

BAB 2

DASAR TEORI

2.1. KAJIAN PUSTAKA

Penelitian [11] membahas sebuah sistem otomatisasi komputer menggunakan teknik *machine learning* dan *deep learning* untuk mendeteksi penyakit pada daun jeruk. Gambar daun jeruk diambil menggunakan *smartphone* dan diproses untuk *resize* gambar, menghilangkan *noise*, meningkatkan kualitas gambar, dan ekstraksi fitur. Deteksi dan klasifikasi dilakukan menggunakan *K-Nearest Neighbor (KNN)* dan *Deep Neural Network (DNN)* sebagai *classifier*. Sistem yang diusulkan mencapai akurasi tinggi dalam mendeteksi penyakit pada jeruk dan dapat membantu petani mendeteksi penyakit pada tahap awal. Algoritma KNN menggunakan metrik jarak untuk mengklasifikasikan data baru berdasarkan set data latihan, sedangkan algoritma DNN menggunakan jaringan saraf multilayer dengan *dropouts*. Kinerja kedua algoritma dievaluasi menggunakan metrik akurasi dan *error*. Hasil penelitian menunjukkan bahwa sistem yang diusulkan mencapai akurasi tinggi dalam mendeteksi penyakit pada daun jeruk. Algoritma KNN mencapai akurasi antara 84,3% hingga 94,4% dengan variasi nilai K, sedangkan algoritma DNN mencapai akurasi pengujian terbaik sebesar 98,89%. Hal ini menunjukkan bahwa kedua algoritma memiliki potensi yang baik dalam klasifikasi penyakit pada jeruk.

Penelitian [12] menggunakan metode *K-Nearest Neighbor (KNN)* untuk mengklasifikasikan buah jeruk segar dan busuk berdasarkan ekstraksi fitur warna dari model RGB (*Red, Green, Blue*) dan HSV (*Hue, Saturation, Value*). Prosesnya melibatkan segmentasi citra, ekstraksi ciri, dan penerapan model KNN untuk mengklasifikasikan buah jeruk berdasarkan citra warna kulit buah jeruk. Selain itu, hasil pengujian akurasi menunjukkan bahwa sistem ini cukup baik dalam mengklasifikasikan buah jeruk segar dan busuk, mencapai akurasi sebesar 88.95%. Metode normalisasi RGB dan perhitungan nilai HSV juga digunakan dalam penelitian ini. Algoritma klasifikasi yang diadopsi adalah KNN, dengan nilai k divariasikan antara 1 hingga 7 untuk mengevaluasi pengaruh nilai k terhadap proses

klasifikasi. Penelitian ini memberikan kontribusi dalam pengembangan sistem pengklasifikasian buah berdasarkan kematangannya menggunakan metode KNN, dengan hasil akurasi yang tinggi, menunjukkan pendekatan yang efektif dalam identifikasi kematangan buah jeruk berdasarkan warna kulit. Terbukti bahwa metode KNN efektif dalam klasifikasi buah berdasarkan citra warna, dan penelitian ini memberikan wawasan tentang pengaruh parameter k dalam proses klasifikasi menggunakan KNN..

Pada [13] membahas mengenai jeruk yang merupakan salah satu jenis buah yang sangat disukai oleh masyarakat karena kandungan nutrisinya yang meliputi vitamin C, vitamin B6, antioksidan, dan serat. Tingginya tingkat konsumsi dan harga jeruk membuat para penjual jeruk kadang-kadang melakukan praktik yang tidak seharusnya. Hal ini terjadi karena kurangnya pengetahuan pembeli dalam membedakan jeruk yang baik dan buruk. Salah satu cara dalam bidang informatika untuk mengenali jeruk yang baik dan buruk adalah dengan menggunakan pengolahan citra. Tujuan dari penelitian ini adalah untuk mengklasifikasikan jeruk lokal berdasarkan bentuk dan teksturnya. Metode yang digunakan dalam penelitian ini adalah metode GLCM (*gray level co-occurrence matrices*) dengan fitur, energi, korelasi, kontras, dan homogenitas untuk mengekstraksi ciri tekstur. Selain itu, metode geometri juga digunakan untuk mengekstraksi ciri bentuk dengan fitur eksentrisitas dan matriks. Untuk menentukan kedekatan antara citra uji dengan citra latih, digunakan metode *K-Nearest Neighbor* berdasarkan fitur tekstur dan bentuk yang diperoleh. Fitur tekstur diperoleh dengan mengambil sudut 0° , 45° , 90° , dan 135° , sedangkan fitur bentuk diperoleh berdasarkan luas dan keliling objek. Nilai k yang digunakan dalam penelitian ini bervariasi dari 1 hingga 5, dan menghasilkan tingkat akurasi tertinggi pada uji dengan $k=1$ sebesar 93,33%, sedangkan akurasi terendah sebesar 86,20% pada uji dengan $k=7$ dan $k=8$.

Pada [14] membahas tentang pengembangan sistem deteksi kualitas buah jeruk menggunakan pemrosesan citra dan klasifikasi SVM (*Support Vector Machine*). Masalah yang diangkat adalah bagaimana melakukan penilaian kualitas buah jeruk secara *non-destruktif* dan otomatis. Penelitian ini memberikan kontribusi yang berharga dalam pengembangan sistem deteksi kualitas buah menggunakan pemrosesan citra dan klasifikasi SVM (*Support Vector Machine*). Dalam upaya

mencapai tujuannya, penelitian ini menggabungkan beberapa metode, termasuk pra-pemrosesan citra dengan Median Filter, segmentasi buah jeruk menggunakan pendekatan hibrida yang menggabungkan metode warna dan *K-means*, serta ekstraksi fitur-fitur yang mencakup warna, bentuk, dan tekstur. Hasil penelitian menunjukkan tingkat akurasi sebesar 67,74% dengan menggabungkan fitur-fitur moment warna, GLCM, dan bentuk.

Sedangkan pada [15] membahas perbandingan metode klasifikasi buah jeruk menggunakan *K-Nearest Neighbor* (KNN) dan *Convolutional Neural Network* (CNN) untuk menentukan tingkat kematangan buah berdasarkan warna kulitnya. Data dikumpulkan melalui studi literatur untuk memperoleh citra buah jeruk dengan perbedaan warna kulit, dan metodologi penelitian meliputi langkah-langkah pengumpulan dataset, pengolahan citra, dan analisis hasil untuk menentukan tingkat akurasi yang lebih unggul. Hasil klasifikasi menunjukkan bahwa algoritma CNN memiliki tingkat akurasi yang lebih tinggi daripada KNN, dengan KNN mencapai 80%, sedangkan CNN mencapai 92%. Metode KNN mendekati objek yang diuji berdasarkan penentuan label yang mendominasi pada kelas tertentu di lingkungan data, sementara CNN melakukan proses klasifikasi dari sampel data uji dengan memanfaatkan konvolusi untuk mengekstraksi fitur-fitur yang relevan. Parameter yang digunakan dalam CNN mungkin mencakup jumlah layer, ukuran filter, jumlah filter, fungsi aktivasi, dan learning rate. Penelitian ini juga mengimplementasikan aplikasi deteksi jeruk dengan fitur kamera, tutorial, dan data, yang dapat menjadi solusi praktis dalam industri pertanian dan pengolahan buah. Dengan hasil akurasi yang tinggi, implementasi metode CNN dalam aplikasi deteksi jeruk dapat memberikan hasil klasifikasi yang akurat dan bermanfaat dalam pemilihan buah jeruk berdasarkan tingkat kematangannya.

