

**SISTEM CERDAS KLASIFIKASI KEMATANGAN DAN
HARGA BUAH PEPAYA BERDASARKAN EKSTRAKSI
FITUR GRAY LEVEL CO-OCCURENCE MATRIX DENGAN
METODE NAIVE BAYES**



Disusun Oleh:

N a m a : Hanung Setyo Sambudi

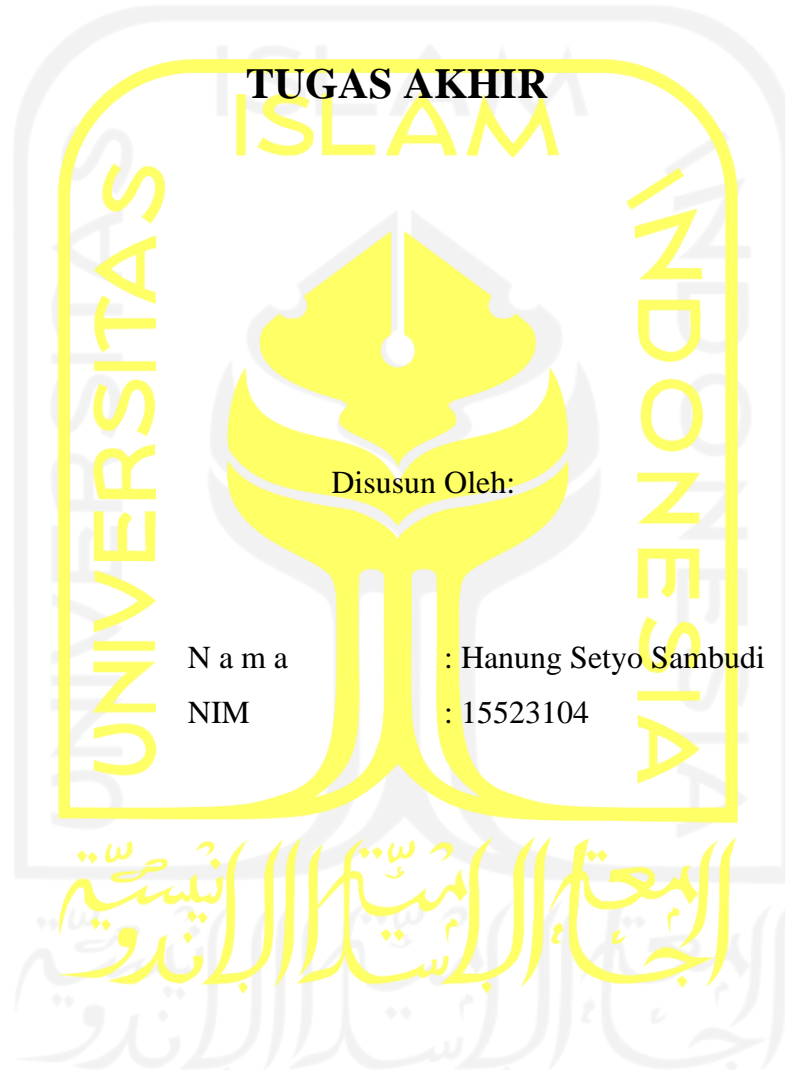
NIM : 15523104

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**SISTEM CERDAS KLASIFIKASI KEMATANGAN DAN
HARGA BUAH PEPAYA BERDASARKAN EKSTRAKSI
FITUR GRAY LEVEL CO-OCCURENCE MATRIX DENGAN
METODE NAIVE BAYES**



Yogyakarta, 17 Juni 2021

Pembimbing,

(Sri Mulyati, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**SISTEM CERDAS KLASIFIKASI KEMATANGAN DAN
HARGA BUAH PEPAYA BERDASARKAN EKSTRAKSI
FITUR GRAY LEVEL CO-OCCURENCE MATRIX DENGAN
METODE NAIVE BAYES**

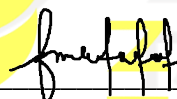
TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 17 Juni 2021

Tim Penguji

Sri Mulyati, S.Kom., M.Kom.



Anggota 1

Ahmad Fathan Hidayatullah, S.T., M.Cs.



Anggota 2

Arrie Kurniawardhani, S.Si., M.Kom.

