpulan piksel dari sebuah citra, semakin dekat media ke suatu objek citra, maka resolusi akan semakin besar (Azad dkk., 2017). Nilai RGB untuk 5 warna pada citra digital ditampilkan pada Tabel 2.2. Bidang pemrosesan citra berwarna mengacu pada pemrosesan komputer digital yang mengolah 3 kanal warna (Gonzalez dan Woods, 2008). Kanal citra berwarna yaitu merah hijau dan biru, secara detail seperti dijelaskan pada Gambar 2.5.

Tabel 2.2 Nilai RGB untuk 5 warna pada citra digital

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255



Gambar 2.5 Kanal citra berwarna

2. Citra Keabuan

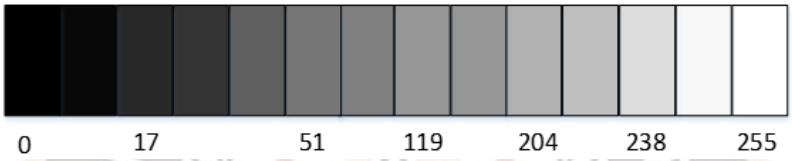
Tahap pertama segmentasi citra adalah mengubah citra berwarna menjadi citra dengan skala keabuan (Bagala dkk., 2020). Citra dengan skala keabuan merupakan citra yang nilai pikselnya mempunyai rentang antara hitam dan putih, sisi hitam merupakan nilai piksel terlemah dan nilai putih merupakan nilai piksel terkuat (He dkk., 2016). Nilai abu-abu sesuai dengan nilai kapasitas kecerahan (Mohan dan Durga, 2016). Citra keabuan merupakan citra dengan tingkat keabuan piksel pada objek citra (Han dkk., 2021). Citra keabuan merupakan hasil proses dari citra RGB, dengan nilai RGB yang dirubah (Song dkk., 2022).

Tingkat keabuan piksel pada objek latar depan sama sekali berbeda dari tingkat keabuan piksel milik latar belakang dalam banyak aplikasi pengolahan citra. Tingkat keabuan ini dapat berfungsi sebagai detektor untuk membedakan antara

latar belakang dan objek latar depan pada citra. Citra keabuan memiliki skala keabuan dengan nilai intensitas paling kecil 0 berwarna hitam hingga warna putih dengan nilai intensitas paling besar 255 (Senthilkumaran dkk, 2016). Formula untuk merubah citra RGB menjadi citra keabuan menggunakan formula 2.1.

$$x = 0{,}299r + 0{,}587g + 0{,}114b \quad (2.1)$$

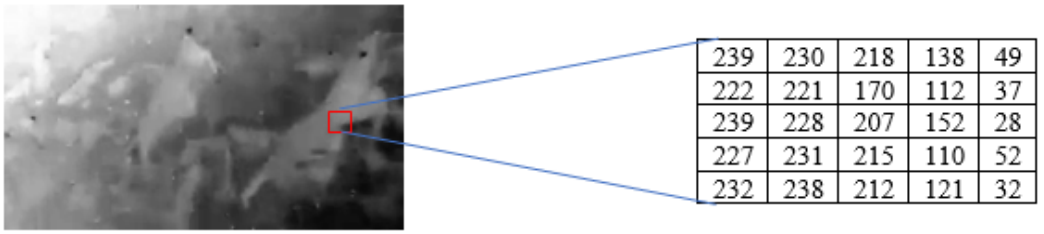
Dengan x = citra keabuan, r = kanal merah, g = kanal hijau, b = kanal biru dari sebuah citra. Representasi citra keabuan dari 0 sampai 255 ditampilkan pada Gambar 2.6, contoh perubahan citra RGB menjadi citra keabuan ditampilkan pada Gambar 2.7 dan contoh nilai piksel pada citra keabuan ditampilkan pada Gambar 2.8.



Gambar 2.6 Representasi nilai citra keabuan

Citra RGB Citra Keabuan

Gambar 2.7 Contoh perubahan citra RGB menjadi citra keabuan



239	230	218	138	49
222	221	170	112	37
239	228	207	152	28
227	231	215	110	52
232	238	212	121	32

Gambar 2.8 Contoh nilai piksel pada citra keabuan

3. Citra Biner

Citra biner merupakan citra dengan nilai piksel 0 dan 1. Teknik batas ambang digunakan untuk proses biner (Du dkk., 2022). Citra biner digunakan untuk memisahkan objek dengan latar belakangnya (Wang dkk., 2021). Mengubah citra skala keabuan ke citra biner menggunakan metode batas ambang (Smith dkk., 1979) (Du dkk., 2022). Metode batas ambang merupakan pemisahan objek antara latar depan dan latar belakang, pengelompokan piksel dibagi menjadi dua yaitu kelompok tingkat keabuan latar depan dan latar belakang (Senthilkumaran dkk, 2016). Metode batas ambang merupakan salah satu metode utama dalam segmentasi citra (Halder dkk., 2016). Formula untuk merubah citra keabuan ke citra biner menggunakan formula 2.2. Perubahan citra keabuan ke citra biner ditampilkan pada gambar 2.9.

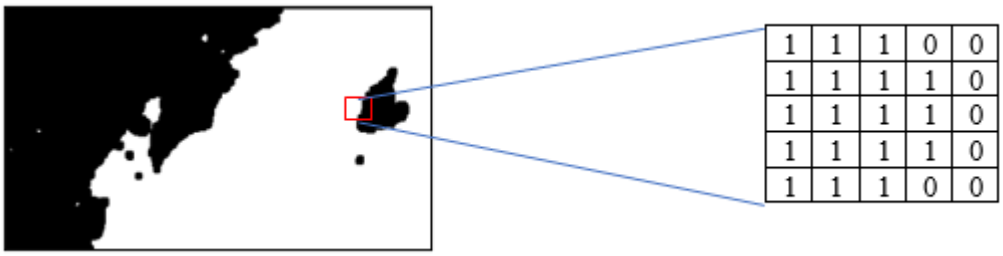
$$g(x,y) = \begin{cases} 1, jika\ f(x,y) \geq T \\ 1, jika\ f(x,y) \geq T \end{cases} \quad (2.2)$$

Dengan $g(x,y) =$ citra biner, $f(x,y) =$ citra keabuan.

Citra Keabuan Citra Biner

Gambar 2.9 Perubahan citra keabuan ke citra biner

Metode Otsu merupakan algoritma yang digunakan untuk memproses citra biner. Metode ini mencari titik pembeda antara batas piksel hitam dan putih. Metode ini memberikan nilai batas ambang kepada piksel dengan warna putih dan hitam. Untuk piksel dengan warna putih, nilai biner = 1 dan untuk piksel berwarna hitam, nilai biner = 0 (Hu dan Wang, 2022). Pada algoritma Otsu, batas ambang bekerja dengan cara menemukan variasi terbesar antar kelas (Halder dkk., 2016). Contoh nilai piksel citra biner ditampilkan pada gambar 2.10.



Gambar 2.10 Nilai piksel pada citra biner

2.2.5 Analisis Citra Digital

Analisis citra digital merupakan metode menganalisis dan memanipulasi citra digital dengan komputer menggunakan operator matematika. Dalam pengolahan citra, masukan berupa citra dan keluaran dapat berupa himpunan karakteristik atau himpunan parameter citra. Sebuah citra terdiri dari berbagai informasi seperti kontur objek, orientasi, ukuran dan warna. Jadi, untuk menemukan informasi bentuk objek, sisi-sisi yang terlibat dalam objek itu harus diidentifikasi. Citra digital dikirim secara elektronik yang ditransfer ke media penyimpanan fisik yang berupa foto dan video (Taoufik dkk., 2021). Sebuah citra terdiri dari berbagai informasi seperti kontur objek, orientasi, ukuran dan warna, dan untuk mendapatkan informasi ini, dibutuhkan proses pengolahan citra (Amer dan Abushaala, 2015).

Analisis citra digital mempunyai bermacam-macam proses, proses yang digunakan pada penelitian ini yaitu: batas ambang, deteksi tepi, deteksi daerah yang diinginkan, deteksi batas (Jain, 1989). Analisis citra digital menghasilkan informasi untuk membantu dalam pengambilan keputusan, analisis citra digital memproses besaran kuantitatif untuk mendapatkan informasi yang ada, dengan mengekstraksi ciri sehingga dapat mengidentifikasi objek pada citra digital.

1. Batas Ambang (*Thresholding*)

Batas ambang merupakan alat yang sederhana namun efektif untuk memisahkan objek latar depan dari latar belakang. Piksel dalam citra dibagi menjadi dua kelompok besar menurut tingkat keabuannya. Teknik batas ambang

merupakan salah satu teknik penting dalam segmentasi citra. Nilai piksel keabuan direpresentasikan pada tingkat keabuan L, dan nilai total piksel dilambangkan dengan N (Halder dkk., 2016). Persamaan yang digunakan yaitu:

$$L = [1,2,3, \dots L] \tag{2.3}$$

$$N = n1 + n2 + n3 + \cdots + nl \tag{2.4}$$

Nilai piksel dibagi menjadi dua kelas, latar belakang C_b dan latar depan C_f, dengan memberi nilai threshold t:

$$Cb = [1,2,3, \dots t] \tag{2.5}$$

$$C_f = [t+1, t+2, t+3, \dots L] \tag{2.6}$$

Formula untuk mengetahui varians latar belakang dan latar depan untuk nilai batas ambang t adalah sebagai berikut:

Latar belakang C_b:

$$\text{Bobot } W_b = \sum_{\mathrm{i=1}}^{\mathrm{t}} \frac{\mathrm{n_i}}{\mathrm{N}} \tag{2.7}$$

$$\text{Rata-rata } \mu_b = \frac{\sum_{i=1}^{t} i * n_i}{\sum_{i=1}^{t} n_i} \tag{2.8}$$

$$\text{Varian } \sigma_b^2 = \frac{\sum_{i=1}^{t} (i-\mu_b)^2 * n_i}{\sum_{i=1}^{t} n_i} \tag{2.9}$$