Dalam penelitian ini, judul yang diambil menunjukkan keterkaitan dengan sejumlah penelitian terdahulu yang secara langsung maupun tidak langsung relevan dengan topik ini. Literatur-literatur tersebut, yang tercantum dalam Tabel 2.1, bukan hanya menjadi landasan teoritis tetapi juga memberikan dukungan empiris yang signifikan. Tabel tersebut merangkum kajian penelitian terdahulu, menyoroti berbagai aspek terkait metode pemanenan dan klasifikasi buah. Dengan demikian, penelitian ini memperoleh kerangka kerja yang kokoh dan terinformasikan.

Tabel 2. 1. Literatur review

No	Judul, Peneliti, Tahun Terbit	Masalah, Solusi, Metode	Perbedaan Dengan Penelitian Yang Akan Dilakukan
1.	<p>Judul : <i>Smartphone assist deep neural network to detect the citrus diseases in Agri-informatics</i></p> <p>Peneliti : Utpal Barman, Ridip Dev Choudhury</p> <p>Tahun Terbit : 2021</p>	<p>Masalah : petani kesulitan dalam mendeteksi penyakit pada daun jeruk secara manual hanya berdasarkan gejala visual eksternal. Hal ini dapat menyebabkan kesalahan identifikasi dan penanganan yang tidak tepat, yang pada akhirnya dapat mengurangi produktivitas dan kualitas buah jeruk.</p> <p>Metode : <i>K-Nearest Neighbor</i>, dan <i>Deep Neural Network</i></p> <p>Solusi : penelitian ini adalah pengembangan sistem deteksi penyakit daun jeruk secara real-time menggunakan metode akuisisi citra dengan smartphone dan algoritma klasifikasi KNN dan DNN, dengan dataset total 300 sampel dan didapatkan akurasi <i>K-Nearest Neighbor (KNN)</i> mencapai akurasi antara 84,3% hingga 94,4% dengan nilai K yang berbeda-beda . Sedangkan metode <i>Deep Neural Network (DNN)</i> mencapai akurasi pengujian terbaik sebesar 98,89%</p>	<p>penelitian yang akan dilakukan, melakukan variasi model yang lebih luas Dengan mengimplementasikan algoritma <i>Distance Weighted K-Nearest Neighbors (DWKNN)</i> dengan ekstraksi fitur <i>Gray Level Co-occurrence Matrix (GLCM)</i>, Dengan melibatkan dataset sebanyak 3852 gambar jeruk, serta dilakukan pengukuran standar deviasi pada model guna mengetahui seberapa konsisten model tersebut melakukan prediksi pada data yang tidak dikenal.</p>
2	<p>Judul : <i>Klasifikasi Buah Jeruk Segar dan Busuk Berdasarkan RGB dan HSV Menggunakan Metode KNN</i></p> <p>Peneliti : Stifani Napitu, Rini Paramita Panjaitan, Putri Aisyah Nulhakim, Muaz Khalik Lubis</p> <p>Tahun Terbit : 2023</p>	<p>Masalah : Buah-buahan di Indonesia merupakan komoditas pertanian dengan tingkat permintaan tinggi karena kaya akan vitamin. Permintaan domestik terhadap buah-buahan, seperti jeruk, apel, dan pisang, mencerminkan popularitasnya di pasar modern dan tradisional. Tingkat produksi dan distribusi yang luas menuntut petani untuk memahami tingkat kesegaran dan risiko pembusukan agar dapat memenuhi permintaan pasar.</p> <p>Metode : <i>K-Nearest Neighbor</i> dengan ekstraksi fitur warna RGB dan HSV</p> <p>Solusi : penggunaan metode <i>K-Nearest Neighbor (KNN)</i> untuk mengklasifikasikan buah jeruk segar dan busuk berdasarkan ekstraksi fitur warna RGB dan HSV. Dataset yang digunakan terdiri dari 146 data <i>training</i> dan 88 data <i>testing</i>. Hasil akurasi dari pengujian</p>	<p>Penelitian ini menerapkan algoritma <i>Distance Weighted K-Nearest Neighbors (DWKNN)</i> untuk memberikan pendekatan klasifikasi yang lebih dinamis dan responsif terhadap perbedaan bobot tetangga berdasarkan jarak. Selain itu, pengintegrasian DWKNN dengan ekstraksi fitur menggunakan <i>Gray Level Co-occurrence Matrix (GLCM)</i>, menekankan fokus pada karakteristik tekstur buah jeruk yang menciptakan model yang adaptif terhadap kompleksitas variasi kulit buah. Dalam evaluasi hasil, penelitian ini melibatkan standar deviasi sebagai metrik tambahan,</p>

