

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian ini dilakukan dengan beberapa tinjauan pustaka serta dasar teori pada penelitian terdahulu yang dikaji dan dipelajari untuk menjadi acuan dan pembandingan dalam menentukan gabungan metode yang akan digunakan pada penelitian ini. Penelitian-penelitian tersebut menggunakan beberapa metode ekstraksi ciri dan metode klasifikasi yang beragam. Penelitian yang pernah dilakukan oleh Lestari dan Awaliyah (2016). Dalam penelitian ini menggunakan metode ekstraksi ciri warna dengan histogram *HSV*, metode *canny* untuk mendapatkan informasi bentuk, dan tekstur orde dua untuk mendapat informasi tekstur dalam sistem temu kembali pada gambar ikan hias. Proses pengujian dilakukan sebanyak 24 kali dengan 6 jenis ikan hias dengan kueri citra ikan hias diposisi horizontal dan citra ikan hias diposisi vertikal. Didapatkan hasil temu kembali citra menggunakan gabungan ciri memiliki nilai rata-rata sama antara citra kueri diposisi horizontal dan vertikal yaitu sebesar 77,083%.

Penelitian lainnya dilakukan oleh Restu, Agus dan Arry (2018) menggunakan ekstraksi ciri *Gray Level Co-occurrence Matrix (GLCM)* untuk klasifikasi mutu jeruk keprok. Pada penelitian ini terdapat 100 dataset gambar dan menggunakan perbandingan 60 gambar untuk data latih dan 40 gambar untuk data uji. Ekstraksi ciri *GLCM* yaitu *contrast*, *homogeneity*, *energy* dan *entropy*. Metode *Support Vector Machine (SVM)* digunakan untuk identifikasi berdasarkan ciri yang telah diekstraksi dengan menggunakan *GLCM*. Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82.5% dengan jumlah data latih sebanyak 20.

Penelitian selanjutnya dilakukan oleh Sri (2013) adalah melakukan identifikasi penyakit dengan melakukan perbandingan algoritma *Backpropagation* dan *K-Nearest Neighbor (K-NN)*. Pada penelitian ini terdapat 82 data pasien yang dibagi sebanyak 72 pasien untuk data *training* dan 10 pasien untuk data *testing*. Dari hasil penelitian

didapatkan bahwa identifikasi terbaik untuk algoritma *K-Nearest Neighbor* (K-NN) dengan menggunakan nilai $K=3$, $K=7$, dan $K=9$ dengan tingkat akurasi sebesar 100% sedangkan untuk algoritma *Backpropagation* didapatkan tingkat akurasi sebesar 90% dengan nilai *learning rate* 0,1 dan nilai toleransi *error* sebesar 0,001.

Penelitian selanjutnya dilakukan oleh Dani Syahid, Jumadi, dan Dian Nursantika (2016) dalam penelitian ini menggunakan citra daun tanaman hias untuk dijadikan objek dari sistem deteksi. Sistem pendeteksi citra daun tanaman hias bekerja dengan membandingkan data citra latih yang telah tersimpan pada *database* terhadap citra yang akan diuji. Setiap input citra akan dilakukan proses konversi *red, green, blue* (RGB) ke dalam ekstraksi fitur warna *hue, saturation, value* (HSV). Setelah didapat nilai HSV, maka dilakukan proses klasifikasi menggunakan metode *K-Nearest Neighbor* (K-NN). Data sampel pada penelitian ini menggunakan 5 klasifikasi citra data latih dengan 10 data citra uji pada setiap data citra latih. Pada penelitian ini, diperoleh hasil dari akurasi sistem pendeteksi citra tanaman dengan hasil sebesar 92%.

Penelitian selanjutnya dilakukan oleh Mungki Astiningrum, Anugrah Nur Rahmanto, dan Welsi Eldayosa(2020) dalam membuat aplikasi identifikasi kualitas sayur bayam menggunakan ekstraksi fitur tekstur *GLCM* dan mencari nilai warna dengan metode *YUV*. Kemudian dilakukan klasifikasinya menggunakan metode *K-NN*. Pada penelitian ini data latih berjumlah 30 data citra daun bayam yang dibagi menjadi 3 kelas yaitu bayam baik, bayam kurang segar, dan bayam buruk. Data uji yang digunakan sejumlah 5 data untuk setiap kelasnya. Pada penelitian ini diperoleh tingkat akurasi terbaik pada nilai $K=7$ dengan nilai akurasi sebesar 86,66%.

