

**PENGENALAN KUALITAS CABAI MENGGUNAKAN
METODE *GRAY LEVEL CO-OCCURRENCE MATRIX*
(GLCM) DAN *K-NEAREST NEIGHBOR* (K-NN)**

Oleh

ARYO ADI PRASETYO MOKOAGOW

T3115163

SKRIPSI



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

PENGESAHAN SKRIPSI

**Pengenalan Kualitas Cabai Menggunakan
Metode *GRAY LEVEL CO-OCCURRENCE MATRIX*
(GLCM) dan *K-NEAREST NEIGHBOR (K-NN)***

Oleh

ARYO ADI PRASETYO MOKOAGOW

T3115163



SKRIPSI

Untuk memenuhi salah satu syarat ujian guna memperoleh gelar sarjana program studi Teknik Informatika, ini telah disetujui oleh Tim Pembimbing

Gorontalo, 28 Juli 2020

Pembimbing I


Zohrahayati, M.Kom
NIDN. 0912117702

Pembimbing II


Hastuti Daldi, M.Kom
NIDN. 0918038803

PERSETUJUAN SKRIPSI

Pengenalan Kualitas Cabai Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (KNN)

Oleh

ARYO ADI PRASETYO MOKOAGOW

T3115163

Diperiksa oleh Panitia Ujian Strata Satu (S1)

Universitas Ihsan Gorontalo

Gorontalo, 06 Agustus 2020

1. Pembimbing I
Zohrahayaty, M.Kom
2. Pembimbing II
Hastuti Dalai, M.Kom
3. Penguji I
Haditsa Annur, M.Kom
4. Penguji II
Sudirman S. Panna, M.Kom
5. Penguji III
Kartika C. Pelangi, M.Kom

Mengetahui:

Dekan Fakultas Ilmu Komputer

Ketua Program Studi


Zohrahayaty, M.Kom
NIDN. 0912117702


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 dipublikasikan 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 diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 28 Juli 2020

Yang Membuat Pernyataan



Aryo Adi Prasetyo Mokoagow

ABSTRACT

The introduction of the quality of chili peppers is a major and very important thing that affects the value of selling price and the use of chili itself. The quality of good chili has a bright red color, as well as having a suitable spicy level. The determination of the quality of chili pepper can be solved using Gray Level CO-Occurrence Matrix method and K-Nearest Neighbor classification. The results showed that the accuracy of the method of Gray Level CO-Occurrence Matrix with a angle of 0° , with a distance of $= 1$ and a value of $k = 7$ on K-NN classification for the introduction of chili quality with the highest accuracy of 80%, with 90 training data and testing 30 chili image data. The result of the calculation above that the value of K with odd numbers provides good accuracy.

Keywords: Quality Chili, Gray Level CO-Occurrence Matrix, K-Nearest-Neighbor

ABSTRAK

Pengenalan kualitas cabai merupakan hal yang utama dan sangat penting yang berpengaruh pada nilai harga jual dan penggunaan cabai itu sendiri. Kualitas cabai yang baik itu memiliki warna merah terang, serta memiliki tingkat pedas yang sesuai. Penentuan kualitas cabai dapat diselesaikan dengan menggunakan metode *Gray Level CO-Occurrence Matrix* dan *Klasifikasi K-Nearest Neighbor*. Hasil penelitian ini menunjukkan bahwa akurasi metode *Gray Level CO-Occurrence Matrix* dengan sudut 0° , dengan jarak =1 dan nilai $K=7$ pada klasifikasi *K-NN* untuk pengenalan kualitas cabai dengan akurasi tertinggi sebesar 80 %, dengan data training 90 dan testing 30 data gambar cabai. Hasil perhitungan di atas bahwa nilai K dengan angka ganjil memberikan akurasi yang baik .

Kata Kunci : Kualitas Cabai, *Gray Level CO-Occurrence Matrix*, *K-Nearest-Neighbor*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul: **“PENGENALAN KUALITAS CABAI MENGGUNAKAN METODE GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN K-NEAREST NEIGHBOR (K-NN)”**, untuk memenuhi salah satu syarat ujian guna mendapatkan gelar sarjana 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 yang setinggi-tingginya kepada:

1. Muh. Ichsan Gaffar, S.E, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Ibu Zohrahayaty, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Gorontalo Sekaligus Pembimbing Utama;
4. Bapak Sudirman S.Panna, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Gorontalo;
8. Ibu Hastuti Dalai, M.Kom, selaku Pembimbing Pendamping;
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada kami;

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 meeka 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 adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semog hasil yang telah dicapai ini dapat bermafaat bagi semua, Aamiin.

Gorontalo, 28 Juli 2020

Penulis

DAFTAR ISI

JUDUL PENELITIAN.....	i
PERSETUJUAN SKRIPSI.....	ii
HALAMAN PENGESAHAN.....	ii
PERNYATAAN SKRIPSI.....	iii
ABSTRACT.....	v
ABSTRAK	vi
KATA PENGANTAR	vii
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 Batasan Masalah	4
1. 6 Manfaat Penelitian	4
1. 6. 1 Manfaat Teoritis.....	4
1. 6. 2 Manfaat Praktis	5
BAB II LANDASAN TEORI.....	6
2. 1 Tinjauan Studi.....	6
2. 2 Tinjauan Pustaka.....	8
2. 2. 1 Kualitas Cabai	8
2. 2. 2 Komputer Vision.....	9
2. 2. 3 Ekstrasi Fitur	9
2. 2. 4 <i>Gray Level Co-Occurrence Matrix</i>	11

2. 2. 5 Klasifikasi	14
2. 2. 6 <i>K-Nearest Neighbor</i> (K-NN)	14
2. 2. 7 Evaluasi Model	18
2. 2. 8 Python	20
2. 3 Kerangka Pikir	21
BAB III METODE PENELITIAN.....	22
3. 1 Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian	22
3. 2 Pengumpulan Data	22
3. 2. 1. Bentuk	22
3. 2. 2. Jumlah	22
3. 2. 3. Sumber Data.....	23
3. 2. 4. Cara Memperoleh Data	23
3. 2. 5. Peralatan Penelitian.....	23
3. 3 Pemodelan.....	24
3. 3. 1 Pra Pengolahan.....	24
3. 3. 2 Ekstraksi Ciri.....	24
3. 3. 3 Data Training	25
3. 3. 4 Training Menggunakan K-NN	25
3. 3. 5 Model	25
3. 3. 6 Data Testing	25
3. 3. 7 Hasil Klasifikasi.....	26
3. 3. 8 Evaluasi.....	26
BAB IV HASIL PENELITIAN	27
4. 1 Hasil Pengumpulan Data	27
4. 2 Hasil Pemodelan	28

4. 2. 1 Pra-Pengolahan	28
4. 2. 2 Ekstraksi Fitur	31
4. 2. 3 Klasifikasi Menggunakan Metode <i>K-NN</i>	35
4. 3 Evaluasi.....	41
4. 3. 1 Confusion Matrix	41
BAB V PEMBAHASAN	44
5. 1.Pembahasan Model	44
5. 2.Pembahasan Sistem.....	46
BAB VI PENUTUP	50
6. 1.Kesimpulan	50
6. 2.Saran	50
DAFTAR PUSTAKA	51

DAFTAR GAMBAR

Gambar 2. 1: Hubungan antara ; (a) citra asli; (b) matrix glcm	11
Gambar 2. 2: Ilustrasi penentuan arah	12
Gambar 2. 3: Menentukan pixel tetangga	12
Gambar 2. 4: Proses membentuk matriks glcm.....	12
Gambar 2. 5: K-nn dengan k tetangga, (a) 1-nn, (b) 3-nn, (c) 5-nn, (d) 7-nn	16
Gambar 2. 6 Kerangka Pikir	21
Gambar 5. 1: Tampilan Menu Utama.....	46
Gambar 5. 2: Tampilan Browse Data Training	46
Gambar 5. 3: Tampilan Browse Data Training	47
Gambar 5. 4: Tampilan Pemasukan Nilai GLCM.....	48
Gambar 5. 5: Tampilan Pemasukan Nilai K-NN	48
Gambar 5. 6: Tampilan Output Hasil Pengenalan Kualitas Cabai	49

DAFTAR TABEL

Tabel 2. 1: Penelitian Terkait	6
Tabel 2. 2: Klasifikasi Data Training.....	16
Tabel 2. 3: Perhitungan Kuadrat Jarak Data Training Dengan Data Testing.....	17
Tabel 2. 4: Penentuan 3 Tetangga Terdekat Dari Data Testing	17
Tabel 2. 5: Klasifikasi Kelas Data Terdekat	18
Tabel 2. 6: Tabel Rumus Confusion Matrix.....	19
Tabel 4. 1 Dataset Gambar Kualitas Cabai	27
Tabel 4. 2 Proses perubahan citra asli menjadi graysacle	28
Tabel 4. 3 Proses normalisasi citra	29
Tabel 4. 4 Proses deteksi kualitas.....	30
Tabel 4. 5 Nilai dari suatu citra	31
Tabel 4. 6 Matrix GLCM	32
Tabel 4. 7 Matrix Ternormalisasi	32
Tabel 4. 8 Data Training.....	35
Tabel 4. 9 Perhitungan Jarak Euclidean Distance	37
Tabel 4. 10 Penentuan 7 tetangga terdekat.....	38
Tabel 4. 11 Hasil penentuan $K=7$	39
Tabel 4. 12 Jumlah data terbanyak menentukan	39
Tabel 4. 13 Hasil perbandingan.....	39
Tabel 4. 14 Hasil data testing $K=7$	41
Tabel 4. 15 Hasil data uji.....	43

DAFTAR LAMPIRAN

Lampiran 1: Dataset.....	54
Lampiran 2: Kode Program.....	63
Lampiran 3: Surat Rekomendasi Penelitian.....	77
Lampiran 4: Surat Rekomendasi Bebas Plagiasi.....	78
Lampiran 5: Hasil Turnitin.....	79
Lampiran 6: Surat Bebas Pustaka.....	82
Lampiran 7: Daftar Riwayat Hidup.....	83

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Salah satu komoditas pertanian yang kaya manfaat dan bernilai ekonomi tinggi adalah cabai [1]. Cabai merupakan salah satu produk olahan hasil perkebunan yang banyak dikonsumsi masyarakat dalam kehidupan sehari-hari. Selain digunakan untuk memenuhi kebutuhan rumah tangga sebagai bumbu masak, cabai juga banyak digunakan sebagai bahan baku industri pangan seperti sambal, saus, variasi bumbu dan lain-lain. Cabai juga dimanfaatkan pada industri obat-obatan dalam pembuatan krim obat gosok antirematik maupun dalam bentuk Koyo Cabai. Penggunaan yang cukup luas tersebut menyebabkan komoditas ini memiliki nilai ekonomis tinggi [2].

Setiap cabai memiliki karakteristik yang berbeda pula, buah cabai memiliki warna hijau, orange, dan merah Ciri yang sering dijadikan indikator dalam mengenali cabai adalah sifat fisik cabai itu sendiri seperti warna, dan bentuk cabai itu sendiri. Karakteristik ini pun yang digunakan sebagai barometer untuk menentukan kualitas cabai. Kualitas cabai merupakan hal yang utama dan sangat penting yang berpengaruh pada nilai harga jual dan penggunaan cabai itu sendiri [3].

Dalam penggunaannya untuk memenuhi kebutuhan, tentunya informasi kualitas cabai diperlukan oleh produsen dalam hal ini petani dan penjual maupun konsumen. Untuk itu perlu dilakukan pengenalan kualitas cabai. Sayangnya masih terdapat kendala yang sering dihadapi masyarakat dalam menentukan kualitas cabai. Salah satu kendalanya yakni identifikasi dengan indra mata manusia memiliki kelemahan yaitu, waktu yang di butuhkan relatif lama serta menghasilkan identifikasi yang beragam karena keterbatasan visual manusia. Selain itu, subjektifitas lainnya yakni tingkat kelelahan dan perbedaan pendapat tentang penilaian terhadap kualitas cabai. Oleh karena itu diperlukan adanya metode yang

memudahkan produsen maupun konsumen dalam menentukan kualitas cabai yang baik secara cepat dan tepat.

Kemajuan teknologi komputer termasuk interaksi antara manusia dengan komputer pada saat ini sudah berkembang sangat pesat. Bahkan dibidang pertanian penggunaan teknologi komputer sudah banyak digunakan. Penerapannya digunakan untuk deteksi penyakit pada tanaman, klasifikasi mutu, penentuan berat dan identifikasi jenis buah-buahan maupun sayuran [4]. 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 untuk melakukan ekstrasi ciri berbasis statistik, perolehan ciri diperoleh dari nilai piksel matrix, yang mempunyai nilai tertentu dan membentuk suatu sudut pola [5]. 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 [6].

Berdasarkan beberapa penelitian yang pernah dilakukan oleh Saifudin dan Abdul Fadilil identifikasi dengan menggunakan metode *GLCM* memberikan hasil yang cukup baik, Pada sistem identifikasi citra kayu berdasarkan tekstur menggunakan metode *Gray Level Co-Occurrence Matrix*(GLCM) dan Klasifikasi menggunakan metode jarak *Euclidian* memberikan nilai hasil akurasi sebesar 82,5% [7]. Pada klasifikasi citra jenis daging berdasarkan tekstur dengan menggunakan metode *Gray Level Co-Occurrence Matrix*(GLCM) dan klasifikasinya menggunakan metode *Support Vector Machine*(SVM) memberikan nilai akurasi sebesar 87,5% pada jarak pengambilan gambar 20 cm, kemudian jarak ketetanggaan piksel 2 pada arah GLCM 135 [8].

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 [9]. Dari beberapa penelitian yang dilakukan oleh F.S sutojo dan D.R.I setiadi pada identifikasi tumbuhan obat herbal berdasarkan citra daun menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM) dan untuk klasifikasinya menggunakan metode *K-Nearest Neighbor* (K-NN) mendapatkan nilai akurasi 83.33% [10], sedangkan pada identifikasi jenis buah apel menggunakan Algoritma K-NN dengan ekstraksi fitur histogram memberikan tingkat akurasi 90% [11]. Hasil ini menunjukkan 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 klasifikasi sehingga dapat meningkatkan akurasi pengenalan kualitas cabai. Sistem di bangun menggunakan *Spyder Python* sebagai editor untuk mengimplementasikan script, dan openCV untuk pemrosesan citranya.