No	Judul, Peneliti, Tahun Terbit	Masalah, Solusi, Metode	Perbedaan Dengan Penelitian Yang Akan Dilakukan
		menunjukkan bahwa sistem ini cukup baik dalam mengklasifikasikan buah jeruk segar dan busuk, dengan akurasi sebesar 88.95%. Metode normalisasi RGB dan perhitungan nilai HSV juga digunakan dalam penelitian ini.	memberikan wawasan tentang seberapa konsisten dan dapat diandalkan model DWKNN dalam berbagai situasi.
3	<p>Judul : Klasifikasi Kualitas Jeruk Lokal Berdasarkan Tekstur Dan Bentuk Menggunakan Metode <i>K-Nearest Neighbor (K-NN)</i></p> <p>Peneliti : Ratri Enggar Pawening, Wali Ja'far Shudiq, Wahyuni</p> <p>Tahun Terbit : 2020</p>	<p>Masalah : Jeruk merupakan salah satu tanaman jenis buah – buahan yang banyak digemari oleh masyarakat, dikarenakan buah jeruk mengandung banyak vitamin C, vitamin B6, Antioksidan dan serat, tingkat konsumsi dan harga yang tinggi banyak dimanfaatkan penjual jeruk untuk berbuat yang tidak seharusnya. Hal ini terjadi karena kurangnya pengetahuan pembeli dalam membedakan jeruk baik dan buruk</p> <p>Metode : GLCM (<i>gray level co – occurrence matrices</i>) dengan fitur <i>energy, Correlation, Contrast, Homogeneity</i> untuk ekstraksi ciri tekstur dan metode geometri untuk ekstraksi ciri bentuk dengan fitur <i>Eccentricity</i> dan <i>matric</i>. Sedangkan untuk menentukan kedekatan antara citra uji dengan citra latih menggunakan metode <i>K –Nearest Neighbor</i> berdasarkan fitur tekstur dan bentuk yang diperoleh</p> <p>Solusi : Fitur fitur tekstur dicari dengan berdasarkan sudut 0°, 45°, 90°, 135°, dan Fitur – fitur bentuk dicari berdasarkan Luas dan Keliling Objek, nilai k yang digunakan dalam penelitian ini adalah 1 sampai 5, yang menghasilkan tingkat akurasi tertinggi pada uji k1 sebesar 93,33% dan akurasi terendah sebesar 86,20% pada uji k7 dan k8.</p>	pada penelitian ini menggunakan algoritma <i>Distance Weighted K-Nearest Neighbors (DWKNN)</i> yang memberikan kerangka kerja klasifikasi yang lebih dinamis, selain itu pada penelitian ini juga menggunakan ekstraksi fitur <i>Gray Level Co-occurrence Matrix (GLCM)</i> dengan fitur yang lebih banyak sehingga memberikan dimensi analisis yang lebih luas terkait dengan tekstur jeruk. penelitian ini lebih difokuskan pada penilaian kualitas jeruk secara menyeluruh. Penggunaan DWKNN dengan ekstraksi fitur GLCM dalam penelitian saya memungkinkan analisis yang lebih holistik terkait dengan variasi kualitas jeruk. Pengukuran standar deviasi diterapkan pada model untuk menilai tingkat konsistensinya dalam melakukan prediksi pada data yang tidak dikenal
4	<p>Judul : <i>Detection of Quality in Orange Fruit Image using SVM ClassifierAssessment</i></p> <p>Peneliti : Hardik Patel, Rashmin Prajapati, Milin Patel</p>	Masalah : deteksi kualitas buah jeruk menggunakan pemrosesan citra. Tujuan dari penelitian ini adalah untuk menyediakan metode non-destruktif dan otomatis dalam menilai kualitas buah jeruk	dalam penelitian ini, fokus utama adalah pada penilaian kualitas jeruk menggunakan algoritma DWKNN dengan ekstraksi fitur GLCM. Pendekatan penelitian ini memberikan

No	Judul, Peneliti, Tahun Terbit	Masalah, Solusi, Metode	Perbedaan Dengan Penelitian Yang Akan Dilakukan
	Tahun Terbit : 2019	<p>Metode : <i>Gray Level Co-occurrence Matrix</i> (GLCM) ,<i>Support Vector Machine</i> (SVM)</p> <p>Solusi : dalam penelitian ini adalah pengembangan sistem deteksi kualitas buah jeruk menggunakan pemrosesan citra dan klasifikasi SVM, dengan dataset sebanyak 5 terdiri dari macam-macam penyakit buah jeruk. Penelitian ini menunjukkan bahwa pendekatan yang diusulkan mencapai akurasi sebesar 67,74% dengan menggunakan kombinasi fitur moment warna, GLCM, dan bentuk</p>	keunggulan dengan fokus lebih spesifik. Pada fitur tekstur dan penggunaan algoritma klasifikasi yang berbeda, dan juga melibatkan dataset sebanyak 3852 gambar jeruk, menunjukkan upaya untuk mencakup variasi yang lebih luas dan merangsang landasan empiris yang lebih kuat. Jumlah data yang lebih besar ini dapat meningkatkan reliabilitas temuan serta memberikan pemahaman yang lebih mendalam tentang kompleksitas kualitas jeruk
5.	<p>Judul : membandingkan klasifikasi pada buah jeruk menggunakan metode <i>convolutional neural network</i> dan <i>k-nearest neighbor</i></p> <p>Peneliti : Kinanthi Putri Siwilopo , Hendra Marcos</p> <p>Tahun Terbit : 2023</p>	<p>Masalah : buah jeruk menjadi salah satu komoditas terbesar di Indonesia. Salah satunya jenis jeruk keprok yang memiliki kandungan vitamin C yang banyak. Namun terkadang konsumen masih kesulitan dalam memilih antara buah jeruk yang matang dan belum matang secara manual.</p> <p>Metode : <i>K-Nearest Neighbor</i> & <i>Convolutional Neural Network</i></p> <p>Solusi : Hasil penelitian menunjukkan bahwa CNN memiliki tingkat akurasi yang lebih tinggi daripada metode <i>K-Nearest Neighbor</i> (KNN) dalam mengklasifikasikan buah jeruk berdasarkan warna kulitnya. Dalam penelitian ini, hasil akurasi klasifikasi pada KNN mencapai 80%, sedangkan CNN mencapai 92% .Selain itu, penelitian ini juga mengimplementasikan aplikasi deteksi jeruk dengan fitur kamera, tutorial, dan data, yang dapat menjadi solusi praktis dalam industri pertanian dan pengolahan buah.</p>	penelitian ini menitikberatkan pada penggunaan algoritma <i>Distance Weighted K-Nearest Neighbors</i> (DWKNN) dengan ekstraksi fitur <i>Gray Level Co-occurrence Matrix</i> (GLCM), yang mengarah pada penilaian kualitas jeruk dengan fokus khusus pada tekstur. Perbedaan kunci terletak pada inklusi standar deviasi sebagai metrik tambahan dalam evaluasi hasil. Penggunaan standar deviasi menjadi elemen penting karena memberikan gambaran lebih jelas tentang konsistensi dan keandalan model DWKNN dalam berbagai situasi.

2.2. DASAR TEORI

2.2.1. Jeruk

Jeruk merupakan salah satu jenis buah yang populer dan sering dikonsumsi di seluruh dunia. Buah jeruk memiliki bentuk bulat atau agak oval dengan kulit yang tebal dan berwarna cerah, seperti oranye, kuning, atau hijau. Dalam daging jeruk terdapat segmen-segmen yang berisi sari buah yang kaya akan vitamin C, serat, dan antioksidan ditampilkan pada gambar 2.1. [16].



Gambar 2. 1. Jeruk

Jeruk memiliki rasa yang segar dan asam yang khas, sehingga sering digunakan untuk minuman jus atau sebagai bahan tambahan dalam masakan. Jeruk juga memiliki manfaat kesehatan, seperti meningkatkan sistem kekebalan tubuh, menjaga kesehatan kulit, dan membantu dalam pencernaan. Jenis-jenis jeruk yang umum meliputi jeruk manis, jeruk mandarin, jeruk bali, dan jeruk nipis. Selain dikonsumsi langsung, jeruk juga digunakan dalam industri makanan dan minuman, serta dalam industri kosmetik dan aromaterapi[17].

Dalam penelitian ini, citra jeruk yang digunakan yaitu jeruk dengan kualitas baik dan layak yang bisa dikonsumsi oleh konsumen dan jeruk dengan kualitas yang tidak baik dan tidak layak untuk dikonsumsi. Sehingga, dari penelitian ini akan menghasilkan *output* berupa jenis kualitas jeruk berdasarkan warna dan tekstur yang akan di uji.