Pada penelitian selanjutnya dilakukan oleh Dhebys Suryai, Wilda Imama Sabilla, Hanif Luhur Wicaksono(2020). Penelitian ini melakukan identifikasi kualitas cabai berdasarkan warna dan tekstur. Setiap citra akan dicari nilai dengan ekstraksi fitur warna dan ekstraksi fitur tekstur menggunakan metode *Gray Level Co-Occurrence* (GLCM) adapun 5 jenis ciri tekstur yang digunakan untuk mendeskripsikan kekerasan dari tekstur, yaitu *Angular Second Moment*(ASM), *Contrast*, *Correlation*, *Inverse*

Difference Moment. Kemudian nilai akan diklasifikasikan menggunakan metode *K-Nearest Neighbor*. Pada penelitian ini diperoleh akurasi sistem sebesar 94,4%.

Dari seluruh penelitian terkait, didapatkan metode sebagai acuan dalam penelitian yang akan dilakukan adalah dengan menggabungkan dari beberapa metode yaitu: ekstraksi fitur warna *HSV*, ekstraksi fitur bentuk *canny*, ekstraksi fitur tekstur *Gray Level Co-occurrence (GLCM)* dan metode klasifikasi dengan menggunakan *K-Nearest Neighbor (K-NN)* sehingga mampu mendeteksi citra stiker pada kendaraan pegawai.

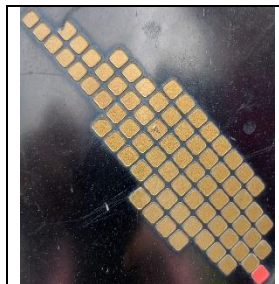
Tabel 2.1 Penelitian terkait

No	Nama Penulis	Metode Penelitian	Hasil Penelitian
1.	Lestari Handayani dan Alwiyah (2016)	Penerapan metode ekstraksi fitur warna <i>HSV</i> , ekstraksi fitur bentuk <i>canny</i> , dan ekstraksi fitur tekstur orde dua	Hasil penelitian yang diperoleh dari temu kembali citra menggunakan gabungan ciri memiliki rata-rata dengan citra kueri posisi horizontal dan kueri citra kueri vertikal yaitu 77,083%
2.	Restu Widodo, Agus Wahyu Widodo, dan Arry Supriyanto (2018)	Metode ekstraksi fitur <i>Gray Level Co-occurrence Matrix (GLCM)</i> yaitu <i>contrast</i> , <i>homogeneity</i> , <i>energy</i> dan <i>entropy</i> . Metode <i>Support Vector Machine (SVM)</i>	Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82,5% .
3.	Sri Redjeki (2013)	Algoritma <i>K-Nearest Neighbor (K-NN)</i> dan algoritma <i>Backpropagation</i>	Dari hasil perbandingan terhadap klasifikasi menggunakan metode <i>K-NN</i> diperoleh hasil terbaik dengan nilai $K=3$, $K=7$ dan $K=9$ tingkat akurasi sebesar 100% sedangkan metode <i>Backpropagation</i> diperoleh tingkat akurasi sebesar 90% dengan nilai <i>learning rate</i> 0,1 dan nilai toleransi <i>error</i> 0,001.
4.	Dani Syahid, Jumadi, Dian Nursantika (2016)	Metode <i>K-Nearest Neighbor (K-NN)</i> , model warna <i>HSV</i>	Penelitian ini mendeteksi citra daun tanaman hias dengan proses konversi <i>RGB</i> ke dalam ekstraksi fitur warna <i>HSV</i> kemudian dilakukan proses klasifikasi menggunakan metode <i>K-NN</i> . Pada penelitian

No	Nama Penulis	Metode Penelitian	Hasil Penelitian
			ini, diperoleh hasil dari akurasi sistem pendeteksi citra daun tanaman hias dengan nilai 92%
5.	Mungki Astiningrum, Anugrah Nur Rahmanto, Welsi Eldayosa (2020)	Metode <i>Gray Level Co-Ocurrence (GLCM)</i> , <i>YUV</i> , dan <i>K-Nearest Neighbor (K-NN)</i>	Pada penelitian ini, data <i>training</i> berjumlah 30 data untuk setiap kelas. Data <i>testing</i> yang digunakan berjumlah 5 data untuk setiap kelasnya. Tingkat akurasi tertinggi diperoleh nilai sebesar 86,66% pada $K=7$.
6.	Dhebys Suryani, Wilda Imama Sabilla, Hanif Luhur Wicaksono (2020)	Metode <i>Gray Level Co-Ocurrence (GLCM)</i> dan <i>K-Nearest Neighbor (K-NN)</i>	Pada penelitian ini menggunakan 72 citra latih yang dibagi ke dalam 3 jenis kelas dan 6 data citra uji. Penelitian ini didapat nilai akurasi terbaik dari nilai $K=1$ dengan total akurasi sebesar 94,4%
7.	Nurul Fauzia Azizah (2020)	Ekstraksi fitur bentuk dengan metode <i>Canny</i> , ekstraksi fitur tekstur dengan metode <i>Gray Level Co-occurrence Matric (GLCM)</i> , ekstraksi fitur warna dengan metode <i>HSV</i> , dan klasifikasi dengan algoritma <i>K-Nearest Neighbour (K-NN)</i>	Pada penelitian ini digunakan penggabungan metode ekstraksi fitur <i>HSV</i> , <i>Canny</i> , <i>GLCM</i> dan Penerapan metode <i>K-NN</i> sehingga mampu mendeteksi citra stiker pada kendaraan pegawai