Berdasarkan berbagai pemaparan di atas, telah membawa peneliti untuk melakukan penelitian dengan judul: “**Pengenalan Kualitas Cabai Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) Dan *K-Nearest Neighbor* (K-NN)**”. Diharapkan dari penelitian ini dapat memberi kontribusi, berupa aplikasi yang dapat membantu masyarakat dalam mengenali kualitas cabai.

1.2 Identifikasi Masalah

1. Terbatasnya fungsi identifikasi dengan indra mata manusia sehingga memberikan hasil yang tidak konsisten dan tidak tepat pada pengenalan kualitas cabai.
2. Perlunya Penambahan Algoritma Klasifikasi untuk meningkatkan akurasi algoritma GLCM pada pengenalan kualitas cabai.

1.3 Rumusan Masalah

1. Bagaimana merekayasa sistem pengenalan kualitas cabai dengan teknologi pengolahan citra?
2. Bagaimana peningkatan akurasi metode GLCM apabila diterapkan metode K-NN dalam proses klasifikasi untuk pengenalan kualitas cabai?

1.4 Tujuan Penelitian

1. Untuk merekayasa sistem pengenalan kualitas cabai dengan teknologi pengolahan citra.
2. Menerapkan metode K-NN dalam proses klasifikasi untuk meningkatkan akurasi metode GLCM pada pengenalan kualitas cabai.

1.5 Batasan Masalah

1. Pengenalan dalam penelitian ini hanya menggunakan cabai rawit berdasarkan data observasi yang ada di lapangan.

1.6 Manfaat Penelitian

1.6.1 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 pengenalan kualitas cabai.

1. 6. 2 Manfaat Praktis

Sebagai solusi untuk mempermudah masyarakat dalam pengenalan kualitas cabai.

BAB II LANDASAN TEORI

2.1 Tinjauan Studi

Penelitian tentang pengenalan kualitas cabai 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.	Abd..Ghofur	Implementasi Metode Klasifikasi <i>Naive Bayes</i> untuk memprediksi kualitas cabai/2016	<i>Naive Bayes</i>	Hasil dari prediksi dari data uji yang berjumlah 30 citra uji, data uji yang berhasil di prediksi adalah 28 citra dari 30 citra data uji, sedangkan data uji yangg tidak berhasil di prediksi adalah 2 citra. Jadi berdasarkan hasil prediksi pada citra cabai di peroleh tingkat akurasi mencapai 93,33 % dengan menggunakan metode <i>Naive Bayes</i>

2.	Angga Aditya Indra Wiratmaka, Imam Fahrur Rozi dan Rosa Andrie Asmara	Klasifikasi Kualitas Tanaman Cabai Menggunakan Metode <i>Fuzzy K-Nearest Neighbor</i> (FKNN)/2017	<i>Fuzzy K-Nearest Neighbor</i>	Dalam penelitian ini akurasi hasil di hitung dari jumlah hasil yang tepat di bagi dengan jumlah data. Penelitian ini menggunakan 100 data sampel dengan 70 data latih dan 30 data uji. Pada proses pengujian menggunakan metode <i>Fuzzy K-Nearest Neighbor</i> dengan 30 data training dan 70 data testing mendapatkan nilai akurasi mencapai 96,67 %.
3.	Saifudin dan Abdul Fadlil	Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Metode <i>Gray Level Co-Occurrence Matrix</i> (GLCM) dengan Klasifikasi Jarak <i>Euclidean</i> /2015	<i>Gray Level Co-Occurrence Matrix</i> (GLCM) dan <i>Euclidean Distance</i>	Hasil dari identifikasi citra kayu dengan ekstraksi ciri <i>Gray Level Co-Occurrence matrix</i> (GLCM) dan Klasifikasi dengan jarak <i>Euclidean Distance</i> menunjukan akurasi yang paling tinggi di dapat pada citra ukuran 30x30 yaitu 82,5% dan akurasi terendah didapat dari citra ukuran 20x20

				dengan nilai akurasi 65,0% dan ukuran citra 10x10 dengan nilai akurasi 75,5%.
4.	Fitria, T.Sutejo dan De Rosal Ignatius Moses Setiadi	Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma <i>Gray Level Co-Occurrence Matrix</i> (GLCM) dan <i>K-Nearest Neighbor</i> (K-NN)/2018	<i>Gray Level Co-Occurrence Matrix</i> (GLCM) dan <i>K-Nearest Neighbor</i> (K-NN)	Penelitian in telah berhasil mengimplementasikan Algoritma GLCM dan K-NN untuk mengekstraksi fitur kontras, korelasi, energi dan homogenitas pada citra daun tumbuhan obat herbal dan mengklasifikasinya berdasarkan jarak terdekat antara citra uji dan citra latih. Algoritma tersebut memberikan hasil nilai akurasi sebesar 83,33%

2. 2 Tinjauan Pustaka

2. 2. 1 Kualitas Cabai

Adapun tipe cabai unggul adalah memiliki pembungaaan dan pembentukan buahnya cepat (umur panen), produktivitasnya tinggi, daya adaptasinya yang luas atau spesifik untuk daerah marginal tertentu (kering, rawa, pantai dan gambut/asam), serta tahan terhadap hama dan penyakit. tak hanya untuk memenuhi hasil secara kuantitas, penentuan cabai unggul juga ditekankan pada kualitas sesuai preferensi konsumen. Para konsumen

menginkan karakter cabai antara lain tingkat kepedasan sesuai, penampilan buah yang baik, mulus, dan warna merah yang terang, dan bebas dari penyakit. Untuk industri pangan, seperti saos sifat-sifat cabai yang diinginkan adalah mempunyai tingkat kepedasan yang tinggi, warna merah yang terang, dan buahnya harus tersedia sepanjang waktu untuk memenuhi kebutuhan industri [12].

2. 2. 2 **Komputer Vision**

Dalam istilah sederhana, *Computer Vision* adalah bagaimana computer/mesin dapat melihat. Computer Vision adalah bidang yang mencakup metode untuk memperoleh, mengolah, menganalisis dan memahami data visual seperti gambar dan video. Tujuan utama dari *Computer Vision* adalah agar computer/mesin dapat meniru kemampuan perseptual mata manusia dan otak atau bahkan dapat mengunggulinya untuk tujuan tertentu.

Computer Vision merupakan *image processing* (pengelolaan citra) dan *machine vision* (visi mesin) ada tumpah tindih yang signifikan dalam berbagai tehknik dan aplikasi yang mencakup dua bidang tersebut. Hal ini menunjukan teknik dasar yang di gunakan dan di kembangkan kurang lebih sama (identik). Komputer vision mencakup teknologi utama dalam menganalisis citra (visual) secara otomatis yang di gunakan dalam bidang lain [13].

Bidang AI (*Artificial Intelligence*) ini merupakan kemampuan suatu komputer yang dapat menangkap signal elektronik dari suatu kamera dan dapat memahami apa yang dilihat oleh kamera tersebut. Penerapan AI misalnya pada komputer yang dipasang dipeluru kendali, sehingga peluru kendali dapat diprogram untuk selalu mengejar sasarannya yang tampak di kamera [14].

2. 2. 3 **Ekstraksi Fitur**

Ekstraksi fitur merupakan suatu pengambilan ciri / fitur dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Fitur ekstraksi dilakukan dengan cara menghitung jumlah titik atau pixels yang ditemui

dalam setiap pengecekan, dimana pengecekan dilakukan dalam berbagai arah *tracing* pengecekan pada koordinat kartesian dari citra yang digital yang dianalisis yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri [15].

Fitur merupakan karakteristik unik dari suatu objek. Fitur dibedakan menjadi dua yaitu fitur “alami” merupakan bagian dari gambar, misalnya kecerahan dan tepi objek. Sedangkan fitur “buatan” merupakan fitur yang diperoleh dengan operasi tertentu pada gambar, misalnya histogram tingkat keabuan. Sehingga ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu objek dari objek yang lain. Ciri yang dapat di ekstrak dari suatu objek dalam citra antara lain adalah warna, bentuk, dan tekstur. Ekstraksi fitur dapat dibedakan menjadi 3 jenis [15]:

a. Ekstraksi Warna

Ruang warna mewakili warna dalam bentuk nilai intensitas. Kita bisa menentukan, memvisualisasikan dan menciptakan warna dengan menggunakan metode ruang warna. Metode yang sering digunakan untuk ekstraksi warna antara lain *Histogram Intersection*, *Zernike Chromaticity Distribution Moments* dan *Color Histogram*.

b. Ekstraksi Bentuk

Bentuk adalah sumber informasi utama yang bisa digunakan untuk pengenalan objek. Tanpa bentuk, objek konten visual tidak bisa dikenali dengan baik. Dua objek tidak bisa memiliki bentuk yang sama persis tapi dengan menggunakan berbagai algoritma kita bisa mengenali bentuk yang serupa dengan mudah. Metode yang sering digunakan untuk ekstraksi bentuk antara lain *Binary Image* dan segmentasi secara horizontal dan vertikal.

c. Ekstraksi Tekstur

Tekstur mengandung informasi penting tentang susunan dasar permukaan objek. Metode yang sering digunakan untuk ekstraksi tekstur antara lain *Gray Level Co-Occurence Matrix (GLCM)*, *Gray Level Run Length Matrix (GLRLM)*, dan *Edge Detection*.

2. 2. 4 Gray Level Co-Occurrence Matrix

Glcmm adalah salah satu metode statistika yang dapat digunakan untuk analisis tekstur. Langkah awal untuk mendapatkan informasi tekstur dari citra adalah menentukan cooccurrence matrix. Co-occurrence matrix menunjukkan hubungan spasial antara gray level dalam citra tekstur. Setiap elemen P dengan posisi (i,j) pada glcmm menunjukkan frekuensi relatif dua piksel gray level i dan j . Cooccurrence matrix dalam penelitian ini dihitung pada arah horizontal 0° dengan jarak 1 piksel, Setiap citra akan menghasilkan sebuah co-occurrence matrix. Co-occurrence matrix final dinormalisasi menggunakan persamaan untuk mengubah matriks glcmm menjadi tabel probabilitas, dapat disimpulkan bahwa setiap citra akan memiliki ukuran matrix kookurensi yang berbeda tergantung pada banyaknya tingkat keabuan pada citra tersebut [16].

100	0	250	100	0
250	100	100	0	250
100	0	100	250	100
250	100	250	250	0
100	0	0	100	0

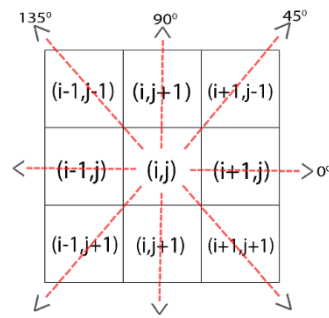
a. Citra asli

	0	100	250
0			
100			
250			

b. Matriks glcmm

Gambar 2. 1: Hubungan antara ; (a) citra asli; (b) matrix glcmm

Gambar 2.1 adalah citra asli dengan ukuran 5x5. Dari table a dapat dilihat ada 3 tingkat keabuan, yaitu 0, 100, dan 250. Karena pada cintra asli terdapat 3 tingkat keabun, maka matriks kookurensi yang akan dibuat adalah matriks dengan ukuran 3x3 seperti pada gambar 2.1 bagian b. Matriks kookurensi dapat dituliskan $p(i,j;d,\theta)$. Dalam membangun hubungan antara pixel diperlukan sudut θ digunakan dalam menentukan arah tetangga dan d menentukan jarak special.



Gambar 2. 2: Ilustrasi penentuan arah

Dari ilustrasi pada gambar 2.2, dapat ditentukan kearah mana suatu pixel akan di hubungkan dengan pixel tetangganya. Dengan ilustrasi ini dapat ditentukan matriks kookurensi dari citra asli pada gambar 2.1 dimana matriks glcm yang dibentuk dengan jarak special $d=1$ dan sudut $\theta=0^\circ$, atau dapat dituliskan $p(i,j;1,0^\circ)$ [10].

	1	2	3	4	5
1	100	0	250	100	0
2	250	100	100	0	250
3	100	0	100	250	100
4	250	100	250	250	0
5	100	0	0	100	0

Gambar 2. 3: Menentukan pixel tetangga

Pada gambar 2.3 menentukan tetangga untuk membentuk hubungan antra pixel berdasarkan aturan $p(i,j;1,0^\circ)$. Karena jarak special dan sudut arah 0° , maka pixel tetangga dari i atau pixel j berada di kanan degan jarak 1, contoh jika i ($_1$ makaj $(_1,2)$).

100	0	250	100	0
250	100	100	0	250
100	0	100	250	100
250	100	250	250	0
100	0	0	100	0

0	100	250
0	1	2
100	6	1
250	1	4

Gambar 2. 4: Proses membentuk matriks glcm

Gambar 2.4 menjelaskan bagaimana membentuk matriks kookurensi berada di sebelah kanan dan matriks asli sebelah kanan. Nilai cell didapatkan dari nilai hubungan antara pixel pada citra asli yang memenuhi sarat $p(i,j;1,0^\circ)$ dimana jarak antara satu pixel dengan pixel tetangga sama dengan 1 dengan arah sudut 0° . Dan yang ditunjukkan pada lingkaran merah dimana nilai $i=0$ dan $j=0$, $d=1$ dan $\theta=0^\circ$, dan pada nilai $i=100$ dan $j=0$, $d=0$ dan $\theta=0^\circ$ ada 6 hubungan pixel yang memenuhi. Cara yang sama untuk mengisi matriks glcm hingga semua nilai matriks glcm ditemukan [17]

Setelah memperoleh matriks glcm, ada fitur yang bisa di peroleh dari matriks tersebut, fitur tersebut antara lain *contrast*, *energy*, *homogeneity*, dan *entropy* [18].

a. Contrast

$$\sum_{i,j=0}^{N-1} P(i,j)(i-j)^2$$

b. Energy

$$\sum_{i,j=0}^{N-1} P(i,j)^2$$

c. Homogeneity

$$\sum_{i,j=0}^{N-1} P(i,j)/R$$

e. Entropy

$$\sum_{i,j=0}^{N-1} (i,j) \log[p(i,j)]$$

Contrast, *energy*, *homogeneity*, dan *entropy* dihitung untuk co-occurrence matrix yang diperoleh sehingga setiap fitur mempunyai sebuah nilai. Informasi fitur untuk setiap citra direpresentasikan sebagai sebuah vektor yang memiliki 3 elemen fitur. Informasi fitur ini menjadi masukan untuk proses klasifikasi dengan knn [16] .