2.2.2. Pengolahan citra

Pengolahan citra adalah suatu proses pengambilan, manipulasi, analisis, dan ekstraksi informasi dari gambar atau citra digital. Tujuan dari pengolahan citra adalah untuk meningkatkan kualitas, kejelasan, dan pemahaman citra yang diperoleh melalui teknik komputasional[18]. Proses pengolahan citra melibatkan

serangkaian langkah atau teknik yang diterapkan pada citra digital. Berikut adalah beberapa tahapan umum dalam pengolahan citra:

1. *Preprocessing* (Pra-pemrosesan): Tahap ini melibatkan langkah-langkah awal untuk mempersiapkan citra sebelum dilakukan analisis lebih lanjut. Pra-pemrosesan meliputi operasi dasar seperti pemotongan (*cropping*), pengubahan ukuran (*resizing*), penajaman (*sharpening*), penghalusan (*smoothing*), dan penyesuaian kontras atau kecerahan.
2. Peningkatan Citra: Langkah ini bertujuan untuk meningkatkan kualitas visual citra dengan memperbaiki kontras, kecerahan, ketajaman, atau pengurangan derau. Metode yang umum digunakan dalam peningkatan citra meliputi penajaman histogram, ekualisasi histogram, filtrasi spasial, dan teknik pengurangan derau.
3. Segmentasi: Segmentasi citra adalah proses membagi citra menjadi wilayah-wilayah atau objek-objek yang berbeda berdasarkan karakteristik tertentu. Tujuan segmentasi adalah untuk mengidentifikasi objek atau fitur penting dalam citra. Metode yang digunakan untuk segmentasi citra meliputi metode berbasis intensitas (*thresholding*), pemisahan berdasarkan tepi (*edge-based*), pemisahan berdasarkan warna (*color-based*), dan pemisahan berdasarkan tekstur (*texture-based*).
4. Pengekstrakan Fitur: Pada tahap ini, fitur-fitur atau karakteristik-karakteristik penting dari citra diekstraksi untuk analisis lebih lanjut. Fitur-fitur ini dapat berupa bentuk, tekstur, warna, atau pola-pola yang dapat digunakan untuk pengenalan objek atau klasifikasi citra.
5. Pengenalan Pola: Langkah ini melibatkan pengenalan atau klasifikasi citra berdasarkan pola-pola atau fitur-fitur yang diekstraksi sebelumnya. Metode yang digunakan dalam pengenalan pola citra termasuk klasifikasi berbasis aturan, pengenalan berbasis pola (*pattern recognition*), pengenalan berbasis jaringan saraf tiruan (*neural networks*), atau pengenalan berbasis pembelajaran mesin (*machine learning*).
6. Analisis dan Interpretasi: Tahap akhir dalam pengolahan citra adalah analisis dan interpretasi hasil yang diperoleh dari tahapan sebelumnya. Ini melibatkan pemahaman dan ekstraksi informasi yang relevan dari citra,

yang dapat digunakan dalam berbagai aplikasi seperti penginderaan jauh, pengolahan medis, pengenalan wajah, survei satelit, dan banyak lagi. Pengolahan citra telah banyak memberikan kontribusi dalam berbagai bidang, termasuk ilmu kedokteran, industri, pertanian[19].

2.2.3. Preprocessing

Pra-pemrosesan (*preprocessing*) adalah tahap awal dalam pengolahan citra yang dilakukan sebelum analisis lebih lanjut. Tujuan dari pra-pemrosesan adalah untuk mempersiapkan citra agar dapat diolah secara efektif dan meningkatkan kualitas visual citra[20].

Berikut adalah beberapa langkah umum dalam pra-pemrosesan citra:

1. Pemotongan (*Cropping*): Pemotongan adalah proses menghapus bagian tidak relevan atau area yang tidak diperlukan dari citra. Hal ini dilakukan untuk fokus pada objek atau area yang penting dalam citra tersebut. Misalnya, menghilangkan bagian latar belakang atau area yang tidak relevan dalam citra medis untuk memfokuskan pada area tertentu yang perlu dianalisis.
2. Pengubahan Ukuran (*Resizing*): Pengubahan ukuran adalah proses mengubah dimensi citra, baik dalam hal resolusi atau skala, agar sesuai dengan kebutuhan analisis atau tampilan yang diinginkan. Misalnya, mengubah ukuran citra menjadi lebih kecil untuk mempercepat waktu komputasi atau mengubah ukuran citra menjadi lebih besar untuk detail visual yang lebih baik.
3. Penajaman (*Sharpening*): Penajaman adalah proses meningkatkan kejelasan atau ketajaman citra dengan memperkuat tepi dan detail dalam citra. Teknik penajaman umum melibatkan penerapan filter yang mengamplifikasi perbedaan intensitas antara piksel-piksel tetangga, sehingga membuat citra terlihat lebih tajam.
4. Penghalusan (*Smoothing*): Penghalusan adalah proses mengurangi derau atau ketidaksempurnaan dalam citra dengan meratakan intensitas piksel-piksel yang berdekatan. Metode penghalusan yang umum digunakan adalah filter rata-rata atau filter *Gaussian*. Penghalusan membantu menghilangkan derau atau ketidaksempurnaan yang dapat mengganggu analisis citra.

5. Penyesuaian Kontras dan Kecerahan: Penyesuaian kontras dan kecerahan melibatkan manipulasi intensitas piksel dalam citra untuk meningkatkan kejelasan dan kualitas visual. Hal ini dapat dilakukan dengan mengubah tingkat kontras secara global atau lokal pada citra. Penyesuaian kontras dan kecerahan membantu mencapai tampilan yang lebih optimal dan memperbaiki perbedaan intensitas yang kurang jelas dalam citra.
6. Konversi gambar ke *array*: Dalam mengolah data gambar, algoritma machine learning tidak dapat langsung mengolah data, melainkan harus dikonversi kedalam bentuk *array*, maka dari itu, setiap gambar yang akan diklasifikasi menggunakan *machine learning*, harus dilakukan konversi kedalam bentuk *array*.
7. Normalisasi : teknik ini digunakan untuk memperpendek jarak angka, sehingga, jika data yang kita olah memiliki nilai yang sangat besar, dengan memperpendek pendekatan nilai menjadi 0 sampai 1, maka akan meningkatkan performa penghitungan yang signifikan. Pada formula 2.1 adalah rumus yg digunakan untuk normalisasi.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

X = nilai sebenarnya

X_{min} = nilai terkecil pada data

X_{max} = nilai terbesar pada data

Preprocessing merupakan langkah penting dalam pengolahan citra karena mempersiapkan citra secara optimal sebelum dilakukan analisis lebih lanjut. Langkah-langkah pra-pemrosesan ini membantu meningkatkan kualitas, kejelasan, dan relevansi informasi dalam citra, sehingga memungkinkan analisis citra yang lebih efektif dan akurat[21].