2.2 Landasan Teori

2.2.1 Stiker



Gambar 2.1 Stiker logo ITERA

Stiker merupakan salah satu alternatif media pembelajaran. Bentuk dari stiker yang dijumpai bervariasi. Manfaat stiker salah satunya adalah dapat dijadikan bahan untuk mengingat suatu kejadian dari tulisan yang terkandung didalamnya, Lebih tepatnya stiker dapat dimanfaatkan sebagai alat komunikasi. Selain itu, stiker dapat dimanfaatkan sebagai tanda bukti diri, iklan, dan seni[4].

2.2.2 Citra

Pengertian citra secara umum adalah merupakan suatu gambar, foto ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek[11]. Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi[12]. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudnya tentu sebuah gambar dapat memberikan informasi yang telah banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual)[13].

2.2.3 Citra Digital

Citra digital yaitu gambar 2 dimensi yang dihasilkan dari suatu gambar analog dua dimensi kontinu yang diubah menjadi gambar diskrit melalui suatu proses pencuplikan (*sampling*) posisi dan kuantisasi intensitas. Citra digital dapat dinyatakan dengan matriks dua dimensi $f(x,y)$. Disini “x” dan “y” yaitu suatu koordinat piksel dalam matriks dan “f” yaitu derajat intensitas piksel tersebut. Citra berwarna bisa dinyatakan dengan berbagai cara, salah satunya dengan cara digunakannya sinyal *RGB* (*Red, Green, Blue*)[14]. Citra digital tersimpan dalam suatu bentuk larik (*array*) angka digital yang merupakan hasil kuantifikasi dari tingkat kecerahan masing-masing piksel penyusun citra. Citra digital yang tersimpan dalam larik dua dimensi tersusun atas unsur-unsur kecil yang disebut dengan piksel. Struktur *array* ini tersusun dalam baris

horizontal yang disebut baris (*lines*) dan kolom vertikal (*samples*). Masing-masing piksel dalam raster citra menyimpan nilai tingkat kecerahan piksel yang diwujudkan sebagai suatu angka digital[11]. Gambar 1 adalah koordinat citra digital terhadap sumbu (x,y) suatu bidang dua dimensi. Citra digital secara matematis dapat ditulis dalam bentuk matriks sebagai berikut [15]:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Keterangan :

M : banyaknya baris pada array citra

N : banyaknya kolom pada array citra

2.2.3.1 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu kanal pada setiap *pixel*, dengan kata lain nilai bagian *red* = *green* = *blue*. Nilai tersebut digunakan untuk menunjukan tingkat intensitas. Warna yang memiliki citra *grayscale* adalah warna keabuan dengan berbagai tingkatan dari hitam hingga putih. Citra *grayscale* dapat diperoleh dari citra *RGB*. Nilai intensitas citra *grayscale* dihitung dari nilai intensitas citra *RGB* dengan menggunakan persamaan (2)[16].

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.2)$$

Keterangan :

$I_R(x, y)$: nilai piksel *red* titik (x,y)

$I_G(x, y)$: nilai piksel *green* titik (x,y)

$I_B(x, y)$: nilai piksel *blue* titik (x,y)

$I_{BW}(x, y)$: nilai piksel *black and white* (x,y)

2.2.3.2 Citra Biner

Citra biner merupakan hasil pengolahan dari *black and white* (citra *grayscale*), dengan menggunakan fungsi sebagai berikut[8].

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (2.3)$$

Dan dalam *floating point*,

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (2.4)$$

Keterangan:

$I_{BW}(x, y)$: nilai piksel *gray* titik (x,y)

$I_{Bin}(x, y)$: nilai piksel *binary* titik (x,y)

T : nilai *threshold*

2.2.3.3 Citra Warna

Citra warna sering disebut juga citra *RGB* atau citra *true color* karena dapat mempresentasikan warna objek menyerupai warna aslinya dengan mengkombinasikan ketiga warna dasar yaitu *red* (R), *green* (G), *blue* (B). Tiap piksel memiliki tiga nilai kanal yang mewakili tiap komponen warna dasar citra[17].