2. 2. 5 Klasifikasi

Klasifikasi adalah penemuan sebuah model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bias digunakan untuk memprediksi kelas dari objek yang label kelasnya. Sedangkan klasifikasi menurut serapannya yang berasal dari belanda, *classificatie*, yang sendirinya berasal dari Bahasa perancis *classification*. Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah di tetapkan [19].

Klasifikasi data terdiri dari dua tahap proses. Pertama adalah Learning (fase training), dimana algoritma klasifikasi dibuat untuk menganalisa data training lalu diintrepresentasikan dalam bentuk rule klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari rule klasifikasi [19].

Model dalam klasifikasi sama artinya dengan suatu model yang menerima masukan kemudian mampu melakukan pemikiran terhadap masukan dan memberikan jawaban sebagai keluaran dari hasil pemikiran terhadap masuk dan memberikan jawaban sebagai keluaran hasil pemikiran, Proses klasifikasi didasarkan pada empat komponen [20] :

1. Kelas adalah variable dependen yang berupa kategorikal yang mempresentasikan “label” yang terdapat pada objek.
2. Predictor, merupakan variable independen yang direpresentasikan oleh karakteristik (attribute) data.
3. Training dataset, Satu set data yang berisi nilai dari dua komponen diatas yang digunakan untuk menentukan kelas yang cocok berdasarkan predictor.
4. Testing Dataset, Berisi data baru yang akan di klasifikasikan oleh model yang telah di buat dan akurasi klasifikasi dievaluasi.

2. 2. 6 K-Nearest Neighbor (K-NN)

Algoritma metode k-nn sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke data training untuk menentukan knnnya. Data

training diproyeksikan ke ruang berdimensi banyak, dimana masing – masing dimensi mempresentasikan fitur dari data. Ruang ini menjadi bagian – bagian berdasarkan klasifikasi data training. Salah satu teknik klasifikasi paling dasar dan sederhana adalah k-nearest neighbor (k-nn) [21]. teknik klasifikasi ini memiliki beberapa kelebihan yaitu :

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

Metode klasifikasi k nearest neighbor melakukan proses pencocokan/pengenalan berdasarkan jumlah tetangga terdekat untuk penentuan kelasnya. Untuk mencari jarak kelas ada beberapa cara yaitu dengan euclidean distance. Tahapan dalam metode klasifikasi *k nearest neighbor* yaitu :

1. Menentukan nilai k
2. Menghitung jarak antara citra testing dengan seluruh citra pada database menggunakan persamaan euclidean distance, persamaan dan menentukan citra terdekat dengan citra testing berdasarkan nilai k.

Rumus dari euclidean distance

$$D_{(a,b)} = \sum_{i=0}^n (x_i - y_i)^2$$

Keterangan:

D (a, b) : jarak Euclidian

x : data 1

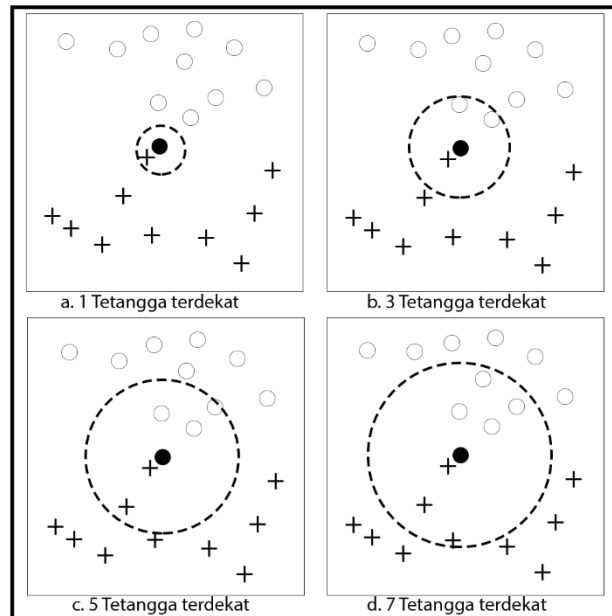
y : data 2

i : fitur

Ke – n : jumlah fitur

3. Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.

4. jika terjadi konflik atau keadaan seimbang pada kelas dengan jumlah anggota yang sama maka digunakan pemecahan konflik.



Gambar 2. 5: K-nn dengan k tetangga, (a) 1-nn, (b) 3-nn, (c) 5-nn, (d) 7-nn

Gambar 2.5 menunjukkan jumlah tetangga yang paling dekat yang dapat dimuat dalam rentang nilai k yang telah ditentukan. Gambar a menunjukkan 1 tetangga terdekat, gambar b menunjukkan 3 tetangga terdekat, gambar c 5 tetangga terdekat, dan d 7 tetangga terdekat. Untuk menentukan jarak tetangga terdekat, terdapat berbagai macam cara salah satunya adalah dengan pendekatan *euclidean distance*. [17].

Contoh Perhitungan K-Nearest Neighbor:

Terdiri dari 2 atribut dengan skala kuantitatif sebagai data training yaitu X_1 dan X_2 serta Y yaitu kelas baik dan buruk seperti pada 2.2 [22].

Tabel 2. 2: Klasifikasi Data Training

X_1	X_2	Y
7	7	Buruk
7	4	Buruk
3	4	Baik

1	4	Baik
---	---	------

Terdapat data testing yaitu $X_1 = 3$ dan $X_2 = 7$, tentukan nilai Y ! Langkah penyelesaian:

1. Tentukan parameter K = jumlah tetangga terdekat, misalkan ditetapkan $K = 3$
2. Hitung jarak antara data baru dengan semua data training seperti pada tabel 2.3.

Tabel 2. 3: Perhitungan Kuadrat Jarak Data Training Dengan Data Testing

X1	X2	Kuadrat Jarak dengan Data Baru (3,7)
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

3. Urutkan hasil kuadrat jarak tersebut secara ascending dan tetapkan tetangga terdekat berdasarkan nilai K (dimana $K = 3$) seperti pada tabel 2.4 dan tabel 2.5.

Tabel 2. 4: Penentuan 3 Tetangga Terdekat Dari Data Testing

X1	X2	Euclidean	Peringkat Jarak	Tetangga Terdekat
7	7	$\sqrt{16} = 4$	3	YA
7	4	$\sqrt{25} = 5$	4	TIDAK
3	4	$\sqrt{9} = 3$	1	YA
1	4	$\sqrt{13} = 3,6$	2	YA

Tabel 2. 5: Klasifikasi Kelas Data Terdekat

X1	X2	EuclideanDistance	Peringkat Jarak	Tetangga Terdekat	Y
7	7	4	3	YA	BURUK
7	4	5	4	TIDAK	-
3	4	3	1	YA	BAIK
1	4	3,6	2	YA	BAIK

4. Berdasarkan hasil dari Tabel 2.5 dimana nilai $K = 3$, mayoritas hasil Y yang diperoleh yaitu “Baik”. Jadi hasil klasifikasi data testing dengan nilai $X1 = 3$ dan $X2 = 7$ adalah $Y =$ kelas Baik. [23]

2. 2. 7 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 [24]. 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 [25].

Tabel 2. 6: Tabel Rumus Confusion Matrix

Klasifikasi Sebenarnya	Hasil Klasifikasi	
	+	-
+	Benar Positif (A)	Benar Negatif (B)
-	Salah (C)	Salah Negatif (D)

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

$$\text{Akurasi} = (A+D)/(A+B+C+D) \quad (1)$$

Presisi didefinisikan sebagai rasio item relevan yang dipilih terhadap semua item yang terpilih. Presisi dapat diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. Rumus presisi adalah:

$$\text{Presisi} = A/(C+A) \quad (2)$$

Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap total jumlah item relevan yang tersedia. *Recall* dihitung dengan rumus:

$$\text{Recall} = A/(A+D) \quad (3)$$

Presisi dan *Recall* dapat diberi nilai dalam bentuk angka dengan menggunakan perhitungan persentase (1-100%) atau dengan menggunakan bilangan antara 0-1. Sistem rekomendasi akan dianggap baik jika nilai presisi dan *recall*nya tinggi.

Kurva ROC menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confusion matrix*. ROC adalah grafik dua dimensi dengan *false positive* sebagai garis horizontal dan *true positive* sebagai garis vertikal. AUC (*the area under curve*) dihitung untuk mengukur perbedaan performansi metode yang digunakan. ROC memiliki tingkat nilai diagnosa yaitu: [26]

- Akurasi bernilai 0,90 – 1,00 = *excellent classification*
- Akurasi bernilai 0,80 – 0,90 = *good classification*
- Akurasi bernilai 0,70 – 0,80 = *fair classification*
- Akurasi bernilai 0,60 – 0,70 = *poor classification*
- Akurasi bernilai 0,50 – 0,60 = *failure*

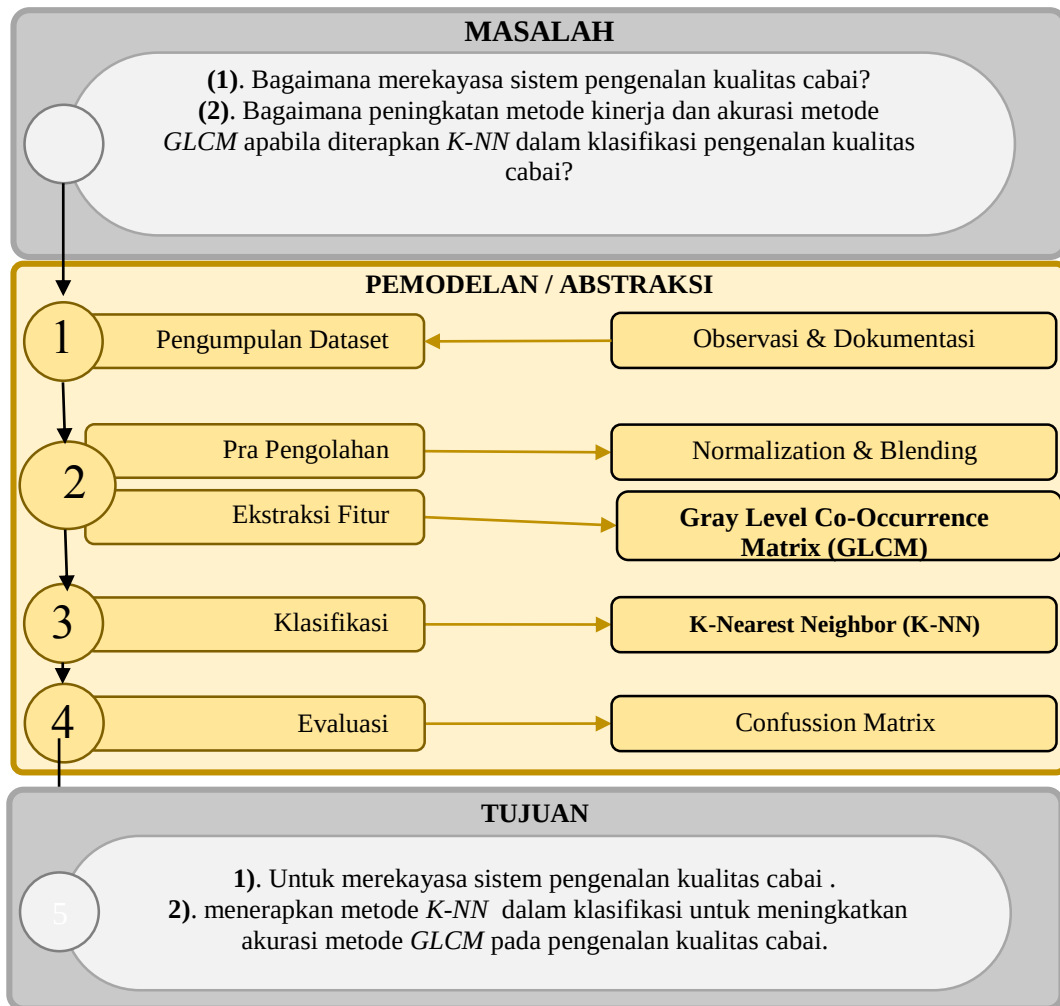
2. 2. 8 Python

Python adalah bahasa pemrograman model skrip (scripting language) yang berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, gui (antarmuka pengguna grafis), dan basis datanya [27].

Beberapa fitur yang dimiliki python adalah:

1. Memiliki kepustakaan yang luas; dalam distribusi python telah disediakan modul-modul.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
4. Berorientasi obyek.
5. Dapat dibangun dengan bahasa python maupun c/c++.

2.3 Kerangka Pikir



Gambar 2. 6 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 karena penelitian ini berfokus penerapannya untuk memberikan solusi atas permasalahan secara praktis. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori. Penelitian ini menggunakan metode penelitian eksperimen. Dengan demikian jenis penelitian ini adalah penelitian eksperimental. Subjek penelitian ini adalah pengenalan kualitas cabai. Penelitian ini dimulai dari maret 2019 sampai dengan Juli 2019.

3.2 Pengumpulan Data

Dalam penelitian ini terdapat bagian yang berhubungan dengan bahan dan data yang akan digunakan. Berikut adalah penjelasan mengenai bahan dan data yang akan digunakan tersebut:

3.2.1. Bentuk

Dalam pengenalan kualitas cabai ini diperlukan banyak data guna melakukan pengenalan cabai ini. Data-data tersebut berupa foto/image cabai yang terdiri dari cabai Kualitas Baik, Sedang, dan tidak baik. jenis data yang digunakan ialah berupa foto cabai:

Cabai rawit. Foto tersebut memiliki format JPG.

3.2.2. Jumlah

Jumlah dataset yang akan digunakan untuk pengenalan kualitas cabai ini ialah sebanyak 120 data foto cabai yang terdiri dari data training 90 dan testing 30. Foto cabai yang diambil terdiri dari 3 class cabai. Setiap class cabai akan diambil 30 foto sebagai data, sehingga 3×30 maka di dapat 90 data training foto cabai.

3. 2. 3. Sumber Data

Data cabai yang digunakan dalam pengenalan kualitas cabai ini diambil di beberapa pasar dan pengepul cabai di sekitaran daerah Gorontalo sendiri. Ada banyak pasar dan pengepul cabai disekitaran daerah Gorontalo ini, salah satunya ialah pasar sabtu dan pengepul cabai yang berada di kabupaten gorontalo.