Latar depan C_f:

$$\text{Bobot } W_f = \sum_{i=1+1}^{L} \frac{n_i}{N} \tag{2.10}$$

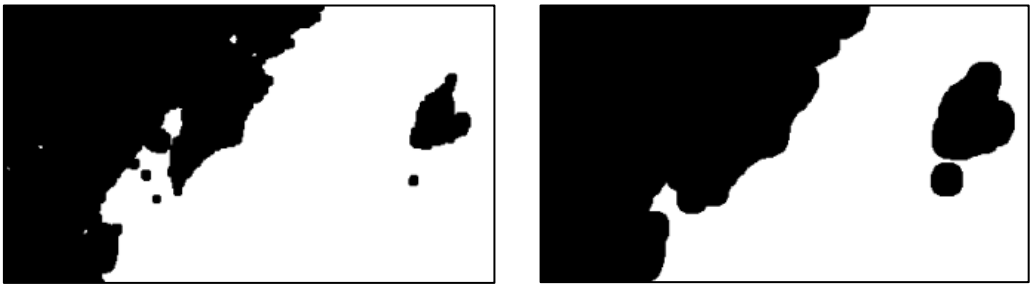
$$\text{Rata-rata } \mu_b = \frac{\sum_{i=1}^{L} i * n_i}{\sum_{i=1}^{L} n_i} \tag{2.11}$$

$$\text{Varian } \sigma_f^2 = \frac{\sum_{i=1+1}^{L} (i-\mu_f)^2 * n_i}{\sum_{i=1+1}^{L} n_i} \tag{2.12}$$

Varian dalam kelas σ_w^2 merupakan penjumlahan dari dua varians dikali dengan bobot. Varian dalam kelas:

$$\sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2 \tag{2.13}$$

Citra biner dengan batas ambang ditampilkan pada Gambar 2.11



Citra biner dengan batas ambang 0,01 Citra biner dengan batas ambang 0,09

Gambar 2.11 Citra biner dengan batas ambang

2. Deteksi Tepi (*Edge Detection*)

Deteksi tepi merupakan metode untuk mendeteksi kemunculan tepi dan lokasinya yang diciptakan oleh variasi intensitas kecerahan warna yang tajam dari suatu citra. Diskontinuitas citra dapat berupa variasi dalam pencahayaan, orientasi permukaan, kedalamannya, dan variasi dalam sifat material. Deteksi tepi bertujuan untuk mendeteksi informasi bentuk objek pada citra. Deteksi tepi adalah langkah penting dalam analisis dan pemrosesan citra, visi komputer, penglihatan manusia, deteksi objek, dan pengenalan pola. Ada berbagai teknik deteksi tepi untuk mendeteksi tepi (Priyadharsini dan Sharmila, 2019).

Deteksi tepi menggunakan detektor tepi bekerja secara berbeda. Beberapa detektor tepi membutuhkan lebih banyak waktu untuk mendeteksi lebih banyak tepi. Deteksi tepi pada suatu citra bergantung pada intensitas, iluminasi, objek, kebisingan dan kekaburan. Salah satu tujuan penting dari pengolahan citra adalah untuk menginterpretasikan isi citra secara efisien dan menemukan informasi yang signifikan. Salah satu langkah penting dalam interpretasi citra adalah menggali informasi tepi dari citra secara tepat. Tepi adalah ciri dasar dari citra dan dapat dibentuk dari garis luar objek. Deteksi tepi umumnya digunakan dalam analisis dan pemrosesan citra. Ada beberapa jenis algoritma untuk mendeteksi tepi (Ansari dkk., 2017).

Saat ini, penelitian dan penerapan teknologi deteksi tepi telah banyak digunakan. Deteksi tepi digunakan pada jaringan saraf konvolusional untuk mengidentifikasi ketebalan objek, selain itu deteksi tepi juga digunakan untuk

pencocokan bentuk, mengekstrak tepi depan dan latar belakang serta menghitung ketebalan menurut piksel regional. Kelebihan metode citra ini adalah akurasi yang tinggi, tidak perlu memodifikasi objek, lebih populer, dan cocok untuk pemantauan setiap saat selama proses (Hu dkk., 2021).

Citra biner Deteksi tepi citra biner

Gambar 2.12 Deteksi tepi pada citra digital

Deteksi tepi ditampilkan pada Gambar 2.12. Deteksi tepi merupakan pendekatan untuk mendeteksi tepi objek citra, yang bertujuan untuk mengukur keberadaan batas area objek pada citra tertentu (Priyadharsini dan Sharmila, 2019). Algoritma Canny merupakan salah satu algoritma yang digunakan untuk deteksi tepi (Xinyi Hu dan Wang, 2022). Canny memproses citra dengan menggunakan filter gaussian (Huang dkk., 2022). Algoritma Canny mendeteksi tepi suatu citra dengan menghilangkan kebisingan di sekitar objek (Yang dkk., 2021).

A. Metode Roberts

Lawrence Roberts telah mengusulkan teknik deteksi tepi Roberts untuk mendeteksi tepi dalam sebuah citra pada tahun 1965. Ini adalah pendekatan yang sederhana dan efisien secara komputasi. Ini mengukur gradien spasial dari sebuah citra. Nilai piksel pada titik tersebut pada citra yang dihasilkan mencirikan estimasi nilai magnitudo absolut dari gradien spasial citra yang dimasukkan pada titik tersebut. Dibutuhkan citra masukan sebagai citra skala keabuan dan menghasilkan tepi yang terlibat dalam citra itu. Kelemahan utama dari teknik ini adalah tidak dapat mendeteksi jenis tepi yang mengalikan 45 derajat dan tidak simetris. Operator Robert berisi sepasang kernel konvolusi 2x2 yang diilustrasikan pada Gambar 2.13.

Salah satu kernel hanya diputar 90 derajat. Turunan parsial dari operator Roberts diberikan sebagai berikut :

$$\frac{\partial f}{\partial x} = f(i,j) - f(i+1,j+1) \quad (2.15)$$

$$\frac{\partial f}{\partial x} = f(i+1,j) - f(i,j+1) \quad (2.16)$$

1	0
0	-1

Gx

0	1
-1	0

Gy

Gambar 2.13 Kernel yang digunakan pada metode Roberts

B. Metode Prewitt

Prewitt telah mengusulkan teknik deteksi tepi Prewitt pada tahun 1970. Ini adalah algoritma yang tepat untuk mengukur orientasi tepi. Teknik ini mengevaluasi arah tepi secara langsung dengan respons maksimum dari kernel. Operator Prewitt ini seperti operator Sobel dan mudah diimplementasikan, sepasang kernel konvolusi 3x3 untuk 8 arah diilustrasikan pada Gambar 2.14, salah satu kernel hanya diputar 90 derajat. Kernel konvolusi 3x3 digunakan untuk mendeteksi gradien dalam arah *X, Y*, filter Prewitt adalah metode cepat untuk deteksi tepi (Amer dan Abushaala, 2015). Turunan parsial dari operator Prewitt diukur sebagai:

$$G_x = (a_2 + \mathrm{ca}_3 + a_4) - (a_0 + 2a_7 + a_6) \quad (2.17)$$

$$G_y = (a_6 + \mathrm{ca}_5 + a_4) - (a_0 + 2a_1 + a_2) \quad (2.18)$$

-1	0	1
-1	0	1
-1	0	1

Gx

1	1	1
0	0	0
-1	-1	-1

Gy

Gambar 2.14 Kernel yang digunakan pada metode Prewitt

C. Metode Sobel

Metode Sobel bergantung pada perbedaan pusat matriks, tetapi rata-rata memberikan bobot lebih pada piksel pusat. Salah satu keunggulan kernel Sobel dibandingkan yang lain adalah memiliki karakteristik peredam kebisingan yang lebih baik. Satu kernel hanya diputar 90 derajat, kernel ini dapat menangani tepi 45 derajat ke sisi piksel. Kernel ini dapat diletakkan secara jelas pada citra masukan untuk memberikan komponen gradien di setiap orientasi. Filter Sobel adalah pendekatan sederhana untuk konsep gradien dengan penghalusan. Kernel konvolusi 3x3 biasanya digunakan untuk mendeteksi gradien dalam arah X dan Y. Operator terdiri dari sepasang kernel konvolusi 3x3 (Amer dan Abushaala, 2015) (Luo dkk., 2021).

Kernel ini dirancang untuk merespon secara maksimal ke tepi yang berjalan secara vertikal dan horizontal, satu kernel untuk dua orientasi tegak lurus. Kernel dapat diterapkan secara terpisah ke citra masukan, untuk menghasilkan pengukuran terpisah dari komponen gradien. Ini kemudian dapat digabungkan bersama untuk menemukan besaran mutlak gradien pada setiap titik. Kernel yang digunakan pada operator Sobel dijelaskan pada Gambar 2.15.