2.2.4. Klasifikasi

Klasifikasi adalah salah satu tugas utama dalam bidang kecerdasan buatan yang melibatkan pengelompokan atau pengkategorian data ke dalam kelas-kelas yang telah ditentukan berdasarkan pola atau karakteristik tertentu. Proses klasifikasi bertujuan untuk membuat prediksi atau keputusan yang cerdas berdasarkan

informasi yang ada dalam data yang dianalisis, Oleh karena itu, klasifikasi berperan penting dalam memahami dan memanfaatkan potensi informasi yang terdapat dalam dataset untuk mendukung pengambilan keputusan yang lebih efektif [22].

2.2.5. Machine Learning

Machine learning adalah suatu cabang dari kecerdasan buatan yang fokus pada pengembangan algoritma dan model komputasional yang dapat belajar dan meningkatkan kinerjanya secara otomatis melalui pengalaman dan data yang diberikan. Tujuan utama dari *machine learning* adalah untuk mengembangkan sistem yang mampu mengenali pola, membuat prediksi, atau mengambil keputusan tanpa harus secara eksplisit diprogram secara khusus[22].

Pada dasarnya, *machine learning* memungkinkan komputer atau sistem untuk belajar dari data yang ada, mengidentifikasi pola-pola yang tersembunyi, dan membuat prediksi atau keputusan berdasarkan pola-pola ini. Hal ini dilakukan melalui proses *training*, di mana model *machine learning* diberikan data pelatihan yang berisi contoh-contoh yang diketahui labelnya (*supervised learning*) atau tidak diketahui labelnya (*unsupervised learning*)[23].

Selama proses training, model *machine learning* akan mengekstraksi fitur-fitur penting dari data dan menggunakan algoritma khusus untuk mengatur parameter atau bobotnya. Dengan demikian, model dapat belajar untuk mengenali pola-pola dan membuat prediksi atau keputusan pada data baru yang belum pernah dilihat sebelumnya[24].

2.2.6. Distance Weighted K-Nearest Neighbor

Distance Weighted K-Nearest Neighbors (DWKNN) adalah variasi dari algoritma KNN yang memperhitungkan bobot jarak antara titik data yang akan diklasifikasikan dengan tetangganya. Dalam KNN tradisional, setiap tetangga diberi bobot yang sama dalam menentukan label kelas. Namun, dalam DWKNN, bobot diberikan berdasarkan jarak antara titik data dan tetangganya[25].

Semakin dekat tetangga dengan titik data yang akan diklasifikasikan, bobot yang lebih besar diberikan ke tetangga tersebut, sedangkan tetangga yang lebih jauh diberi bobot yang lebih kecil. Dengan memberikan bobot berdasarkan jarak, *Distance Weighted K-Nearest Neighbors* dapat meningkatkan keakuratan klasifikasi dengan memberikan pengaruh yang lebih besar pada tetangga yang lebih

dekat. Hal ini berguna dalam situasi di mana tetangga yang lebih dekat dianggap lebih relevan dan memiliki informasi yang lebih signifikan dalam menentukan klasifikasi[25].

Algoritma DWKNN adalah sebuah metode klasifikasi yang berbasis pada jarak dan bobot. Pada DWKNN, setiap data pelatihan memiliki atribut serta label yang diketahui. Saat melakukan klasifikasi pada data yang belum diketahui labelnya, DWKNN mengidentifikasi *k-nearest neighbors* terdekat dari data tersebut berdasarkan jarak *Euclidean* atau jarak lain yang telah ditentukan. Kemudian, bobot diberikan kepada setiap tetangga berdasarkan jaraknya, di mana tetangga yang lebih dekat diberi bobot yang lebih tinggi. Bobot ini kemudian digunakan dalam proses voting untuk menentukan label dari data yang belum diketahui[26].

Metode DWKNN ini memiliki keunggulan dalam menangani ketidakhomogenan data dan pengaruh *outlier*. Dengan memberikan bobot pada setiap tetangga berdasarkan jaraknya, DWKNN dapat memberikan penekanan pada tetangga-tetangga yang lebih dekat dan mengurangi pengaruh tetangga yang lebih jauh. Hal ini memungkinkan DWKNN untuk memberikan klasifikasi yang lebih akurat dan adaptif terhadap karakteristik data yang kompleks. Selain itu, DWKNN juga dapat diaplikasikan pada berbagai jenis data dan masalah klasifikasi. Metode ini dapat digunakan dalam klasifikasi biner maupun multikelas, serta dapat diintegrasikan dengan berbagai metode ekstraksi fitur untuk meningkatkan kinerja klasifikasi[27].

Persamaan yang digunakan dalam algoritma DWKNN adalah sebagai berikut:

1. Menghitung jarak antara sampel yang akan diklasifikasikan dengan setiap sampel dalam dataset, yang ditampilkan pada formula (2.2):

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (2.2)$$

di mana x_i dan x_j adalah vektor fitur dari dua sampel yang akan dibandingkan, dan n adalah jumlah fitur.

2. Menghitung bobot jarak menggunakan *inverse square weighting* dihitung menggunakan persamaan (2.3) :

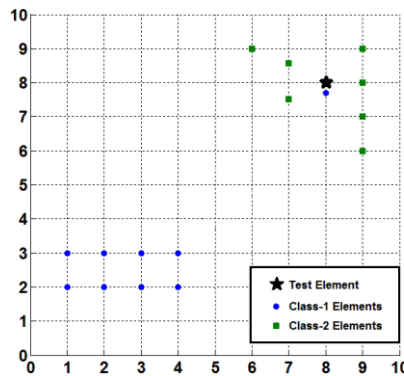
$$w(x_i, x_j) = \frac{1}{d(x_i, x_j)^2} \quad (2.3)$$

di mana $w(x_i, x_j)$ adalah bobot jarak antara dua sampel.

- Mengklasifikasikan sampel berdasarkan mayoritas kelas dari tetangga terdekat dengan mempertimbangkan bobot jarak yang dilakukan dengan menggunakan persamaan (2.4):

$$y(x_i) = \arg \max_c \sum_{x_j \in N_i} w(x_i, x_j) \cdot I(y(x_j) = c) \quad (2.4)$$

di mana $y(x_i)$ adalah label kelas yang diprediksi untuk sampel x_i , N_i adalah himpunan tetangga terdekat dari x_i , c adalah kelas yang mungkin, dan I adalah fungsi indikator yang mengembalikan 1 jika pernyataan di dalamnya benar, dan 0 jika salah[26].