3. 2. 4. Cara Memperoleh Data

Banyaknya data yang digunakan tidak terlepas dari cara memperoleh data tersebut. Data berupa foto cabai ini diperoleh dengan cara survey ke beberapa pasar dan pengepul, kemudian membeli cabai yang diperlukan. Lalu dilakukan pengambilan foto secara manual foto dengan kamera Smartphone realmi 5i dengan kamera 24 MP. Jarak pengambilan foto dilakukan setinggi 20 cm.

3. 2. 5. Peralatan Penelitian

Dalam pembuatan tugas akhir pengenalan kualitas cabai ini tidak lari dari beberapa peralatan yang digunakan. Peralatan tersebut digunakan untuk membantu dalam proses dibuatnya tugas akhir ini. Peralatan tersebut terdiri sebagai berikut:

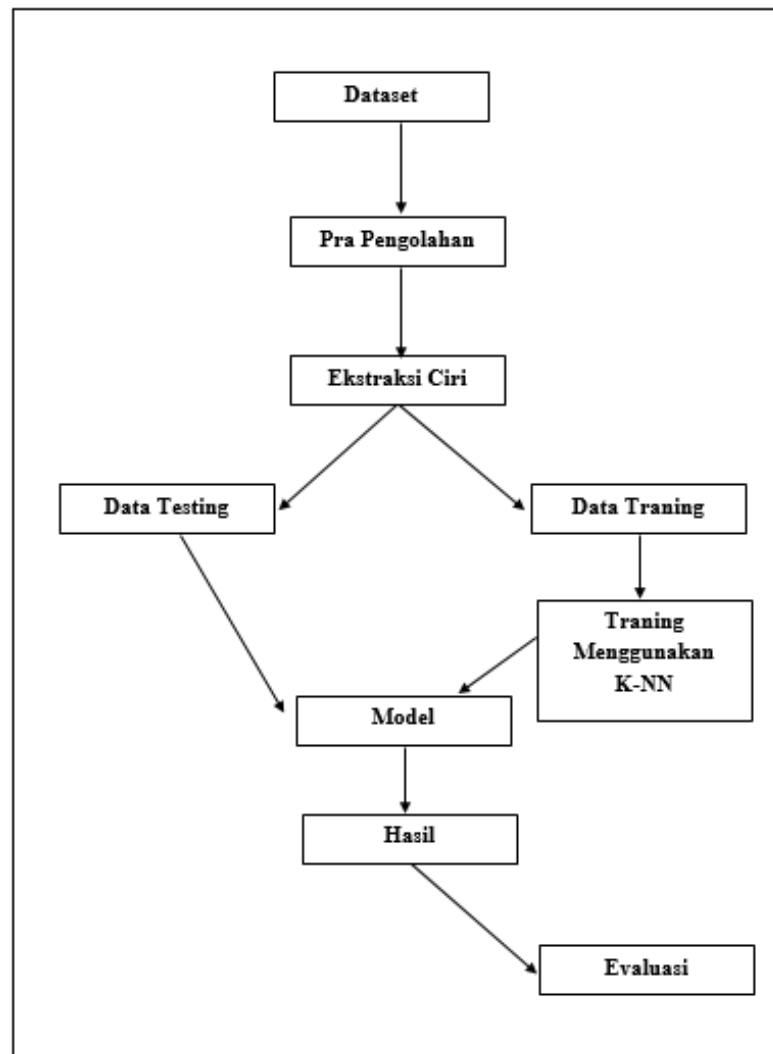
1) Laptop

Merk : Asus
 Type : A455L
 Processor : Intel Core i3-5005U 2.0 GHz
 RAM : 4.00 GB (3.87 GB useble)
 HDD : 500 GB
 OS : Windows 10 Pro 64-bit.

Terdapat beberapa aplikasi di laptop yang digunakan untuk membantu pengerjaan tugas akhir ini. Aplikasi tersebut diantaranya:

a) Python

3.3 Pemodelan



Gambar 3. 1 Pemodelan Sistem

3.3.1 Pra Pengolahan

Sebelum data diolah, terlebih dahulu dilakukan intensity adjustment dan segmentasi. Hal ini dilakukan karena intensity adjustment bertujuan untuk meningkatkan citra yang gelap dan proses segmentasi ialah proses pemisahan antara objek yang dikehendaki dan objek yang tidak dikehendaki.

3.3.2 Ekstraksi Ciri

Ekstraksi ciri berfungsi sebagai deteksi fitur dari suatu citra. Ciri yang dapat digunakan membedakan citra satu dengan citra yang lain, di antaranya adalah ciri

bentuk, ukuran, geometri, tekstur, dan warna. Pada tahap ini digunakan Gray Level Co-occurrence Matrix. Masing-masing citra diekstrak cirinya berdasarkan parameter-parameternya tertentu dan dikelompokkan pada kelas tertentu, parameter – parameter yang digunakan berupa Contrasts, Energy, Homogeneity, dan Entropy.

3. 3. 3 Data Training

Data Training berupa data yang telah terekstrak cirinya yang selanjutnya akan dilatih menggunakan algoritma K-Nearest Neighbor, jumlah data training yang digunakan berupa 10% dari sampel cabai dengan citra cabai yang berbeda . Algoritma ini akan menentukan/ mencari bobot yang terbaik. Data training ini berupa hasil dari ekstraksi Gray Level Co-occurrence Matrix terhadap pengenalan kualitas cabai.

3. 3. 4 Training Menggunakan K-NN

Training menggunakan K-NN yaitu menjadikan data training menjadi data inputan. Algoritma KNN bekerja berdasarkan jarak terpendek dari training dan test dalam penentuannya, kemudian mengambil mayoritas hasil ketentuan yang telah didapatkan untuk dijadikan prediksi dari test.

3. 3. 5 Model

Model merupakan hasil dari proses training dari algoritman K-NN menggunakan data training.

3. 3. 6 Data Testing

Data testing merupakan data yang telah terekstraksi cirinya, untuk menguji data yang telah dilatih, jumlah data training yang digunakan berupa 90% dari sampel cabai. Data testing digunakan untuk mengetahui keberhasilan classifier berhasil melakukan klasifikasi, data testing merupakan ciri dari hasil ekstraksi dari Gray Level Co-occurrence Matrix terhadap pengenalan kualitas cabai.

3. 3. 7 Hasil Klasifikasi

Hasil klasifikasi merupakan output, pada data testing yang didapatkan dari proses klasifikasi yang menggunakan algoritma K-NN berdasarkan model yang diperoleh dari data training.

3. 3. 8 Evaluasi

- Evaluasi bertujuan untuk mengetahui hasil kinerja dari metode yang digunakan, evaluasi dilakukan pada data testing dan output yang dihasilkan akan dimasukan ke dalam Confusion Matrix untuk menghitung nilai akurasi dan Cross Validation untuk menguji atribut inputan.

BAB IV HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Dataset diambil dengan menggunakan kamera smartphone dengan resolusi 24 MP dengan ukuran masing-masing gambar. Berikut ini beberapa data yang telah dikumpulkan :

Tabel 4. 1 Dataset Gambar Kualitas Cabai

Baik			
			
Sedang			
			
Tidak baik			
			

4.2 Hasil Pemodelan

4.2.1 Pra-Pengolahan

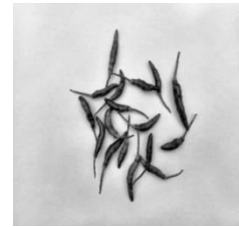
Pra-pengolahan citra (image pre-processing), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra cabai akan di konversi agar diperoleh data citra cabai sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra cabai dari permasalahan luminisasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari system pengenalan kualitas cabai. Pra-pengolahan dalam penelitian ini dilakukan dalam 3 tahap yaitu:

1. Pengubahan citra warna ke *grayscale*

Pra-pengolahan pertama yang akan dilakukan adalah merubah citra training atau citra *testing* yang awalnya berbentuk citra dari RGB (Red, Green, Blue) menjadi citra bentuk 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. Berikut gambar proses perubahan citra warna ke *grayscale* :

Tabel 4. 2 Proses perubahan citra asli menjadi grayscale

Tingkat Kualitas	Image Asli	Image Grayscale
Baik		

Sedang		
Tidak baik		

2. Normalisasi citra

Setelah citra asli di konversi ke citra abu-abu maka pra-pengolahan selanjutnya adalah normalisasi citra dengan ekualisasi histogram. Ekualisasi histogram adalah sebuah proses yang mengubah distribusi nilai derajat keabuan pada sebuah citra sehingga menjadi seragam. Tujuan dari ekualisasi histogram adalah untuk memperoleh penyebaran histogram yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relative sama. Berikut gambar normalisasi citra abu-abu dengan histogram ekualisasi.

Tabel 4. 3 Proses normalisasi citra

Tingkat Kualitas	Citra Grayscale	Citra Histogram
Baik		

Sedang		
Tidak baik		

3. Deteksi tingkat kualitas

Setelah citra histogram didapatkan maka pra-pengolahan selanjutnya adalah pendektesian citra cabai terhadap citra dengan otsu thresholding. Otsu thresholding merupakan salah satu metode segmentasi citra yang memisahkan antara objek dan background dalam suatu citra berdasarkan pada perbedaan tingkat kecerahannya atau gelap terangnya. Region citra cenderung gelap akan dibuat semakin gelap (hitam sempurna dinilai intensitas sebesar 0), sedangkan region citra yang cenderung terang akan dibuat semakin terang(putih sempurna dinilai intensitas sebesar 1).

Tabel 4. 4 Proses deteksi kualitas

Tingkat Kualitas	Citra Histogram	Hasil Segmentasi Citra	Citra Otsu
Baik			



4. 2. 2 Ekstraksi Fitur

Proses ekstraksi fitur ini dilakukan untuk mengetahui setiap ciri dari sebuah tekstur dari masing-masing jenis cabai yang kemudian akan dirubah yang kemudian di rubah ke dalam kode biner agar mudah dikenali oleh system dengan menggunakan salah satu metode ekstraksi fitur yaitu metode GLCM dan dari hasil ekstraksi fitur kemudian akan menghasilkan sekumpulan data dari setiap tekstur jenis cabai. Berikut contoh perhitungan matrix 10x10 dengan ukuran

Tabel 4. 5 Nilai dari suatu citra

	0	1	2	3	4	5	6	7	8	9
0	218	216	217	216	216	215	217	217	217	218
1	217	217	217	217	217	216	216	217	217	218
2	217	217	217	217	217	217	217	217	217	218
3	217	217	217	218	217	217	217	217	217	218
4	217	217	217	218	216	216	216	217	217	218
5	217	216	217	218	217	216	217	217	217	217
6	217	214	217	216	217	216	218	217	217	217
7	217	215	217	217	217	216	218	217	217	217
8	216	214	216	217	217	217	217	218	218	217
9	216	213	216	216	216	216	216	217	216	217

Tabel 4.5 adalah citra asli dengan Ukuran 10x10. Dari tabel dapat di lihat ada 6 tingkat keabuan, yaitu 213, 214, 215, 216, 217, 218. Karena pada citra asli terdapat 6 tingkat keabuan, maka matrix Co-Occurrence yang akan di buat adalah matrix dengan ukuran 6x6 dengan sudut 0° seperti pada tabel 4.6.

Tabel 4. 6 Matrix GLCM

	213	214	215	216	217	218
213	0	0	0	1	0	0
214	0	0	0	1	1	0
215	0	0	0	1	2	0
216	1	1	1	5	11	4
217	0	1	2	11	17	12
218	0	0	0	4	12	1

Jumlah total nilai GLCM pada table 4.6 Matrix GLCM = 78

Tabel 4. 7 Matrix Ternomalisasi

i/j	213	214	215	216	217	218
213	0	0	0	0,012	0	0
214	0	0	0	0,012	0,012	0
215	0	0	0	0,012	0,025	0
216	0,012	0,012	0,012	0,064	0,141	0,051
217	0	0,012	0,025	0,141	0,217	0,153
218	0	0	0	0,051	0,153	0,012

Setelah memperoleh matrix yang ternomalisasi, ada fitur yang di peroleh dari matrix tersebut, fitur tersebut yaitu *Contras*, *Energy*, *Homogeneity*, dan *Entropy*.

a. Menghitung Nilai *Contras*

$$(213-216)^2(0,012) +$$

$$(214-216)^2(0,012) + (214-217)^2(0,012) +$$

$$(215-216)^2(0,012) + (215-217)^2(0,025) +$$

$$(216-213)^2(0,012) + (216-214)^2(0,012) + (216-215)^2(0,012) + (216-216)^2(0,064) + (216-217)^2(0,141) + (216-218)^2(0,051) +$$

$$(217-214)^2(0,012) + (217-215)^2(0,025) + (217-216)^2(0,141) + (217-217)^2(0,217) + (217-218)^2(0,153) +$$

$$(218-216)^2(0,051) + (218-217)^2(0,153) + (218-218)^2(0,012)$$

$$Contras = 0,108 + 0,156 + 0,112 + 0,513 + 0,502 + 0,357$$

$$Contras = \mathbf{1,748}$$

b. Menghitung Nilai *Energy*

$$\begin{aligned} Energy = & ((0,012)^2) + ((0,012)^2) + ((0,012)^2) + ((0,012)^2) + ((0,025)^2) \\ & + ((0,012)^2) + ((0,012)^2) + ((0,012)^2) + ((0,064)^2) + ((0,141)^2) \\ & + ((0,051)^2) + ((0,012)^2) + ((0,025)^2) + ((0,141)^2) + ((0,217)^2) \\ & + ((0,153)^2) + ((0,051)^2) + ((0,153)^2) + ((0,012)^2) \end{aligned}$$

$$\begin{aligned} Energy = & 0,001 + 0,001 + 0,001 + 0,001 + 0,006 + 0,001 + 0,001 + 0,001 \\ & + 0,004 + 0,019 + 0,002 + 0,001 + 0,006 + 0,019 + 0,047 \\ & + 0,023 + 0,002 + 0,023 + 0,001 \end{aligned}$$

$$Energy = \mathbf{0,16}$$

c. Menghitung Nilai *Homogeneity*

$$\begin{aligned} Homogeneity = & ((0,012) / 1 + (213-216)) + \\ & ((0,012) / 1 + (214-216)) + ((0,012) / 1 + (214-217)) + \\ & ((0,012) / 1 + (215-216)) + ((0,025) / 1 + (215-217)) + \\ & ((0,012) / 1 + (216-213)) + ((0,012) / 1 + (216-214)) + \\ & ((0,012) / 1 + (216-215)) + ((0,064) / 1 + (216-216)) + \\ & ((0,141) / 1 + (216-217)) + ((0,051) / 1 + (216-218)) + \\ & ((0,012) / 1 + (217-214)) + ((0,025) / 1 + (217-215)) + \\ & ((0,141) / 1 + (217-216)) + ((0,217) / 1 + (217-217)) + \\ & ((0,153) / 1 + (217-218)) + \\ & ((0,051) / 1 + (218-216)) + ((0,153) / 1 + (218-217)) + \end{aligned}$$