2.2.4 Pengolahan Citra Digital

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk manipulasi dan modifikasi citra dengan berbagai cara[18]. Pengolahan citra (*image processing*), yaitu suatu proses citra menggunakan komputer agar citra yang dihasilkan mendapat kualitas yang baik dari sebelumnya[14].

Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital

juga mencakup semua data dua dimensi. Operasi-operasi pada pengolahan citra diterapkan bila[11]:

1. Perbaikan atau modifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*). Contoh: perbaikan kontras gelap/terang. Perbaikan tepian objek, penajaman, pemberian warna semu, dan lain-lain.
2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan (*image restoration*). Contoh: perbaikan kesamaran (*debluring*) citra tampak kabur karena pengaturan fokus lensa tidak tepat/ kamera goyang, penghilangan *noise*.
3. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*). Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukan ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam mengidentifikasikan objek (*image analysis*). Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh: pendeteksian tepi objek.
5. Sebagian citra perlu digabung dengan bagian citra yang luas lain (*image reconstruction*). Contoh: beberapa foto rontgen digunakan untuk membentuk ulang gambar tubuh.
6. Citra perlu dimampatkan (*image compression*) contoh: suatu file citra berbentuk *BMP* berukuran 258 KB dimampatkan dengan metode *JPEG* menjadi berukuran 49 KB.
7. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (*steganografi* dan *watermarking*).

2.2.5 Computer Vision

Sebagai salah satu disiplin ilmu, *computer vision* berkaitan dengan teori di balik sistem bautan bahwa ekstraksi informasi dari gambar. Data gambar dapat mengambil banyak bentuk, seperti urutan video, pandangan dari beberapa kamera, atau data multi-dimensi

dari *scanner* medis subdomain dari *computer vision* termasuk adegan rekonstruksi, deteksi *event*, pelacakan video, pengenalan objek, belajar, *indexing*, estimasi gerak, dan pemulihan citra[11].

2.2.6 Preprocessing

Preprocessing gambar latih merupakan langkah untuk meningkatkan kualitas gambar. Gambar mengalami transformasi untuk menghasilkan fitur penting dari sebuah gambar. Pengubahan aras warna menjadi gambar abu-abu (*grayscale*) juga akan menurunkan tingkat komputasi pada tahap pengambilan fitur[19]. Tujuan utama *pre-processing* adalah menghilangkan bagian-bagian dari citra digital yang tidak relevan untuk proses klasifikasi[20].

2.2.6.1 Resize Citra

Resize citra merupakan proses untuk melakukan perubahan luas dari citra menjadi lebih besar ataupun lebih kecil dari ukuran aslinya dengan melakukan perubahan terhadap ukuran dapat mengakibatkan pergeseran pada nilai warna sehingga mengubah konten digital yang ada didalamnya[21].

2.2.6.2 Mean Filter

Mean Filter adalah intensitas pada beberapa *pixel* lokal dimana setiap *pixel* akan digantikan nilainya dengan rata-rata dari nilai intensitas *pixel* tersebut dengan *pixel-pixel* tetangganya, dan jumlah *pixel* tetangga yang dilibatkan tergantung pada filter yang dirancang. Secara matematis, hal ini dapat dinyatakan dengan persamaan:[22]

$$h(x, y) = \frac{1}{M} \sum_{(k,l) \in N} f(k, l) \quad (2.5)$$

Keterangan :

H : filter rata-rata

M : jumlah *pixel* pada jendela sebesar N*N

F : filter (*pixel*)

K : intensitas *pixel*

Cara mencari nilai *mean* diatas adalah:

1. Baca nilai *pixel* yang akan diproses beserta *pixel-pixel* tetangganya.
2. Gantikan nilai dengan bobot rata-rata (pada kernel ukuran 3 x 3 bobot maksimum adalah 1/9 dan 1/25 untuk kernel 5 x 5).
3. Pilih nilai pada bagian tengah untuk nilai yang baru bagi *pixel* (x,y).

