

**PENERAPAN METODE GLCM (*GRAY LEVEL CO-
OCCURRENCE MATRIX*) DAN K-NN (*K-NEAREST
NEIGHBOR*) UNTUK MENDETEKSI TINGKAT
KEMATANGAN BUAH BELIMBING BINTANG**

**OLEH
QURNIA SHANDY
T3116036**

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020**

PERSETUJUAN SKRIPSI

PENERAPAN METODE GLCM (*GRAY LEVEL CO-OCCURRENCE MATRIX*) DAN K-NN (*K-NEAREST NEIGHBOR*) UNTUK MENDETEKSI TINGKAT KEMATANGAN BUAH BELIMBING BINTANG

Oleh

QURNIA SHANDY

T3116036

SKRIPSI

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar sarjana

Program Studi Teknik Informatika

Ini telah disetujui oleh Tim Pembimbing

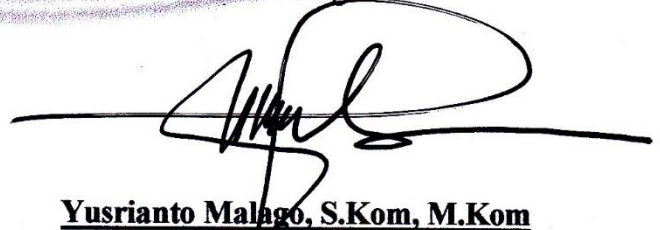
Gorontalo, 14 Juli 2020

Pembimbing I



Sudirman S. Panna, M.Kom
NIDN : 99 099128 52

Pembimbing II



Yusrianto Malaga, S.Kom, M.Kom
NIDN : 09 091089 01

PENGESAHAN SKRIPSI

PENERAPAN METODE GLCM (*GRAY LEVEL CO-OCCURRENCE MATRIX*) DAN K-NN (*K-NEAREST NEIGHBOR*) UNTUK MENDETEKSI TINGKAT KEMATANGAN BUAH BELIMBING BINTANG

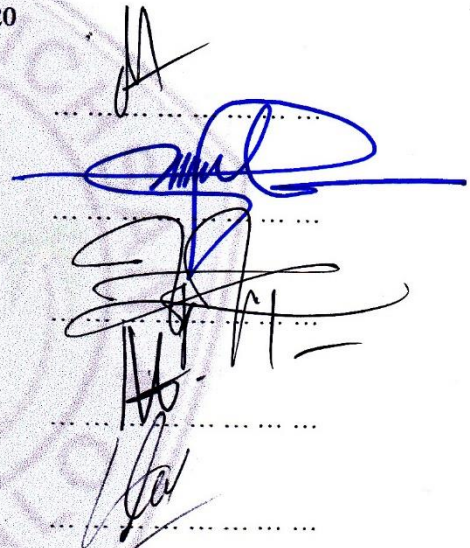
Oleh

QURNIA SHANDY

T3116036

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo
Gorontalo, 6 Agustus 2020

1. Pembimbing I
Sudirman S.Panna, M.Kom
2. Pembimbing II
Yusrianto Malago, S.Kom, M.Kom
3. Penguji I
Zohrahayati, M.Kom
4. Penguji II
M. Efendi Lasulika, M.Kom
5. Penguji III
Husdi, M.Kom



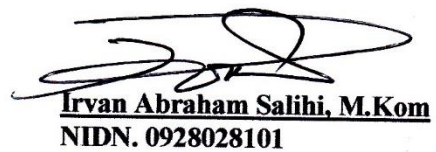
Mengetahui :

Dekan Fakultas Ilmu Komputer



Zohrahayati, M.Kom
NIDN. 0912117702

Ketua Program Studi



Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam Karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 14 Juli 2020

Yang membuat pernyataan



METERAI
TEMPEL
TGL
19E5BAHF539644088
6000
ENAM RIBU RUPIAH

Qurnia Shandy

ABSTRACT

Starfruit is one of horticultural plant type that has high economic value and rather popular in the society. In use, the information of the star fruit maturity level is needed by the agricultural industry. Yet, for knowing the maturity level of the starfruit, people are still doing manually so that it takes more time if it is done in large quantities. Determination of starfruit maturity level detection need is needed to be done with more [accurate, dependable, efficient, fast, or optimal, etc] in order to get high accuracy value. Determination of starfruit maturity level detection can be finished by using Gray Level Co-occurrence Matrix method and K-Nearest Neighbor as the model of the clarification that has not ever been tried before but it is proven to be able and reliable in finishing problems such as determination of the Bintangstarfruit maturity level. The results of the research showed that the accuracy of K-NN model for determination of starfruit maturity level is 90% by using test at the K-NN model trial, amount K-5, and direction GLCM = 0° and 135° with range = 1. This research used 10 testing data and 50 training data, which were 25 ripe categories and 25 raw categories. Thus, the application of Gray Level Co-occurrence Matrix method n and K-Nearest Neighborn problem above successfully got high accuracy value as high as 90% by using Confusion Matrix test.

Keywords: *Starfruit, application, Gray Level Co-Occurrence Matrix, K-Nearest neighbor.*

ABSTRAK

Buah Belimbing Bintang salah satu jenis tanaman hortikultura yang memiliki nilai ekonomis tinggi dan cukup banyak digemari oleh masyarakat. Dalam penggunaannya tentu informasi tingkat kematangan buah belimbing sangat diperlukan oleh industri pertanian. Namun untuk mengetahui tingkat kematangan buah belimbing bintang masih dilakukan secara manual sehingga waktu yang dibutuhkan relatif lama jika dilakukan dengan jumlah yang banyak. Penentuan deteksi tingkat kematangan buah belimbing bintang perlu dilakukan dengan lebih [akurat, handal, efisien, efektif, cepat, atau optimal, dll] agar didapatkan nilai akurasi yang tinggi. Penentuan deteksi tingkat kematangan buah belimbing dapat diselesaikan dengan menggunakan metode *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor* sebagai model Klasifikasinya yang belum pernah di uji coba sebelumnya namun terbukti dapat dan handal dalam menyelesaikan masalah seperti penentuan tingkat kematangan buah belimbing bintang. Hasil penelitian menunjukan bahwa akurasi model *K-NN* untuk penentuan tingkat kematangan buah belimbing bintang sebesar 90% dengan menggunakan pengujian Pada percobaan model *K-NN*, jumlah $K=5$, dan arah $GLCM = 0^\circ$ dan 135° dengan jarak = 1 penelitian ini menggunakan 10 data testing dan 50 data training, yaitu kategori matang dan mentah dengan masing-masing 25 data.

Dengan demikian, Penerapan metode *Gray Level Co-occurrence Matrix* dan *K-Nearest Neighbor* pada masalah diatas berhasil mendapatkan nilai akurasi yang tinggi sebesar 90% dengan menggunakan pengujian *Confusion Matrix*.

Kata Kunci : Belimbing Bintang, Penerapan, *Gray Level Co-Occurrence Matrix*, *K-Nearest Neighbor*.

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan Skripsi ini dengan judul **“Penerapan Metode GLCM (*Gray Level Co-occurrence Matrix*) Dan K-NN (*K-Nearest Neighbor*) Untuk Mendeteksi Tingkat Kematangan Buah Belimbing”**, untuk memenuhi salah satu syarat Ujian Akhir guna memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa Skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan setinggi-tingginya kepada:

1. Mohammad Ichsan Gaffar S.E, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Dr. Abd Gaffar Latjoke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Zohrahayati, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Sudirman S. Panna, M.Kom, selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo sekaligus Dosen Pembimbing Utama, yang telah membimbing penulis selama mengerjakan usulan Skripsi ini;
5. Irma Surya Kumala Idris, M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Sudirman Melangi, M.Kom, selaku Pembantu Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Yusrianto Malago, S.Kom, M.Kom, selaku Pembimbing II yang telah membimbing penulis selama mengerjakan Skripsi ini;

9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
10. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
12. Kepada semua pihak yang ikut membantu dalam penyelesaian Skripsi ini yang tak sempat penulis sebutkan satu persatu.

Semoga Allah SWT, melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, 14 Juli 2020

Penulis

DAFTAR ISI

PERSETUJUAN SKRIPSI	ii
PENGESAHAN SKRIPSI	iii
PERNYATAAN SKRIPSI	iv
ABSTRACT	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1. 1 Latar Belakang	1
1. 2 Identifikasi Masalah	4
1. 3 Rumusan Masalah	4
1. 4 Tujuan Penelitian.....	4
1. 5 Manfaat Penelitian.....	4
1. 5. 1 Manfaat Teoritis.....	4
1. 5. 2 Manfaat Praktis	4
BAB II LANDASAN TEORI	5
2. 1 Tinjauan Studi	5

2. 2 Tinjauan Pustaka	6
2.2.1 Buah Belimbing	6
2.2.2 Citra (<i>image</i>)	7
2.2.3 Pengolahan Citra	8
2.2.4 Akuisisi Citra	8
2.2.5 Pra-processing	8
2.2.6 Ekstrasi Fitur	9
2.2.7 Klasifikasi	9
2.2.8 GLCM (Gray Level Co-Occurrence Matrix)	9
2.2.9 K-Nearest Neighbor	12
2.2.10 Evaluasi Model	16
2. 3 Kerangka Pikir	18
BAB III METODE PENELITIAN	19
3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian	19
3.2 Pengumpulan Data	19
3.3 Pemodelan	20
3.3.1 Pra Pengolahan Data	20
3.3.2 Ekstrasi Ciri	21
3.3.3 Data Training	21
3.3.4 Training K-NN	21
3.3.5 Model Klasifikasi	21
3.3.6 Data Testing	21

3.3.7 Hasil Klasifikasi	21
3.3.8 Evaluasi Model.....	22
BAB IV HASIL PENELITIAN.....	23
4. 1 Hasil Pengumpulan Data	23
4. 2 Hasil Pemodelan.....	23
4.2.1 Pengambilan Citra Buah Belimbing.....	23
4.2.2 Pra-Pengolahan.....	23
4.2.3 Ekstrasi Fitur	27
4.2.4 Klasifikasi.....	31
4.2.5 Evaluasi Model.....	34
BAB V PEMBAHASAN	36
5.1 Pembahasan Sistem	36
5.2 Pembahasan Model.....	38
BAB VI PENUTUP	40
6.1 Kesimpulan.....	40
6.2 Saran	40
DAFTAR PUSTAKA	41

DAFTAR GAMBAR

Gambar 2.1 : Contoh arah GLCM.....	10
Gambar 2.2 : Penentuan awal matriks GLCM berbasis dua piksel.....	11
Gambar 2.3 : Diagram Hasil Pengujian.....	16
Gambar 2.4 : Kerangka Pikir.....	18
Gambar 3.1 : Pemodelan	20
Gambar 4.1 : Diagram histogram	25
Gambar 4.2: Hasil Output dari proses konversi RGB-Grayscale-Histogram	27
Gambar 4.3: Hasil Segmentasi	27
Gambar 5.1 Tampilan Menu Utama	36
Gambar 5.2 Tampilan Proses Load Data Training.....	36
Gambar 5.3 Tampilan Proses Load Data Testing.....	37
Gambar 5.4 Tampilan Hasil Pengenalan	37

DAFTAR TABEL

Tabel 2.1: Penelitian Terkait	5
Tabel 2.2: Sample Data Training.....	13
Tabel 2.3: Sample Data Testing	14
Tabel 2.4: Sampel Hasil Pengujian	15
Tabel 2.5: Hasil Pengujian	15
Tabel 2.6: Model Confusion matrix	17
Tabel 4.1: Data gambar Tingkat kematangan buah belimbing	23
Tabel 4.2: Citra Keabuan 4x4	25
Tabel 4.3: Perhitungan Dengan Langkah Dan Rumus	26
Tabel 4.4: Data Citra 4x4 Setelah Dilakukan Histogram Equalization.....	26
Tabel 4.5: Matrix Hasil Segmentasi.....	28
Tabel 4.6: Matrix GLCM	28
Tabel 4.7: Matrix Normalisasi	29
Tabel 4.8: Hasil Training Data set	31
Tabel 4.9: Perhitungan Jarak Euclidean Distance	31
Tabel 4.10: Hasil pengurutan dari nilai terkecil	32
Tabel 4.11: Hasil Pengurutan dari nilai terkecil menggunakan K=5	33
Tabel 4.12: Hasil Pengenalan.....	34
Tabel 4.13: Hasil Data Uji.....	35

DAFTAR LAMPIRAN

Lampiran 1: KODE PROGRAM	44
Lampiran 2: RIWAYAT HIDUP PENELITI	57
Lampiran 3: SURAT REKOMENDASI PENELITIAN.....	58
Lampiran 4: BUKTI PENERIMAAN SOFTCOPY SKRIPSI	59
Lampiran 5: SURAT PERNYATAAN	60
Lampiran 6: SURAT REKOMENDASI BEBAS PLAGIASI.....	61
Lampiran 7: HASIL TURNITIN	62

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Buah-buahan merupakan suatu Komoditas yang menguntungkan karena keanekaragaman varietas dan di dukung oleh iklim yang sesuai, sehingga menghasilkan berbagai buah-buahan yang sangat bervariasi dan menarik[1]. Produksi buah segar terus ditingkatkan karena kebutuhan masyarakat untuk mengkonsumsi buah juga meningkat[2]. Namun, apabila setelah panen tidak di tangani dengan baik, kualitas hasil panen buah-buahan akan menurun secara bertahap[1].