-1	-2	-1
0	0	0
1	2	1

Gx

-1	0	1
-2	0	2
-1	0	1

Gy

Gambar 2.15 Kernel yang digunakan pada metode Sobel

Turunan parsial untuk operator sobel adalah sebagai berikut:

$$G_x = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6) \quad (2.19)$$

$$G_x = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2) \quad (2.20)$$

Besaran Gradien adalah sebagai berikut:

$$|G| = \sqrt[2]{G_x^2} + G_y^2 \quad (2.21)$$

Sudut Orientasi diukur sebagai berikut:

$$\theta = arctan\left(\frac{Gx}{Gy}\right) - \frac{3\pi}{4} \quad (2.22)$$

D. Metode Laplacian of Gaussian (LoG)

Cara kerja metode LoG yaitu dengan menghaluskan citra terlebih dahulu kemudian menghitung deteksi tepi laplacian. Proses ini menghasilkan citra tepi ganda. Ini menempatkan tepi kemudian mencari persimpangan nol antara tepi ganda. Metode deteksi tepi LoG berisi sepasang konvolusi kernel 3x3 yang diilustrasikan pada Gambar 2.16 (Radha, 2011). LoG didasarkan pada turunan orde kedua, menggunakan formula sebagai berikut:

$$V^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad (2.23)$$

Dengan V merupakan turunan orde kedua, dan x,y merupakan sumbu axis

1	1	1
1	-8	1
1	1	1

Gx

-1	2	-1
2	-4	2
-1	2	-1

Gy

Gambar 2.16 Kernel yang digunakan untuk LoG

E. Metode Canny

Canny merupakan salah satu metode deteksi tepi, operator Canny dapat mengontrol tepi citra secara detail dan dapat menekan kebisingan secara efisien. Metode ini mengikuti langkah-langkah berikut: 1) Menghaluskan citra, menggunakan filter dengan nilai yang teridentifikasi dari sigma yang mengurangi kebisingan. 2) Pada setiap titik, arah tepi dan gradien lokal dihitung berdasarkan titik tepi, ini merupakan titik dengan kekuatan maksimum dalam arah gradien. 3) Titik tepi meningkatkan gelombang dalam magnitudo gradien citra. Keluaran dari operator Canny menunjukkan bahwa metode ini telah mendeteksi tepi palsu dan mengabaikan tepi yang benar. Salah satu karakteristik detektor tepi Canny adalah

properti penghubung tepi (Priyadharsini dan Sharmila, 2019). Kernel yang dipakai pada metode Canny di jelaskan pada Gambar 2.17. Formula metode Canny adalah sebagai berikut :

$$G(x,y) = \frac{1}{2x\delta^2} \exp\left(-\frac{x^2+y^2}{2x\delta^2}\right) \quad (2.24)$$

Dengan G merupakan besaran gradient, dan x, y merupakan koordinat sebuah piksel.

-1	0	1
-2	0	2
-1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

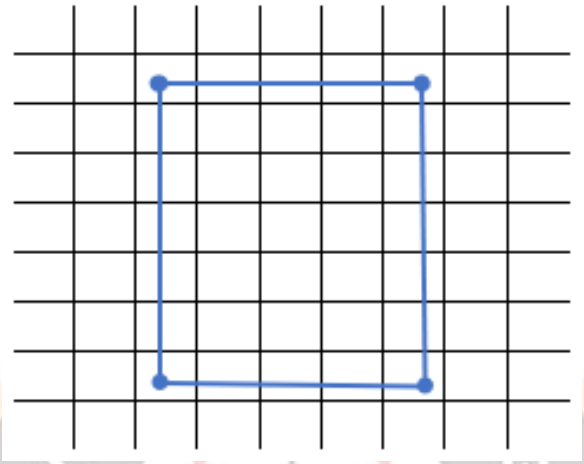
Gambar 2.17 Kernel yang digunakan pada metode Canny

3. Daerah yang Diinginkan (Region of Interest - ROI)

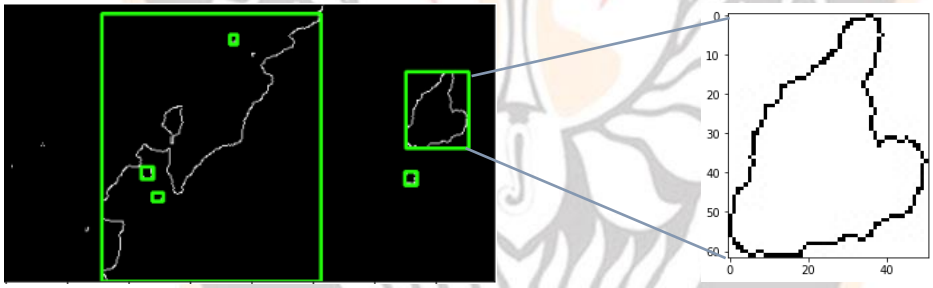
Daerah yang Diinginkan (Region of Interest - ROI) merupakan metode yang digunakan untuk memberikan pembatasan area tertentu pada citra digital (Abdellatif dkk., 2022). Hal ini dilakukan agar proses pengolahan citra hanya terletak pada area yang terdeteksi sebagai udang. ROI juga berfungsi untuk pengurangan derau, sehingga bisa mendapatkan hasil yang lebih akurat. ROI yang digunakan pada penelitian ini adalah persegi untuk memberikan batas minimal objek udang, citra diambil batas minimal sesuai dengan ROI yang terdeteksi. Teknik berbasis daerah bergantung pada pola umum nilai intensitas dalam sekelompok piksel. Klaster disebut sebagai pengelompokan region menurut peran dan fungsinya. (Jan dkk., 2020).

ROI merupakan suatu area yang ditandai dengan garis yang mengelilingi area tersebut. garis ROI terdiri dari empat segmen garis yang menghubungkan empat titik sudut seperti ditunjukkan pada Gambar 2.18. Implementasi ROI pada citra ditunjukkan pada Gambar 2.19. Perhitungan statistik ROI dilakukan melalui segmen garis lurus yang saling terhubung. Formula untuk perhitungan ROI adalah sebagai berikut:

$$area = \left(\pi - \left(n * \sin\left(\frac{\pi}{n}\right) x \cos\left(\left(\frac{\pi}{n}\right)\right)\right)\right) x \frac{100\%}{\pi} \quad (2.25)$$



Gambar 2.18 Segmen garis ROI



Deteksi ROI pada citra digital ROI yang dipilih

Gambar 2.19 ROI pada citra digital

2.2.6 Deteksi Blob

Deteksi blob (*binary large object*) merupakan pendeteksian area dalam suatu citra yang mempunyai sifat kecerahan yang berbeda dengan sekitarnya, deteksi blob merupakan metode pendeteksian properti geometri suatu citra menggunakan teknik gumpalan berbentuk lingkaran, kotak, atau yang lain. Perbedaan antara deteksi blob dan deteksi tepi adalah pada penandaan area pendeteksian, deteksi blob menggunakan gumpalan untuk mengetahui diskontinuitas pencahayaan citra biner, sehingga lebih bisa diukur luasan wilayah dari piksel dalam suatu citra.

Detektor blob didasarkan pada representasi skala area dari citra, untuk sebuah citra digital. Berbagai jenis operator diferensial multi skala menghasilkan berbagai ukuran blob dalam domain ruang skala. Kernel *Gaussian* multi skala merupakan algoritma sering digunakan. Operator *Laplacian of Gaussian* (LoG) yang dinormalisasi menghasilkan metode berbasis kernel Gaussian, dan versi lainnya adalah metode *Difference of Gaussians* (*DoG*) dan *Determinant of Hessian* (*DoH*) (Li dan Shui, 2020). Informasi tambahan blob seperti warna atau tekstur disertakan ke proses yang pendeteksian blob konvensional. Deskriptor digunakan untuk meningkatkan fitur vektor blob yang akan diekstraksi dari citra asli menggunakan batas setiap blob di semua skala. Fitur yang digunakan dalam blob di antaranya warna dan tekstur, vektor fitur citra asli diekstrak sehingga pemburaman tidak memengaruhi vektor fitur. Stabilitas fitur ini ditandai dengan persamaan 2.26. Dengan $f = [f1, f2, f3, \dots, fn], g = [g1, g2, g3, \dots, gn]$ adalah vektor fitur yang sesuai dengan dua kandidat blob dan *dist* adalah Jarak Euclidean (Inthajak dkk., 2010)

$$S = \frac{1}{dist(f,g)} \tag{2.26}$$

LoG diberi masukan citra f(x,y), citra ini diproses dengan kernel *Gaussian*, menggunakan persamaan 2.27.

$$g(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}} \tag{2.27}$$

pada skala t tertentu diberikan representasi ruang skala L(x,y;t) = g(x,y,t) * f(x,y), kemudian diterapkan operator *Laplacian* :

$$\nabla^2 L = L_{xx} + L_{yy} \tag{2.28}$$

Untuk DoG, dari representasi skala L (x,y,t) :

$$\partial_t L = \frac{1}{2}\nabla^2 L \tag{2.29}$$

Jika mengikuti operator LoG $\nabla^2_{norm} L(x, y; t)$ dapat juga dihitung sebagai perbedaan batas penghalusan *Gaussian* dua citra :

$$\nabla^2_{norm} L(x, y; t) \approx \frac{t}{\Delta t}(L(x, y; t + \Delta t) - L(x, y; t)) \tag{2.30}$$

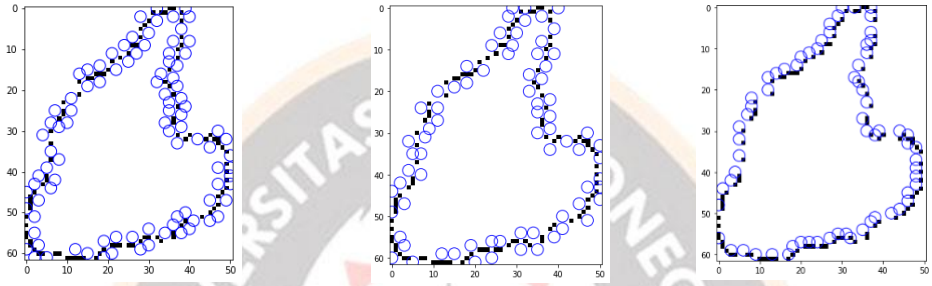
Untuk *DoH* :

$$detH_{norm}L = t^2(L_{xx}L_{yy} - L_{xy}^2) \quad (2.31)$$

Dengan HL adalah matriks *Hessian*

$$(x, y; t = argmaxlocal_{(x,y,t)}((detH_{norm}L)(x, y; t)) \quad (2.32)$$

Hasil detektor blob menggunakan LoG, DoG dan DoH yang diambil dari citra original, dan hanya mendeteksi satu ROI ditampilkan pada Gambar 2.20.