Salah satu dari bentuk *Low Pass Filter* adalah filter rata-rata yang merupakan filter dalam bentuk matriks yang berukuran M x N, dan nilainya adalah sama untuk setiap elemen, dan karena bersifat *LPF* maka jumlah seluruh elemen adalah satu dan dituliskan dengan persamaan:

$$H(i,j) = \frac{1}{m \times n}, 1 \leq i \leq m, 1 \leq j \leq n \quad (2.6)$$

Keterangan:

H : filter rata-rata

i,j : koordinat *pixel* pada citra

m : lebar citra (*pixel*)

Filter rata-rata berukuran 5 x 5 adalah:

$$H = \begin{bmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{bmatrix} \quad (2.7)$$

Mean Filter merupakan salah satu algoritma memperhalus citra dengan cara perhitungan nilai intensitas rata-rata pada koordinat *pixel* pada citra. Algoritma yang umum digunakan adalah *Arithmetic* dan *Geometric Mean Filter* proses yang dilakukan adalah menghitung rata-rata nilai dari citra yang rusak $g(x,y)$ pada sebuah blok area citra yang didefinisikan oleh $S(x,y)$. Nilai dari citra $f(x,y)$ yang diperbaiki pada tiap titik (x,y) hanya dihitung dengan menggunakan *pixel* dalam menggunakan *pixel* dalam daerah yang didefinisikan oleh $S(x,y)$ dengan rumus:

$$f(x,y) = \frac{1}{M} \sum_{(s,t) \in S_{xy}} g(s,t) \quad (2.8)$$

Pada algoritma *Geometric Mean Filter* proses yang dilakukan adalah sama dengan *Arithmetic Mean Filter* dengan rumus:

$$f(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{1/mn} \quad (2.9)$$

Keterangan:

x,y : koordinat *pixel* pada citra

mn : dimensi citra (*pixel*)

s,t : nilai intensitas *pixel*

$\prod$: perkalian nilai *pixel* yang terkena filter

Pada algoritma ini setiap *pixel* yang diperbaiki oleh hasil kali masing-masing *pixel* dengan *subimage window*, kemudian dipangkatkan dengan $1/mn$.

2.2.7 Segmentasi

Segmentasi adalah salah satu cara yang sering digunakan dalam memilih citra dalam memilah citra dalam data, membagi citra menjadi bagian yang diharapkan termasuk objek yang dianalisis. Segmentasi sering dideskripsikan sebagai proses analogi

terhadap proses pemisahan latar depan dan latar belakang. Memilih bentuk dalam sebuah citra berguna dalam pengukuran atau pemahaman citra. Secara tradisional, segmentasi didefinisikan sebagai proses pendefinisian jangkauan nilai gelap atau terang pada citra yang sebenarnya, memilih piksel dalam jangkauan ini sebagai latar depan dan menolak sisanya sebagai latar belakang. Dengan demikian, citra terbagi atas dua bagian utuh, atau warna yang membatasi setiap wilayah. Salah satu metode yang efektif dalam segmentasi citra biner adalah memeriksa hubungan piksel dengan tetangganya dan memberikan label. Metode ini disebut pelabelan komponen[23].

Pembagian citra menjadi beberapa daerah, berdasarkan sifat-sifat tertentu dari citra yang dapat dijadikan pembeda, disebut juga segmentasi citra. Suatu daerah dalam citra adalah sekumpulan piksel yang terkoneksi satu sama lain dan mempunyai sifat yang secara umum sama. Dalam citra ideal, sebuah daerah akan dibatasi dengan kurva tertutup, artinya objek yang berada didalam citra itu tampil utuh, tidak terpotong atau menyentuh tepi bingkai citra. Pada prinsipnya, segmentasi daerah dan deteksi tepi membuahkan hasil yang sama, yaitu menjadi pusat perhatian dalam menginterpretasikan suatu citra[23].

2.2.7.1 Segmentasi Warna HSV

Segmentasi warna merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Segmentasi citra dengan deteksi warna *HSV* menurut Gunanto (2009) menggunakan dasar seleksi warnapada model warna *HSV* dengannilai toleransi tertentu.

Pada metode segmentasi dengan deteksi citra warna *HSV* menurut Giannakopoulos (2008), dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Citra digital menggunakan model warna *RGB* sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna *RGB* ke *HSV*. Untuk membentuk segmen sesuai dengan warna yang diinginkan

maka ditentukan nilai toleransi pada setiap dimensi warna *HSV* kemudian nilai toleransi pada setiap dimensi warna *HSV*, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses *adaptive threshold*. Hasil dari proses *threshold* tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan. Proses segmentasi menurut Giannakopoulos (2008).

1. Tentukan citra *HSV* yang menjadi objek deteksi nilai warna *HSV* yang menjadi acuan (hasil proses peatihan data) dan nilai toleransi *HSV* yang digunakan.
2. Transpose citra *RGB* ke *HSV*.
3. Lakukan filter warna *HSV* pada citra berdasarkan nilai acuan (T) dan nilai toleransi(tol). Dengan x sebagai warna *HSV* pada piksel yang ada maka warna yang tidak termasuk dalam rentang $T-tol < x < T+tol$ diberi warna hitam.
4. *Transpose* Kembali citra ke *RGB*, tampilkan hasil filter.