$$((0,012) / 1 + (218-218))$$

$$\begin{aligned} \text{Homogeneity} = & 0,012 + -3 + 0,012 + -2 + 0,012 + -3 + 0,012 + -1 + 0,025 + \\ & -2 + 0,012 + 3 + 0,012 + 2 + 0,012 + 1 + 0,064 + 0,041 + -1 \\ & + -1 + 0,051 + -2 + 0,012 + 3 + 0,025 + 2 + 0,141 + 1 + 0,217 \\ & + 0,153 + -1 + 0,051 + 2 + 0,153 + 1 + 0,012 \end{aligned}$$

$$\text{Homogeneity} = \mathbf{0,029}$$

d. Menghitung nilai *Entropy*

$$\begin{aligned} \text{Entropy} = & (0,012 \cdot \log 0,012) + \\ & (0,012 \cdot \log 0,012) + (0,012 \cdot \log 0,012) + \\ & (0,012 \cdot \log 0,012) + (0,025 \cdot \log 0,025) + \\ & (0,012 \cdot \log 0,012) + (0,012 \cdot \log 0,012) + (0,012 \cdot \log 0,012) + \\ & (0,064 \cdot \log 0,064) + (0,141 \cdot \log 0,141) + (0,051 \cdot \log 0,051) + \\ & (0,012 \cdot \log 0,012) + (0,025 \cdot \log 0,025) + (0,141 \cdot \log 0,141) + \\ & (0,217 \cdot \log 0,217) + (0,153 \cdot \log 0,153) + \\ & (0,051 \cdot \log 0,051) + (0,153 \cdot \log 0,153) + (0,012 \cdot \log 0,012) \\ \text{Entropy} = & -0,0230 + -0,0230 + -0,0230 + -0,0230 + -0,0400 + -0,0230 + \\ & -0,0230 + -0,0230 + -0,0764 + -0,1199 + -0,0659 + -0,0230 + \\ & -0,0400 + -0,1199 + -0,1439 + -0,1247 + -0,0659 + -0,1247 + \\ & -0,0230 \end{aligned}$$

$$\text{Entropy} = \mathbf{-0,8405}$$

Empat ciri yang didapatkan dari matrix *Gray Level CO-Occurrence Matrix* inilah yang digunakan untuk tahap klasifikasi dengan metode *K-NN*.

4. 2. 3 Klasifikasi Menggunakan Metode K-NN

Tabel 4. 8 Data Training

No	Data Training	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
0		187,046829	0,7356871	0,8328789	2,5316541	Baik
1		169,240579	0,7409364	0,826774	2,5285543	Baik
2		237,081838	0,7596116	0,8311238	2,4100201	Baik
3		222,522772	0,6759658	0,779278	3,0794075	Baik
.....						
30		206,56811	0,6857478	0,7719108	3,0703443	Sedang
31		270,210881	0,7297357	0,7970623	2,7287097	Sedang
32		261,76792	0,7155541	0,7945168	2,8024768	Sedang
33		271,380029	0,6364197	0,7309582	3,4865087	Sedang

....						
86		136,73639 6	0,822956	0,859212	1,890699 5	Tidak baik
87		144,4806	0,815795 8	0,852302 5	1,970160 6	Tidak baik
88		157,77734 2	0,794281 8	0,834572 5	2,16476	Tidak baik
89		130,05133 7	0,806509 5	0,848392 7	2,028906 6	Tidak baik
No	Data Testing	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
1		182,52725	0,7136943	0,8054117	2,7511538	?

1. Menentukan parameter nilai K

Untuk Menentukan Kelas dari data uji maka terlebih dahulu di tentukan nilai K, Di sini peneliti menggunakan nilai K=7.

2. Perhitungan jarak *Euclidean Distance*

Menghitung jarak antara citra testing dengan seluruh citra pada data training menggunakan rumus perhitungan jarak Euclidean Distance seperti di bawah ini:

$$d = \sqrt{(187,046829 - 182.52725)^2 + (0.7356871 - 0.7136943)^2 + (0.8328789 - 0.8054117)^2 + (2.5316541 - 2.7511538)^2}$$

$$d = \sqrt{(4,519579)^2 + (0.0219928)^2 + (0.0274672)^2 + (-0.2194997)^2}$$

$$d = \sqrt{20,426594337 + 0.0004836833 + 0.0007544471 + 0.0481801183}$$

$$d = \sqrt{20,476012586} = 4,525042827$$

Tabel 4. 9 Perhitungan Jarak Euclidean Distance

Data train i	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{(187,046829 - 182,52725)^2 + \dots (2,5316541 - 2,7511538)^2}$	4,525042827
1	$d = \sqrt{(169,240579 - 182,52725)^2 + \dots (2,5285543 - 2,7511538)^2}$	13,28858407
2	$d = \sqrt{(237,081838 - 182,52725)^2 + \dots (2,4100201 - 2,7511538)^2}$	54,5556768
3	$d = \sqrt{(222,522772 - 182,52725)^2 + \dots (3,0794075 - 2,7511538)^2}$	39,99689241
...	...	...
30	$d = \sqrt{(206,56811 - 182,52725)^2 + \dots (3,0703443 - 2,7511538)^2}$	24,04301521
31	$d = \sqrt{(270,210881 - 182,52725)^2 + \dots (2,7287097 - 2,7511538)^2}$	87,68363209
32	$d = \sqrt{(261,76792 - 182,52725)^2 + \dots (2,8024768 - 2,7511538)^2}$	79,24068362
33	$d = \sqrt{(271,380029 - 182,52725)^2 + \dots (3,4865087 - 2,7511538)^2}$	88,85588378
...	...	...
86	$d = \sqrt{(136,736396 - 182,52725)^2 + \dots (1,8906995 - 2,7511538)^2}$	45,799103
87	$d = \sqrt{(144,4806 - 182,52725)^2 + \dots (1,9701606 - 2,7511538)^2}$	38,054834
88	$d = \sqrt{(157,777342 - 182,52725)^2 + \dots (2,16476 - 2,7511538)^2}$	24,75700527
89	$d = \sqrt{(130,051337 - 182,52725)^2 + \dots (2,0289066 - 2,7511538)^2}$	52,48098616

Tabel 4. 10 Penentuan 7 tetangga terdekat

Data Training	Jarak Ke Masing-masing Data Latih	Kelas
19	0,576523351	Baik
24	0,724067116	Baik
12	1,132220196	Baik
71	1,14196396	Tidak baik
59	1,385867715	Sedang
11	1,409554321	Baik
9	1,805827458	Baik
47	2,380974301	Sedang
.....		
31	87,68363029	Sedang
16	88,32434954	Baik
33	88,85588378	Sedang
28	89,9863492	Baik
38	92,14435993	Sedang
49	93,07816007	Sedang
43	100,7622044	Sedang
39	102,940071	Sedang

Tabel 4.10 Hasil pengurutan dari nilai terkecil berdasarkan perhitungan jarak dari masing-masing data latih ke data uji.

- Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak, dengan jumlah $K=7$ sesuai yang di tentukan sebelumnya. dapat dilihat pada tabel 4.11

Tabel 4. 11 Hasil penentuan $K=7$

Data	Jarak Ke Masing-masing Data Latih	Kelas
19	0,576523351	Baik
24	0,724067116	Baik
12	1,132220196	Baik
71	1,14196396	Tidak baik
59	1,385867715	Sedang
11	1,409554321	Baik
9	1,805827458	Baik

Tabel 4. 12 Jumlah data terbanyak menentukan

Kelas	Jumlah data
Baik	5
Sedang	1
Tidak baik	1

Berdasarkan hasil dari tabel 4.12 Jumlah data terbanyak menentukan hasil klasifikasi K -NN.

Tabel 4. 13 Hasil perbandingan

No	Data Training	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
0		187,046829	0,7356871	0,8328789	2,5316541	Baik
1		169,240579	0,7409364	0,826774	2,5285543	Baik

2		237,081838	0,7596116	0,8311238	2,4100201	Baik
3		222,522772	0,6759658	0,779278	3,0794075	Baik
....						
30		206,56811	0,6857478	0,7719108	3,0703443	Sedang
31		270,210881	0,7297357	0,7970623	2,7287097	Sedang
32		261,76792	0,7155541	0,7945168	2,8024768	Sedang
33		271,380029	0,6364197	0,7309582	3,4865087	Sedang
....						
86		136,73639 6	0,822956	0,859212	1,890699 5	Tidak baik
87		144,4806	0,815795 8	0,852302 5	1,970160 6	Tidak baik
88		157,77734 2	0,794281 8	0,834572 5	2,16476	Tidak baik

89		130,05133 7	0,806509 5	0,848392 7	2,028906 6	Tidak baik
No	Data Testing	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
1		182,52725	0,7136943	0,8054117	2,7511538	Baik

Hasil yang didapatkan dari klasifikasi menggunakan metode *K-NN* untuk data uji adalah “Baik”.

4.3 Evaluasi

4.3.1 Confusion Matrix

Pada penelitian ini menggunakan *confusion matrix* sebagai metode dalam perhitungan akurasi pada pengenalan kualitas cabai. Evaluasi kinerja pengenalan kualitas cabai di dasarkan dari jumlah pengujian objek salah dan benar yang di deteksi. Dapat dilihat dari tabel berikut:

Tabel 4. 14 Hasil data testing K=7

No	Data Testing	Klasifikasi	Hipotesis	No	Data Testing	Klasifikasi	Hipotesis
1.		Baik	Baik	16		Sedang	Sedang
2		Baik	Tidak baik	17		Sedang	Tidak baik

3		Baik	Baik	18		Sedang	Sedang
4		Baik	Baik	19		Sedang	Sedang
5		Baik	Baik	20		Sedang	Sedang
6		Baik	Baik	21		Tidak baik	Tidak baik
7		Baik	Tidak baik	22		Tidak baik	Baik
8		Baik	Baik	23		Tidak baik	Tidak baik
9		Baik	Baik	24		Tidak baik	Tidak baik
10		Baik	Baik	25		Tidak baik	Tidak baik
11		Sedang	Sedang	26		Tidak baik	Tidak baik
12		Sedang	Sedang	27		Tidak baik	Tidak baik

13		Sedang	Sedang	28		Tidak baik	Baik
14		Sedang	Sedang	29		Tidak baik	Baik
15		Sedang	Sedang	30		Tidak baik	Tidak baik

Tabel 4. 15 Hasil data uji

Data Aktual	Hasil Klasifikasi		
	Baik	Sedang	Tidak baik
Baik	8	0	2
Sedang	0	9	1
Tidak Baik	3	0	7

Jumlah data yang diklasifikasi dengan benar = 24

Jumlah data yang diklasifikasi salah = 6

$$\text{Akurasi} = \frac{\text{Data uji Dikenali}}{\text{Data uji}} \times 100 \% \text{ Akurasi} = \frac{24}{30} \times 100\% = 80 \%$$

BAB V

PEMBAHASAN

5. 1. Pembahasan Model

Pada penelitian ini data yang digunakan adalah data real peneliti yang di dapat dengan mengambil gambar secara langsung dengan menggunakan kamera smartphone realme 5i dengan resolusi kamera 24MP. Data yang digunakan berjumlah 120 dataset gambar cabai, yang terdiri dari 90 data training dan testing 30. Berdasarkan hasil penelitian data yang diambil terdiri dari 3 kategori baik, sedang, dan tidak baik. Selanjutnya citra cabai di ekstrasi ciri menggunakan *Gray Level CO-Occurrence Matrix*. Setelah di dapatkan ciri dari citra cabai selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra cabai dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model *K-NN*.

Pada percobaan model *K-NN*, dengan jumlah nilai $K = 1$, dan Arah *GLCM* = 45° dengan jarak = 2, maka mendapatkan nilai akurasi sebesar =56 %, dengan data testing 30 dan training 90 dengan kategori baik, sedang dan tidak baik dengan masing-masing 30 data. Sedangkan pada percobaan model *K-NN*, dengan jumlah $K=2$, dan Arah *GLCM* = 90° dengan jarak = 2, maka mendapatkan nilai akurasi = 50%, dengan menggunakan dataset yang sama seperti dalam percobaan model sebelumnya.

Pada percobaan model *K-NN*, dengan jumlah $K = 3$, dan Arah *GLCM* = 135° dengan jarak = 2, Maka Mendapatkan Nilai Akurasi Sebesar = 73 %, dengan Data testing 30 training 90 dengan kategori baik, sedang, dan tidak baik dengan masing-masing 30 data. Pada percobaan model *K-NN*, dengan jumlah $K = 4$, dan Arah *GLCM* = 0° dengan jarak = 1, Maka Mendapatkan Nilai Akurasi sebesar =63% dengan menggunakan dataset yang sama seperti dalam percobaan model sebelumnya

Pada percobaan dengan jumlah $K = 5$, dan Arah $GLCM = 0^\circ$ dengan jarak = 1, maka mendapatkan nilai akurasi tertinggi Sebesar = 83 %, dengan data training 90, yaitu data cabai kualitas baik = 30, data kualitas sedang = 30, dan data Kualitas tidak baik = 30 Data Testing = 30. Pada percobaan model $K-NN$, dengan jumlah $K = 6$, dan Arah $GLCM = 0^\circ$ dengan jarak = 1, Maka Mendapatkan Nilai Akurasi sebesar =66% dengan menggunakan dataset yang sama seperti dalam percobaan model sebelumnya.

Sedangkan pada percobaan dengan jumlah $K=7$, dan arah $GLCM=0^\circ$ dengan jarak =1, mendapatkan nilai akurasi sebesar 80 %, dengan data training 90 data training dan 30 data testing.

Dalam hal ini terdapat kendala saat penelitian pengenalan kualitas cabai . Hal ini terjadi karena penggunaan data sampel cabai harus diperhatikan dari segi pencahayaan dan posisi dari data sampel, karena ada beberapa sampel data yang tidak sesuai dikenali. Nilai K dari klasifikasi $K-NN$ sangat berpengaruh untuk pengenalan kualitas cabai, hasil yang di dapatkan untuk metode $GLCM$ dengan $K-NN$ dengan arah 0° , jarak 1 dan nilai $K=5$ lebih banyak mengenali kualitas dan akurasi tertinggi. Hasil yang di dapatkan dari pengujian metode akurasi yang diperoleh sangat baik, hasil dari perhitungan akurasi di atas bahwa nilai K dengan angka ganjil memberikan nilai akurasi yang sangat baik ketimbang dengan nilai K dengan angka genap yang memberikan akurasi cukup rendah.