Salah satu buah yang cukup banyak digemari yaitu buah belimbing. Belimbing atau belimbing manis adalah tanaman buah berbentuk khas yang berasal dari Indonesia, India dan Sri Lanka. Belimbing manis memiliki nama latin *Averrhoa carambola*. Buah belimbing berwarna kuning kehijauan. Saat baru tumbuh buahnya berwarna hijau dan jika dipotong buahnya mempunyai penampang yang berbentuk bintang dan berbiji kecil yang berwarna coklat. buah belimbing memiliki rasa yang manis dan sedikit rasa asam serta mengandung banyak vitamin A dan vitamin C[3]. Buah belimbing merupakan salah satu jenis tanaman hortikultura yang memiliki nilai ekonomis tinggi. Hal tersebut disebabkan karena buah belimbing tidak hanya digunakan sebagai bahan pangan yang dikonsumsi dalam bentuk segar, namun juga beraneka ragam bentuk olahan sampai dengan bahan obat alami atau herbal[4].

Dalam Penggunaannya untuk memenuhi kebutuhan, tentunya informasi tingkat kematangan buah belimbing sangat diperlukan oleh industri pertanian pada umumnya. Sayangnya masih terdapat kendala yang sering dihadapi dalam menentukan kematangan buah belimbing, salah satu kendalanya yakni identifikasi yang masih dilakukan secara manual. Identifikasi tingkat kematangan ini dilakukan dengan mengamati secara langsung. Identifikasi secara manual sangat di pengaruhi oleh subjektifitas orang yang mengamati. Kelemahannya waktu yang dibutuhkan relatif lama jika identifikasi buah belimbing dilakukan dengan jumlah yang banyak.

Selain itu subjektifitas lainnya tingkat kelelahan dan perbedaan persepsi tentang penilaian terhadap tingkat kematangan buah belimbing. Oleh karena itu diperlukan adanya metode yang memudahkan industri dalam menentukan tingkat kematangan buah belimbing secara efisien.

Kemajuan teknologi komputer termasuk interaksi antara manusia dengan komputer pada saat ini sudah berkembang pesat. Bahkan dibidang pertanian penggunaan teknologi komputer sudah banyak digunakan. Penerapannya digunakan untuk deteksi penyakit pada tanaman, mendeteksi tingkat kemtangan buah, klasifikasi mutu, penentuan berat dan identifikasi jenis buah-buahan maupun sayuran[5]. Teknologi pengolahan citra memungkinkan untuk memilah dan mengamati produk pertanian dan perkebunan tersebut secara otomatis dengan bantuan aplikasi pengolah citra.

Seiring perkembangannya, metode-metode yang digunakan dalam teknologi pengolahan citra pun sudah cukup banyak. Salah satunya adalah metode *Gray Level Co-Occurrence Matrix*(GLCM). *Gray Level Co-Occurrence Matrix*(GLCM) merupakan suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri, perolehan ciri diperoleh dari nilai piksel matrix, yang mempunyai nilai tertentu dan membentuk suatu sudut pola[5]. Metode GLCM menggunakan beberapa fitur pendekatan statistik seperi energi, entropi, kontras, dan sebagainya[6]. Kelebihan metode *Gray Level Co-Occurrence Matrix*(GLCM) yaitu telah terbukti menjadi descriptor yang efektif, serta memiliki akurasi dan waktu komputasi yang lebih baik dari metode ekstrasi tekstur lainnya[5].

Berdasarkan beberapa penelitian yang pernah dilakukan identifikasi tingkat kematangan buah strawberry menggunakan metode GLCM memberikan hasil yang cukup baik dengan akurasi sebesar 80%[6]. Pada penelitian lain sistem untuk mengklasifikasi jenis daun jambu air menggunakan metode GLCM dan dan klasifikasinya menggunakan metode ANN mendapatkan nilai akurasi mencapai 81,25% pada identifikasi jenis daun jambu tersebut[7]. Kekurangan metode GLCM adalah kurang optimal dalam klasifikasi, sehingga dapat mengurangi akurasi untuk mendeteksi kematangan. Oleh karena itu perlu adanya penambahan metode

klasifikasi untuk peningkatan akurasi untuk mendeteksi kematangan buah belimbing.

Pada penelitian ini untuk meningkatkan akurasi pengenalan metode *Gray Level Co-Occurrence Matrix*(GLCM) dapat dikombinasikan dengan metode *K-Nearest Neighbor* dalam proses klasifikasi. K-NN merupakan salah satu teknik klasifikasi yang paling dasar dan sederhana. Metode ini memiliki kelebihan yakni pelatihan yang sangat cepat, sederhana dan mudah di pelajari, tahan terhadap data yang memiliki derau dan efektif jika data pelatihan besar[8]. Dari beberapa penelitian yang dilakukan, pada Penentuan kematangan buah salak pondoh di pohon berbasis pengolahan citra dan klasifikasinya menggunakan metode *K-Nearest Neighbor* (K-NN) mendapatkan nilai akurasi 93%[9], sedangkan pada identifikasi jenis buah apel menggunakan Algoritma K-NN dengan ekstrasi fitur histogram memberikan tingkat akurasi 90%[10]. Begitu pula pada aplikasi klasifikasi tingkat kematangan buah jambu biji merah menggunakan algoritma *K-Nearest Neighbor* (K-NN) dan ekstrasi fiturnya menggunakan *Hue, Saturation, Value* (HSV), tingkat akurasinya mencapai 84% [11].

Hasil ini menunjukan bahwa penggunaan metode K-NN baik dalam meningkatkan akurasi klasifikasi jika dikombinasikan dengan metode lain. Dengan demikian metode K-NN dapat dikombinasikan dengan metode GLCM pada proses mendeteksi tingkat kematangan sehingga dapat meningkatkan akurasi pengenalan (Identifikasi) tingkat kematangan buah belimbing. Sistem di bangun menggunakan *Spyder Python* sebagai editor untuk mengimplementasikan script, dan *openCV* untuk pemrosesan citranya.

Berdasarkan berbagai pemaparan di atas, maka peneliti tertarik untuk melakukan penelitian dengan judul: “**Penerapan Metode GLCM (Gray Level Co-occurrence Matrix) Dan K-NN (K-nearest Neighbor) Untuk Mendeteksi Tingkat Kematangan Buah Belimbing.**” Diharapkan peniltian ini dapat memberikan kontribusi, berupa model GLCM dan K-NN yang paling akurat untuk mendeteksi tingkat kematangan buah belimbing.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan maka identifikasi masalah adalah :

1. Terbatasnya fungsi identifikasi secara manual sehingga memberikan hasil yang tidak konsisten untuk mendeteksi tingkat kematangan buah belimbing.
2. Perlunya metode algoritma untuk menentukan tingkat kematangan buah belimbing secara efisien.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah, maka dapat dirumuskan masalah penelitian sebagai berikut :

Bagaimana kinerja dan efektifitas Sistem Deteksi tingkat kematangan buah belimbing menggunakan metode GLCM dan KNN?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah maka tujuan penilitan ini adalah :

Menguji Coba Algoritma GLCM dan KNN untuk memperoleh Akurasinya yang terbaik pada deteksi kematangan buah belimbing.

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Memberikan masukan bagi perkembangan ilmu pengetahuan dan teknologi, khususnya pada bidang ilmu komputer, yaitu berupa penerapan metode K-NN untuk mengoptimalkan kinerja metode GLCM pada sistem identifikasi tingkat kematangan buah belimbing.

1.5.2 Manfaat Praktis

Sebagai solusi untuk mempermudah industri dalam mengidentifikasi tingkat kematangan buah belimbing.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Penelitian tentang sistem Mendeteksi kematangan buah dan metode yg digunakan telah dilakukan oleh beberapa peneliti sebelumnya. Berikut beberapa penelitian terkait:

Tabel 2.1: Penelitian Terkait

NO	PENELITI	JUDUL/TAHUN	METODE	HASIL
1.	Dedi Tri Hermanto, Selvy Megira, Devina Ninosari, Kusrini [6].	Identifikasi Tingkat Kematangan Buah Strawberry Menggunakan Gray Level Co-Occurance Matrix (GLCM) Dan Laplacian Filter/2018	Gray Level Co-Occurance Matrix (GLCM) dan Laplacian Filter	Penelitian ini akan menghasilkan identifikasi tingkat kematangan buah strawberry dengan kategori matang , mentah dan busuk. Berdasarkan hasil pengujian, sistem dapat melakukan identifikasi tingkat kematangan buah strawberry dengan penambahan fitur RGB didapatkan akurasi sebesar 80% sedangkan tingkat akurasi tanpa menambahkan fitur RGB sebesar 60%

2.	Suhendri, Putri Rahayu[7].	Metode Grayscale Co- occurrence Matrix (GLCM) Untuk Klasifikasi Jenis Daun Jambu Air Menggunakan Algoritma Neural Network/2019	Gray Level Co- Occurance Matrix (GLCM) dan Algoritma Neural Network	Hasil penelitian ini menunjukkan bahwa tingkat akurasi pada daun jambu bol adalah 81,25%, daun buntan 3 Hijau 75%, dan daun citra 80% dan total nilai akurasi keseluruhan 78.89%.
3.	Pawit Rianto, Agus Harjoko[9].	Penentuan Kematangan Buah Salak Pondoh Di Pohon Berbasis Pengolahan Citra Digital/2017	Backpropaga tioon <i>dan</i> K- Nearest Neighbor (K- NN)	Dari hasil pengujian menunjukkan bahwa keberhasilan sistem ini sebesar 92% jika menggunakan algoritma klasifikasi backpropagation dan 93% dengan algoritma k-Nearest Neighbor

2. 2 Tinjauan Pustaka

2.2.1 Buah Belimbing

Belimbing atau belimbing manis merupakan tanaman buah berbentuk cukup unik yang berasal dari Indonesia, India dan Sri Lanka. Belimbing manis memiliki nama latin *Averrhoa carambola* yang termasuk ke dalam family Oxalidaceae dengan ordo Oxalidales. Buah belimbing berwarna kuning ke hijauan. Saat baru tumbuh, buahnya berwarna hijau dan jika dipotong buahnya mempunyai penampang yang berbentuk bintang dan berbiji kecil yang berwarna coklat. Buah

belimbing memiliki rasa yang manis dan sedikit rasa asam yang segar serta mengandung banyak vitamin A dan C. Belimbing adalah salah satu tanaman yang cukup banyak digunakan oleh masyarakat sebagai obat tradisional. Biasanya masyarakat menggunakan belimbing wuluh dan belimbing manis sebagai obat darah tinggi atau hipertensi[3].

2.2.2 Citra (*image*)

Citra adalah representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu penyimpanan[12]. Citra merupakan komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Selain itu mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi[13]. Secara umum terdapat 3 jenis citra yaitu : [13][14].

1. Citra Biner

Citra biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih (0 dan 1). Nilai 0 menyatakan warna hitam dan 1 menyatakan warna putih.

2. Citra Keabuan

Citra jenis ini menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu abu . citra keabuan biasa disebut juga dengan citra *grayscale* yaitu citra yang nilai pixelnya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih.

3. Citra Berwarna

Citra Berwarna merupakan citra yang menyajikan warna dalam bentuk RGB (R = Merah), (G = Hijau), (B = Biru). Setiap komponennya menggunakan 8 bit, nilainya berkisar dari (0 – 255). Dengan demikian warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna.

2.2.3 Pengolahan Citra

Pengolahan citra digital merupakan bagian dari perkembangan teknologi dalam konsep dan penalaran, manusia ingin mesin (komputer). Pengolahan citra digital mempelajari tentang teknik-teknik mengolah citra. Citra adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Digital mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan computer. Secara sistematis citra merupakan fungsi kontinyu (continue) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan computer digital, maka suatu citra harus dipresentasikan secara numeric dengan nilai-nilai diskrit. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra[15].

2.2.4 Akuisisi Citra

Akuisisi citra adalah proses pengambilan citra, dimana sebuah objek. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tempak menjadi citra digital. Beberapa alat yang dapat digunakan seperti : kamera *digital*, *scanner*, foto sinar-x atau sinar infra merah, dan alat penangkapan bayang lainnya. Hasil dari akuisisi citra ini ditentukan oleh kemampuan *sensor* atau alat untuk mendigitalisasikan sinyal yang terkumpul pada sensor atau alat tersebut [16].

2.2.5 Pra-processing

Pra-processing bertujuan untuk mempermudah proses dalam melakukan identifikasi citra digital. Tahap ini terdiri dari tahap pengubahan warna latar dari putih menjadi hitam untuk meningkatkan tingkat akurasi sistem. Warna ini kemudian dikonversikan menjadi citra keabuan (*greyscale*). Tahap terakhir dari pra-pengolahan adalah peningkatan kontras citra untuk meningkatkan kecerahan pada citra, agar citra lebih mudah dikenali[17].

2.2.6 Ekstrasi Fitur

Ekstrasi Fitur merupakan suatu pengambilan ciri atau fitur suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Fitur adalah karakteristik dari suatu objek yang akan berguna untuk proses klasifikasi. Ekstrasi fitur dilakukan dengan cara menghitung jumlah titik atau pixel yang ada pada citra[18].