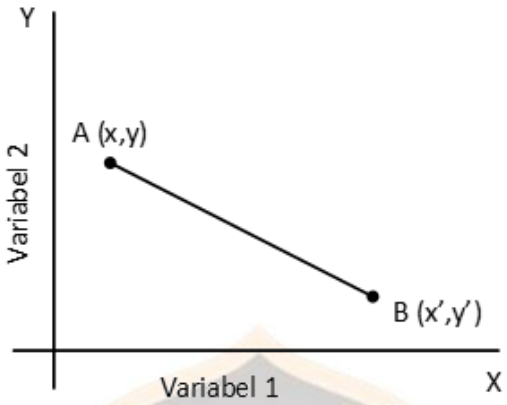
LoG DoG DoH

Gambar 2.20 Detektor Blob dengan tiga metode

2.2.7 Jarak Euclidean (*Euclidean Distance*)

Jarak Euclidean merupakan metode untuk mengukur jarak dari 2 buah titik. Metode ini merupakan pengembangan metode Pitagoras (Bourdeau dkk., 2021). Jarak Euclidean bisa diterapkan pada objek baik 2 dimensi (2D) maupun 3 dimensi (3D). Penerapan Jarak Euclidean pada objek 2D diterapkan pada citra digital yaitu perhitungan jarak antara koordinat awal (x,y) dan koordinat akhir (x',y') (Gandla dkk., 2020) seperti ditunjukkan pada Gambar 2.20. Persamaan Jarak Euclidean dihitung menggunakan persamaan 2.33 (Andries dkk., 2020).

$$d(a, b) = \sqrt{(x' - x)^2 + (y' - y)^2} \quad (2.33)$$

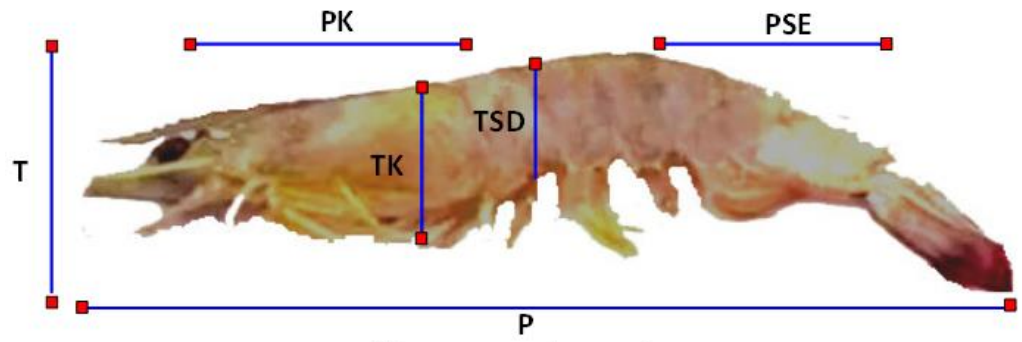


Gambar 2.21 Jarak Euclidean antara 2 titik koordinat

Jarak Euclidean merupakan bagian dari jarak matriks yang digunakan untuk mengukur kesamaan dan kedekatan antara dua titik data. Semakin kecil Jarak Euclidean antara dua matriks citra, maka citra tersebut semakin mirip. Tingkat kemiripan bisa dihitung menggunakan ciri pada citra digital, seperti ciri warna, ciri tekstur, ciri bentuk dan ciri ukuran (Barrett, 2005).

2.2.8 Ciri Morfometrik Udang

Ekstraksi ciri citra udang merupakan pencarian informasi ciri unik yang terdapat pada citra udang, ciri citra bisa diambil dari ciri warna, tekstur, bentuk (Patel, 2016). Ciri morfometrik merupakan karakteristik bentuk spesies laut. Berbagai karakteristik digunakan untuk mengidentifikasi spesies binatang laut seperti morfologi, fisiologis dan perilaku. Karakteristik morfometrik penting untuk mengidentifikasi spesies udang dan kekhasan habitatnya serta kriteria ekologi. Berbagai karakteristik morfologi, fisiologis, perilaku dan biokimia digunakan untuk mengidentifikasi dan mengklasifikasikan udang. Meskipun pengukuran morfometrik digunakan secara langsung, mereka biasanya disajikan sebagai proporsi standar. Bentuk dan struktur organisme laut sangat bervariasi antara ikan, dan udang (Ap, 2016).



Udang tampak samping

Udang tampak atas

Gambar 2.22 Ciri morfometrik udang

Tabel 2.3 Ciri morfometrik udang

Ciri	Symbol
Panjang udang	P
Tinggi udang	T
Lebar udang	L
Panjang karapas udang	PK
Tinggi karapas udang	TK
Panjang segmen keenam udang	PSE
Tinggi segmen kedua udang	TSD

Ciri morfometrik yang digunakan adalah panjang udang (P), tinggi udang (T), lebar udang (L), panjang karapas udang (PK), tinggi karapas udang (TK), panjang segmen keenam udang (PSE), tinggi segmen kedua udang (TSD) (Ap, 2016) (Zhang dkk., 2020). Secara detail ciri morfometrik dijelaskan pada Gambar 2.22 dan Tabel 2.3.

2.2.9 Kalibrasi Nilai Ciri Morfometrik

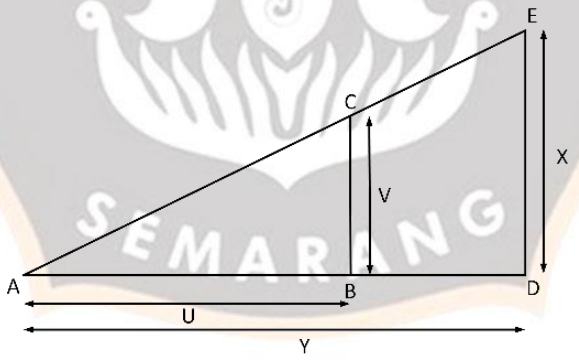
Kalibrasi nilai ciri morfometrik merupakan proses untuk membandingkan nilai ukuran objek sebenarnya, dengan nilai ukuran ciri morfometrik udang yang di observasi, proses ini merupakan proses pengoreksian nilai ciri morfometrik,

agar sebanding dengan ukuran sebenarnya pada semua ciri morfometrik citra digital yang di dapat (Zhang dkk., 2020).

Proses kalibrasi diperlukan untuk memperkecil nilai kesalahan antara nilai citra hasil tangkapan kamera dengan nilai sebenarnya. Pada penelitian ini proses kalibrasi citra menggunakan data dari luar citra, kalibrasi berbasis data empiris dengan penyesuaian data lapangan. Teknik metode geometri diterapkan untuk penyesuaian (Swamidoss dkk., 2021).

1. Kemiripan Segitiga (Triangle Similarity - TS)

Kemiripan segitiga (Triangle Similarity - TS) merupakan 2 buah segitiga yang mempunyai persamaan bentuk, namun berbeda ukuran. Dua segitiga yang mempunyai persamaan bentuk, memiliki besaran sudut yang sama di masing-masing sudut segitiga (Klinaku dan Berisha, 2019). Triangle 2D dalam citra tidak boleh tumpang tindih, segitiga ini merupakan 2 bidang yang berbeda (Hu dkk., 2021). Triangle similarity ditampilkan pada Gambar 2.23 dan persamaan triangle similarity ditampilkan pada persamaan 2.34 dan 2.35 (Yin dkk., 2022).



Gambar 2.23 Kemiripan Segitiga ABC-ADE

$$\angle CAB = \angle EAD \tag{2.34}$$

$$\frac{AB}{AD} = \frac{CB}{ED} \tag{2.35}$$

Panjang fokal (*focal length*) merupakan jarak lensa dan objek citra pada kamera. Panjang fokal diukur dalam satuan centimeter (cm). Panjang fokal

merupakan parameter lensa pada kamera yang memperlihatkan objek citra (Mazur dkk., 2022). Panjang fokal merupakan jarak pusat optik dan sensor kamera, yang digunakan untuk mengatur kefokusan citra yang ditangkap kamera (Cheng dkk., 2022). Panjang fokal mengatur kefokusan dengan mengatur jarak pusat optik sehingga cahaya yang masuk bisa mempertajam citra di kamera (Wang dkk., 2022). Persamaan untuk menghitung panjang fokal ditampilkan pada persamaan 2.36. Dengan P adalah panjang objek yang ditangkap kamera dalam satuan piksel, D panjang asli objek, W adalah jarak asli objek ke kamera, P' adalah panjang objek baru yang ditangkap kamera, D' adalah estimasi jarak antara objek dengan kamera.

$$F = (P * D)/W \tag{2.36}$$

$$D' = (W * F)/P' \tag{2.37}$$

2. Faktor Koreksi (Correction Factor - CF)

Faktor koreksi (Correction Factor - CF) merupakan metode faktor koreksi perhitungan ciri yang menggunakan rasio antara citra dari ukuran komputer dengan ukuran sebenarnya (Zhang dkk., 2020) (Fan dkk., 2022). Faktor koreksi yang digunakan adalah nilai rasio dari ukuran panjang udang asli yang diukur menggunakan alat ukur dengan ukuran panjang udang yang tertangkap kamera dengan satuan centimeter / piksel (Schrader dkk., 2021). Dari nilai faktor koreksi yang didapat, digunakan untuk menghitung semua nilai ciri citra udang (V_f). Untuk perhitungan faktor koreksi dan nilai ciri citra udang menggunakan formula 2.38 dan 2.39 (Karimi dkk., 2018).

$$CF = \frac{p_t}{p_m} \tag{2.38}$$

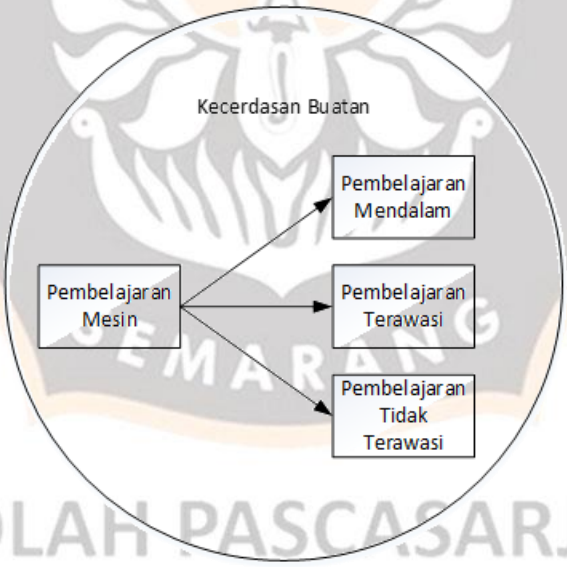
$$V_f = V_o \; x \; CF \tag{2.39}$$

Dengan *CF* adalah faktor koreksi, p_t adalah nilai panjang udang asli yang diukur menggunakan alat ukur dengan satuan centimeter (cm), p_m adalah nilai panjang udang dengan satuan piksel. V_f adalah nilai final ciri citra udang, V_o adalah nilai ciri citra udang dengan satuan piksel.