Gambar 2. 2. Ilustrasi Algoritma DWKNN

Ada dua kelas dalam set pelatihan seperti yang terlihat pada contoh set data pada gambar 2.2 dan *outlier* dari salah satu kelas terlalu dekat dengan elemen uji. *Outlier* ini dapat menyebabkan elemen uji diklasifikasikan secara tidak akurat dalam metode klasifikasi yang menggunakan pembobotan berdasarkan jarak. Untuk menghindari hal ini, jarak elemen dalam himpunan pelatihan ke pusat kelas mereka digunakan dalam pembobotan ini. Namun, jika *outlier* berada sedekat mungkin dengan elemen uji, elemen tes, pembobotan berdasarkan jarak ganda ini tidak berguna. Selain itu, metode yang didasarkan pada pembobotan dengan jarak ganda memperpanjang waktu proses. Selain itu, satuan jarak dan pembobotan derajat harus dipilih dengan benar. Dengan kata lain, dalam dalam klasifikasi berbobot jarak, masalah seperti pemilihan unit jarak dan kebutuhan klasifikasi berdasarkan jarak yang terjadi. Bahkan jika semua pemilihan dilakukan dengan benar, tingkat

keberhasilannya mungkin rendah karena kedekatan yang ekstrim dari *outlier* dengan elemen uji[28].

DWKNN adalah modifikasi dari algoritma KNN yang memperkenalkan pembobotan jarak untuk meningkatkan kinerja dalam melakukan klasifikasi. Dalam algoritma KNN, klasifikasi dilakukan dengan mencari *k-nearest neighbors* (tetangga terdekat) dari data uji berdasarkan jarak *Euclidean* atau metrik jarak lainnya. Namun, algoritma KNN memiliki kelemahan yaitu tidak memberikan bobot yang berbeda pada setiap tetangga, sehingga tetangga yang lebih dekat dengan data uji memiliki pengaruh yang sama dengan tetangga yang lebih jauh[29].

DWKNN memperbaiki kelemahan ini dengan memberikan bobot yang berbeda pada setiap tetangga berdasarkan jaraknya dengan data uji. Semakin dekat tetangga dengan data uji, semakin besar bobot yang diberikan pada tetangga tersebut. Bobot diberikan dengan menggunakan fungsi *inverse distance weighting* (IDW) yang dapat diatur sesuai kebutuhan[30]. Tujuan dari pemberian bobot pada DWKNN adalah untuk meningkatkan akurasi dan kestabilan klasifikasi, terutama dalam kasus-kasus di mana data memiliki sebaran yang tidak merata atau tidak seimbang. Dengan adanya pembobotan jarak, DWKNN dapat mengenali pola dan struktur yang lebih kompleks dalam data, sehingga meningkatkan kemampuan untuk mengklasifikasikan data dengan lebih tepat[27].

Penerapan DWKNN dalam klasifikasi kualitas jeruk dengan ekstraksi fitur GLCM memiliki potensi untuk memberikan hasil yang lebih baik dalam mengenali kualitas jeruk berdasarkan citra. Dengan memanfaatkan informasi tekstur yang diekstraksi dari GLCM dan pemberian bobot jarak pada algoritma DWKNN, diharapkan penelitian ini dapat memberikan kontribusi positif dalam bidang klasifikasi produk pertanian, khususnya dalam meningkatkan efisiensi dan akurasi dalam memanen dan memilah kualitas jeruk[31].

2.2.7. Gray Level Co-occurrence Matrix

Ekstraksi ciri (*feature extraction*) adalah proses mengidentifikasi, memilih, atau menghasilkan fitur-fitur penting dari citra yang dapat digunakan dalam analisis dan pengenalan pola. Tujuan dari ekstraksi ciri adalah mengubah representasi citra menjadi representasi yang lebih informatif dan kompak[32].

GLCM adalah salah satu metode ekstraksi fitur yang umum digunakan dalam pemrosesan citra. GLCM digunakan untuk menggambarkan hubungan spasial antara pasangan piksel dalam citra *grayscale*. Keuntungan menggunakan GLCM termasuk kemampuannya untuk mengekstrak informasi tekstur yang kompleks dan penting dari citra, yang dapat digunakan untuk analisis dan pengenalan pola yang lebih akurat dengan merepresentasikan frekuensi kemunculan pasangan intensitas piksel tertentu dengan jarak dan arah tertentu. Dalam proses ekstraksi fitur GLCM, citra *grayscale* dibagi menjadi beberapa jendela piksel yang bergerak sepanjang citra[31].

Gambar 2.3. menunjukkan bahwa pergerakan arah GLCM pada 0° . Proses ini menghitung *array* GLCM dengan arah yang menunjukkan pergerakan kearah kanan. Secara *default* penghitungan GLCM diimplementasikan dengan menghitung dari 1 tetangga terdekat berdasarkan arahnya, sehingga perbedaan dari *array* GLCM yang ada, diawali dengan arah penghitungan yang disesuaikan dengan derajat atau arah GLCM.

3	5	7	$\rightarrow 0^\circ$
2	4	1	
6	3	5	

Gambar 2. 3. Array GLCM 0°

Pergerakan *array* kekanan dengan mengambil 1 tetangga terdekat, menghasilkan hanya ada kolom 1 dan 2 yang mempunyai tetangga, kolom 1 mempunyai tetangga kolom 2, dan kolom 2 mempunyai tetangga kolom 3, sedangkan kolom 3 tidak mempunyai tetangga lainnya, karena ukuran matrix 3×3 . Konsep ini diterapkan ke seluruh fitur GLCM, bedanya hanyalah arah dan penentuan jumlah tetangga terdekatnya.

Gambar 2.4. menunjukkan gerakan arah GLCM dengan sudut 45° . Proses ini menunjukkan gerakan yang mengarah diagonal ke pojok kanan atas *array*. Dengan mengambil 1 tetangga terdekat, maka penghitungan akan diambil berdasarkan baris 2 dan 3 pada kolom 1 dan 2.

3	5	7	$\nearrow 45^\circ$
2	4	1	
6	3	5	

Gambar 2. 4. Array GLCM 45°

Gambar 2.5. menunjukkan gerakan *array* GLCM 90°, proses ini mengarah kearah atas. Sehingga penghitungan *array* akan dimulai dari arah bawah ke atas, dengan tetangga nilai yang memiliki tetangga terdekat adalah baris 2 dan 3 pada seluruh kolom yang ada pada *array*.

90°

3	↑	7
2	↑	1
6	↑	5

Gambar 2. 5. Array GLCM 90°

Arah 135 menunjukkan gerak diagonal ke pojok kiri atas ditampilkan pada gambar 2.6. Tetangga terdekatnya adalah baris 2 kolom 2 dan 3, serta baris 3 kolom 2 dan 3. Dari seluruh derajat yang diimplementasikan pada penelitian ini, selanjutnya akan dibahas lebih mendalam pada tabel 4.1.

135°

3	5	7
2	4	1
6	3	5

Gambar 2. 6. Array GLCM 135°

Kemudian, untuk setiap jendela piksel, GLCM dihitung dengan menghitung frekuensi kemunculan pasangan intensitas piksel yang berdekatan berdasarkan jarak dan arah yang ditentukan. Dari GLCM, berbagai fitur statistik dapat dihitung, seperti energi, kontras, korelasi, dan homogenitas, yang dapat memberikan informasi tentang tekstur citra. Dengan menggunakan fitur-fitur ini, GLCM dapat membantu dalam menggambarkan karakteristik tekstur dan pola dalam citra, dan digunakan dalam berbagai aplikasi seperti pengenalan pola, deteksi objek, dan klasifikasi citra[33].