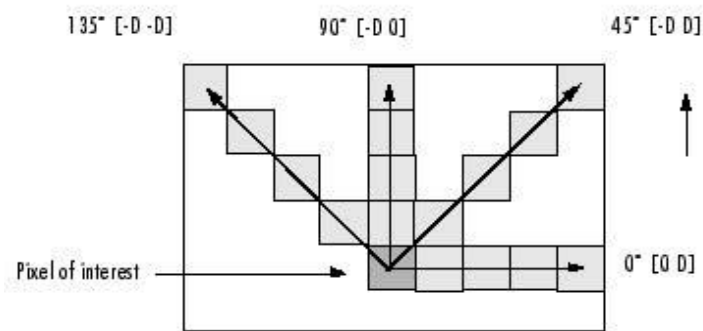
2.2.7 Klasifikasi

Klasifikasi merupakan suatu metode untuk mengelompokkan sebuah objek ke dalam kelompok atau kelas tertentu[19]. Klasifikasi termasuk ke dalam *supervised learning* karena menggunakan sekumpulan data untuk di analisis terlebih dahulu, kemudian pola dari analisis tersebut digunakan untuk pengklasifikasian data uji. Proses klasifikasi data terdiri dari pembelajaran dan klasifikasi. Pada pembelajaran data training dianalisis menggunakan algoritma klasifikasi, selanjutnya pada klasifikasi digunakan data testing untuk memastikan tingkat akurasi dari rule klasifikasi yang digunakan. Teknik klasifikasi dibagi menjadi lima kategori berdasarkan perbedaan konsep matematika, yaitu berbasis statistik, berbasis jarak, berbasis pohon keputusan, berbasis jaringan syaraf, dan berbasis rule[20]. Proses ini dilakukan agar data atau citra dapat dikategorikan dalam suatu kelas tertentu yang telah ditentukan. Algoritma klasifikasi yang banyak digunakan secara luas, yaitu K-Nearest Neighbor, Decision atau Classification Trees, Bayesian Classifiers atau Naïve Bayes classifiers, Neural Networks, Analisa Statistik, Algoritma Genetika, Rough Sets, Metode Rule Based, Memory Based Reasoning, dan Support Vector Machines (SVM)[21]

2.2.8 GLCM (Gray Level Co-Occurrence Matrix)

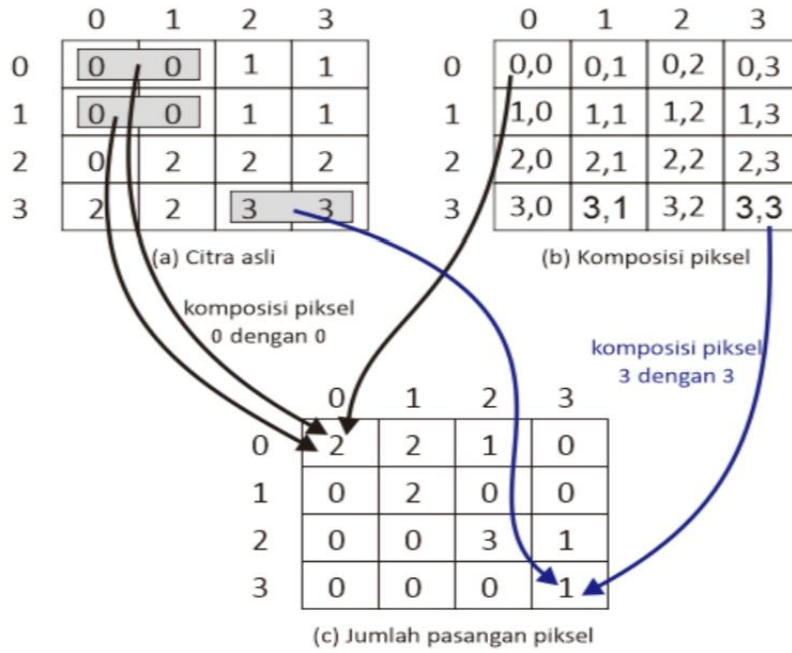
GLCM (*Gray Level Co-occurrence Matrix*) pertama kali diusulkan oleh Haralick pada tahun 1979 dengan 28 fitur untuk menjelaskan pola spasial[22]. Metode GLCM (*Gray Level Co-occurrence Matrix*) merupakan salah satu metode untuk ekstraksi tekstur pada citra. Ekstraksi tekstur dilakukan untuk mengambil informasi pokok dari suatu citra sebelum digunakan ke proses berikutnya. Metode GLCM menggunakan beberapa fitur pendekatan statistik seperti energi, entropi,

kontras, dan sebagainya. Langkah pertama untuk menghitung fitur-fitur GLCM adalah dengan mengubah citra RGB menjadi citra berskala keabuan. Langkah kedua adalah menciptakan matrik co-occurrence dan dilanjutkan dengan menentukan hubungan spasial antara piksel referensi dan piksel tetangga berdasarkan sudut θ dan jarak d . Langkah selanjutnya adalah menciptakan matrik simetris dengan menambahkan matrik co-occurrence dengan matriks transposenya. Kemudian dilakukan normalisasi terhadap matrik simetris dengan menghitung probabilitas setiap element matrik. Langkah terakhir adalah dengan menghitung fitur GLCM. Setiap fitur dihitung dengan satu piksel jarak di empat arah, yaitu 0° , 45° , 90° , dan 135° untuk mendeteksi co-occurrence[23]. Keempat arah tersebut dapat dilihat dari gambar 2.1.



Gambar 2.1: Contoh arah GLCM[24].

Sebagai keperluan ilustrasi, arah dari ketetanggan piksel dapat di pilih ke timur(kanan). Berupa untuk mempresentasikan hubungan ini yaitu adalah nilai (1,0), yang merepresentasikan hubungan dua piksel yang berjajar horizontal dengan piksel bernilai 1 diikuti dengan piksel bernilai 0. Dengan menggunakan acuan tersebut, jumlah kelompok piksel yang memenuhi hubungan derajat tersebut dihitung, diilustrasikan pada Gambar 2.2.



Gambar 2.2: Penentuan awal matriks GLCM berbasis dua piksel[24].

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut [5] :

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah 0° , 45° , 90° atau 135° .
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasi matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri, yaitu:
 - a) *Kontras*

$$\sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2) \quad (1)$$

- b) *Homogeneity*

$$\sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \quad (2)$$

c) *Energy*

$$\sum_{i_1} \sum_{i_2} p^2(i_1, i_2) \quad (3)$$

d) *Entropi*

$$\sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2) \quad (4)$$

2.2.9 K-Nearest Neighbor

Algoritma K-Nearest Neighbor (K-NN) merupakan metode klasifikasi yang menggunakan algoritma supervised dimana hasil query instance diklasifikasikan berdasarkan mayoritas dari kategori pada K-NN. Algoritma ini adalah salah satu algoritma pembelajaran mesin sederhana, Prinsip kerja K-Nearest Neighbor (K-NN) adalah proses mencari tetangga paling dekat terhadap sampel data yang diuji. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru berdasarkan atribut dan training sample. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample [11][24].

Metode K-NN memiliki beberapa kelebihan yaitu: [8]

1. Pelatihan sangat cepat,
2. Sederhana dan mudah dipelajari,
3. Tahan terhadap data pelatihan yang memiliki derau, dan
4. Efektif jika data pelatihan besar

Metode K-NN melakukan proses pencocokan/pengenalan berdasarkan jumlah tetangga terdekat untuk menentukan kelasnya. Untuk mencari jarak kelas ada beberapa cara yaitu euclidean distance. Adapun penerapan algoritma K-Nearest Neighbor terdiri langkah-langkah berikut : [25]

1. Menentukan parameter k (jumlah tetangga paling dekat)

2. Menghitung kuadrat jarak euclid (query instance) masing-masing obyek terhadap data sampel yang diberikan.

Rumus euclidean distance : [11]

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (5)$$

Keterangan:

x_1 = Sampel Data

x_2 = Data Uji / Testing

i = Variabel Data

d = Jarak

p = Dimensi Data

3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak euclid terkecil.
4. Mengumpulkan kategori Y (*Klasifikasi nearest neighbor*)
5. Dengan menggunakan kategori *nearest neighbour* yang paling mayoritas maka diprediksi nilai query instan yang telah dihitung.

Adapun contoh penerapan dari perhitungan manual dari K-NN : [26]

Contoh data, terdapat beberapa data yang berasal dari survey questioner tentang klasifikasi jenis rempah-rempah berdasarkan fitur warna RGB tekstur. Proses pengujian klasifikasi dilakukan dengan menggunakan Algoritma K-Nearest Neighbor dengan nilai K=1, K=3 dan K=5, metode pengukuran jarak yang digunakan adalah Euclidean Distance. secara keseluruhan data training yang digunakan adalah 30 untuk masing-masing jenis rempah-rempah sehingga total keseluruhan berjumlah 120 data.

Tabel 2.2 : Sample Data Training[26]

Nama	<i>red</i>	<i>green</i>	<i>blue</i>	<i>energy</i>	<i>homogeneity</i>
Jahe 1	243.947	238.967	231.500	0,75108	0,98817
Jahe 2	250.102	247.829	244.378	0,84991	0,98974

Kencur 1	251.516	249.977	248.058	0,93425	0,99570
Kencur 2	247.648	245.583	243.968	0,90663	0,99389
Kunyit 1	216.296	207.255	200.842	0,56372	0,98040
Kunyit 2	198.486	184.850	176.708	0,34896	0,95906
Lengkuas 1	230.962	221.110	213.744	0,66720	0,99041
Lengkuas 2	248.632	244.912	240.438	0,86982	0,99466

Tabel 2.3: Sample Data Testing[26]

Nama	<i>red</i>	<i>green</i>	<i>blue</i>	<i>energy</i>	<i>homogeneity</i>
Jahe 1	240.915	235.502	228.249	0,74449	0,98763
Jahe 6	249.537	246.454	240.995	0,80756	0,98988
Jahe 8	250.110	247.861	244.917	0,89477	0,99255
Kencur 1	248.707	246.5	244.557	0.90695	0.9934
Kencur 5	251.905	251.059	250.285	0,96216	0,99771
Kencur 6	250.997	249.764	248.898	0,94807	0,99628
Kunyit 3	244.339	241.190	237.806	0,83032	0,99002
Kunyit 5	247.306	244.755	242.526	0,86992	0,99196
Kunyit 8	247.402	244.262	241.149	0.8629	0.99367
Lengkuas 1	229.378	219.924	211.976	0,64473	0,98368
Lengkuas 2	237.728	230.507	223.552	0,73537	0,98993
Lengkuas 3	247.495	244.549	241.371	0.87983	0.99575

Setelah diperoleh data yang digunakan untuk pengujian kemudian dilakukan proses klasifikasi menggunakan Algoritma *K-Nearest Neighbour* dengan langkah-langkah sebagai berikut :

1. Menentukan parameter k (jumlah tetangga paling dekat). Parameter k (jumlah tetangga paling dekat untuk melakukan klasifikasi jenis rempahrempah diatas adalah menggunakan nilai k=1, k=3, k=5.
2. Menghitung kuadrat jarak euclidean masing-masing objek terhadap data sample yang diberikan.

Sebagai contoh perhitungan diambil data testing yang pertama yaitu data testing Jahe1, lalu dikalikan terhadap seluruh data training yang ada:

$$d = \sqrt{(243.947 - 240.915)^2 + (238.967 - 235.502)^2 + (231.500 - 228.249)^2 + (0,75108 - 0,74449)^2 + (0,98817 - 0,98763)^2} = 5636,33303$$

Jadi jarak Euclidean Distance yang diperoleh adalah sebesar 5636,33303. Data testing jahe1 tersebut dihitung Euclidean Distance nya keseluruhan data training yang ada, kemudian dirangking dengan mengambil 5 neighbor yang terdekat, hasil dari perkalian data testing jahe1 terhadap 120 data training dapat dilihat pada tabel berikut ini.

Tabel 2.4: Sampel Hasil Pengujian[26]

Ranking	Euclidean Distance	Nama Kelas
1	1784,19422	Jahe
2	1872,70873	Lengkuas
3	1938,29461	Lengkuas
4	2042,767731	Lengkuas
5	2174,98	Jahe

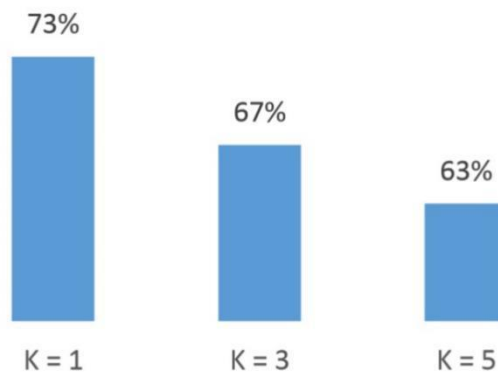
Dari hasil pengujian diatas dapat dilihat bahwa jika menggunakan nilai K=1 maka proses klasifikasi benar, jika menggunakan nilai K=3 maka proses klasifikasi salah, dan jika menggunakan nilai K=5 maka proses klasifikasi juga salah. Pada penelitian ini dilakukan 30 kali pengujian dengan menggunakan data training masing-masing jenis rempah-rempah terhadap 120 data training, hasil dari keseluruhan pengujian dapat dilihat pada table 2.5.