2.2.10 Pembelajaran Mesin (*Machine Learning*)

Pembelajaran mesin merupakan bidang ilmu dari kecerdasan buatan, pembelajaran mesin mempunyai 3 metode dalam mengolah data, yaitu pembelajaran terawasi (*supervised learning*), pembelajaran tidak terawasi (*unsupervised learning*), dan pembelajaran mendalam (*deep learning*), seperti dijelaskan pada Gambar 2.24 (Zhao dkk., 2021).

Pembelajaran terawasi merupakan metode pembelajaran mesin yang mengenali pola masukan dan label keluaran, pembelajaran mesin dilatih untuk mengenali data masukan yang mempunyai keterhubungan dengan keluaran, dimana keluaran ini sudah ditentukan sebelumnya. Fungsi dari pembelajaran terawasi yaitu untuk memproduksi keluaran dari pengalaman yang sudah dilakukan (Madi dkk., 2019). Metode ini menggunakan dua jenis variabel yaitu variabel masukan dan variabel keluaran, yang termasuk dalam metode ini adalah klasifikasi dan regresi (Swathi dan Kodukula, 2022).



Gambar 2.24 Tiga metode mengolah data dalam pembelajaran mesin

Pembelajaran mesin (*machine learning*) digunakan untuk melakukan proses prediksi dan klasifikasi. Pembelajaran mesin yang digunakan yaitu *Multiple Linear Regression* (MLR), *Support Vector Machine* (SVM), *Random Forest* (RF),

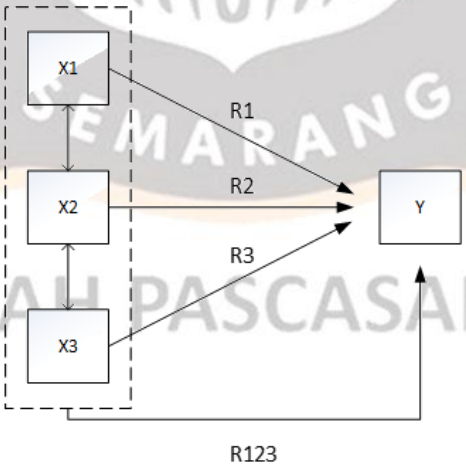
Decision Tree (DT), *K-Nearest Neighbor* (KNN), *Back Propagation Neural Network* (BPNN), dan *Principal Component Regression* (PCR) (Islam dkk., 2021).

1. *Multiple Linear Regression* (MLR)

Multiple Linear Regression (MLR) adalah algoritma statistik yang bertujuan untuk menentukan hasil prediksi dari variabel Y. Y adalah hasil prediksi dari setiap masukan X (Yoshikazu dkk., 2022) (Kamboj dkk., 2022). Rumus umum algoritma MLR dinyatakan dalam formula 2.40. Dengan Y adalah variabel dependen, $x\_1, x\_2 \ldots, x\_n$ adalah variabel independen, $b\_1, b\_2, \ldots, b\_n$ adalah koefisien regresi, a adalah konstanta, dan e adalah nilai kesalahan (Reyhaneh dkk., 2021).

$$Y = a + b_1 x_1 + b_2 x_2 + \cdots b_n x_n + e \tag{2.40}$$

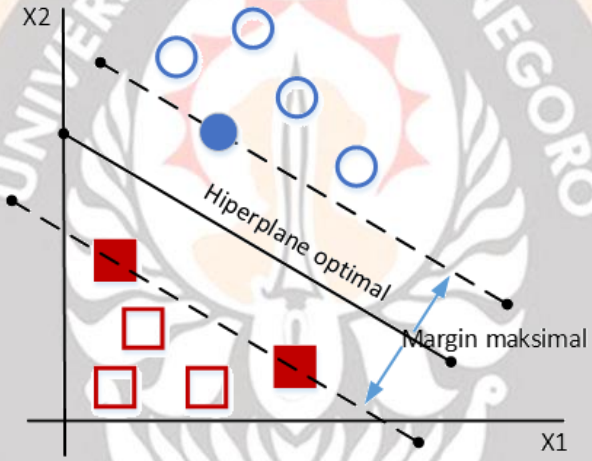
Analisis *Multiple Linear Regression* digunakan untuk mengetahui keterhubungan antara lebih dari satu variabel bebas dan variabel terikat, seperti dijelaskan pada Gambar 2.25, dimana X1, X2, X3 merupakan variabel bebas, Y merupakan variabel terikat, R1 merupakan koefisien regresi X1 terhadap Y, R2 merupakan koefisien regresi X2 terhadap Y, R3 merupakan koefisien regresi X3 terhadap Y, R123 merupakan koefisien regresi X1, X2, X3 terhadap Y.



Gambar 2.25 Skema *Multiple Linear Regression*

2. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah metode pembelajaran terawasi yang dapat digunakan untuk regresi dan klasifikasi. SVM memiliki beberapa kernel, antara lain kernel linier, kernel polinomial, dan kernel fungsi basis radial (Radial Basis Function - RBF) (Mou dkk., 2022). SVM merupakan metode yang digunakan untuk mencari pemisah jarak antar kelas. SVM telah didemonstrasikan dengan hasil yang lebih baik untuk mengolah data berdimensi tinggi dengan nilai numerik. Penggunaan parameter yang tepat dapat meningkatkan kinerja SVM (Mustaqeem, 2021).



Gambar 2.26 Skema *Support Vector Machine*

SVM mencari fungsi pemisah (*hyperplane*) untuk memisahkan dua atau lebih kelas dari data yang dijadikan masukan, hyperplane merupakan garis pemisah pada data dua dimensi, maupun bidang datar pada data tiga dimensi. SVM untuk regresi bertujuan untuk memprediksi keluaran dari data baru, dengan nilai keluaran bernilai kontinyu, sedangkan SVM untuk klasifikasi bertujuan untuk mengklasifikasi data masukan dengan hasil keluaran merupakan kelas, skema SVM dijelaskan pada Gambar 2.26. SVM menggunakan beberapa kernel untuk mentransformasikan data, kernel yang digunakan adalah:

Kernel linier

$$K(x_i, x) = x_i^T x \tag{2.41}$$

Dengan x_i merupakan data latih (*training*), dan x adalah data uji.

Kernel polynomial

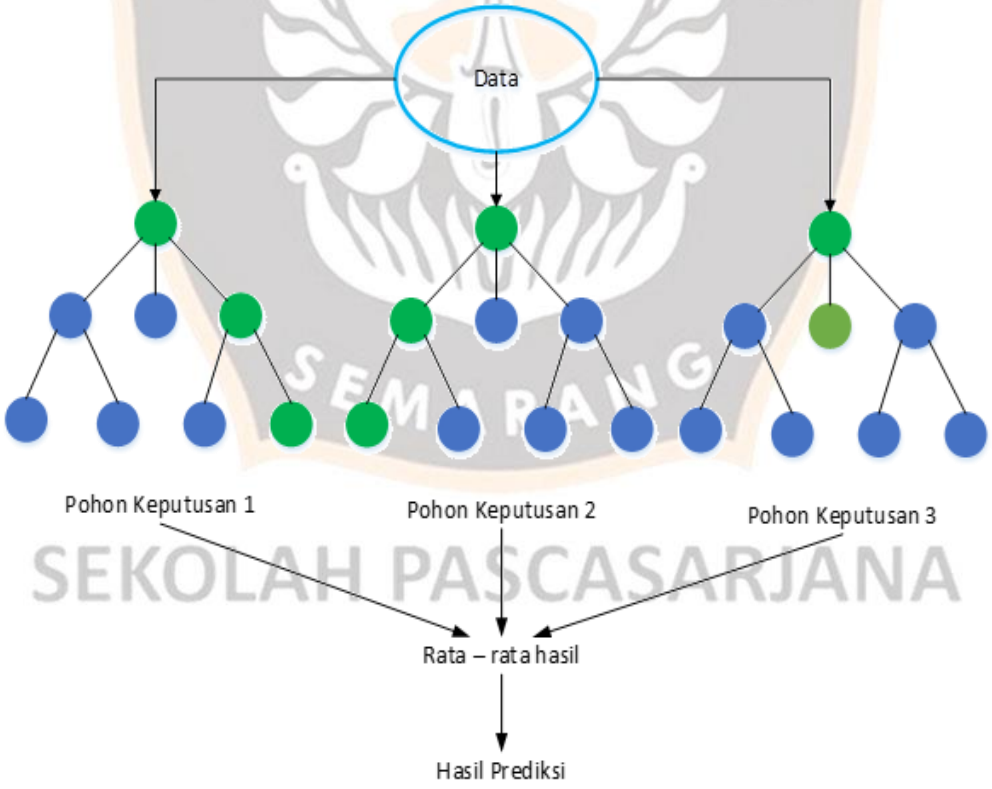
$$K(x_i, x) = (y(x_i^T x) + r)^p \quad (2.42)$$

Kernel sigmoid

$$K(x_i, x) = tanh(y(x_i^T x) + r) \quad (2.43)$$

3. *Random Forest* (RF)

Random Forest (RF) adalah algoritma pembelajaran mesin yang merupakan kumpulan pohon keputusan. RF dapat digunakan untuk regresi dan klasifikasi pada data besar. RF bekerja dengan membangun pohon keputusan untuk mendapatkan hasil regresi (Gao dan Zhou, 2020). RF menggabungkan banyak pohon dalam memproses pelatihan data, jumlah pohon yang digunakan mempengaruhi tingkat akurasi yang dihasilkan, semakin banyak pohon yang digunakan, maka akurasi akan semakin tinggi (Yoshikazu dkk., 2022), seperti dijelaskan pada gambar 2.27.