2.2.8 Ekstraksi Fitur

Fitur adalah objek yang memiliki karakteristik atau sifat. Ekstraksi fitur merupakan proses pengambilan nilai fitur atau ciri dari sebuah gambar[24]. Ekstraksi fitur dilakukan dengan cara menghitung jumlah titik atau *pixels* yang ditemui dalam setiap pengecekan, dimana pengecekan dilakukan dalam berbagai arah *tracing* pengecekan pada koordinat kartesian dari citra digital yang dianalisis, yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri[25]. Ekstraksi warna dilakukan perhitungan kemunculan masing-masing warna pada piksel agar didapatkan warna apa saja yang mendominasi citra. Pada ekstraksi fitur tekstur secara statistik, dihitung berdasarkan distribusi statistik dari kombinasi intensitas piksel pada posisi tertentu *relative* terhadap lainnya dalam suatu matriks citra. Bergantung pada jumlah piksel atau titik intensitas dalam masing-masing kombinasi[jurnal19]. Ekstraksi bentuk yang dimaksud dengan tepi mencirikan batas-batas objek yang berguna untuk proses identifikasi objek didalam citra. Tujuannya adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek didalam citra[jurnal 22].

2.2.9 Ekstraksi Fitur Warna HSV

HSV merupakan salah satu cara untuk mendefinisikan warna yang didasarkan pada roda warna. *Hue* mengukur sudut sekitar roda warna (merah pada 0° , 120° di hijau, biru di 240°). *Saturation* yang menunjukkan pada roda warna sehingga menunjukkan proporsi antara gelap (pusat) untuk warna keputih murni (di luar). *Value* menunjukkan nilai kecerahan. *Hue* memiliki nilai antara 0° hingga 360° [26]. Nilai HSV yang dijadikan tolak ukur penelitian ini dilakukan tahap normalisasi kedalam angka yang lebih sederhana. Hal ini bertujuan untuk mempersingkat waktu komputasi, namun tidak mengurangi akurasi pencarian secara signifikan. Kuantisasi dilakukan dengan persamaan (2.10) dibawah ini [26] :

$$\begin{array}{l}
 H \left\{ \begin{array}{l} 0ifh \in [316, 20] \\ 1ifh \in [21, 40] \\ 2ifh \in [41, 75] \\ 3ifh \in [76, 155] \\ 4ifh \in [156, 190] \\ 5ifh \in [191, 270] \\ 6ifh \in [271, 295] \\ 7ifh \in [296, 315] \end{array} \right. \quad \begin{array}{l} S \left\{ \begin{array}{l} 0ifs \in [0, 0, 2] \\ 1ifs \in [0, 2, 0, 7] \\ 2ifs \in [0, 7, 1] \end{array} \right. \\ V \left\{ \begin{array}{l} 0ifv \in [0, 0, 2] \\ 1ifv \in [0, 2, 0, 7] \\ 2ifv \in [0, 7, 1] \end{array} \right. \end{array} \quad (2.10)
 \end{array}$$

2.2.10 Deteksi Tepi

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua daerah pada suatu citra [9]. Dalam analisis deteksi sebuah citra/gambar. Pada kondisi awal sebuah citra harus disederhanakan kedalam bentuk yang lebih sedikit keragamannya. Kemudian langkah berikutnya adalah citra diolah atau dibentuk suatu pola yang lebih mudah untuk dikenali dibandingkan dengan citra sebelum dilakukan proses penyederhanaan. Metode deteksi merupakan suatu metode yang digunakan dengan menghitung nilai-nilai piksel

dari pusat daerah, dengan mengevaluasi beberapa piksel disekeliling suatu citra. Pada objek dimensi satu perubahan dapat diukur dengan menggunakan fungsi turunan (*derivative function*), perubahan mencapai maksimal pada saat nilai turunan pertamanya mencapai nilai maksimal atau nilai turunan keduanya (*2ndderivative*) bernilai 0. Deteksi tepi (*edge detction*) adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berada. Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek citra, tujuannya adalah :

1. Untuk menandai bagian yang menjadi detail citra.
2. Untuk memperbaiki detail dari citra yang kabur terjadi karena *error*.
3. Adanya efek dari proses akuisisi.
4. Untuk mengubah citar 2D menjadi bentuk kurva titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

2.2.10.1 Deteksi Tepi *Canny*

Metode *canny* merupakan salah satu algortima deteksi tepi *modern*. Metode *canny* ini tidak memiliki operator khusus, Metode *Canny* akan mendeteksi tepi dengan mencari gradien maksimumlokal dari citra. Gradien dihitung menggunakan turunan dari filter *gaussian*. Metode *Canny* menggunakan dua ambang, yang berguna untuk mendeteksi tepi yang kabur, tepi yang kabur atau lemah, dan tepi yang kabur terlihat pada *output* yang terhubung ke tepi yang jelas. Metode ini berbeda dengan metode sebelumnya, sebelum proses deteksi tepi harus dilakukan proses penghilangan *noise*. Metode *Canny* yang lebih lama akan mendeteksi tepi yang kurang terdefinisi, yang tidak dapat diperoleh dengan metode lain[23].