Tabel 2.5: Hasil Pengujian[26]

Nilai K	Jumlah Kebenaran	Persentasi Kebenaran
1	23/30	73%

3	20/30	67%
5	19/30	63%

Untuk lebih jelasnya perbandingan hasil pengujian dapat dilihat pada diagram seperti yang ditunjukkan pada gambar 2.3 :



Gambar 2.3: Diagram Hasil Pengujian

Dari hasil pengujian diatas dapat dilihat bahwa nilai K terbaik untuk melakukan klasifikasi jenis rempah-rempah berdasarkan fitur warna RGB dan tekstur adalah 1 dengan nilai akurasi 73%, yang kedua adalah nilai K=3 dengan nilai akurasi 67% dan yang ketiga adalah nilai K=5 dengan nilai akurasi 63%.

2.2.10 Evaluasi Model

Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun. Beberapa cara untuk melakukan evaluasi adalah dengan menghitung akurasi (*accuracy*), presisi (*precision*), dan *recall* dari hasil analisis sistem. Metode yang digunakan pada penelitian ini adalah *Confusion Matrix*. *Confusion matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Evaluasi dengan *confusion matrix* menghasilkan nilai akurasi, presisi dan *recall*.

Akurasi dalam klasifikasi adalah persentase ketepatan record data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi.

Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar[27]

Tabel 2.6: Model Confusion matrix [28]

<i>Correct Classification</i>	<i>Classified as</i>	
	<i>Predicted “+”</i>	<i>Predicted “-“</i>
<i>Actual “+”</i>	<i>TP</i>	<i>FN</i>
<i>Actual “-“</i>	<i>FP</i>	<i>TN</i>

Perhitungan akurasi dengan tabel *confusion matrix* adalah sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} * 100\% \quad (6)$$

$$Presisi = \frac{TP}{TP+FP} * 100\% \quad (7)$$

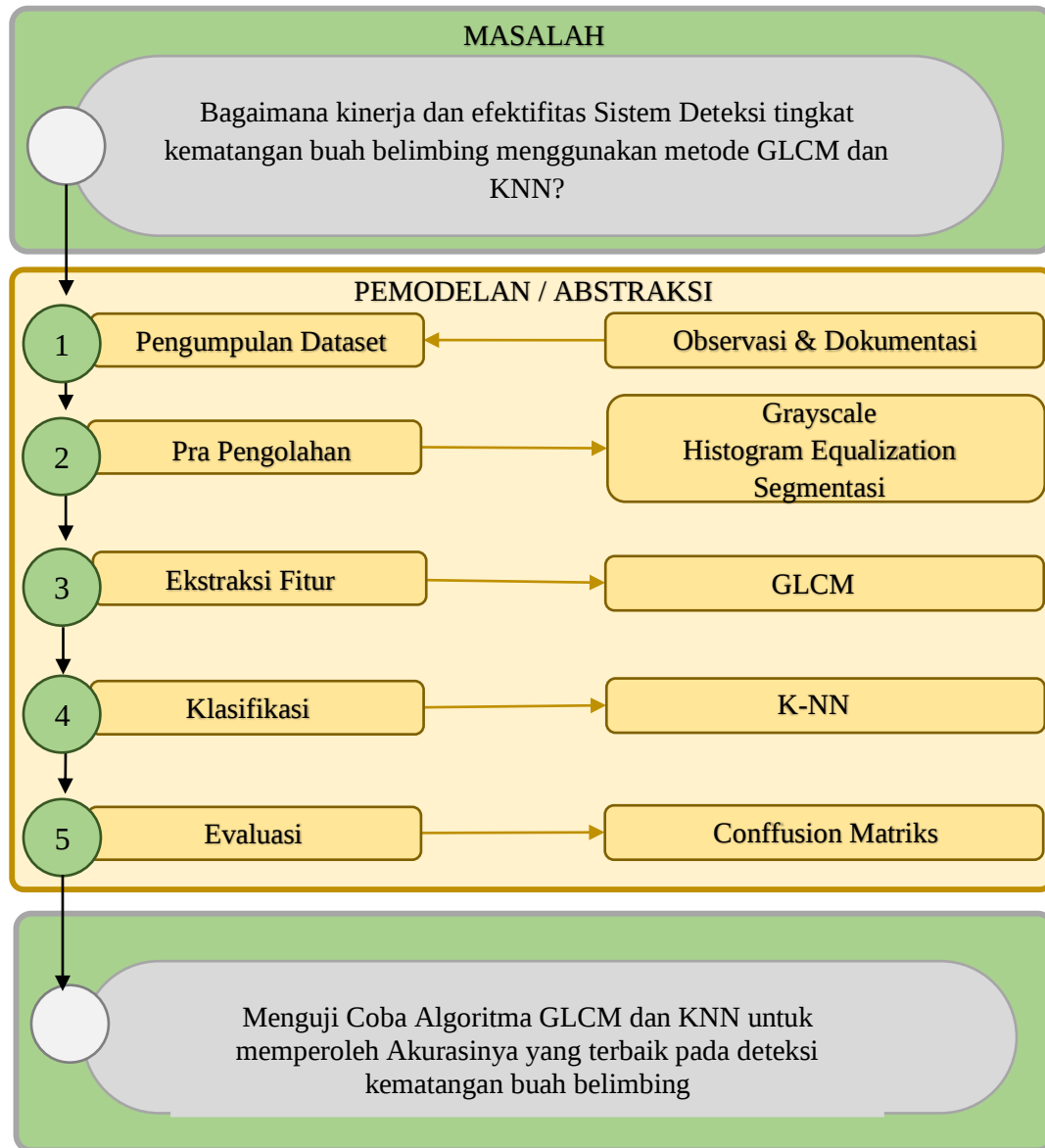
$$Recall = \frac{TP}{TP+FN} * 100\% \quad (8)$$

Keterangan :

- True Positive (TP) = adalah jumlah record datapositif yang diklasifikasikan sebagai nilai positif
- True Negative (TN) = adalah jumlah record data negatif yang diklasifikasikan sebagai nilai negative
- False Positive (FP) = adalah jumlah record data negatif yang diklasifikasikan sebagai nilai positif
- False Negative (FN) = adalah jumlah record data positif yang diklasifikasikan sebagai nilai positif

2.3 Kerangka Pikir

Gambar 2.4: Kerangka Pikir



BAB III

METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang di olah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori.

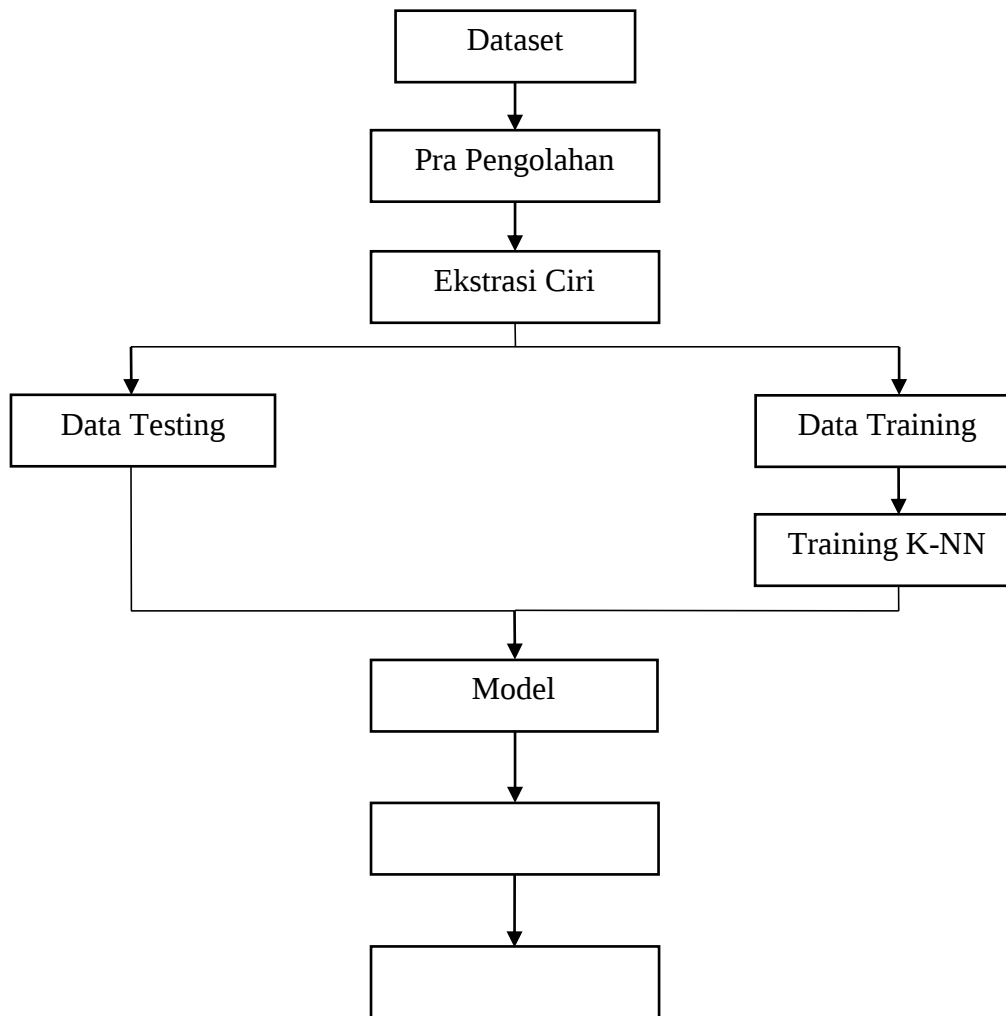
Penelitian ini menggunakan metode penelitian *eksperimental*. Dengan demikian ini adalah penelitian *eksperimental*.

Subjek penelitian ini adalah *klasifikasi* pada objek Buah Belimbing. Penelitian ini dimulai dari Juli 2019 sampai dengan November 2019.

3.2 Pengumpulan Data

Data primer penelitian ini adalah citra Buah Belimbing, dimana peneliti mengambil gambar secara langsung menggunakan Kamera Smarthphone Samsung Galaxy A20 dengan resolusi 13 MP. Jumlah data yang digunakan yaitu 60 file dalam format JPG dengan tingkat kematangan (Mentah dan Matang).

3.3 Pemodelan



Gambar 3. 1: Pemodelan

3.3.1 Pra Pengolahan Data

Sebelum data diolah, terlebih dahulu dilakukan *histogram equalization*. Hal ini dilakukan karena *Histogram equalization* adalah proses yang mengubah distribusi nilai derajat keabuan pada sebuah citra sehingga menjadi seragam (uniform). Selanjutnya dilakukan proses segmentasi citra, segmentasi citra merupakan proses pengolahan yang bertujuan memisahkan wilayah (*region*) objek dengan wilayah latar belakang agar objek mudah di analisis dalam rangka mengenali objek yang banyak melibatkan persepsi visual.

3.3.2 Ekstrasi Ciri

Ekstrasi ciri berfungsi sebagai pendeteksi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan objek satu dengan objek lainnya, diantaranya adalah ciri bentuk, ukuran, ciri geometri, ciri tekstur, dan warna. Pada penelitian ini menggunakan ekstrasi ciri tekstur *Gray Level Co-Occurrence Matrix* (GLCM). Masing-masing objek diekstrak cirinya berdasarkan parameter-parameter tertentu dan dikelompokkan pada kelas tertentu. Nilai dari parameter tersebut kemudian dijadikan sebagai data masukan dalam proses identifikasi klasifikasi.

3.3.3 Data Training

Data training berupa data yang telah terekstrak cirinya yang selanjutnya akan dilatih menggunakan algoritma *K-Nearest Neighbor* (K-NN), jumlah data training yang digunakan berupa 10% dari sample citra buah belimbing dengan tingkat kematangan yang berbeda. Data training ini berupa hasil ekstrasi dari *Gray Level Co-Occurrence Matrix* (GLCM).

3.3.4 Training K-NN

Training K-NN bekerja berdasarkan jarak terdekat dari training dan test dalam penentuannya, kemudian mengambil mayoritas hasil ketentuan yang telah didapatkan untuk dijadikan prediksi dari test.

3.3.5 Model Klasifikasi

Merupakan hasil dari proses training dari algoritma *K-Nearest Neighbor* (K-NN) menggunakan data training.

3.3.6 Data Testing

Data training merupakan data yang telah terekstrasi cirinya yang digunakan untuk menguji data yang telah ditraining. Jumlah data yang digunakan berupa 90% dari sample citra buah belimbing dengan tingkat kematangan yang berbeda-beda.

3.3.7 Hasil Klasifikasi

Hasil klasifikasi merupakan hasil output yang didapatkan pada data testing dari proses klasifikasi yang menggunakan *K-Nearest Neighbor* (K-NN)

berdasarkan model yang di dapatkan dari data training. Atau hasil yang didapatkan dari proses klasifikasi yang dilakukan model.

3.3.8 Evaluasi Model

Model yang telah di hasilkan kemudian dievaluasi dengan menggunakan *confusion matrix* Untuk mengetahui tingkat akurasinya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah data real dengan cara mengambil foto secara langsung menggunakan kamera Smartphone Samsung A20 dengan resolusi kamera 13 Megapixel. Dengan jumlah data 60 citra gambar buah belimbing dengan tingkat kematangan yang berbeda (30 Matang, 30 Mentah).