Secara matematis, untuk mendapatkan nilai piksel kookurensi seperti pada formula 2.5 dimana d adalah jarak antara dua piksel yaitu (x_1, y_1) dan (x_2, y_2) .

$$p = \sum_{x=1}^K \sum_{y=1}^K \begin{cases} 1, & \text{if } I(x, y) = i \text{ and } I(x + d_x, y + d_y) = j \\ 0, & \text{lainnya} \end{cases} \quad (2.5)$$

Untuk mendapatkan fitur statistik orde dua, kita dapat menghitung probabilitas hubungan tetangga antara dua piksel pada jarak dan orientasi sudut

tertentu. Setiap elemen dalam GLCM merepresentasikan hubungan antara dua piksel dengan tingkat keabuan i dan j pada jarak d dan orientasi sudut θ . Fungsi orientasi sudut digunakan untuk menentukan arah hubungan tetangga antara piksel-piksel dalam citra. Dalam contoh gambar 2.3 sampai gambar 2.6 menggambarkan orientasi sudut citra dengan jarak antarpiksel sebesar satu ($d = 1$) dan orientasi dalam empat arah yaitu horizontal (0°), diagonal (45° dan 135°), serta vertikal (90°)[34].

Setelah menghasilkan matriks *co-occurrence* berdasarkan jarak dan orientasi sudut, langkah selanjutnya adalah melakukan operasi penambahan matriks kookurensi dengan matriks transposenya untuk menciptakan simetri terhadap sumbu diagonal. Selanjutnya, setiap elemen dalam matriks hasil tersebut, yang direpresentasikan oleh $P(i, j)$, dinormalisasi dengan membagi setiap elemen dengan jumlah total pasangan piksel. Hal ini menghasilkan matriks $P_n(i, j)$ yang dinormalisasi[34].

GLCM adalah salah satu metode ekstraksi fitur yang digunakan dalam pengolahan citra dan analisis tekstur. Metode ini digunakan untuk mengekstrak informasi tentang hubungan spasial antara piksel-piksel dalam citra berwarna abu-abu. GLCM bekerja dengan menghitung probabilitas kemunculan pasangan piksel dengan tingkat keabuan tertentu, pada jarak, dan orientasi tertentu[35].

Pada proses ekstraksi fitur GLCM, citra awal diubah menjadi citra dengan tingkat keabuan diskrit, biasanya dengan mengubah skala tingkat keabuan menjadi beberapa level. Kemudian, dilakukan perhitungan untuk setiap pasangan piksel dalam citra untuk menghitung seberapa sering pasangan tersebut muncul pada jarak dan orientasi tertentu. Hasil perhitungan ini akan membentuk sebuah matriks, yang disebut GLCM, yang berisi probabilitas kemunculan pasangan-pasangan piksel tersebut [36].

Fitur-fitur tekstur seperti kontras, kekasaran, energi, dan entropi dapat dihitung dari GLCM. Fitur-fitur ini memberikan informasi tentang variasi tingkat keabuan, tekstur, dan struktur dalam citra. Fitur-fitur ini dapat digunakan sebagai representasi dari citra dan digunakan dalam berbagai aplikasi, termasuk klasifikasi dan analisis citra [37].

2.2.7.1. ASM (*Angular Second Moment*)

ASM mengukur keseragaman atau repetisi dalam tekstur gambar. Pada persamaan 2.6 ASM dihitug dengan menjumlahkan kuadrat semua elemen matriks GLCM. Nilai ASM mencapai maksimum ketika distribusi tingkat abu-abu dalam gambar adalah konstan dan teratur, sehingga menciptakan pola repetitif yang seragam. Nilai yang lebih tinggi dari ASM mengindikasikan bahwa gambar memiliki tekstur yang lebih seragam dan teratur [35].

$$ASM = \sum_{i,j=0}^{levels-1} P_{i,j}^2 \quad (2.6)$$

i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

2.2.7.2. Contrast (*Kontras*)

Kontras mengukur sejauh mana perbedaan intensitas antara piksel-piksel yang berdekatan dalam citra. Semakin besar kontras, semakin besar perbedaan intensitas, dan semakin kasar citra tampaknya. Kontras dihitug dengan menjumlahkan kuadrat perbedaan intensitas antara piksel-piksel yang berdekatan dalam GLCM seperti pada persamaan 2.7 [35].

$$Contrast = \sum_{i,j=0}^{levels-1} P_{i,j} (i - j)^2 \quad (2.7)$$

i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

2.2.7.3. Correlation

Korelasi mengukur tingkat ketergantungan linier antara intensitas piksel dalam GLCM. Semakin mendekati 1, semakin kuat korelasinya, dan semakin sejajar distribusi intensitasnya. Pada persamaan 2.8 adalah rumus untuk menghitung korelasi [35].

$$Correlation = \sum_{i,j=0}^{levels-1} P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (2.8)$$

i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

μ_i = Rata – rata penghitungan baris

μ_j = Rata – rata penghitungan kolom

σ_i = Penghitungan rata rata baris dikurangi kolom

σ_j = Penghitungan rata rata kolom dikurangi baris

2.2.7.4. *Dissimilarity* (Disimilaritas)

Persaman 2.9 dimana disimilaritas mengukur rata-rata besarnya perbedaan intensitas antara pasangan piksel dalam citra. Semakin tinggi nilai disimilaritas, semakin banyak perbedaan intensitas antara pasangan piksel, dan citra tampak semakin kasar [35].

$$Dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j} |i - j| \quad (2.9)$$

i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

2.2.7.5. *Energy* (Energi)

Energy dalam konteks GLCM (Gray Level Co-occurrence Matrix) mencerminkan seberapa mirip nilai piksel dalam gambar. Ketika nilai piksel mirip satu sama lain, energi akan memiliki nilai tinggi, mengindikasikan homogenitas dalam distribusi warna. Sebaliknya, nilai rendah menandakan heterogenitas. Nilai maksimum dari *energy* adalah 1 yang artinya distribusi piksel dalam kondisi konstan atau bentuk nya yang berperiodik (tidak acak). Dari persamaan 2.10 Secara konseptual, energi memiliki kemiripan secara konsep dengan *Angular Second Moment* (ASM), perbedaanya adalah merujuk pada akar dari *Angular Second Moment* (ASM), yang memberikan nilai yang lebih rendah dan lebih terkait dengan seberapa seragam dan teratur pola tekstur dalam gambar [35].

$$Energy = \sqrt{\sum_{i,j=0}^{levels-1} P_{i,j}^2} \quad (2.10)$$

i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

2.2.7.6. Homogeneity (Homogenitas)

Homogenitas mengukur sejauh mana intensitas piksel-piksel berdekatan cenderung serupa. Semakin tinggi nilai homogenitas, semakin mendekati intensitas yang seragam antara pasangan piksel dalam citra. Persamaan 2.11 adalah rumus dari homogenitas [35].