Gambar 2.27 Skema *Random Forest*

RF merupakan algoritma pembelajaran terawasi yang bisa digunakan untuk menyelesaikan permasalahan klasifikasi dan regresi, karakteristik metode ini setiap pohon tumbuh pada sampel *bootstrap* yang berbeda diambil dari data latih secara acak, dalam setiap pemisahan simpul selama pembentukan pohon keputusan, Sebagian sampel dari variabel m dipilih dari kumpulan data asli, kemudian yang terbaik akan digunakan dalam simpul tersebut. untuk RF yang terdiri dari pohon N dihitung menggunakan formula sebagai berikut:

$$l(y) = argmax_c(\sum_{n=1}^{n} I_{h_n(y)=c}) \quad (2.44)$$

Dengan *I* adalah fungsi indikator dan h_n adalah pohon ke-*n* dari RF.

4. Decision Tree (DT)

Decision Tree (DT) adalah algoritma pembelajaran mesin yang berlaku untuk membuat aturan keputusan seperti struktur pohon. DT merupakan metode pembelajaran terawasi yang dapat digunakan untuk klasifikasi dan regresi (Ghane dkk., 2022). DT menggunakan aturan untuk membuat keputusan seperti struktur pohon, memiliki cabang dalam pengambilan keputusan agar hasil yang didapatkan lebih maksimal (Spirkovska, 1993).

Elemen-elemen DT yaitu akar (*root*) merupakan tujuan akhir yang dihasilkan, ranting (*branches*) merupakan berbagai pilihan tindakan, simpul keputusan (`decision` node) digunakan untuk membuat `keputusan` berdasarkan fitur dari data yang diberikan, simpul daun (*leaf node*) merupakan hasil atas setiap tindakan, seperti dijelaskan pada Gambar 2.28. Pohon keputusan dimulai dengan satu simpul (*node*), kemudian simpul tersebut bercabang untuk menyatakan pilihan-pilihan yang ada. Langkah membuat pohon keputusan adalah sebagai berikut:

1. Pohon dibangun dalam suatu metode rekursif, seluruh contoh pelatihan dimulai dari simpul root, lalu dilakukan pengujian. Percabangan yang benar berdasarkan hasil pengujian, apakah simpul daun ditemukan atau tidak, jika tidak kembali ke langkah 1.
2. Atribut-atribut berada dalam suatu kategori

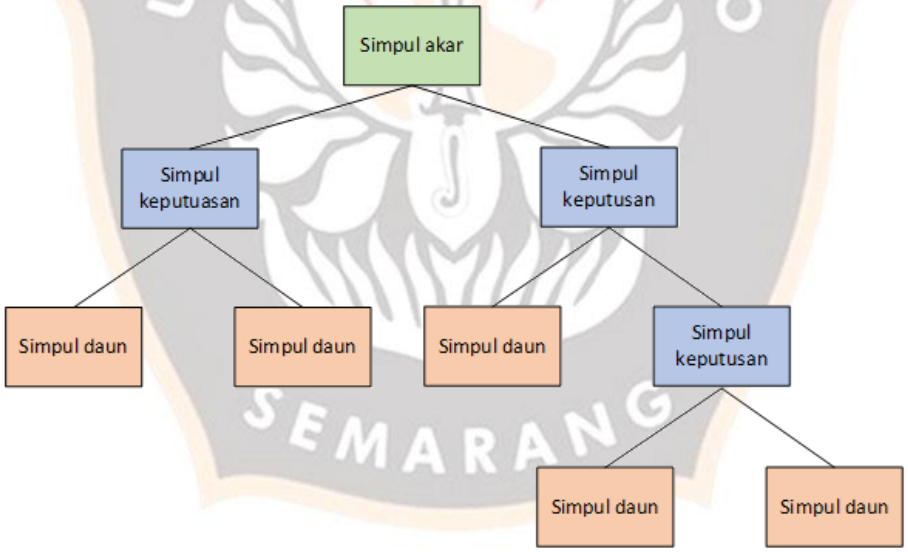
3. Menentukan node terpilih menggunakan nilai entropi dari setiap kriteria dengan data sampel yang ditentukan. Node terpilih adalah kriteria dengan entropi yang paling kecil, Pencarian nilai entropy menggunakan persamaan 2.45.
4. Atribut uji dipilih berdasarkan heuristik atau pengukuran statistik seperti Gain informasi (*information Gain*). Gain informasi merupakan ukuran efektivitas suatu atribut dalam mengklasifikasi data, perhitungan Gain menggunakan persamaan 2.46.

$$Entropy\ (S) = \sum_{i=1}^{n} -pi * log_2\ pi \quad (2.45)$$

Dengan *S* = himpunan kasus, *n* = jumlah partisi *S*, *pi* = proporsi dari *Si* terhadap *S*.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|si|}{|s|} * Entropy(si) \quad (2.46)$$

Dengan *S* = himpunan kasus, *A* = atribut, *n* = jumlah partisi atribut *a*, $|Si|$ = jumlah kasus pada partisi ke *i*, $|s|$ = jumlah kasus dalam *S*.



Gambar 2.28 Skema *Decision Tree*

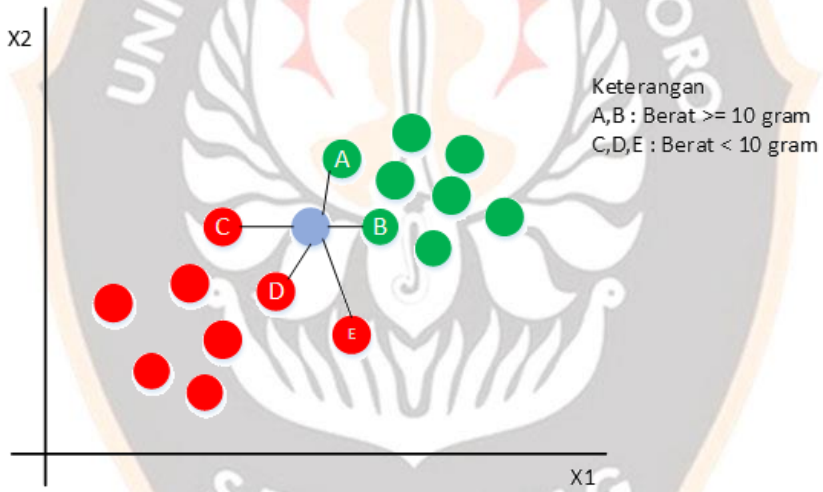
5. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) adalah algoritma yang mengklasifikasikan data berdasarkan kesamaan dengan data lainnya. Untuk kasus regresi KNN, ini memberikan pengenalan K terdekat dalam regresi tetangga (Wang dkk., 2022). KNN merupakan algoritma yang bekerja menggunakan K yang terdekat, yang

dijadikan sebagai acuan untuk menentukan kelas. KNN memproses regresi dan klasifikasi berdasarkan kemiripan dengan data lainnya (Wang dkk., 2022). Dalam *K-Nearest Neighbor* data point yang berdekatan disebut *neighbor* atau tetangga, seperti dijelaskan pada Gambar 2.29. Tahapan metode KNN adalah sebagai berikut:

1. Penentuan jumlah tetangga (K) yang akan digunakan untuk penentuan kelas.
2. Hitung jarak dari data baru ke masing-masing data point pada dataset.
3. Ambil sejumlah K data dengan jarak terdekat, kemudian tentukan kelas dari data baru tersebut.

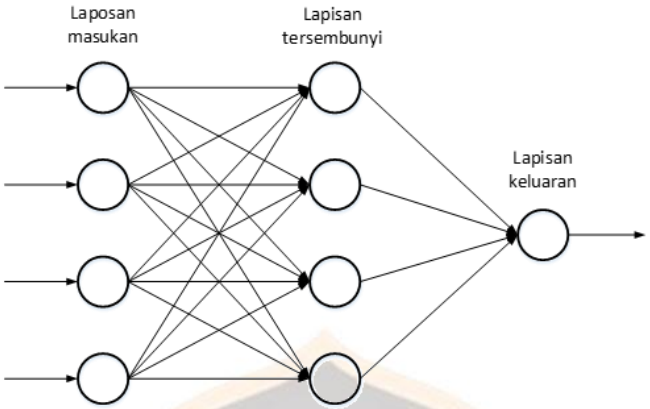
Untuk menghitung jarak antara dua titik pada algoritma KNN, menggunakan metode Jarak Euclidean pada persamaan 2.33.



Gambar 2.29 Skema *K-Nearest Neighbor*

6. *Backpropagation Neural network* (BPNN)

Back Propagation Neural Network (BPNN) adalah metode pembelajaran terawasi, yang menggunakan kesalahan keluaran untuk mengubah nilai bobotnya ke belakang. BPNN merupakan model yang meniru cara kerja otak manusia, terdiri dari lapisan masukan, lapisan tersembunyi dan lapisan keluaran, untuk mendapatkan hasil yang baik BPNN memerlukan pelatihan dengan waktu yang lama dan data yang besar (Luca dan Gallo, 2020).