Tabel 4.1: Data gambar Tingkat kematangan buah belimbing

Mentah				
Matang				

4.2 Hasil Pemodelan

4.2.1 Pengambilan Citra Buah Belimbing

Pada penelitian ini, dataset yang digunakan adalah citra buah belimbing bintang. Data yang akan digunakan adalah sebanyak 60 data dengan rincian 30 citra untuk buah belimbing yang sudah matang dan 30 citra untuk buah yang masih mentah. Pengambilan citra dilakukan dengan menggunakan alas kertas HVS yang memiliki warna latar putih (*background*). Warna latar putih bertujuan untuk mempermudah proses pra pengolahan dan ekstraksi ciri dimana warna latar tidak mempengaruhi nilai dari masing masing nilai ciri citra.

4.2.2 Pra-Pengolahan

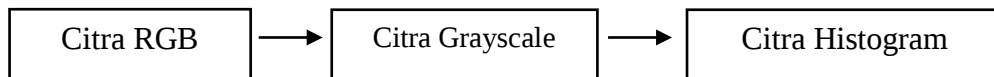
Pra-pengolahan citra (*image pre-processing*), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra buah belimbing yang sudah ada dikonversi agar diperoleh data citra belimbing yang

sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem deteksi tingkat kematangan.

Proses awal yang dilakukan yaitu merubah citra *training* atau citra *testing* yang awalnya citra dari RGB menjadi citra *grayscale*, perubahan ini dilakukan karena citra *grayscale* memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0.

Setelah citra asli dikonversi ke citra *grayscale* maka pra-pengolahan selanjutnya adalah histogram ekualisasi. Histogram ekualisasi adalah proses yang mengonversikan bagian nilai derajat keabuan pada sebuah citra sehingga menjadi sejenis. Maksud dari proses *histogram equalization* adalah untuk mencapai diseminasi histogram yang menyeluruh sehingga setiap derajat keabuan mendapat jumlah pixel yang relatif sama.

Berikut tahapan-tahapan yang dilakukan dalam pra pengolahan :



Rumus Konversi Citra RGB ke Grayscale:

$$s = \frac{R + B + G}{3}$$

Dimana: R = Red, B = Blue, G = Green

HISTOGRAM EQUALIZATION

Histogram Equalization adalah suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Pada dasarnya metode ini akan digunakan untuk memperlebar range tingkat keabuan, sehingga akan meningkatkan kontras citra. Dari hasil penurunan rumus perataan Histogram secara matematis didapat transformasi sebagai berikut:

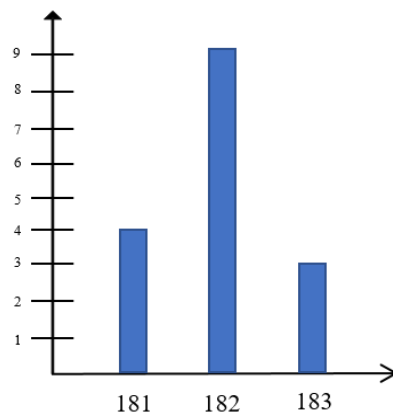
$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

$$= \sum_{j=0}^k \frac{n_j}{n}$$

Contoh Perhitungan manual Citra Keabuan ke citra Histogram

Tabel 4.2: Citra Keabuan 4x4

x,y	0	1	2	3
0	183	181	182	183
1	182	181	182	182
2	182	181	182	182
3	182	181	183	182



Gambar 4.1 : Diagram histogram

Setelah didapat histogram asli dari citra diatas, maka dilakukan perhitungan dengan langkah dan rumus pada tabel dibawah :

Tabel 4.3: Perhitungan Dengan Langkah Dan Rumus

Derajat Keabuan (j)	181	182	183
Kemunculan	4	9	3
$s = \sum_{j=0}^k \frac{n_j}{n}$	4/16	9/16	3/16

Hasil dari perhitungan Perataan Histogram diatas telah menunjukan sejumlah data citra baru seperti pada tabel diatas, dan disimpulkan seperti berikut :

Tabel 4.4: Data Citra 4x4 Setelah Dilakukan Histogram Equalization

0,1875	0,25	0,5625	0,1875
0,5625	0,25	0,5625	0,5625
0,5625	0,25	0,5625	0,5625
0,5625	0,25	0,1875	0,5625





Gambar 4.2: Hasil Output dari proses konversi RGB-Grayscale-Histogram

SEGMENTASI CITRA

Pada tahap ini dilakukan pembagian atau pemisahan objek gambar dengan background gambar dan dengan menggunakan segmentasi. Berikut merupakan Contoh dari gambar hasil segmentasi :

Citra Asli	Nilai R	Nilai G
Nilai B	Hasil Otsu	Hasil Segmentasi

Gambar 4.3: Hasil Segmentasi

4.2.3 Ekstrasi Fitur

Fitur ekstrasi Grey Level Co-occurrence Matrix (GLCM) merupakan matrix yang menggambarkan frekuensi munculnya pasangan dua pixel dengan

intensitas tertentu dalam jarak dapat berupa nilai 1,2,3 dan seterusnya sedangkan orientasi arah sudut dapat berupa 0° , 45° , 90° , 135° .

Contoh : Nilai pada jarak 1 dan orientasi arah sudut 0°

Nilai pada table 4.5 berupa matrix 4 x 4, dari matrix 255 x 255 yang disederhanakan untuk memudahkan perhitungan.

Tabel 4.5: Matrix Hasil Segmentasi

i/j	0	1	2	3
0	183	180	181	183
1	183	182	182	185
2	183	181	182	183
3	183	185	183	183

Setelah menentukan arahnya selanjutnya membentuk matriks kookurens dengan cara menghitung frekuensi kemunculan pasangan nilai keabuan piksel referensi dan piksel tetangga pada jarak dan arah yang ditentukan. Perhitungan menggunakan nilai orientasi dengan arah sudut 0° . 0° yaitu nilai gambar yang memiliki nilai yang sama dan bersebelahan. Hasilnya sebagai berikut:

Tabel 4.6: Matrix GLCM

i/j	180	181	182	183	185
180	0	1	0	1	0
181	1	0	1	2	0
182	0	1	1	2	1
183	1	2	2	1	0
185	0	0	1	0	0
Jumlah Total Nilai GLCM = 18					

Langkah selanjutnya yaitu menormalisasikan matrix dengan cara menjumlahkan seluruh nilai yang ada dalam matriks.

Tabel 4.7: Matrix Normalisasi

i/j	180	181	182	183	185
180	0	0,057	0	0,057	0
181	0,057	0	0,057	0,111	0
182	0	0,057	0,057	0,111	0,057
183	0,057	0,111	0,111	0,057	0
185	0	0	0,057	0	0

Langkah terakhir adalah menghitung ciri statistik GLCM yaitu :

a. Menghitung nilai Contras

Contras =

$$(180-181)^2(0,057) + (180-183)^2(0,057) + (181-180)^2(0,057) + (181-182)^2(0,057) + (181-183)^2(0,111) + (182-181)^2(0,057) + (182-182)^2(0,057) + (182-183)^2(0,111) + (182-185)^2(0,057) + (183-180)^2(0,057) + (183-181)^2(0,111) + (183-182)^2(0,111) + (183-183)^2(0,057) + (185-182)^2(0,057)$$

$$\text{Contras} = 0,057 + 0,513 + 0,057 + 0,057 + 0,444 + 0,057 + 0 + 0,111 + 0,513 + 0,513 + 0,444 + 0,111 + 0 + 0,513$$

$$\text{Contras} = \mathbf{3,903}$$

b. Menghitung nilai Energy

$$\text{Energy} = ((0,057)^2) + ((0,057)^2) + ((0,057)^2) + ((0,057)^2) + ((0,111)^2) + ((0,057)^2) + ((0,057)^2) + ((0,111)^2) + ((0,057)^2) + ((0,057)^2) + ((0,111)^2) + ((0,111)^2) + ((0,057)^2) + ((0,057)^2)$$

$$\text{Energy} = 0,003 + 0,003 + 0,003 + 0,003 + 0,012 + 0,003 + 0,003 + 0,012 + 0,003 + 0,003 + 0,012 + 0,012 + 0,003 + 0,003$$

$$\text{Energy} = \mathbf{0,078}$$

c. Menghitung nilai Homogeneity

Homogeneity =

$$\begin{aligned} & ((0,057)/1 + (180-181)) + ((0,057)/1 + (180-183)) + ((0,057)/1 + (181-180)) + \\ & ((0,057)/1 + (181-182)) + ((0,111)/1 + (181-183)) + ((0,057)/1 + (182-181)) + \\ & ((0,057)/1 + (182-182)) + ((0,111)/1 + (182-183)) + ((0,057)/1 + (182-185)) + \\ & ((0,057)/1 + (183-180)) + ((0,111)/1 + (183-181)) + ((0,111)/1 + (183-182)) + \\ & ((0,057)/1 + (183-183)) + ((0,057)/1 + (185-182)) \end{aligned}$$

Homogeneity =

$$\begin{aligned} & 0,057 + -1 + 0,057 + -3 + 0,057 + 1 + 0,057 + -1 + 0,111 + -2 + 0,057 + 1 + \\ & 0,057 + 0 + 0,111 + -1 + 0,057 + -3 + 0,057 + 3 + 0,111 + 2 + 0,111 + 1 + 0,057 \\ & + 0 + 0,057 + 3 \end{aligned}$$

Homogeneity = **1,014**

d. Menghitung nilai Entropy

Entropy =

$$\begin{aligned} & (0,057 \log 0,057) + (0,057 \log 0,057) + \\ & (0,057 \log 0,057) + (0,057 \log 0,057) + (0,111 \log 0,111) + \\ & (0,057 \log 0,057) + (0,057 \log 0,057) + (0,111 \log 0,111) + (0,057 \log 0,057) + \\ & (0,057 \log 0,057) + (0,111 \log 0,111) + (0,111 \log 0,111) + (0,057 \log 0,057) + \\ & (0,057 \log 0,057) \\ & = - 0,0709 + -0,0709 + \\ & -0,0709 + -0,0709 + -0,1059 + \\ & -0,0709 + -0,0709 + -0,1059 + -0,0709 + \\ & -0,0709 + -0,1059 + -0,1059 + -0,0709 + \\ & -0,0709 \end{aligned}$$

Entropy = **-1,1326**

Empat ciri yang di dapatkan dari matrix CO-Occurrence inilah yang akan digunakan untuk tahap klasifikasi.

4.2.4 Klasifikasi

Tabel 4.8: Hasil Training Data set

Data Training	X1	X2	X3	X4	Kelas
0	846.923	0.39773	0.541476	5.116527	Matang
1	460.09	0.204053	0.435188	6.2194	Matang
2	766.5394	0.238777	0.438966	6.270957	Matang
3	793.1135	0.249219	0.480671	5.803614	Matang
4	558.86677	0.2469538	0.4777707	5.873895	Matang
.....					
25	604.16987	0.2237504	0.3712644	6.461477	Mentah
26	476.34625	0.1636906	0.3778651	6.457394	Mentah
27	556.15086	0.2091028	0.4119541	6.450568	Mentah
28	639.73773	0.2107961	0.3906127	6.360824	Mentah
29	672.81456	0.1983076	0.3731912	6.697155	Mentah
.....					
Data Testing	101.72656	0.5457467	0.8240597	3.4306211	?

Untuk menentukan kelas dari data uji maka terlebih dahulu di tentukan nilai K=5. Setelah itu setiap jarak dari data latih ke data uji dihitung menggunakan rumus perhitungan jarak euclidean distance seperti dibawah ini:

$$\begin{aligned}
 d &= \sqrt{(846,9237 - 101,727)^2 + (0,39773 - 0,54575)^2} \\
 &\quad + (0,54148 - 0,82406)^2 + (5,11653 - 3,43062)^2 \\
 &= \sqrt{(745,196)^2 + (-0,14802)^2 + (-0,28258)^2 + (1,68591)^2} \\
 &= \sqrt{555.317,078416 + 0,0219099204 + 0,0798514564 + 2,8422925281} \\
 &= \sqrt{555.320,0224699} = 745.1984537
 \end{aligned}$$

Tabel 4.9: Perhitungan Jarak Euclidean Distance

Data Training	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{(846,9237 - 101,727)^2 + \dots (5,11653 - 3,43062)^2}$	745.1984537
1	$d = \sqrt{(460.09 - 101,727)^2 + \dots (6.2194 - 3,4306)^2}$	358.3746869

2	$d = \sqrt{(766.5394 - 101,727)^2 + \dots (6.270957 - 3,4306)^2}$	664.8191203
3	$d = \sqrt{(793.1135 - 101,727)^2 + \dots (5.803614 - 3,4306)^2}$	691.391126
4	$d = \sqrt{(558.86677 - 101,727)^2 + \dots (5.873895 - 3,4306)^2}$	457.1469707
...	...	
25	$d = \sqrt{(604.16987 - 101,727)^2 + \dots (6.461477 - 3,4306)^2}$	363.9680626
26	$d = \sqrt{(476.34625 - 101,727)^2 + \dots (6.457394 - 3,4306)^2}$	399.6762387
27	$d = \sqrt{(556.15086 - 101,727)^2 + \dots (6.450568 - 3,4306)^2}$	546.6944768
28	$d = \sqrt{(639.73773 - 101,727)^2 + \dots (6.360824 - 3,4306)^2}$	545.0805135
29	$d = \sqrt{(672.81456 - 101,727)^2 + \dots (6.697155 - 3,4306)^2}$	502.4527663
...	...	