$$Homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (2.11)$$

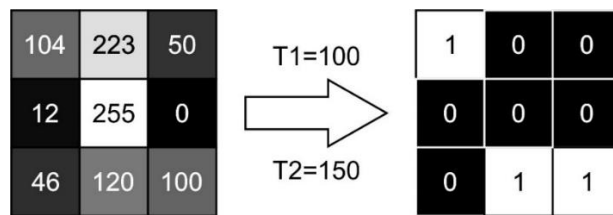
i = Nilai baris

j = Nilai kolom

P = Probabilitas kemunculan nilai *array* GLCM

2.2.8. Greyscale

Greyscale (skala keabuan) adalah salah satu mode representasi warna pada citra digital yang sangat umum digunakan. Dalam mode *greyscale*, setiap piksel pada citra direpresentasikan oleh sebuah angka tunggal yang menunjukkan tingkat keabuan atau kecerahan dari piksel tersebut. Angka tersebut biasanya berada dalam rentang nilai 0 hingga 255, di mana 0 melambangkan warna hitam (piksel gelap) dan 255 melambangkan warna putih (piksel terang). Rentang nilai ini mencerminkan perbedaan tingkat kecerahan antara piksel-piksel dalam citra, di mana nilai yang lebih tinggi menunjukkan kecerahan yang lebih besar[38].



Gambar 2. 7. Ilustrasi greyscale

Mode *greyscale* ditampilkan pada gambar 2.7 digunakan untuk ekstraksi fitur. Dengan mengonversi citra warna menjadi citra *greyscale*, kita dapat menyederhanakan kompleksitas citra dan memfokuskan pada analisis konten kecerahan dan tekstur. Ini memungkinkan berbagai algoritma pemrosesan dan pengenalan pola untuk diterapkan dengan lebih mudah dan efisien[39]. Pada citra *greyscale*, metode analisis yang umum digunakan adalah ekstraksi fitur menggunakan matriks keabuan seperti *Gray Level Co-occurrence Matrix* (GLCM).

GLCM adalah salah satu teknik yang digunakan untuk menggambarkan hubungan statistik antara pasangan piksel dengan tingkat keabuan tertentu dalam citra. Ini memungkinkan pengukuran statistik yang bermanfaat, seperti energi, kontras, dan homogenitas, yang dapat membantu dalam mengidentifikasi pola dan tekstur citra[40].

Secara keseluruhan, mode *greyscale* merupakan fondasi penting dalam pengolahan dan analisis citra karena memungkinkan penyederhanaan citra serta ekstraksi fitur yang lebih efisien. Dengan menggunakan metode analisis seperti GLCM pada citra *greyscale*, dapat memperoleh informasi penting tentang citra yang dapat digunakan untuk berbagai aplikasi, termasuk klasifikasi, deteksi objek, dan pengenalan pola[40].

2.2.9. Confusion Matrix

Confusion matrix adalah sebuah metode evaluasi yang digunakan dalam pemodelan klasifikasi untuk mengevaluasi performa suatu model. *Confusion matrix* menggambarkan jumlah prediksi yang benar (*true positives* dan *true negatives*) dan jumlah prediksi yang salah (*false positives* dan *false negatives*) yang dilakukan oleh model pada data pengujian. Matriks ini dapat memberikan informasi tentang tingkat keakuratan, sensitivitas, spesifisitas, dan presisi dari model klasifikasi. *Confusion matrix* memainkan peran penting dalam evaluasi performa model klasifikasi, dan penggunaannya dapat ditemukan dalam berbagai bidang penelitian. Matriks ini memberikan gambaran yang lebih komprehensif tentang bagaimana model klasifikasi berperforma, dan kutipan jurnal tersebut menunjukkan penggunaan *confusion matrix* dalam konteks evaluasi performa pada data medis[41].

Confusion matrix adalah konsep penting dalam evaluasi klasifikasi yang mengukur kinerja model *machine learning* dengan membandingkan prediksi klasifikasinya dengan kelas sebenarnya. Ini sangat berguna dalam kasus di mana kelas-kelas tidak seimbang atau ketika jenis kesalahan yang berbeda memiliki implikasi yang berbeda[42].

Dalam konteks evaluasi model klasifikasi, *confusion matrix*, seperti yang ditampilkan pada Tabel 2.2, menyajikan informasi yang penting. *Confusion matrix* menggambarkan hasil klasifikasi model dengan membagi prediksi menjadi empat kategori:

1. *True Positives* (TP): Instansi yang benar-benar diklasifikasikan sebagai kelas positif.
2. *True Negatives* (TN): Instansi yang benar-benar diklasifikasikan sebagai kelas negatif.
3. *False Positives* (FP): Instansi yang salah diklasifikasikan sebagai kelas positif.
4. *False Negatives* (FN): Instansi yang salah diklasifikasikan sebagai kelas negatif.

Tabel 2. 2. Confussion matrix

<i>Confussion Matrix</i>		
Aktual	Prediksi	
	<i>True</i>	<i>False</i>
<i>True</i>	TP	FP
<i>False</i>	FN	TN

Metrik-metrik ini memungkinkan perhitungan berbagai indikator kinerja, seperti akurasi, presisi, *recall* (sensitivitas), spesifisitas, dan *F1-score*. Akurasi mengukur proporsi instansi yang diklasifikasikan dengan benar dibandingkan dengan total instansi. Presisi mengukur proporsi prediksi positif yang benar dalam semua prediksi positif, mengindikasikan seberapa baik model ini berkinerja ketika memprediksi hasil positif. *Recall* menghitung proporsi prediksi positif yang benar dalam semua instansi positif yang sebenarnya, mengindikasikan kemampuan model untuk mengidentifikasi instansi positif dengan benar. Spesifisitas mengukur proporsi prediksi negatif yang benar dalam semua instansi negatif yang sebenarnya. *F1-score* adalah rata-rata harmonis dari presisi dan *recall*, memberikan metrik seimbang untuk situasi di mana ada trade-off antara presisi dan *recall*[43]. Dalam melakukan evaluasi lebih mendetail dari masing masing model baik pada skema menggunakan GLCM dan tanpa GLCM, perlu adanya formula standar deviasi untuk mengidentifikasi seberapa jauh perbedaan antara akurasi latih dan akurasi. Standar deviasi berfungsi untuk mengambil kesimpulan terhadap performa model yang telah dilakukan proses training, dengan membandingkan akurasi latih dan akurasi validasi, jika hasil standar deviasi mendekati 0 maka performa model mendekati *good fit*, sebaliknya, jika hasil standar deviasi menjauhi angka 0 maka

model memiliki performa *overfitting* / *underfitting*[44]. Persamaan 2.12 merupakan formula standar deviasi.

$$\sigma = \sqrt{\frac{\sum(X_i - \mu)^2}{N}} \quad (2.12)$$

N = banyaknya percobaan

X = setiap nilai dari populasi

μ = nilai rata-rata

i = indeks setiap percobaan