Gambar 2.30 Skema *Backpropagation Neural Network*

Arsitektur BPNN terdiri dari neuron-neuron yang terhubung dalam lapisan masukan, lapisan tersembunyi dan lapisan keluaran. Lapisan masukan merupakan unit-unit yang bertugas menerima masukan data, lapisan tersembunyi merupakan unit-unit yang mengolah nilai data masukan, akan tetapi prosesnya tidak dapat diamati secara langsung, lapisan keluaran merupakan unit-unit keluaran hasil pengolahan data, arsitektur BPNN ditampilkan secara jelas pada Gambar 2.30. Langkah-langkah metode BPNN adalah sebagai berikut:

1. Inisialisasi bobot
2. Selama kondisi berhenti tidak terpenuhi, kerjakan langkah 3-8
3. Umpan maju (*feed forward*), tiap unit pada lapisan masukan menerima masukan x_i dan diteruskan ke lapisan tersembunyi

   $$(x_i, i = 1,2,2 \dots n) \quad (2.47)$$

4. Tiap unit pada lapisan tersembunyi z_j menghitung bobot sinyal masukan menggunakan persamaan 2.48

   $$(z_j, j = 1,2,3 \dots p)$$

   $$z_{injk} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad (2.48)$$

Dengan z adalah unit tersembunyi, v_{0j} adalah bobot bias unit masukan ke-j, x_i adalah unit masukan ke-i, v_{ij} adalah bobot unit masukan ke unit tersembunyi. Fungsi aktivasi yang digunakan menggunakan persamaan 2.49

$$z_j = f(z_{in_j}) \quad (2.49)$$

Dengan z_j adalah unit ke-j pada lapisan tersembunyi, z_{in_j} merupakan keluaran untuk unit z_j.

5. Tiap unit keluaran y_k menjumlahkan isyarat masukan yang telah dihitung dengan bobot menggunakan persamaan 2.50

$$(y_k, k = 1, \dots, m)$$
$$(y_{in_k} = w_{0k} + \sum_{k=1}^{p} z_j w_{jk}) \quad (2.50)$$

Dengan y_{in_k} adalah keluaran untuk unit y_k, w_{0k} adalah bobot bias untuk unit ke-*k,* z_j adalah unit ke-j pada lapisan tersembunyi, w_{jk} adalah bobot unit tersembunyi ke unit keluaran. Dengan menerapkan fungsi aktivasi menggunakan persamaan 2.49

6. Umpan maju (*back forward*), tiap unit keluaran $(y_k, k = 1, \dots, m)$ menerima pola pelatihan masukannya, dengan menghitung nilai kesalahan setiap lapisan dengan persamaan 2.51.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (2.51)$$

Dengan δ_k adalah faktor koreksi bobot w_{jk}, t_k adalah target, y_k adalah unit keluaran ke-*k*, y_{in_k} adalah keluaran untuk unit y_k. Perhitungan koreksi bobot bias menggunakan persamaan 2.52.

$$\Delta w_{jk} = \alpha \delta_k x_j$$
$$\Delta w_{0k} = \alpha \delta_k \quad (2.52)$$

Dengan Δw_{jk} adalah selisih antara $w_{jk}\ (t)$ dengan $w_{jk}\ (t+1)$, Δw_{0k} adalah bobot bias untuk unit tersembunyi ke-*k*, α adalah tingkat pembelajaran (*learning rate*), δ_k adalah faktor koreksi bobot w_{jk}, *x* adalah masukan.

7. Tiap unit tersembunyi z_j menjumlahkan delta masukannya dari unit-unit yang berada pada lapisan diatasnya dengan persamaan 2.53.

$$(z_j, j = 1, \dots, p)$$
$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk} \quad (2.53)$$

Dengan δ_k adalah faktor koreksi bobot w_{jk}, w_{jk} adalah bobot unit tersembunyi ke unit keluaran. Hitung nilai kesalahan setiap lapisan dengan persamaan 2.54.

$$\delta_j = \delta_{in_j} f(x_{in_j}) \quad (2.54)$$

Dengan δ_j adalah faktor koreksi bobot v_{ij}, δ adalah faktor koreksi, *x* adalah input. Hitung koreksi bobot dan bias dengan persamaan 2.55.

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.55)$$

Dengan Δv_{ij} adalah bobot unit masukan ke unit tersembunyi, α adalah tingkat pembelajaran, δ_j adalah faktor korelasi bobot v_{ij}, x_i adalah unit masukan ke-*i*

8. Perbaiki bobot dan bias (*update weight*), tiap unit keluaran y_k memperbaharui bobot dan bias *j,* dihitung dengan persamaan

   $y_k, k = 1, \dots, m; j = 0,1, \dots p$

   $$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (2.56)$$

Dengan w_{jk} adalah bobot unit tersembunyi ke unit keluaran, Δw_{jk} adalah selisih bobot unit tersembunyi ke unit keluaran. Tiap unit tersembunyi z_j memperbaharui bobot dan bias dihitung dengan persamaan 2.57.

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad (2.57)$$

Dengan v_{ij} adalah bobot unit masukan ke unit tersembunyi, Δv_{ij} adalah selisih bobot unit input ke unit tersembunyi.

9. Uji syarat berhenti

7. Principal Component Regression (PCR)

Principal Component Regression (PCR) adalah pendekatan pemodelan prediktif yang melibatkan penggunaan metode *Principal Component Analysis* (PCA) dan MLR, yang semakin banyak digunakan di berbagai aplikasi dan industri (Meerasri dan Sothornvit, 2022). Metode ini juga digunakan untuk memodelkan prediksi dengan beberapa variabel independen (Reyhaneh dkk., 2021). PCA merupakan metode yang digunakan untuk mereduksi dimensi pada citra digital, metode ini digunakan untuk memudahkan proses pengolahan citra. Proses PCR diawali dengan pereduksian dimensi data citra, setelah data direduksi, kemudian dilanjutkan mengolah data menggunakan metode MLR (Chuen dan Aziz, 2021) (Reyhaneh dkk., 2021).

Gambar 2.31 Skema *Principal Component Regression*

Seperti yang telah dijelaskan pada Gambar 2.31, metode PCR menggunakan analisis komponen utama dengan tahapan sebagai berikut:

1. Pengujian variabel menggunakan Bartlett's Test dan Kaiser-Meyer-Olkin (KMO)

$$KMO = \frac{\sum_i[\sum_{i\neq k} r_{ik}^2]}{\sum_i \sum_{i\neq k} r_{ik}^2 + \sum_i \sum_{i\neq k} a_{ik}^2} \quad (2.58)$$

Dimana r_{ik}^2adalah kuadrat matriks korelasi sederhana, a_{ik}^2 adalah kuadrat matriks korelasi parsial (Shrestha, 2021).

2. Standarisasi data

$$z = \frac{x-\mu}{\sigma} \quad (2.59)$$

Nilai simpangan baku dihitung menggunakan persamaan 2.60

$$S = \sqrt{\frac{\sum(x_i-\mu)^2}{n}} \quad (2.60)$$

Dimana S adalah nilai simpangan baku, x_i merupakan nilai x ke i dari satu variabel, μ merupakan nilai rata-rata x, dan n merupakan jumlah data (Jamal dkk., 2014)

3. Menghitung matriks kovarian / korelasi

$$Cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)\left(y_i - \mu_y\right) \quad (2.61)$$

Dimana μ_x adalah rata-rata dari sampel x, μ_y adalah rata-rata dari sampel y, x_i adalah nilai observasi ke i dari variabel x, y_i adalah nilai observasi ke i dari variabel y (Jian dkk., 2022) (Wang dkk., 2021).

4. Menghitung nilai eigen dan vektor eigen

$$Determinan\ (A - \lambda I) = 0 \quad (2.62)$$

Dengan A adalah matriks mxn, λ adalah nilai eigen, I adalah matriks identitas (Ersöz dkk., 2017). Untuk proses perhitungan vektor eigen, menggunakan persamaan 2.63

$$AX = \lambda X$$

$$AX - \lambda X = 0$$

$$(A - \lambda)x = 0$$

$$(A - \lambda I)x = 0, x \neq 0 \quad (2.63)$$

Dimana A adalah matriks nxn yang memiliki n nilai eigen λn, λ adalah nilai eigen, x adalah matriks non zero, I adalah matriks identitas. Nilai eigen matriks X disimbolkan dengan $\lambda_1, \lambda_2, \lambda_3 \ldots \lambda_n$ dan vektor eigen disimbolkan dengan $x_1, x_2, x_3 \ldots x_n$ (Kumar dan Verma, 2021).

Rotasi matriks komponen

$$X^*_{mxn} = X_{mxn} * T_{nxn} \quad (2.64)$$

5. Menghitung nilai *principal component* (PC)

$$PC(\%) = \frac{nilai\ eigen}{varian\ kovarian}\ x\ 100\% \quad (2.65)$$

2.2.11 Metode Evaluasi

1. Sensitivitas Deteksi Tepi

Sensitivitas pada deteksi tepi diukur menggunakan tingkat kesalahan (*error rate*) (Collins, 2007)**.** Tingkat kesalahan diukur menggunakan selisih jumlah piksel pada citra asli hasil deteksi tepi dengan citra kebisingan deteksi tepi, kemudian dibagi dengan jumlah piksel citra asli hasil deteksi tepi.

$$P = \frac{|n_N - n_R|}{n_R} \quad (2.66)$$

dengan P = nilai kesalahan, n_N = jumlah piksel pada deteksi tepi citra dengan kebisingan, n_R = jumlah piksel pada deteksi tepi citra referensi. Nilai *P* yang besar menyatakan sensitivitas yang tinggi detektor tepi terhadap kebisingan (Qidwai dan Chen, 2020).