Tabel 4.10: Hasil pengurutan dari nilai terkecil berdasarkan perhitungan jarak dari masing-masing data latih ke data Uji

Data	Jarak Ke Masing-masing Data Latih	Kelas
36	278.1381102	Mentah
13	288.2122132	Matang
16	314.7734160	Matang
48	329.8817425	Mentah
14	336.8540320	Matang
1	358.3746869	Matang
21	363.9680626	Matang
19	365.3757257	Matang
26	374.6323839	Mentah
40	396.7110182	Mentah
...	...	...

Tabel 4.11: Hasil Pengurutan dari nilai terkecil menggunakan K=5

Data	Jarak Ke Masing-masing Data Latih	Kelas
36	278.1381102	Mentah
13	288.2122132	Matang
16	314.7734160	Matang
48	329.8817425	Mentah
14	336.8540320	Matang

Kelas	Jumlah Data
Matang	3
Mentah	2

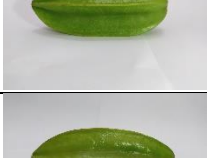
Jumlah data terbanyak menentukan kelas dari data uji

Data Training	X1	X2	X3	X4	Kelas
0	846.923	0.39773	0.541476	5.116527	Matang
1	460.09	0.204053	0.435188	6.2194	Matang
2	766.5394	0.238777	0.438966	6.270957	Matang
3	793.1135	0.249219	0.480671	5.803614	Matang
4	558.86677	0.2469538	0.4777707	5.873895	Matang
....					
25	604.16987	0.2237504	0.3712644	6.461477	Mentah
26	476.34625	0.1636906	0.3778651	6.457394	Mentah
27	556.15086	0.2091028	0.4119541	6.450568	Mentah
28	639.73773	0.2107961	0.3906127	6.360824	Mentah
29	672.81456	0.1983076	0.3731912	6.697155	Mentah
....					
Data Testing	101.72656	0.5457467	0.8240597	3.4306211	Matang

Hasil yang didapatkan dari perhitungan jarak euclidean distance untuk data uji adalah “Matang”

4.2.5 Evaluasi Model

Tabel 4.12: Hasil Pengenalan

Image Testing	Klasifikasi	Hipotesis
	Matang	Matang
	Matang	Matang
	Matang	Matang
	Matang	Matang
	Matang	Matang
	Mentah	Mentah
	Mentah	Mentah
	Mentah	Matang
	Mentah	Mentah

	Mentah	Matang
---	--------	--------

Tabel 4.13: Hasil Data Uji

Kelas	Matang	Mentah
Matang	5	0
Mentah	1	4

Jumlah data yang di klasifikasikan dengan benar = 9

Jumlah data yang di klasifikasikan salah = 1

$$Akurasi = \frac{9}{10} \times 100\% = 90\%$$

$$Presisi = \frac{5}{6} \times 100\% = 83\%$$

$$Presisi = \frac{5}{5} \times 100\% = 100\%$$

Keterangan :

True Positive (TP) = 5

True Negative (TN) = 4

False Positive (FP) = 1

False Negative (FN) = 0

BAB V

PEMBAHASAN

1.1 Pembahasan Sistem

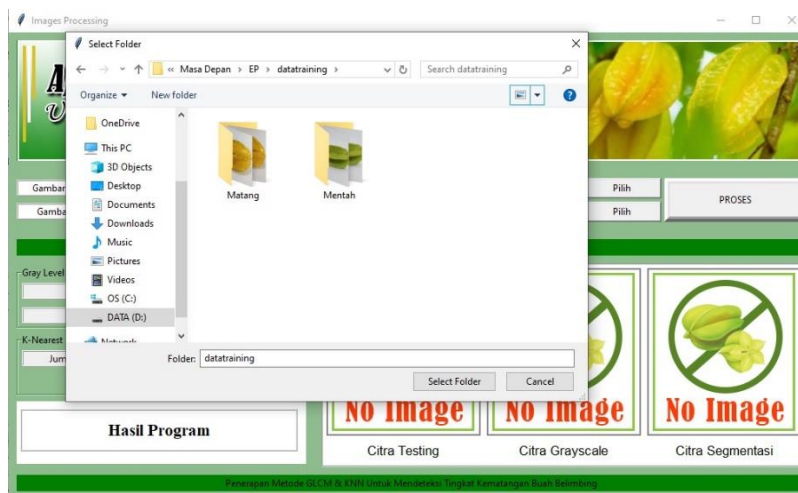
1. Tampilan Menu Utama

Form menu utama merupakan form utama yang muncul saat program dijalankan. Pada form menu utama menampilkan menu – menu yang ada pada sistem yaitu menu load training, menu load testing, dan proses.



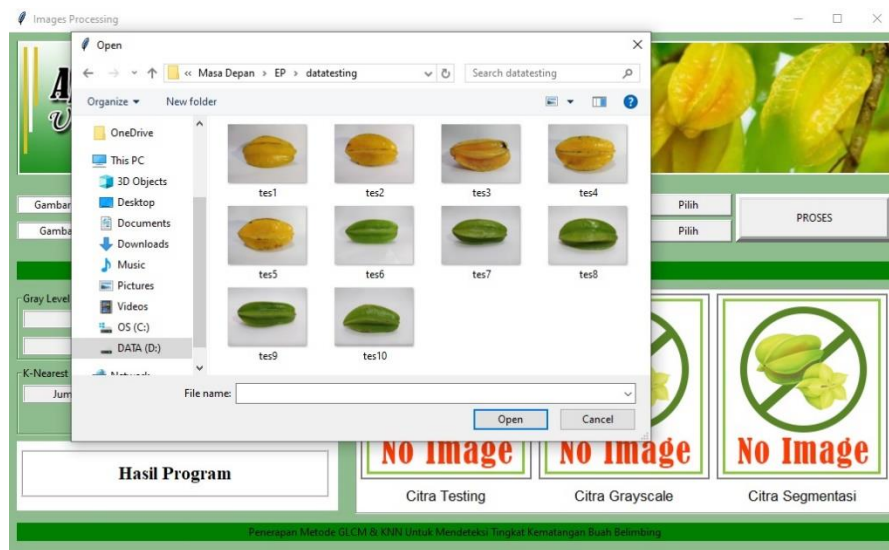
Gambar 5.1 Tampilan Menu Utama

2. Tampilan Proses Load Data Training



Gambar 5.2 Tampilan Proses Load Data Training

3. Tampilan Proses Load Data Testing



Gambar 5.3 Tampilan Proses Load Data Testing

4. Tampilan Hasil Pengenalan



Gambar 5.4 Tampilan Hasil Pengenalan

1.2 Pembahasan Model

Data yang digunakan dalam penelitian ini terdiri dari data real peneliti yang di dapat dengan mengambil gambar secara langsung dengan menggunakan kamera smarthphone Samsung A20 dengan resolusi kamera 13 MP. Masing masing 50 data training dan 10 data testing sehingga total dataset yang di gunakan pada penelitian ini yaitu 60 data set.

Berdasarkan hasil penelitian data yang di ambil terdiri dari 2 kategori yaitu Matang dan Mentah. Selanjutnya citra buah belimbing di ekstrasi ciri menggunakan *Gray Level Co-occurrence Matrix*. Setelah di dapatkan ciri dari citra buah belimbing selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra pada buah belimbing dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model K-NN.

Percobaan pertama dilakukan eksperimen menggunakan arah GLCM = 0° dengan jarak = 1, pada percobaan model K-NN dengan jumlah nilai K-1 mendapatkan nilai akurasi 30% dengan 10 data testing dan 50 data training, dengan kategori matang dan mentah dengan masing-masing 25 data. Selanjutnya percobaan model K-NN dengan jumlah nilai K-3 mendapatkan nilai akurasi sebesar 80% dan Percobaan menggunakan model K-NN dengan jumlah K-5 mendapatkan nilai akurasi tertinggi sebesar 90%.

Percobaan selanjutnya menggunakan arah GLCM = 45° dengan jarak = 1, pada percobaan model K-NN dengan jumlah nilai K-1 mendapatkan nilai akurasi 20%, Selanjutnya percobaan model K-NN dengan jumlah nilai K-3 mendapatkan nilai akurasi sebesar 50% dan Percobaan menggunakan model K-NN dengan jumlah K-5 mendapatkan nilai akurasi 50%.

Sedangkan pada percobaan menggunakan arah GLCM = 90° dengan jarak = 1, pada percobaan model K-NN K-1, K-2, dan K-3 Mendapatkan Nilai Akurasi yang sama yaitu sebesar 50%.

Percobaan terakhir dilakukan eksperimen menggunakan arah GLCM = 135° dengan jarak = 1, pada percobaan model K-NN dengan jumlah nilai K-1

mendapatkan nilai akurasi 30%. percobaan model K-NN dengan jumlah nilai K-3 mendapatkan nilai akurasi sebesar 80% dan Percobaan menggunakan model K-NN dengan jumlah K-5 mendapatkan nilai akurasi tertinggi sebesar 90% dengan menggunakan dataset yang sama seperti dalam percobaan model sebelumnya.

Pada tahap Confusion Matrix terdapat kendala pengenalan tingkat kematangan, beberapa data keluar dengan hasil yang tidak sesuai. Penambahan jumlah nilai K sangat berpengaruh untuk akurasi tingkat kematangan buah belimbing bintang. Hasil yang di dapatkan dari pengujian metode GLCM dan K-NN akurasi yang di peroleh sudah cukup baik, dari perhitungan hasil akurasi di atas bahwa semakin banyak nilai K semakin tinggi nilai akurasi yang di peroleh sedangkan percobaan penambahan nilai dari Arah dan Jarak GLCM menunjukan perubahan nilai akurasi yg tidak stabil . Percobaan pada Jumlah K-5 menggunakan arah GLCM = 0° dan 135° dengan jarak = 1 mendapatkan nilai akurasi Tertinggi. Dari hasil perhitungan akurasi untuk pengujian Klasifikasi mendapatkan nilai akurasi 90%. Pada penelitian ini penggunaan metode K-NN sangat baik untuk tahap klasifikasi.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik kesimpulan bahwa:

Kinerja model deteksi tingkat kematangan buah belimbing bintang menggunakan metode *K-Nearest Neighbor* dengan fitur ekstraksi *Gray Level Co-occurrence Matrix* setelah diukur menggunakan *Confusion Matrix* menghasilkan akurasi sebesar 90%.

6.2 Saran

1. Untuk penggunaan metode *K-Nearest Neighbor* pada deteksi tingkat kematangan buah belimbing bintang disarankan untuk menggunakan jumlah dataset yang banyak, agar pola tingkat akurasi semakin baik.
2. Sistem deteksi tingkat kematangan buah belimbing bintang ini dapat dikembangkan dengan model pengenalan secara *realtime*.
3. Perlu diadakan pengembangan dengan penggunaan metode lain seperti *Gray Level Co-occurrence Matrix* dan *Artificial Neural Network*, *Support Vector Machine* atau metode klasifikasi lainnya sebagai perbandingan untuk mendapatkan hasil terbaik.

DAFTAR PUSTAKA

- [1] D. R. Radityo, M. R. Fadillah, Q. Igwahyudi, and S. Dewanto, “Alat Penyortir Dan Pengecekan Kematangan Buah,” vol. 20, no. 2, pp. 88–92.
- [2] F. Y. Mulato, “Klasifikasi Kematangan Buah Jambu Biji Merah (Psidium Guajava) Dengan Menggunakan Model Fuzzy,” 2015.
- [3] B. E. Permadi, “Rancang Bangun Alat Sortir Kematangan Buah Belimbing Berdasarkan Ukuran Dan Warna Dengan Mikrokontroler Arduino.”
- [4] F. Y. Manik and K. S. Saragih, “Klasifikasi Belimbing Menggunakan Naïve Bayes Berdasarkan Fitur Warna RGB,” vol. 11, no. 1, 2017.
- [5] R. Widodo, A. W. Widodo, and A. Supriyanto, “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu,” vol. 2, no. 11, pp. 5769–5776, 2018.
- [6] D. T. Hermanto, S. Megira, and D. Ninosari, “Identifikasi Tingkat Kematangan Buah Strawberry Menggunakan Gray Level Co-Occurance Matrix (Glcm) Dan Laplacian Filter,” pp. 25–30, 2018.
- [7] P. R. Suhendri, “Metode Grayscale Co-occurrence Matrix (GLCM) Untuk Klasifikasi Jenis Daun Jambu Air Menggunakan Algoritma Neural Network,” vol. 01, no. 01, pp. 15–22, 2019.
- [8] M. M. Siti Mutrofin, Abidatul Izzah, Arrie Kurniawardhani, “Optimasi Teknik Klasifikasi Modified K Nearest Neighbor Menggunakan Algoritma Genetika,” no. September, pp. 130–134, 2014.
- [9] P. Rianto and A. Harjoko, “Penentuan Kematangan Buah Salak Pondoh Di Pohon Berbasis Pengolahan Citra Digital,” vol. 11, no. 2, pp. 143–154, 2017.
- [10] I. A. Halela, B. Nurhadiyono, S. Si, M. Kom, and F. Z. Rahmanti, “Identifikasi Jenis Buah Apel Menggunakan Algoritma K-Nearest Neighbor (KNN) dengan Ekstraksi Fitur Histogram,” pp. 1–8.
- [11] M. K. Achdiyat Risvana, Prajanto Wahyu Adi, M. Kom, Setia Astuti, S. Si, “Klasifikasi Tingkat Kematangan Buah Jambu Biji Merah Dengan Algoritma K-Nearest Neighbor,” no. x, pp. 1–9, 2012.
- [12] Iswahyudi Catur, “Prototype Aplikasi Untuk Mengukur Kematangan Buah Apel,” vol. 3, pp. 107–112, 2010.