Berikut adalah langkah-langkah dalam melkaukan deteksi tepi *Canny* [1]:

1. Menghilangkan noise yang ada pada citra dengan mengimplemtasikan filter *Gaussian*. Hasilnya citra akan tampak sedikit buram. Hal ini dimaksudkan

untuk mendapatkan tepian citra sebenarnya. Bila tidak dilakukan maka garis-garis halus juga akan kedeteksi sebagai tepian.

2. Melakukan deteksi tepi dengan salah satu operator deteksi tepi *canny* dengan melakukan pencarian secara horizontal (G_x) dan secara vertikal (G_y). Hasil dari kedua operator digabungkan untuk mendapatkan hasil gabungan tepi horizontal dan vertikal dengan rumus persamaan(2.11) :

$$|G| = |G_x| + |G_y| \quad (2.11)$$

3. Menentukan arah tepian yang ditentukan dengan menggunakan rumus persamaan(2.12).

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.12)$$

4. Memperkecil garis tepi yang muncul dengan menerapkan *non maximum suppression* sehingga menghasilkan garis tepian yang lebih ramping.
5. Binerisasi dengan menerapkan dua nilai ambang.

2.2.11 Gray Level Co-Occurrence (GLCM)

Metode *GLCM* merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi. Metode ini juga untuk tabulasi tentang frekuensi kombinasi nilai piksel yang muncul pada suatu citra. Untuk melakukan analisis citra berdasarkan distribusi statistika dari intensitas pikselnya, dapat dilakukan dengan mengekstraksi fitur tekstur[17].

Ekstraksi fitur *GLCM*, yaitu *contrast*, *homogeneity*, *energy*, dan *correlation*[6] :

1. Pembentukan matriks awal *GLCM* dari pasangan dua piksel yang berjajar sesuai dengan arah 0°
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal *GLCM* dengan nilai transposnya.

3. Menormalisasi Matriks *GLCM* dengan membagi elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri, yaitu *contrast, homogeneity, energy, correlation*.

a. *Contrast*

Nilai kontras menunjukkan besar kecilnya perpotongan (momen inersia) elemen matriks citra. Jika jauh dari diagonal utama, nilai kontras akan lebih besar. Secara visual, nilai kekontrasan merupakan ukuran perubahan antara tingkat keabuan pada suatu area citra.[17]. Perhitungan kontras dapat dihitung dengan persamaan(2.13) , dimana P_{ij} adalah probabilitas suatu *pixel* dari tingkat keabuan pada i ke tingkat keabuan pada j [16].

$$f1 = \sum_{i=1}^M \sum_{j=1}^N (i-j)^2 P_{ij} \quad (2.13)$$

Keterangan :

M : banyaknya baris pada array citra

N : banyaknya kolom pada array citra

P : matriks *GLCM* normalisasi.

i : indeks baris matriks P.

j : indeks kolom matriks P.

b. *Homogeneity*

Homogeneity merupakan tingkat kehomogenan suatu piksel dengan piksel lain dan dapat dihitung dengan menggunakan persamaan(14), dimana P_{ij} merupakan penyebaran probabilitas *pixel* dengan tingkat keabuan pada i hingga tingkat keabuan pada j [16].

$$f2 = \sum_{i=1}^M \sum_{j=1}^N \frac{P_{ij}}{1+|i-j|} \quad (2.14)$$

Keterangan :

M : banyaknya baris pada array citra

N : banyaknya kolom pada array citra

P : matriks GLCM normalisasi

i : indeks baris matriks P

j : indek kolom matriks P

c. *Energy*

Energy adalah hasil penjumlahan elemen *co-occurrence matrix* yang dapat dihitung dengan menggunakan persamaan(2.15), dimana P_{ij} merupakan penyebaran probabilitas pixel dengan tingkat keabuan pada i hingga tingat pada j [16].

$$f_3 = \sum_{i=1}^M \sum_{j=1}^N P_{ij}^2 \quad (2.15)$$

Keterangan :