5. 2. Pembahasan Sistem

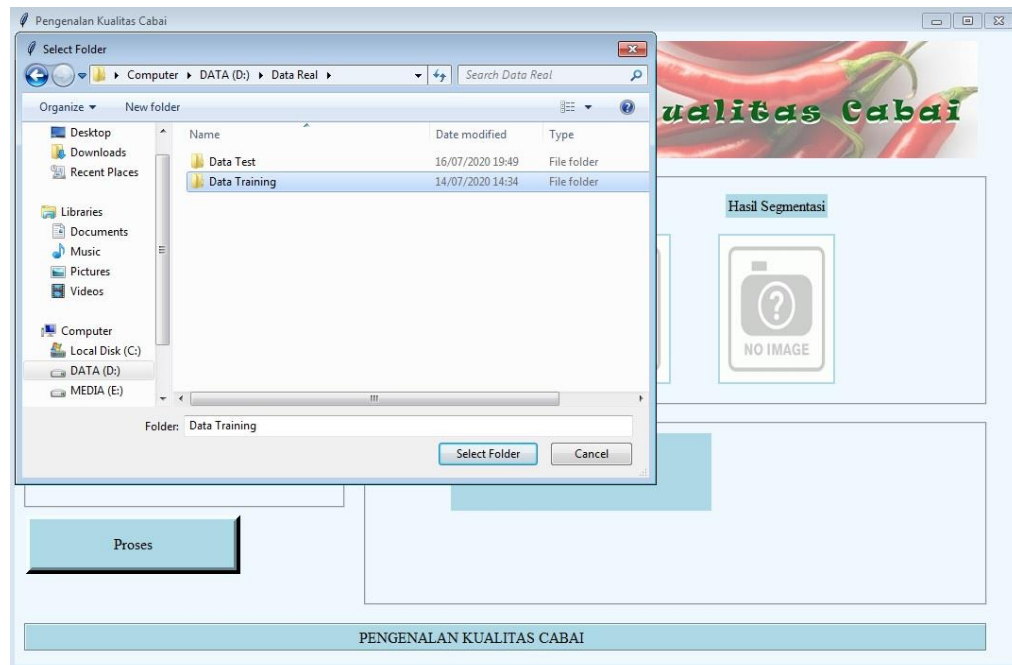
1. Tampilan Menu Utama



Gambar 5. 1: Tampilan Menu Utama

Tampilan pertama saat program dijalankan.

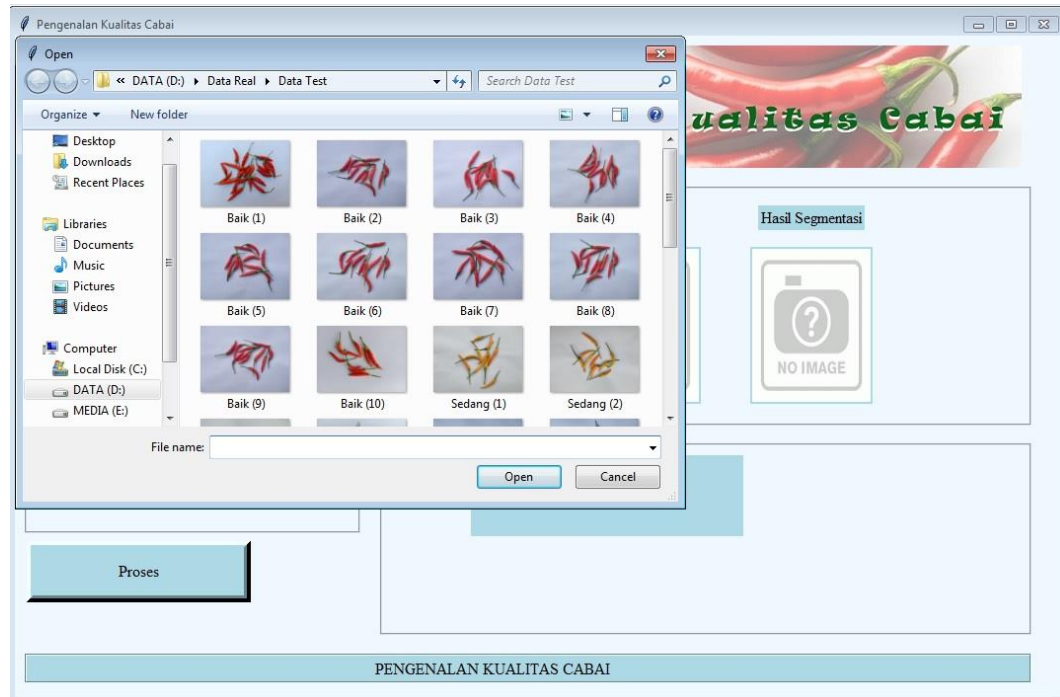
2. Tampilan Browse Data training



Gambar 5. 2: Tampilan Browse Data Training

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data traning.

3. Tampilan Browse Data testing



Gambar 5. 3: Tampilan Browse Data Training

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data testing

4. Tampilan Pemasukan Nilai *GLCM*

The screenshot shows the application window titled "Pengenalan Kualitas Cabai". It features a header with a chili pepper image and the title "APLIKASI Pengenalan Kualitas Cabai". The main interface is divided into several sections:

- Input Gambar:** Contains two rows of buttons for "Image Training" and "Image Testing", each followed by a file path field ("D:/Data Real/Data Trair" and "D:/Data Real/Data Test") and a "Browse" button.
- Gray Level Co-Occurrence Matrix:** Includes a dropdown menu set to "GLCM" and a "Jarak" (Distance) field with a list of values: 0, 45, 90, and 135.
- K - Neural Network:** Features a "Jumlah Nilai K" dropdown menu.
- Proses:** A large blue button at the bottom left.
- Output Area:** On the right, there are three placeholders for "Image Testing", "Grayscale", and "Hasil Segmentasi", each with a "NO IMAGE" label. Below these is a large "Output" box.
- Footer:** A blue bar at the bottom with the text "PENGENALAN KUALITAS CABAI".

Gambar 5. 4: Tampilan Pemasukan Nilai *GLCM*

Tampilan ini akan muncul saat pengguna telah memasukan nilai dan memilih perhitungan pada *GLCM*.

5. Tampilan Pemasukan Nilai *K-NN*

This screenshot shows the same application window as Gambar 5.4, but with the "Jumlah Nilai K" dropdown menu set to "1". The "Proses" button remains at the bottom left. The right side of the interface, including the "Image Testing", "Grayscale", and "Hasil Segmentasi" placeholders and the "Output" box, is identical to the previous image. The footer bar also remains the same.

Gambar 5. 5: Tampilan Pemasukan Nilai *K-NN*

Tampilan ini akan muncul saat pengguna telah memasukan nilai K.

6. Tampilan Output Hasil Pengenalan Kualitas Cabai



Gambar 5. 6: Tampilan Output Hasil Pengenalan Kualitas Cabai

Tampilan ini akan muncul saat pengguna mengklik/memilih tombol button proses setelah menginput data traning, data testing, arah dan jarak *GLCM* serta nilai k. Tampilah ini adalah hasil output program aplikasi pengenalan kualitas cabai.

BAB VI

PENUTUP

6. 1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada pengenalan kualitas cabai dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik suatu kesimpulan bahwa:

1. Secara umum bahwa aplikasi sistem pengenalan kualitas cabai menggunakan metode *Gray Level Co-Occurrence Matrix* dan *K-Nearest Neighbor* yang dibuat pemodelannya sudah berhasil.
2. Dari hasil yang diperoleh peningkatan akurasi metode *Gray Level Co-Occurrence Matrix* (GLCM) yang dikombinasikan dengan metode *K-Nearest Neighbor* (K-NN) untuk pengenalan kualitas cabai menunjukkan kinerja yang baik, dengan akurasi 80% dengan sudut GLCM = 0° dan jarak = 1 dengan nilai K=7 dengan data training 90 dan data testing 30. Dari hasil perhitungan diatas bahwa nilai K dengan angka ganjil memberikan akurasi yang baik.

6. 2. Saran

Setelah melakukan Penelitian pengenalan kualitas cabai, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu sebagai berikut:

1. Penulis berharap pada penelitian selanjutnya dapat menggunakan data sampel yang lebih banyak agar hasil akurasi lebih tinggi.
2. Penelitian selanjutnya dapat menggunakan metode Kasifikasi seperti *Artificial Neural Network* dan *Naive bayes*.

DAFTAR PUSTAKA

- [1] Yuvida Anindita Fitrianingtyas and Cahya Rahmad, “Sistem Pakar Deteksi Hama Dan Penyakit Pada Tanaman Cabai Dengan Metode Naive Bayes,” *Prosiding Seminar Informatika Aplikatif Polinema* , no. 1, ISSN: 2460-1160, 2015.
- [2] Hestiani Wulandari, Anang Kurnia and Bambang Sumantri, “Penerapan Analisis Regresi Spline Untuk Menduga Harga Cabai Di Jakarta,” *Indonesian Journal of Statistics and Its Applications*, vol. 1, p. 2, 2017.
- [3] Abd. Ghofur, “Implementasi Metode Klasifikasi Naive Bayes Untuk Memprediksi Kualitas Cabai,” *Jurnal Ilmiah Informatika*, vol. 1, 2016.
- [4] Restu Widodo, Agus Wahyu Widodo and Arry Supriyanto, “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 11, ISSN: 2548-964X, November 2018.
- [5] Anita Ahmad Kasim and Agus Harjoko, “Klasifikasi Citra Batik Menggunakan Jaringan Syaraf Tiruan Berdasarkan Gray Level Co-Occurrence,” *Journal Aplikasi Teknologi Informasi*, Juni 2014.
- [6] Fernando Robertide Siqueira, William Robson Schwartz and Helio Pedrini, “Multi-scale Gray level co-occurrence matrices for texture description,” *journalhomepage: www.elsevier.com/locate/neucom*, 2013.
- [7] Saifudin and Abdul Fadlil, “Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Co-Occurrence Matrix Dengan Klasifikasi Jarak Euclidean,” no. ISSN: 1410-233, Program Studi Teknik Informatika, Fakultas Teknologi Industri Universitas Ahmad Dahlan Yogyakarta, 2013.
- [8] Neneng and Yusra Fernando , “Klasifikasi Jenis Daging Berdasarkan Analisis Citra Teksture Gray Level Co-occurrence Matrix dan Warna,” *Website : jurnal.umj.ac.id/index.php/semnastek*, no. ISSN : 2407 – 1846, Teknik Informatika, STMIK Teknokrat, Bandar Lampung, November 2017.
- [9] Siti Mutrofin, Arrie Kurniawardhani, Abidatul Izzah and Mukhamad Masrur, “Optimasi Teknik Klasifikasi Modified k Nearest Neighbor Menggunakan Algoritma Genetika,” *JURNAL GAMMA*, vol. 10, no. ISSN 0216-9037, 2015.

- [10] Fittria Shofrotun Ni'mah, T. Sutojo and De Rosal Ignatius Moses Setiadi, "Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurrence Matrix dan K-Nearest Neighbor," *Journal Teknologi dan Sistem Komputer*, vol. 6, no. -ISSN:2338-0403, Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang, Indonesia, April 2018.
- [11] Ion Ataka Halela, Bowo Nurhadiyono, S.Si, M.Kom and Farah Zakiyah Rahmanti, "Identifikasi Jenis Buah Apel Menggunakan Algoritma K-Nearest Neighbor (KNN) dengan Ekstraksi Fitur Histogram," Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Jl. Nakula 1 No. 5-11, Jawa Tengah, 2017.
- [12] Kasdi Subagyo, "Budidaya dan Pasca Panen Cabai Merah," Balai Pengkajian Teknologi Pertanian, Jawa Tengah, 2010.
- [13] Puji Aswari And Nova Eka Diana, "Identifikasi Emosi Berdasarkan Action Unit Menggunakan Metode Bezier Curve," vol. 20, no. 1, ISSN. 1410-2331, 2016.
- [14] Yogi Adi Nugraha, "Implementasi Sistem Otomatis Pada Robot Kapal Berbasis Computer Vision Untuk Kontes Kapal Cepat Tsk Berawak Nasional," *Telekontran*, vol. 2, no. 1, November 2014.
- [15] Rizki Anggasari Suandari and Rossa Andrie Asmara, "Pengenalan Ekspresi Wajah Berdasarkan Ekstraksi Fitur Alis, Mata Dan Mulut," *Proseeding Seminar Informatika Aplikasi Polinema*, no. ISSN. 2460-1160, Malang, 2015.
- [16] Amalia, "Pengenalan Citra Tanda Tangan Menggunakan Citra Tanda Tangan Menggunakan Gray Level Co-Occurrence Matrix Dan Probabilistic Neural Network," *Journal Tekno*, vol. XIV, no. 1, 2014.
- [17] Tobias d.s, "Deteksi Glaukoma Pada Citra Fundus Retina Dengan Metode K-Nearest Neighbor," 2016.
- [18] Azwar, "Integrasi Ekstraksi Fitur Local Binary Pattern Dan Gray Level Co-Occurrence Matrix Untuk Pengenalan Ekspresi Mulut," vol. IX, no. 1, 2017.
- [19] Andri, Yesi Novaria Kunang and Sri Murniati, "Implementasi Teknik Data Mining Untuk Memprediksi Tingkat Kelulusan Mahasiswa Pada Universitas Bina Darma Palembang," *Seminar Nasional Informatika*, no. Jurusan Sistem Informasi, Universitas Bina Darma Palembang, Mei 2013.