- [13] N. Nafi'iyah, "Algoritma Kohonen Dalam Mengubah Citra Graylevel Menjadi Citra Biner," vol. 9, no. 2, pp. 49–55, 2015.
- [14] B. Citra *et al.*, "Binerisasi Citra Tangan Dengan Metode Otsu," vol. 3, no. 2, pp. 11–13, 2004.
- [15] R. D. Kusumanto and A. N. Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB," vol. 2011, no. Semantik, 2011.
- [16] M. T. I Putu Gangga Sugi Pradnyana, Ledya Novamizanti, S.Si., M.T., Hilman Fauzi, S.T., "Perancangan Sistem Pendeteksi Genangan Air Potensi Perkembangbiakan Nyamuk Melalui Foto Citra Udara Dengan Metode Gray Level Co-Occurrence Matrix (GLCM)," vol. 2, no. 2, pp. 2977–2984, 2015.
- [17] F. Shofrotun, T. Sutojo, D. R. Ignatius, and M. Setiadi, "Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurence Matrix dan K-Nearest Neighbor," vol. 6, no. November 2017, pp. 51–56, 2018.
- [18] R. Munarto, E. Permata, and R. Salsabilla, "Klasifikasi Kualitas Biji Jagung Manis Berdasarkan Fitur Warna Menggunakan Fuzzy Logic," pp. 5–12, 2014.
- [19] M. A. Naufal, "Implementasi Metode Klasifikasi K-Nearest Neighbor (K-Nn) Untuk Pengenalan Pola Batik Motif Lampung," 2017.
- [20] D. Sartika and D. I. Sensuse, "Perbandingan Algoritma Klasifikasi Naive Bayes , Nearest Neighbour , dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian," vol. 1, no. 2, pp. 151–161, 2017.
- [21] H. Leidiyana, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," vol. 1, no. 1, pp. 65–76, 2013.
- [22] E. P. Satrio, T. Sutojo, S. Si, and M. Kom, "Klasifikasi Tenun Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-Occurrence Matrices (Glcm)," no. 5.
- [23] H. Wijayanto, "Klasifikasi Batik Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-occurrence Matrices (GLCM)," no. 5.
- [24] R. A. Surya, A. Fadlil, and A. Yudhana, "Ekstraksi Ciri Citra Batik Berdasarkan Tekstur Menggunakan Metode Gray Level Co Occurrence

- Matrix,” vol. 2, no. 1, pp. 146–150, 2016.
- [25] J. S. A. Arwansyah, “Implementasi Algoritma KNN Dalam Memprediksi Curah Hujan dan Temperatur Untuk Tanaman Padi,” vol. VIII, no. 1, pp. 11–20, 2019.
- [26] E. T. L. Kaharuddin, Kusrini, “Klasifikasi Jenis Rempah-Rempah Berdasarkan Fitur Warna Rgb Dan Tekstur Menggunakan Algoritma K-Nearest Neighbor,” vol. 4, no. 1, 2019.
- [27] E. R. Paramita Mayadewi, “Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining,” no. November, pp. 2–3, 2015.
- [28] M. F. Rahman, M. I. Darmawidjaja, and D. Alamsah, “Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN),” vol. 11, pp. 36–45, 2017.

Lampiran 1: KODE PROGRAM

1. Pembuatan Form

```
import tkinter as tk

import tkinter.filedialog as fdialog

import tkinter.ttk as ttk

from PIL import Image, ImageTk

import glcm

import cv2

class aplikasiku(tk.Frame):

    def __init__(self, parent):

        self.parent = parent

        self.frames()

        self.widget()

    def frames(self):

#frame judul

        self.frame1 = tk.Frame(self.parent, width=1004, height=150, borderwidth=0,
background="green", relief='raised')

        self.frame1.grid(row=0, column=0, padx=10, pady=10, columnspan=2,
sticky="NSEW")

        self.frame1.grid_propagate(0)

#frame input gambar

        self.frame2 = tk.Frame(self.parent, width=1024, height=80, borderwidth=0,
padx=10, pady=10, background="darkseagreen", relief='raised')
```

```

self.frame2.grid(row=1, column=0, columnspan=2, sticky="NSEW")

self.frame2.grid_propagate(0)

#frame teks proses

self.frame3 = tk.Frame(self.parent, width=1024, height=25, borderwidth=0,
background="green", relief='raised')

self.frame3.grid(row=2, column=0, columnspan=2, padx=10, pady=10,
sticky="NSEW")

self.frame3.grid_propagate(0)

#frame prapengolahan

self.frame4 = tk.LabelFrame(self.parent, height=80, width=370,
bg="darkseagreen", text="Gray Level Co-Occurrence Matrix", fg="black")

self.frame4.grid(row=3, column=0, padx=10, sticky="W")

self.frame4.grid_propagate(0)

self.frame5 = tk.LabelFrame(self.parent, height=80, width=370,
bg="darkseagreen", text="K-Nearest Neighbor", fg="black")

self.frame5.grid(row=4, column=0, padx=10, sticky="W")

self.frame5.grid_propagate(0)

self.frame7 = tk.LabelFrame(self.parent, height=260, width=620,
bg="white")

self.frame7.grid(row=3, column=1, rowspan=3, padx=10, sticky="E")

self.frame7.grid_propagate(0)

#frame hasil akhir

self.frame6 = tk.Frame(self.parent, height=80, width=370, bg="white")

self.frame6.grid(row=5, column=0, padx=10, sticky="W")

self.frame6.grid_propagate(0)

```

```
#frame judul penelitian
```

```
self.frame8 = tk.Frame(self.parent, width=1000, height=25, borderwidth=0,  
background="green", relief='raised')
```

```
self.frame8.grid(row=6, column=0, columnspan=3, padx=10, pady=10,  
sticky="NSEW")
```

```
self.frame8.grid_propagate(0)
```

```
def widget(self):
```

```
#frame 1
```

```
=====
```

```
self.PathTestingImg ='D:\Masa Depan\EP\Logo.jpg'
```

```
self.imT = Image.open(self.PathTestingImg)
```

```
self.imT = ImageTk.PhotoImage(self.imT)
```

```
self.label_dataTrain = tk.Label(self.frame1, width=1004, height=150,  
background="white", image=self.imT)
```

```
self.label_dataTrain.pack(side="top")
```

```
#frame 2
```

```
=====
```

```
#data training
```

```
#label
```

```
self.lbl_dataTrain = tk.Label(self.frame2, text='Gambar Latih', relief='ridge',  
width=15, bg="white")
```

```
self.lbl_dataTrain.grid(row=1, column=0, sticky="W")
```

```
#entry
```

```

value_datatraining = tk.StringVar(self.frame2, value="")

self.ent_dataTrain = tk.Entry(self.frame2, textvariable=value_datatraining,
relief='ridge', width=100)

self.ent_dataTrain.grid(row=1, column=1, padx=3, pady=3, sticky="W")

#button

self.button_datatrain = tk.Button(self.frame2, text='Pilih', width=13,
command=lambda:value_datatraining.set(fdialog.askdirectory()))

self.button_datatrain.grid(row=1, column=2, padx=2, pady=2, sticky="W")

#data testing

#label

self.lbl_dataTest = tk.Label(self.frame2, text='Gambar Uji', relief='ridge',
width=15, bg="white")

self.lbl_dataTest.grid(row=2, column=0, sticky="W")

#entry

value_datatesting = tk.StringVar(self.frame2, value="")

self.ent_dataTest = tk.Entry(self.frame2, textvariable=value_datatesting,
relief='ridge', width=100)

self.ent_dataTest.grid(row=2, column=1, padx=3, pady=3, sticky="W")

#button

self.button_datatesting = tk.Button(self.frame2, text='Pilih', width=13,
command=lambda:value_datatesting.set(fdialog.askopenfilename()))

self.button_datatesting.grid(row=2, column=2, padx=2, pady=2, sticky="W")

#proses

#button

```

```
self.button_proses = tk.Button(self.frame2, text='PROSES', width=23,
command=self.proses)
```

```
self.button_proses.grid(row=1, column=3, rowspan=2, padx=2, pady=2,
sticky="NSEW")
```

```
self.button_proses.configure(borderwidth=7, relief="raised",
anchor="center")
```

```
#frame 3 Teks Proses Perhitungan
```

```
=====
```

```
self.lbl_text = tk.Label(self.frame3, text="Proses Ekstraksi dan Klasifikasi",
background="green", anchor="center")
```

```
self.lbl_text.grid()
```

```
self.lbl_text.pack()
```

```
#frame 4 GLCM
```

```
=====
```

```
#label
```

```
self.lbl_Arah = tk.Label(self.frame4, text='Arah', relief='ridge', width=20)
```

```
self.lbl_Arah.grid(row=1, column=0, padx=5, pady=5, sticky="W")
```

```
#combobox
```

```
self.valdist = tk.StringVar(self.frame4, value="")
```

```
self.cb_Arah = ttk.Combobox(self.frame4, textvariable=self.valdist,
width=30, state='readonly')
```

```
self.cb_Arah['values']=('0','45','90','135')
```

```
self.cb_Arah.current(0)
```

```
self.cb_Arah.grid(row=1, column=2, padx=2, pady=2, sticky="W")
```

```
#label
```

```
self.lbl_Jarak = tk.Label(self.frame4, text='Jarak', relief='ridge', width=20)
```

```

self.lbl_Jarak.grid(row=2, column=0, padx=5, pady=5, sticky="W")

#entry

value_jarak = tk.StringVar(self.frame4, value="")

self.ent_jarak = tk.Entry(self.frame4, textvariable=value_jarak, relief='ridge',
width=33)

self.ent_jarak.grid(row=2, column=2, padx=3, pady=3, sticky="W")

#frame 5 KNN
=====

self.lbl_nilaiK = tk.Label(self.frame5, text='Jumlah Nilai K', relief='ridge',
width=20)

self.lbl_nilaiK.grid(row=1, column=0, padx=5, pady=5, sticky="W")

self.valdist = tk.StringVar(self.frame5, value="")

self.cb_nilaiK = ttk.Combobox(self.frame5, textvariable=self.valdist,
width=30, state='readonly')

self.cb_nilaiK['values']=('1','2','3','4','5')

self.cb_nilaiK.current(0)

self.cb_nilaiK.grid(row=1, column=2, padx=2, pady=2, sticky="W")

#form 7 Gambar
=====

self.asli = 'noimage.png'

self.noimageAsli = Image.open(self.asli)

self.noimageAsli = self.noimageAsli.resize((193, 210), Image.ANTIALIAS)

self.noimageAsli = ImageTk.PhotoImage(self.noimageAsli)

self.CitraAsli = tk.Label(self.frame7, bg="gray", image=self.noimageAsli)

self.CitraAsli.grid(row=1, column=0, padx=4, pady=4, sticky="NSEW")

```

```

self.Citrateks = tk.Label(self.frame7, text="Citra Testing", font="arial 13 ",
background="white")

self.Citrateks.grid(row=2, column=0, padx=3, pady=3)

self.gray = 'noimage.png'

self.noimageGray = Image.open(self.asli)

self.noimageGray = self.noimageGray.resize((193, 210),
Image.ANTIALIAS)

self.noimageGray = ImageTk.PhotoImage(self.noimageGray)

self.CitraGray = tk.Label(self.frame7, bg="gray", image=self.noimageGray)

self.CitraGray.grid(row=1, column=1, padx=4, pady=4)

self.Citrateks = tk.Label(self.frame7, text="Citra Grayscale", font="arial 13
", background="white")

self.Citrateks.grid(row=2, column=1, padx=3, pady=3)

self.hist = 'noimage.png'

self.noimageHist = Image.open(self.asli)

self.noimageHist = self.noimageHist.resize((193, 210), Image.ANTIALIAS)

self.noimageHist = ImageTk.PhotoImage(self.noimageHist)

self.CitraHist = tk.Label(self.frame7, bg="gray", image=self.noimageHist)

self.CitraHist.grid(row=1, column=2, padx=4, pady=4)

self.Citrateks = tk.Label(self.frame7, text="Citra Segmentasi", font="arial 13
", background="white")

self.Citrateks.grid(row=2, column=2, padx=3, pady=3)

```

```

#frame 8 text judul

```

```

=====

```

```

        self.lbl_text = tk.Label(self.frame8, text="Penerapan Metode GLCM & KNN
Untuk Mendeteksi Tingkat Kematangan Buah Belimbing", background="green",
anchor="center")

```

```

        self.lbl_text.grid()

```

```

        self.lbl_text.pack()

```

```

#frame 6 hasil akhir
=====

```

```

        self.lbl_hasil = tk.Label(self.frame6, text="Hasil Program", font="Times 16
bold", relief="ridge", height=2, width=29, background="white")

```

```

        self.lbl_hasil.grid(row=1, columnspan=3, sticky="N", padx=5, pady=10)