2. Kesalahan Kuadrat Rata-Rata (Mean Square Error - MSE)

Kesalahan Kuadrat Rata-Rata (Mean Square Error - MSE) dalam citra merupakan pengukuran nilai piksel dari informasi citra asli dengan citra yang terdegradasi. Secara umum, MSE dihitung sebagai kesalahan kuadrat rata-rata antara citra asli dan citra terdegradasi. Kesalahan tersebut dapat dihitung sebagai

perbedaan antara citra asli dan citra terdegradasi. Nilai MSE yang lebih rendah menunjukkan kualitas citra yang tinggi dan terbaik (Gupta dkk., 2018).

$$MSE = \frac{1}{n}|x - x'|^2 = \frac{1}{n}\sum_{i=1}^{n}(x - x')^2 \quad (2.67)$$

3. Kinerja Titik Kunci (*Key Point Performance*)

Titik kunci merupakan titik gumpalan yang terdapat pada detektor blob, titik kunci dihitung pada wilayah entropi tinggi tanpa membagi citra, titik kunci dicocokkan pada penggunaan filter algoritma untuk mengetahui nilai piksel pada citra. Titik kunci yang baik adalah yang bisa lebih banyak mendeteksi wilayah entropi (Niyishaka dan Bhagvati, 2020).

Titik blob didefinisikan dari operasional diferensial geometri, mengarah ke deskriptor gumpalan yang kovarian dengan rotasi, dan penskalaan dalam domain citra (Lindeberg, 1998). Pencocokan fitur merupakan proses untuk menemukan titik-titik di area citra dari detektor blob di setiap metode yang digunakan. Setiap titik dikodekan sebagai deskriptor biner *D(i)* (Niyishaka dan Bhagvati, 2020)

$$Hd = \sum_{k=1}^{z} XOR(\beta D_k(i), \beta D_k(j)) \quad (2.68)$$

4. Kesalahan Kuadrat Rata-Rata Akar (Root Mean Square Error - RMSE)

Kesalahan Kuadrat Rata-Rata Akar (Root Mean Square Error - RMSE) merupakan metode yang digunakan untuk mengevaluasi model regresi dari suatu model. RMSE yang baik adalah yang menghasilkan nilai yang kecil. Semakin kecil nilai RMSE maka nilai yang diobservasi semakin akurat. RMSE biasa digunakan untuk menghitung kesalahan pada prediksi data kuantitatif (Chen dkk., 2019).

$$RMSE = \sqrt{\frac{1}{N}\sum_{I=1}^{N}(Y_{da} - Y_{dp})^2} \quad (2.69)$$

Dengan Y_{da} adalah nilai asli, Y_{dp} adalah nilai prediksi

5. Kesalahan Absolut Rata-Rata (Mean Absolute Error - MAE)

Kesalahan Absolut Rata-Rata (Mean Absolute Error - MAE) merupakan rata-rata selisih mutlak nilai sebenarnya (aktual) dengan nilai prediksi. MAE

digunakan untuk mengukur keakuratan suatu model statistik dalam melakukan prediksi. MAE yang baik adalah yang nilainya kecil. Semakin kecil nilai MAE, maka nilai prediksi semakin akurat (Chen dkk., 2019).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_{da} - Y_{dp}\right| \tag{2.70}$$

Dengan Y_{da} adalah nilai asli, Y_{dp} adalah nilai prediksi

6. Koefisien Determinasi (Coefficient of Determination - R^2)

Koefisien determinasi merupakan metode untuk mengetahui pengaruh variabel independen terhadap variabel dependen pada statistik. Koefisien determinasi menjelaskan seberapa pengaruh variabel bebas terhadap variabel terikat. R^2 yang bagus adalah yang menghasilkan nilai mendekati 1, hal ini berarti kemampuan variabel bebas dalam menjelaskan keterkaitan dengan variabel terikat adalah tinggi (Pourmohammad dkk., 2020).

$$R^2 = \frac{\left(\sum_{i=1}^{N}(Y_{da}-Y_{dp})(Y_{dp}-Y_{dw})\right)^2}{\sum_{i=1}^{N}(Y_{da}-Y_{dp})^2 \sum_{i=1}^{N}(Y_{dp}-Y_{dw})^2} \tag{2.71}$$

Di mana Y_{da} adalah nilai asli, Y_{dp} adalah nilai prediksi, Y_{dw} , adalah nilai rata-rata prediksi, dan n adalah jumlah data yang tersedia.

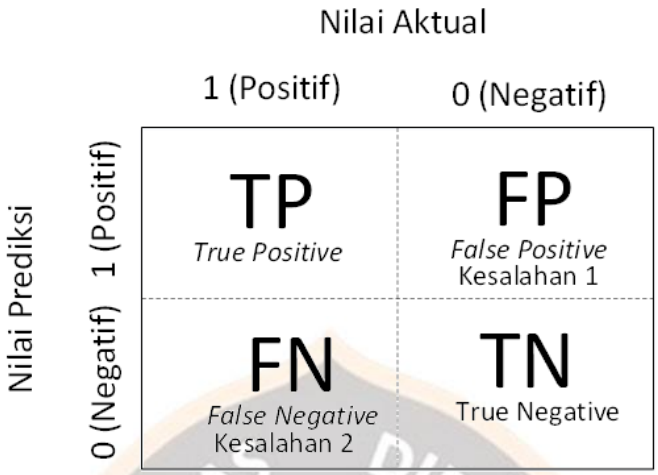
7. *Confusion Matrix*

Confusion Matrix merupakan metode evaluasi untuk klasifikasi pada pembelajaran mesin. Cara kerja *Confusion Matrix* adalah dengan nilai aktual dengan nilai prediksi. Empat istilah yang digunakan pada *Confusion Matrix* yaitu Positif Benar (*True Positive*), Negatif Benar (*True Negative*), Positif Salah (*False Positive*), dan Negatif Salah (*False Negative*). *Confusion Matrix* menggunakan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda, seperti dijelaskan pada Gambar 2.32. *Confusion Matrix* juga digunakan untuk menghitung akurasi, presisi dan sensitivitas *(recall)* dengan persamaan sebagai berikut:

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.72}$$

$$presisi = \frac{TP}{TP+FP} \tag{2.73}$$

$$sensitivitas = \frac{TP}{TP+FN} \tag{2.74}$$



Gambar 2.32 *Confusion Matrix*

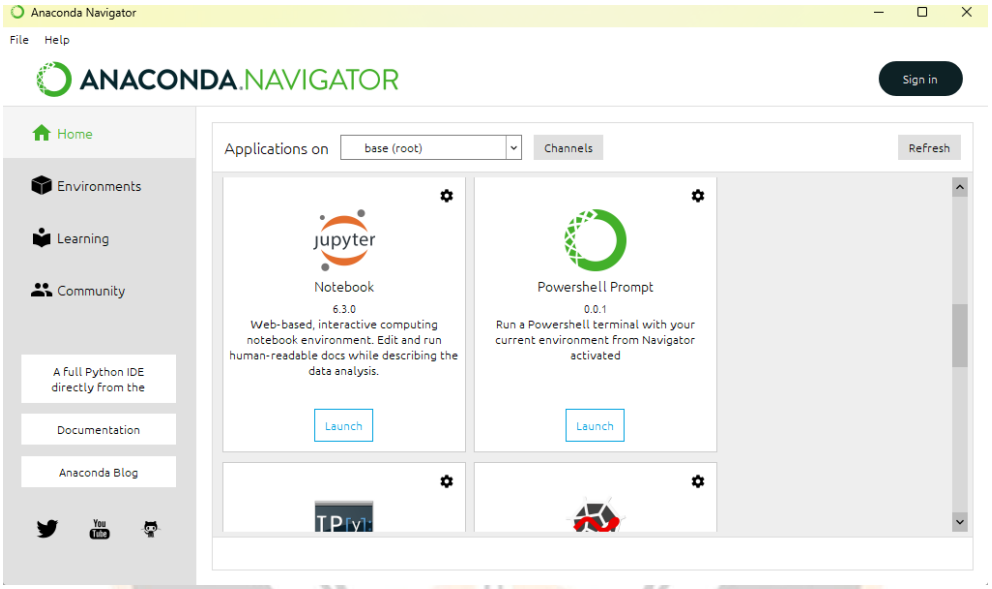
2.2.12 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini yaitu bahasa pemrograman Python. Python merupakan bahasa pemrograman tingkat tinggi yang bisa digunakan untuk pemrograman berorientasi objek. Dalam penelitian ini juga menggunakan pustaka OpenCV. Pustaka ini digunakan untuk mengolah citra digital baik citra maupun video. OpenCV banyak digunakan pada aplikasi visi komputer karena kemampuannya yang handal dalam mengolah citra digital. Tahapan penelitian dilakukan dengan pembuatan modul–modul perangkat lunak yang digunakan dalam penelitian (Lutz, 2009).

Selain OpenCV, pustaka lain yang digunakan yaitu Matplotlib, Scikit, Pandas, Numpy. Pustaka ini mempunyai fungsi masing – masing, Matplotlib mempunyai fungsi untuk bekerja dengan diagram dan grafik, Scikit mempunyai fungsi untuk bekerja dengan pembelajaran mesin, Pandas mempunyai fungsi untuk bekerja dengan pembelajaran mesin, Numpy mempunyai fungsi untuk bekerja dengan data numerik dan himpunan (Rossum dkk, 2018).

Python yang digunakan berjalan pada platform Anaconda Navigator yang bersifat sumber terbuka (*open source),* Anaconda Navigator merupakan platform yang bisa berjalan pada sistem operasi Windows, Linux dan Mac OS. Beberapa paket pustaka yang dibutuhkan untuk analisis citra digital sudah tersedia pada Anaconda Navigator. Di dalam Anaconda Navigator terdapat Jupyter yang

mendukung bahasa pemrograman Python. Jupyter merupakan aplikasi berbasis sumber terbuka yang bisa digunakan untuk menulis program dengan menggunakan Python. Secara detail Anaconda Navigator ditampilkan pada Gambar 2.33.



Gambar 2.33 Perangkat lunak Anaconda Navigator