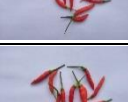
- [20] Gorunescu and Florin, "Data Mining: Concepts and Techniques," no. Verlag Berlin Heidelberg: Springer , 2011.
- [21] Ahsanti N., "Implementasi Algoritma K-Nearest Neighbor Dalam Sistem Case Based Reasoning Untuk Pembentukan Identitas Jawaban Otomatis Dan Pencari Kemiripan Jawaban Dari Soal-Soal Algorithim," 2016.
- [22] Ivan Jaya, Ainul Hizriadi and Evi Sersanti Purba, "Klasifikasi Surat Laporan Kehilangan Kepolisian Menggunakan Algoritma K-Nearest Neighbor," vol. 10, no. 2, Program Studi Teknologi Informasi, Universitas Sumatera Utara, Oktober 2018.
- [23] P. Mayadewi And E. Rosely, "Prediksi Nilai Proyek Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining," *Seminar Nasional Sistem Informasi*, 2015.
- [24] S. Heranurweni, "Pengenalan Wajah Menggunakan Learning Vector Quantization (LVQ)," *Prosiding Seminar Nasional Sains dan Teknologi*, 2010.
- [25] J. Han and M. Kember, "Data Mining Concepts & Techniques 2nd Edition," no. San Fransisco: Elsevier, 2006.
- [26] D. M. Powers, "Evaluation From Precision, Recall, and F-Measure to ROC, Infomedness Markedness and Correlation," *Journal Of Machine Learning Technologies* , 2011.
- [27] T. R. Perkasa, H. Widyantara and P. Susanto, "Rancang Bangun Pendeteksi Gerak Menggunakan Metode Image Subtraction Pada Single Board Computer (SBC)," *Jcones*, vol. III, no. 2, 2014.

LAMPIRAN

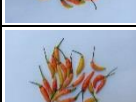
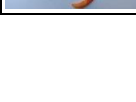
Lampiran 1: Dataset

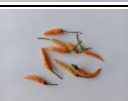
Total seluruh dataset yang digunakan berjumlah 120 data real yang terdiri dari 90 data training dan 30 data testing.

1. Data Training

NO	Data Training	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
0		187,04683	0,735687	0,832879	2,531654	Baik
1		169,24058	0,740936	0,826774	2,528554	Baik
2		237,08184	0,759612	0,831124	2,41002	Baik
3		222,52277	0,675966	0,779278	3,079407	Baik
4		230,3941523	0,6699928	0,7744366	3,151633	Baik
5		185,3570984	0,717244	0,8133303	2,711706	Baik
6		191,0585433	0,7633352	0,8424062	2,326267	Baik
7		133,5742955	0,7348831	0,846579	2,495626	Baik
8		139,9435776	0,7134079	0,8317291	2,67781	Baik
9		180,7759314	0,7647105	0,8407752	2,315228	Baik
10		143,8065893	0,784174	0,8635662	2,116894	Baik
11		181,1183144	0,7130118	0,8225514	2,713203	Baik
12		183,6517303	0,7257124	0,8267773	2,621256	Baik

13		179,5487591	0,758622	0,8414407	2,364896	Baik
14		158,7192557	0,7327237	0,8234537	2,585095	Baik
15		160,1781472	0,7183712	0,8309518	2,658227	Baik
16		270,8504478	0,6653396	0,7685848	3,198754	Baik
17		191,0846685	0,7429944	0,8275643	2,519898	Baik
18		155,5725366	0,7631339	0,8515185	2,292426	Baik
19		182,0419617	0,7482737	0,8324076	2,44302	Baik
20		207,6353449	0,748661	0,8315823	2,459232	Baik
21		188,6092331	0,7466745	0,828595	2,491735	Baik
22		193,5902534	0,7338428	0,8208433	2,573731	Baik
23		146,4061856	0,7329519	0,8265489	2,56044	Baik
24		183,2075012	0,7427339	0,8276852	2,505804	Baik
25		178,7766124	0,7480667	0,8328092	2,467029	Baik
26		169,0870344	0,7560835	0,8391129	2,387447	Baik
27		139,7213662	0,7669463	0,8515063	2,269595	Baik
28		272,5127735	0,6738015	0,7731896	3,133995	Baik

29		163,1063937	0,7422808	0,8343306	2,488089	Baik
30		206,5681101	0,6857478	0,7719108	3,0703443	Sedang
31		270,2108807	0,7297357	0,7970623	2,7287097	Sedang
32		261,7679196	0,7155541	0,7945168	2,8024768	Sedang
33		271,3800294	0,6364197	0,7309582	3,4865087	Sedang
34		157,8225641	0,8176727	0,8595514	1,9182112	Sedang
35		234,2630119	0,7361209	0,8066362	2,6457452	Sedang
36		188,0382419	0,7433757	0,8201206	2,5515477	Sedang
37		246,4471086	0,7313556	0,8027563	2,69532	Sedang
38		274,6710262	0,6873246	0,7708504	3,0771934	Sedang
39		285,4665357	0,6792407	0,7611861	3,1501604	Sedang
40		213,0713932	0,7438344	0,8149889	2,5751968	Sedang
41		218,3897019	0,7711414	0,8271328	2,3339511	Sedang
42		213,8430408	0,742499	0,8143503	2,5869804	Sedang
43		283,2894365	0,7212427	0,7825114	2,8119935	Sedang
44		237,1979246	0,7346055	0,8049602	2,6604456	Sedang

45		187,1957104	0,7957549	0,8436665	2,1167916	Sedang
46		243,8582342	0,7141139	0,7959463	2,7996028	Sedang
47		184,824266	0,794455	0,8426397	2,1307802	Sedang
48		149,2444608	0,8632337	0,8922843	1,4895188	Sedang
49		275,6054076	0,7303411	0,7978748	2,723818	Sedang
50		200,1722897	0,7445643	0,8170613	2,4696318	Sedang
51		255,4953803	0,6686174	0,7626849	3,1295666	Sedang
52		149,3844648	0,8082876	0,8609453	1,9095425	Sedang
53		242,2508613	0,7312715	0,8074412	2,5732634	Sedang
54		223,9105562	0,7645038	0,8326347	2,2964873	Sedang
55		169,3713669	0,7338815	0,8210154	2,5431316	Sedang
56		246,6989164	0,7331588	0,8076948	2,5688484	Sedang
57		199,3159811	0,7898648	0,8498527	2,0707176	Sedang
58		246,4650897	0,7198182	0,8065605	2,6673693	Sedang
59		181,1611478	0,7382488	0,817386	2,5195536	Sedang
60		127,9944045	0,8071165	0,8492726	2,022402	Tidak baik

61		139,3053404	0,7585621	0,8156701	2,467619	Tidak baik
62		122,6045643	0,7918765	0,8446891	2,139097	Tidak baik
63		129,9924727	0,8061447	0,8475193	2,035021	Tidak baik
64		165,0623875	0,7938927	0,8333638	2,172496	Tidak baik
65		131,694616	0,793018	0,8431506	2,130781	Tidak baik
66		139,3375293	0,7957153	0,8366732	2,160705	Tidak baik
67		129,191163	0,7932722	0,8450354	2,126165	Tidak baik
68		175,5424575	0,7846902	0,8267026	2,263384	Tidak baik
69		132,3746816	0,7926487	0,8420255	2,112147	Tidak baik
70		136,2044531	0,7954575	0,8382472	2,159056	Tidak baik
71		183,5609343	0,784973	0,825311	2,271457	Tidak baik
72		122,3870152	0,7529535	0,8189897	2,463977	Tidak baik
73		162,2707145	0,7937829	0,8343242	2,169327	Tidak baik
74		146,383317	0,78272	0,8241449	2,274693	Tidak baik
75		139,9148011	0,7584369	0,8161484	2,463548	Tidak baik
76		159,9381419	0,7939674	0,8344955	2,166136	Tidak baik

77		136,9756796	0,7870791	0,8291244	2,230717	Tidak baik
78		120,7583135	0,805323	0,8474782	2,053081	Tidak baik
79		167,3876844	0,8094849	0,8435449	2,041133	Tidak baik
80		136,8870221	0,7874354	0,8303267	2,2275027	Tidak baik
81		101,6773954	0,8721024	0,8990136	1,3997753	Tidak baik
82		136,5794634	0,8214182	0,8580528	1,9101366	Tidak baik
83		125,0640559	0,8044479	0,8468942	2,0593851	Tidak baik
84		139,1234127	0,758123	0,8169226	2,4643729	Tidak baik
85		151,9133497	0,8015874	0,8444577	2,0836981	Tidak baik
86		136,7363959	0,822956	0,859212	1,8906995	Tidak baik
87		144,4806002	0,8157958	0,8523025	1,9701606	Tidak baik
88		157,7773421	0,7942818	0,8345725	2,16476	Tidak baik
89		130,0513369	0,8065095	0,8483927	2,0289066	Tidak baik

2. Data Testing

NO	Data Testing	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Kelas
0		159,62536	0,713968	0,803822	2,768729	?
1		193,38208	0,712391	0,807353	2,75858	?
2		147,95112	0,711139	0,813388	2,707439	?
3		182,52725	0,713694	0,805412	2,751154	?
4		178,025455	0,7411972	0,8226465	2,514649	?
5		154,7768112	0,669834	0,8008488	3,020885	?
6		172,6231623	0,72627	0,8182856	2,594977	?
7		154,6104912	0,6854944	0,8142189	2,881012	?
8		191,8176442	0,6775895	0,7839512	3,013344	?
9		173,0297233	0,7097886	0,8078319	2,742195	?
10		259,3789781	0,7697121	0,8491691	2,242321	?
11		230,2356871	0,7531654	0,8407555	2,36262	?
12		245,9985694	0,7726168	0,8522398	2,195154	?
13		239,3097419	0,7258999	0,8342048	2,549849	?

14		250,6630514	0,7006477	0,7776107	2,967579	?
15		205,235204	0,7163826	0,8314265	2,581276	?
16		214,0740369	0,7587396	0,8480692	2,280396	?
17		116,7754985	0,7619747	0,8807703	2,148097	?
18		271,590463	0,7444071	0,8273094	2,474169	?
19		242,0859155	0,7815357	0,8552538	2,143127	?
20		133,8511574	0,7920264	0,8573554	2,111947	?
21		142,8157124	0,7776856	0,8498266	2,232234	?
22		167,1556439	0,7439328	0,8242627	2,554956	?
23		147,0608924	0,7385173	0,8317658	2,556421	?
24		140,1970982	0,7784326	0,8510551	2,217371	?
25		126,9582463	0,7545686	0,8351424	2,418547	?
26		125,0733339	0,7430238	0,8340913	2,520963	?
27		97,44452653	0,8478172	0,8927099	1,587978	?
28		162,063661	0,7484981	0,8125932	2,51246	?
29		164,5959921	0,7750943	0,8478119	2,258262	?

Lampiran 2: KODE PROGRAM

1. Pembuatan From

```

import tkinter as tk
import tkinter.filedialog as fdialog
import tkinter.ttk as ttk
from PIL import Image, ImageTk
import training
import cv2

class aplikasiku(tk.Frame):
    def __init__(self, parent):
        tk.Frame.__init__(self)
        self.parent = parent
        self.frames()
        self.form()

    def frames(self):
        #frame judul
        self.FrameAtas = tk.Frame(self.parent, width=1010, height=150,
background="aliceblue", relief='raise')
        self.FrameAtas.grid(row=0, column=0, columnspan=3, padx=9, pady=10,
sticky="NSEW")
        self.FrameAtas.grid_propagate(0)

        #frame input gambar

```

```

        self.MainFrame1 = tk.LabelFrame(self.parent, height=120,
background="aliceblue", text="Input Gambar", font="Times", fg="black")

        self.MainFrame1.grid(row=1, column=0, columnspan=3, padx=10,
sticky="NSEW")

        self.MainFrame1.grid_propagate(0)


#glcm

        self.MainFrame2 = tk.LabelFrame(self.parent, height=120,
background="aliceblue", text="Gray Level Co-Occurrence Matrix",
font="Times", fg="black")

        self.MainFrame2.grid(row=2, column=0, padx=10, pady=10,
sticky="NSEW")

        self.MainFrame2.grid_propagate(0)


#knn

        self.MainFrame3 = tk.LabelFrame(self.parent, height=100,
background="aliceblue", text="K - Neural Network", font="Times", fg="black")

        self.MainFrame3.grid(row=3, column=0, padx=10, sticky="NSEW")

        self.MainFrame3.grid_propagate(0)


#
        self.MainFrame12 = tk.LabelFrame(self.parent, height=10,
background="lightblue", font="Times", fg="black")

#
        self.MainFrame12.grid(row=4, column=0, rowspan=1, padx=10,
sticky="NSEW")

#
        self.MainFrame12.grid_propagate(0)

#

#proses

        self.MainFrame4 = tk.Frame(self.parent, height=100,
background="aliceblue")

        self.MainFrame4.grid(row=5, column=0, padx=10, pady=6,
sticky="NSEW")

        self.MainFrame4.grid_propagate(0)

```

```

        #hasil

        self.MainFrame5 = tk.LabelFrame(self.parent, background="aliceblue",
font="Times", fg="black")

        self.MainFrame5.grid(row=1, column=1, rowspan=2, columnspan=2,
padx=10, pady=9, sticky="NSEW")

        self.MainFrame5.grid_propagate(0)


        #kaki

        self.MainFrame6 = tk.LabelFrame(self.parent, height=30, width=1020,
background="aliceblue")

        self.MainFrame6.grid(row=6, column=0, columnspan=3, padx=10, pady=10,
sticky="NSEW")

        self.MainFrame6.grid_propagate(0)


        #hasil

        self.MainFrame7 = tk.LabelFrame(self.parent, background="aliceblue",
font="Times", fg="black")

        self.MainFrame7.grid(row=3, column=1, rowspan=3, columnspan=2,
padx=10, pady=9, sticky="NSEW")

        self.MainFrame7.grid_propagate(0)


    def form(self):

#        frame
#        logo

        self.PathTestingImg ='D:/Cabai4.jpg'

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

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

```

```

        self.label1 = tk.Label(self.FrameAtas, width=1010, height=120,
                                bg="aliceblue", image=self.imT)
        self.label1.pack(side="top")

#frame 1

#label data training

self.label1 = tk.Label(self.MainFrame1, text="Image Training",
                        font="Times", relief="ridge", width=12, background="lightblue")
self.label1.grid(row=3, column=0, sticky="NSEW", padx=8, pady=5)

#textbox data training

self.nilai = tk.StringVar(self.MainFrame1, value="")
self.txtrain = tk.Entry(self.MainFrame1, textvariable = self.nilai, width=20)
self.txtrain.grid(row=3, column=1, sticky="NSEW", pady=5)

#button browse training

self.path = tk.Button(self.MainFrame1, text="Browse", font="Times",
                        width=6, background="lightblue",
                        command=lambda:self.nilai.set(fdialog.askdirectory()))
self.path.grid(row=3, column=5, padx=5)

#label data testing

self.label2 = tk.Label(self.MainFrame1, text="Image Testing", font="Times",
                        relief="ridge", background="lightblue")
self.label2.grid(row=4, column=0, sticky="NSEW", padx=8, pady=10)

#textbox data testing

self.nilai1 = tk.StringVar(self.MainFrame1, value="")
self.txtest = tk.Entry(self.MainFrame1, textvariable = self.nilai1)
self.txtest.grid(row=4, column=1, sticky="NSEW", pady=10)

#button browse tetsting