```

```

def proses(self):

```

```

    foldertraining = self.ent_dataTrain.get()

```

```

    imagetesting = self.ent_dataTest.get()

```

```

    arah = self.cb_Arah.get()

```

```

    jarak = self.ent_jarak.get()

```

```

    K = self.cb_nilaiK.get()

```

```

    a = glcm.analisis_citra()

```

```

    hasil, img_segmen, gray, = a.proses_rekognisi(foldertraining, imagetesting,
arah, jarak, K)

```

```

    self.lbl_hasil.configure(text=hasil)

```

```

    imgAsli = Image.open(self.ent_dataTest.get())

```

```

    imgAsli = imgAsli.resize((193, 210), Image.ANTIALIAS)

```

```

    imgAsli = ImageTk.PhotoImage(imgAsli)

```

```

    self.CitraAsli.configure(image=imgAsli)

```

```

    self.CitraAsli.image_names = imgAsli

```

```

imgGray = Image.fromarray(gray)

imgGray = imgGray.resize((193, 210), Image.ANTIALIAS)

imgGray = ImageTk.PhotoImage(imgGray)

self.CitraGray.configure(image=imgGray)

self.CitraGray.image_names = imgGray

imgHist = cv2.cvtColor(img_segmen, cv2.COLOR_BGR2RGB)

imgHist = Image.fromarray(imgHist)

imgHist = imgHist.resize((193, 210), Image.ANTIALIAS)

imgHist = ImageTk.PhotoImage(imgHist)

self.CitraHist.configure(image=imgHist)

self.CitraHist.image_names = imgHist

def main():

    root = tk.Tk()

    root.geometry('1024x600+50+50')

    root.configure(background='darkseagreen')

    app = aplikasiku(root)

    app.parent.title("Images Processing")

    root.mainloop()

if __name__ == '__main__':

    main()

```

2. Image Processing

```
import cv2
import numpy as np
from skimage import feature
from scipy.stats import entropy
from imutils import paths
import os
from scipy import stats
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA

class analisis_citra:
    def segmentasi(self, img):
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray,(5,5),0)
        ret, alpha = cv2.threshold(np.uint8(blur), 0, 255,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
        alpha[alpha==255] = 1
        rect = cv2.boundingRect(alpha)
        r = img[:, :, 0] * alpha
        g = img[:, :, 1] * alpha
        b = img[:, :, 2] * alpha
        hasil = cv2.merge((r,g,b))
        x,y,w,h = rect
        result = hasil[y:y+h, x:x+w]
        return result, gray

    def fitur_glcmm(self, img, Arah, Jarak):
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```

matrix_glcmm = feature.greycomatrix(img, [Jarak], [Arah],
symmetric=False, normed=True)

contrast = feature.greycomprops(matrix_glcmm, 'contrast')[0,0]
energy = feature.greycomprops(matrix_glcmm, 'energy')[0,0]
homogeneity = feature.greycomprops(matrix_glcmm, 'homogeneity')[0,0]
ent = entropy(matrix_glcmm.flatten())
hasil_fitur = [contrast, energy, homogeneity, ent]
return hasil_fitur

def get_img_path(self, folderPath):
    imPath = []
    for path in paths.list_images(folderPath):
        imPath.append(path)
    return imPath

def get_label(self, folderPath):
    label = []
    imPath = self.get_img_path(folderPath)
    for item in imPath:
        label.append(os.path.split(os.path.dirname(item))[-1])
    return label

def training_data(self, folderPath, Arah, Jarak):
    pathList = self.get_img_path(folderPath)
    images = []
    for imPath in pathList:
        img = cv2.imread(imPath)
        img = cv2.resize(img, (255, 255))
        images.append(img)
        data_train = []
    for img in images:

```

```

        hasil_segmen, gray = self.segmentasi(img)
        ekstrasi = self.fitur_glcml(hasil_segmen, Arah, Jarak)
        data_train.append(ekstrasi)
    label_train = self.get_label(folderPath)
    return data_train, label_train

def testing_data(self, img, Arah, Jarak):
    data_test = []
    img = cv2.imread(img)
    hasil_segmen, gray = self.segmentasi(img)
    hasil_ekstrak_glcml = self.fitur_glcml(hasil_segmen, Arah, Jarak)
    data_test.append(hasil_ekstrak_glcml)
    return data_test, hasil_segmen, gray

def encode_data(self, data):
    scaler = MinMaxScaler(feature_range=(0,1))
    hasil_transform = scaler.fit_transform(data)
    return hasil_transform

def proses_rekognisi(self, folder_train, image_test, Arah, Jarak, K):
    data_train, y_train = self.training_data(folder_train, Arah, Jarak)
    data_test, hasil_segmen, gray = self.testing_data(image_test, Arah, Jarak)
    allData = self.encode_data(data_train + data_test)
    pca = PCA(n_components = 2)
    pca.fit(allData)
    data_decomposition = pca.transform(allData)
    x_train = data_decomposition[0:-int(len(data_test))]
    x_test = data_decomposition[-int(len(data_test)):]
    clf = KNeighborsClassifier(n_neighbors=int(K), weights='uniform',
                              algorithm='auto', metric='euclidean')
    clf.fit(x_train, y_train)

```

```
final_hasil = clf.predict(x_test)
return final_hasil[0], hasil_segmen, gray
```

Lampiran 2: RIWAYAT HIDUP PENELITI



Nama : Qurnia Shandy
Nim : T3116036
Tempat, Tanggal Lahir : Sidomulyo, 24 Oktober 1998
Email : gurniashandy9@gmail.com

Riwayat pendidikan :

1. Tahun 2010, menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Sidomulyo, sekarang SDN 3 Boliyohuto, Kecamatan Boliyohuto Kabupaten Gorontalo
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Menengah Pertama 1 Boliyohuto, Kecamatan Boliyohuto Kabupaten Gorontalo
3. Tahun 2015, menyelesaikan Pendidikan di Sekolah Menengah Kejuruan 1 Mootilango, Kecamatan Mootilango, Kabupaten Gorontalo
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.

Lampiran 3: SURAT REKOMENDASI PENELITIAN



PEMERINTAH KABUPATEN GORONTALO
KECAMATAN BOLIYOHUTO

Jln. Raja Tolangohula Desa Sidomulyo

REKOMENDASI

Nomor : 420 / 122 / K.Bol / VII/2020

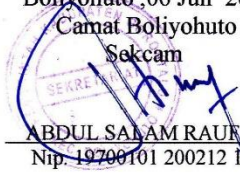
Berdasarkan Surat Kementerian Riset , Teknologi dan Pendidikan Tinggi, Lembaga Penelitian (Lemlit) Universitas Ichsan Gorontalo Nomor : 1808/PIP/ LEMLIT – UNISAN /GTLO/XI / 2019 Tanggal 12 November 2019 Perihal Permohonan Izin Penelitian dengan ini kami memberikan Rekomendasi Kepada :

Nama : **Qurnia Shandy**
NIM : T3116036
Fakultas : Fakultas Ilmu Komputer
Program Studi : Teknik Informatika
Lokasi Penelitian : Kecamatan Boliyohuto Kabupaten Gorontalo
Judul Penelitian : Penerapan Metode GLCM (GRAY LEVEL COOCCURENCE MATRIX) dan KNN (K- NEAREST NEIGHBOR) Untuk mendeteksi Tingkat Kematangan Buah Belimbing Bintang .

Untuk melakukan Penelitian sehubungan dengan Penulisan / Penyusunan Skirpsi pada Program Studi S1 Teknik Informatika Fakultas Ilmu Komputer universitas Ichsan Gorontalo. Demikian Permohonan ini, atas perhatian dan Kerjasamanya di sampaikan terima kasih

Boliyohuto, 06 Juli 2020

Camat Boliyohuto
Sekcam


ABDUL SALAM RAUF S.Pt
Nip. 19700101 200212 1013

Lampiran 4: BUKTI PENERIMAAN SOFTCOPY SKRIPSI



Pustikom
Universitas Ichsan Gorontalo

BUKTI PENERIMAAN SOFTCOPY SKRIPSI **PENGECEKAN SIMILARITY TURNITIN**

Nama Mahasiswa : QURNIA SHANDY
NIM : T3116036
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Penerapan Metode GLCM (Gray Level Co- Occurrence Matrix) Dan K-NN (K-Nearest Neighbor) Untuk Mendeteksi Tingkat Kematangan Buah Belimbing Bintang

Nama File (Pdf) : SKRIPSI - T3116036 - QURNIA SHANDY - PENERAPAN_2020
No. HP/WA : 082266375527
e-Mail : qurnia.shandy9@gmail.com

Tgl. Terima :

23/6/2020					
24%					

Hasil Pengecekan :

Diterima/Diperiksa Oleh,


Sudirman S. Panna, M.Kom
085340910769

Lampiran 5: SURAT PERNYATAAN



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO**

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

1. Nama : Sudirman S. Panna, M.Kom
Sebagai : Pembimbing I
2. Nama : Yusrianto Malago, M.Kom
Sebagai : Pembimbing II

Dengan ini Menyatakan bahwa :

- Nama Mahasiswa : QURNIA SHANDY
NIM : T3116036
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Penerapan Metode GLCM (Gray Level Co- Occurrence Matrix) Dan K-NN (K-Nearest Neighbor) Untuk Mendeteksi Tingkat Kematangan Buah Belimbing Bintang

Setelah kami melakukan pengecekan kembali antara softcopy skripsi dari hasil pemeriksaan aplikasi Turnitin dengan hasil Similarity sebesar 24% oleh Tim Verifikasi Plagiasi di Pustikom dengan Skripsi Aslinya, isinya SAMA dan format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi.

Demikian surat pernyataan ini dibuat untuk mendapatkan Surat Rekomendasi Bebas Plagiasi.

Pembimbing I

Sudirman S. Panna, M.Kom
NIDN. 0924038205

Gorontalo, Juni 2020
Pembimbing II

Yusrianto Malago, M.Kom
NIDN. 0909108901

Mengetahui
Ketua Program Studi,

Irvan A. Salih, M.Kom
NIDN. 0928028101

Catatan Perbaikan :

- ☐ Penggunaan tanda petik dua tidak Wajar
- ☐ Penulisan Rumus masih berbentuk gambar
- ☐ Beberapa Paragraf berbentuk gambar
- ☐ Beberapa kata tidak lengkap hurufnya / beberapa kata digabung tanpa spasi
- ☐ _____

Lampiran 6: SURAT REKOMENDASI BEBAS PLAGIASI



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO**

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0264/UNISAN-G/S-BP/IV/2020

Yang bertanda tangan di bawah ini:

Nama : Sunarto Taliki, M.Kom
NIDN : 0906058301
Unit Kerja : Pustikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasisw : QURNIA SHANDY
NIM : T3116036
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Penerapan Metode GLCM (Gray Level Co- Occurrence Matrix) Dan K-NN (K-Nearest Neighbor) Untuk Mendeteksi Tingkat Kematangan Buah Belimbing Bintang

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 24%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 25 Juni 2020
Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip

Lampiran 7: HASIL TURNITIN

PENERAPAN METODE GLCM (GRAY LEVEL CO-OCCURRENCE MATRIX) DAN K-NN (K-NEAREST NEIGHBOR) UNTUK MENDETEKSI TINGKAT KEMATANGAN BUAH BELIMBING BINTANG

ORIGINALITY REPORT

24%	20%	8%	20%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas 17 Agustus 1945 Surabaya Student Paper	2%
2	docplayer.info Internet Source	2%
3	Submitted to Universitas Brawijaya Student Paper	2%
4	eprints.dinus.ac.id Internet Source	2%
5	www.scribd.com Internet Source	2%
6	j-ptiik.ub.ac.id Internet Source	1%
7	digilib.unila.ac.id Internet Source	1%

jurnal.mdp.ac.id

8	Internet Source	1%
9	Submitted to Sriwijaya University Student Paper	1%
10	teknotugasku.blogspot.com Internet Source	1%
11	mti.uad.ac.id Internet Source	1%
12	publikasiilmiah.ums.ac.id Internet Source	1%
13	journal.ugm.ac.id Internet Source	1%
14	ifanefendi24.blogspot.com Internet Source	1%
15	voi.stmik-tasikmalaya.ac.id Internet Source	1%
16	id.123dok.com Internet Source	1%
17	Submitted to Udayana University Student Paper	<1%
18	Submitted to Universitas Diponegoro Student Paper	<1%
19	repository.ubaya.ac.id Internet Source	<1%

20	maghfirahmaulani82.blogspot.com Internet Source	<1 %
21	lp2m.asia.ac.id Internet Source	<1 %
22	Submitted to Politeknik Negeri Jember Student Paper	<1 %
23	Submitted to Universitas Dian Nuswantoro Student Paper	<1 %
24	Submitted to Aston University Student Paper	<1 %
25	pemrogramanmatlab.wordpress.com Internet Source	<1 %
26	Budy Santoso, Azminuddin I. S. Azis, Andi Bode. "Pengendalian Lampu Lalu Lintas Cerdas di Persimpangan Empat Ruas yang Kompleks Menggunakan Algoritma Adaptive Neuro Fuzzy Inference System", Jurnal Edukasi dan Penelitian Informatika (JEPIN), 2020 Publication	<1 %
27	Submitted to Myongji University Graduate School Student Paper	<1 %
28	ayuniindya.wordpress.com Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 25 words