M : banyaknya baris pada array citra

N : banyaknya kolom pada array citra

P : matriks GLCM normalisasi

i : indek baris matriks P

j : indeks kolom matriks P

2.2.12 *Machine Learning*

Secara definisi *machine learning* adalah cabang dari ilmu kecerdasan buatan yang berfokus pada pembangunan dan studi sebuah sistem agar mampu belajar dari data-data yang diperolehnya. Menurut Arthur Samuel, *machine learning* adalah bidang studi yang memberikan kemampuan program komputer untuk belajar tanpa secara eksplisit diprogram. Untuk bisa mengaplikasikan Teknik-teknik *machine learning* maka harus ada data. Tanpa data maka algoritma *machine learning* tidak dapat bekerja. Data yang ada biasanya dibagi menjadi dua, yaitu *training data* dan *testing data*. Adapun data latih digunakan sebagai data untuk melatih algoritma sedangkan data uji digunakan untuk mengetahui kinerja dari model yang telah dilakukan *training* terhadap data baru yang berbeda denga data latih [1].

2.2.12.1 *K-Nearest Neighbor (K-NN)*

Algoritma K-Nearest Neighbor (K-NN) merupakan metode yang menggunakan algoritma supervised learning. Perbedaan antara pembelajaran terawasi dan pembelajaran tidak terawasi adalah bahwa pembelajaran terawasi bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang ada dengan data baru. Dalam unsupervised learning, data belum memiliki pola. Tujuan unsupervised learning adalah untuk menemukan pola dalam data. Tujuan dari algoritma K-NN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel pelatihan. Hasil sampel uji baru diklasifikasikan menurut sebagian besar kategori di K-NN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma K-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Jarak yang digunakan adalah jarak *euclidean distance*. Jarak *euclidean distance* adalah jarak yang paling umum digunakan pada data numerik.

$$D_{ij} = \sqrt{\sum_{k=1}^n (X_i - Y_j)^2} \quad (2.16)$$

Keterangan :

D_{ij} : jarak Eulidean antara I dengan j

X_i : data pada x ke-i yang akan dilakukan tahap perhitungan.

y_j : data pada y ke-j yang akan dilakukan tahap perhitungan

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor* antara lain[27] :

1. Menentukan parameter K
2. Menghitung jarak antara data yang akan dievaluasi dengan semua pelatihan
3. Mengurutkan jarak yang terbentuk
4. Menentukan jarak terdekat sampai urutan K
5. Memasangkan kelas yang bersesuaian

6. Mencari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

Pada umumnya, rasio pembagian *dataset* (*training: validation: testing*) adalah (80%: 10%: 10%) atau (90%: 5%: 5%). *Validation set* pada umumnya bisa tidak digunakan apabila *dataset* berukuran kecil (hanya dibagi menjadi *training* dan *testing set* saja). Dalam kasus ini, pembagian *dataset* menjadi *training* dan *testing set* pada umumnya memiliki rasio (90% : 10%), (80% : 20%), (70% : 30%), atau (50% : 50%)[28]. Dalam menentukan nilai *K* sebaiknya dilihat dari jumlah klasifikasi bila jumlahnya genap maka sebaiknya menggunakan nilai *K* yang ganjil, dan sebaliknya jika jumlah klasifikasi jumlahnya ganjil maka sebaiknya dalam menggunakan nilai *K* yang genap, karena jika tidak begitu maka sistem kemungkinan tidak akan mendapatkan jawaban[29].

2.2.12.2 Indeks Pengukuran

Indeks pengukuran bertujuan untuk mengetahui tingkat kesuksesan terhadap proses-proses yang telah dilakukan[30]. Perhitungan akurasi dilakukan untuk menguji tingkat keakuratan pada metode yang diteliti untuk membandingkan hasil pemrosesan dengan target keluaran sesungguhnya. Nilai akurasi dari hasil klasifikasi dapat diperoleh dengan menghitung jumlah klasifikasi yang benar dan sesuai target dibagi dengan jumlah klasifikasi yang berbeda dengan target dari semua kelas. Nilai-nilai tersebut disajikan dalam bentuk *confusion matrix*[30].

Tabel 2.2 *Confusion matrix*

		Actual values	
		True	False
Predict Values	True	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	False	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

Perhitungan nilai akurasi, *recall*, dan *precision* digunakan persamaan seperti berikut:

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (2.17)$$

$$Recall = \frac{(TP)}{(TP + FN)} \times 100\% \quad (2.18)$$

$$Precision = \frac{(TP)}{(TP + FP)} \times 100\% \quad (2.19)$$

Keterangan :

TP (*true positive*) : jumlah data benar pada target yang terklasifikasi salah pada sistem

TN (*true negative*) : jumlah data salah pada target yang terklasifikasi benar pada sistem

FN (*false negative*) : representasi data salah pada target yang terklasifikasi benar pada sistem

FP (*false positive*) : representasi data benar pada target yang terklasifikasi salah pada sistem