```



```

        self.path1 = tk.Button(self.MainFrame1, text="Browse",
                                font="Times",background="lightblue",
                                command=lambda:self.nilai1.set(fdialog.askopenfilename()))

        self.path1.grid(row=4, column=5, sticky="NSEW", padx=5, pady=5)


#frame 2
#COMBOBOX GLCM
v = tk.IntVar(self.MainFrame2)

self.label = tk.Label(self.MainFrame2, text="GLCM", width=12,
                        font="Times", relief="ridge", background="lightblue")
self.label.grid(row=5, column=0, padx=8 , pady=10)


#    arah GLCM
arah = tk.StringVar(self.MainFrame2, value="")

self.cbarahglcm = ttk.Combobox(self.MainFrame2,
                                values=("0", "45", "90", "135"), textvariable = arah, width=10)
self.cbarahglcm.grid(row=5, column=1, sticky="NSEW", pady=10)


#    jarak GLCM
self.label1 = tk.Label(self.MainFrame2, text="Jarak", font="Times",
                        relief="ridge", width=12, background="lightblue")
self.label1.grid(row=6, column=0, sticky="NSEW", padx=8, pady=5)

jarak = tk.StringVar(self.MainFrame2, value="")
self.tbjarak = tk.Entry(self. MainFrame2, textvariable = jarak, width=30)
self.tbjarak.grid(row=6, column=1, sticky="NSEW", padx=8, pady=5 )


#frame 3
#klasifikasi KNN
#label iterasi

```

```

        self.label = tk.Label(self.MainFrame3, text="Jumlah Nilai K", font="Times",
relief="ridge", width=12, background="lightblue")

        self.label.grid(row=0, column=0, sticky="NSEW", padx=8, pady=5)

#textbox iterasi
        self.valdist = tk.StringVar(self.MainFrame3, value="")

        self.cb_nilaiK = ttk.Combobox(self.MainFrame3, textvariable=self.valdist,
width=30)

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

        self.cb_nilaiK.grid(row=0, column=1, sticky="NSEW", pady=5 )

#frame 4
#button proses
        self.path = tk.Button(self.MainFrame4, text="Proses",
font="Times",height=2, width=23, background="lightblue",
command=self.proses)

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

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

#frame 5
#output
#label hasil image testing
        self.label = tk.Label(self.MainFrame5, text="Image Testing", font="Times",
background="lightblue")

        self.label.grid(row=0, column=0, padx=3, pady=17)

#image hasil testing
        self.noimage = 'D:/noimage.jpg'

        self.noimageTest = Image.open(self.noimage)

```

```

self.noimageTest = ImageTk.PhotoImage(self.noimageTest)
self.labelTest = tk.Label(self.MainFrame5, bg = "lightblue",
image=self.noimageTest)
self.labelTest.grid(row=1,column = 0, padx=25)

#label Grayscale
self.label7 = tk.Label(self.MainFrame5, text="Grayscale", font="Times",
background="lightblue")
self.label7.grid(row=0, column=2, padx=35, pady=5)

#image grayscale
self.noimage1 = 'D:/noimage.jpg'
self.noimageGLCM = Image.open(self.noimage1)
self.noimageGLCM = ImageTk.PhotoImage(self.noimageGLCM)
self.labelGLCM = tk.Label(self.MainFrame5, bg = "lightblue",
image=self.noimageGLCM)
self.labelGLCM.grid(row=1,column = 2, padx=25)

#label hasil segmentasi
self.label4 = tk.Label(self.MainFrame5, text="Hasil Segmentasi",
font="Times", background="lightblue")
self.label4.grid(row=0, column=3, padx=30, pady=5)

#image hasil segmentasi
self.noimage2 = 'D:/noimage.jpg'
self.noimageHiz = Image.open(self.noimage2)
self.noimageHiz = ImageTk.PhotoImage(self.noimageHiz)
self.labelHiz = tk.Label(self.MainFrame5, bg = "lightblue",
image=self.noimageHiz)
self.labelHiz.grid(row=1,column = 3, padx=25)

```

```

#frame 6

#label kaki

self.label = tk.Label(self.MainFrame6, text="PENGENALAN KUALITAS
CABAI", font="Times")

self.label.grid(row=2, column=2, sticky="NSEW", padx=350, pady=1)

self.label.config(background="aliceblue")

self.MainFrame6.grid_propagate(0)

#    frame 7

#label keterangan hasil output

self.label12 = tk.Label(self.MainFrame7, text="Output", width=30,height=4,
font="Times")

self.label12.grid(row=2, column=0, columnspan=5, padx=90, pady=10)

self.label12.config(background="Lightblue")

# self.label13 = tk.Label(self.MainFrame7, text="Akurasi",
width=30,height=3, font="Times")

# self.label13.grid(row=3, column=0, columnspan=5, padx=50, pady=10)

def proses(self):

    # tic = time.time()

    folder_train = self.txtrain.get()

    image_test = self.txtest.get()

    arah = self.cbarahglcm.get()

    jarak = self.tbjarak.get()

    K = self.cb_nilaiK.get()

    #hasil output akhir

    a = training.ekstraksiFitur()

```

```
Hasil, img_segmen, gray, = a.proses_rekognisi(folder_train, image_test,
jarak, arah, K)
```

```
self.label12.configure(text = Hasil)
```

```
#hasil image 1
```

```
imgTest = Image.open(self.txtest.get())
```

```
imgTest = imgTest.resize((165, 165), Image.ANTIALIAS)
```

```
imgTest = ImageTk.PhotoImage(imgTest)
```

```
self.labelTest.configure(image=imgTest)
```

```
self.labelTest.image_names = imgTest
```

```
#hasil image 2
```

```
imgGray = Image.fromarray(gray)
```

```
imgGray = imgGray.resize((165, 165), Image.ANTIALIAS)
```

```
imgGray = ImageTk.PhotoImage(imgGray)
```

```
self.labelGLCM.configure(image=imgGray)
```

```
self.labelGLCM.image_names = imgGray
```

```
#Hasil Image 3
```

```
img = cv2.cvtColor(img_segmen, cv2.COLOR_BGR2RGB)
```

```
img = Image.fromarray(img)
```

```
img = img.resize((165,165), Image.ANTIALIAS)
```

```
img = ImageTk.PhotoImage(img)
```

```
self.labelHiz.configure(image=img)
```

```
self.labelHiz.image_names = img
```

```
def main():
```

```

root = tk.Tk()
root.geometry('1060x670+50+50')
root.configure(background='aliceblue')
app = aplikasiku(root)
app.parent.title("Pengenalán Kualitas Cabai")
root.mainloop()

```

```

if __name__ == '__main__':
    main()

```

2. Proses Rekognisi

```

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

class ekstraksiFitur:
    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, 225,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

```

```

#cv2.imshow('hasil otsu', alpha)
alpha[alpha==225] = 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, jarak, arah):
    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, jarak, arah):
    pathList = self.get_img_path(folderPath)
    images = []
    for imPath in pathList:
        img = cv2.imread(imPath)
        images.append(img)
    data_train = []
    for img in images:
        hasil_segmen, gray = self.segmentasi(img)
        ekstraksi = self.fitur_glcml(hasil_segmen, jarak, arah)
        data_train.append(ekstraksi)
    label_train = self.get_label(folderPath)
    return data_train, label_train

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

```



```

def proses_rekognisi(self, folder_train, image_test, jarak, arah, k):
    x_train, y_train = self.training_data(folder_train, jarak, arah)
    x_test, hasil_segmen, gray = self.testing_data(image_test, jarak, arah)
    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, hasil_segmen, gray

```

3. GLCM

```

import cv2
import numpy as np
from skimage import feature
from scipy.stats import entropy

class ekstrasiFitur:
    def fitur_glcml(self, img):
        matrix_glcml = feature.greycomatrix(img, [1], [0], symmetric=False,
normed=True)
        contrast = feature.greycomprops(matrix_glcml, 'contrast')[0,0]
        energy = feature.greycomprops(matrix_glcml, 'energy')[0,0]
        homogeneity = feature.greycomprops(matrix_glcml, 'homogeneity')[0,0]
        ent = entropy(matrix_glcml.flatten())
        hasil_fitur = [contrast, energy, homogeneity, ent]
        return hasil_fitur

def main():
    a = ekstrasiFitur()

```

```
img = cv2.imread('baik (9).jpg')
img = cv2.resize(img, (225, 225))
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
fitur_glcmm = a.fitur_glcmm(gray)
print(fitur_glcmm)
cv2.waitKey(0)
cv2.destroyAllWindows()

if __name__ == '__main__':
    main()
```

Lampiran 3: SURAT REKOMENDASI PENELITIAN



PEMERINTAH PROVINSI GORONTALO DINAS PERTANIAN

Jln. Moh. Tayeb Gobel Kompleks Perkantoran Provinsi Gorontalo
Desa Ayula Utara Kec, Bulango Selatan Kab, Bone Bolango

SURAT KETERANGAN

Nomor : 570 / 856 / DP - SK / VII / 2020

Sekretaris Dinas Pertanian Provinsi Gorontalo dengan ini menerangkan bahwa :

N a m a : **ARYO MOKOAGOW**

Nim : T31151163

Program Studi : S-1 Teknik Informatika

Telah selesai mengikuti penelitian dengan judul Skripsi IDENTIFOKASI JENIS DAN KUALITAS CABAI MENGGUNAKAN METODE GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN K-NEARES NEIGHBOR (KNN). Kualitas Baik di Dinas Pertanian Provinsi Gorontalo mulai tanggal, 24 Maret 2019 yang dilaksanakan oleh Mahasiswa Universitas Ichsan Gorontalo.

Demikian Surat Keterangan ini diberikan untuk dipergunakan sebagaimana mestinya.


Kepala Dinas
SEKRETARIS,
HI. RELIANTO MAMONTO, SP. M.Si
NIP. 196406041988031002

Tembusan disampaikan kepada Yth :

1. Kepala Dinas Pertanian Provinsi Gorontalo (sebagai laporan);
2. Ketua Lembaga Penelitian Universitas Ichsan Gorontalo;
3. Pertinggal.

Lampiran 4: 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. 0407/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 Mahasiswa : ARYO ADI PRASETYO MOKOAGOW
NIM : T3115163
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Pengenalan Kualitas Cabai Menggunakan Metode Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 29%, 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, 21 Juli 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 5: HASIL TURNITIN

PENGENALAN KUALITAS CABAI MENGGUNAKAN METODE GRAY LEVEL CO-OCCURRENCE MATRIX DAN K-NEAREST NEIGHBOR

ORIGINALITY REPORT

29%	30%	7%	18%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.usd.ac.id Internet Source	2%
2	anzdoc.com Internet Source	2%
3	www.scribd.com Internet Source	2%
4	ilkom.unnes.ac.id Internet Source	2%
5	Submitted to Universitas Brawijaya Student Paper	2%
6	docplayer.info Internet Source	2%
7	repository.usu.ac.id Internet Source	2%
8	Submitted to Universitas Jenderal Achmad Yani Student Paper	2%

9	jurnal.pnl.ac.id Internet Source	1 %
10	media.neliti.com Internet Source	1 %
11	text-id.123dok.com Internet Source	1 %
12	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	1 %
13	jtsiskom.undip.ac.id Internet Source	1 %
14	digilib.uinsgd.ac.id Internet Source	1 %
15	igprp2210.blogspot.com Internet Source	1 %
16	ojs.uniska-bjm.ac.id Internet Source	1 %
17	jurnal.unived.ac.id Internet Source	1 %
18	j-ptiik.ub.ac.id Internet Source	1 %
19	medium.com Internet Source	1 %
20	ejournal.amiki.ac.id Internet Source	

		1 %
21	journal.stats.id Internet Source	<1 %
22	rafdhi.blogspot.com Internet Source	<1 %
23	rachmannulhakim19.blogspot.com Internet Source	<1 %
24	milyunima.wordpress.com Internet Source	<1 %
25	unsri.portalgaruda.org Internet Source	<1 %
26	id.123dok.com Internet Source	<1 %
27	ejournal.catursakti.ac.id Internet Source	<1 %
28	jip.polinema.ac.id Internet Source	<1 %
29	Submitted to Sriwijaya University Student Paper	<1 %
30	smai.fti.mercubuana-yogya.ac.id Internet Source	<1 %

Lampiran 6: SURAT BEBAS PUSTAKA



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN GORONTALO**

UPT. PERPUSTAKAAN PUSAT

SURAT KEPUTUSAN MENDIKS RI NO. 84/D/0/2001

Jln. Achmad Nadjamuddin No.17 Telp. (0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No.024/perpus_fikom/VII/2020

Perpustakaan Selasa, 21 Juli 2020 Fakultas Ilmu komputer (FIKOM) Universitas Ichsan
Gorontalo dengan ini menerangkan bahwa :

Nama : Aryo Adi Prasetyo Mokoagow
Nim : T3115163
No anggota : M201986

Terhitung sejak tanggal 21 Juli 2020, dinyatakan telah bebas dari pinjaman buku dan koleksi
lainnya dipergustakaan Fakultas Ilmu komputer.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 21 Juli 2020
Kepala Perpustakaan
Fakultas Ilmu Komputer

Apriyanto Alhamad, M.Kom
NIDN 09240486

Lampiran 7: DAFTAR RIWAYAT HIDUP

Nama : Aryo Adi Prasetyo Mokoagow

NIM : T3115163

Tempat Tanggal/lahir : Bongkudai 05-04-1997

Email : Aryomokoagow@gmail.com

Riwayat Pendidikan :

1. Peneliti lulus di Taman Kanak-kanak di TK Beringin pada tahun 2003
2. Peneliti lulus di Madrasah Ibtidaiyah Negeri Bongkudai induk pada tahun 2009
3. Peneliti lulus di Madrasah Tsanawiyah Negeri Bongkudai induk pada tahun 2012
4. Peneliti lulus pendidikan SMA Negeri 1 Modayag pada tahun 2015.
5. Peneliti Masuk dan di terimah di Universitas Ichsan Gorontalo pada Tahun 2015