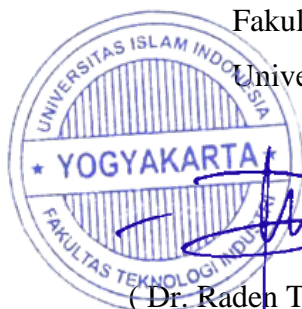


Mengetahui,

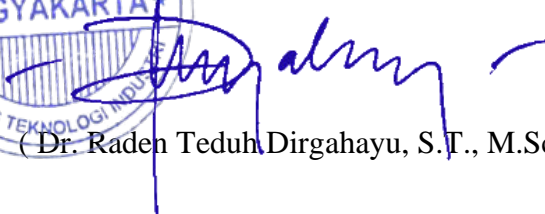
Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)



HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Hanung Setyo Sambudi

NIM : 15523104

Tugas akhir dengan judul:

SISTEM CERDAS KLASIFIKASI KEMATANGAN DAN HARGA BUAH PEPAYA BERDASARKAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURENCE MATRIX DENGAN METODE NAIVE BAYES

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 17 Juni 2021



(Hanung Setyo Sambudi)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah memberikan ridho, rahmat serta hidayah-Nya atas nikmat sehat, limpahan ilmu dan kekuatan yang tak terkira kepada saya, sehingga tak henti hentinya selalu bersyukur atas segala yang telah diberikan serta memohon ampun atas setiap kesalahan yang telah di perbuat baik sengaja maupun tidak disengaja.

Shalawat serta salam senantiasa tercurahkan kepada junjungan kita Nabi agung Muhammad SAW yang telah membawa kita semua dari zaman yang gelap ke zaman yang terang benderang bagi akhlak dan kehidupan manusia sehingga menuntun ke jalan kebenaran. Tugas akhir ini saya persembahkan kepada:

1. Kedua orang tua yang sangat saya cintai, Bapak Sunardi dan Ibu Dwi Puji Rahayu yang selalu memberikan doa, semangat dan dukungan baik secara langsung maupun tidak langsung.
2. Adek saya Nuri Fery Prasetyanti yang membantu saya selama pengerjaan Tugas Akhir ini serta semangat selalu yang diberikan kapan pun dan di manapun.
3. Kepada Keluarga besar, Sahabat, dan Teman-teman semua yang selalu berbagi kisah suka maupun duka, semoga ini menjadi sedikit kebanggaan bagi kalian semua.
4. Ibu Sri Mulyati, S.Kom., M.Kom., Selaku Dosen Pembimbing Tugas Akhir saya.
5. Bapak Galang Prihadi Mahardhika, S.Kom., M.Kom., Selaku Dosen Pembimbing Akademik saya.
6. Keluarga besar Setya Bhakti Tegalsari yang sudah menjadi tempat saya mengembangkan diri dan menjadi bermanfaat bagi masyarakat di sekitar tempat tinggal saya.
7. Keluarga besar Strava Boyolali dan teman pehobi sepeda saya yang sudah mengisi kejenuhan disaat penulisan Tugas Akhir ini.

Penulis menyadari bahwa semua ini tidak cukup untuk membalas budi dan kebaikan mereka semua. Semoga dengan prestasi kecil ini dapat membuat kebanggaan tersendiri terutama kepada kedua orang tua tercinta. Aamiin

HALAMAN MOTO

Bekerjalah untuk akhiratmu seolah-olah kamu akan mati esok hari, dan bekerjalah untuk kehidupan duniamu seolah-olah kamu akan hidup selamanya.

(Ali bin Abi Thalib)

“Tidak menggantungkan kebahagiaan kepada orang lain karena pada dasarnya kebahagiaan itu kita sendiri yang membuatnya”

(Hanung Setyo Sambudi)



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobbil'alamin, Puji syukur kami panjatkan ke hadirat Allah SWT, atas berkat rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Sistem Cerdas Klasifikasi Kematangan Dan Harga Buah Pepaya Berdasarkan Ekstraksi Fitur *Gray Level Co-Occurence Matrix* Dengan *Metode Naive Bayes*” dengan baik. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita baginda Nabi agung Muhammad SAW, karena berkat perjuangan beliau kita sekarang berada di zaman yang terang benderang seperti pada saat ini.

Laporan Tugas Akhir ini dibuat sebagai salah satu syarat agar memperoleh gelar sarjana Teknik Informatika Universitas Islam Indonesia serta sebagai sarana penulis untuk mengimplementasikan secara langsung ilmu yang telah didapat selama masa perkuliahan di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Penulis menyadari bahwa selama pelaksanaan Tugas Akhir dan penyusunan laporan ini tidak bisa lepas dari bimbingan, dorongan serta bantuan baik secara langsung maupun tidak langsung dari berbagai pihak yang terlibat, oleh karena itu perkenankanlah penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini dengan baik, semoga Tugas Akhir ini kelak dapat bermanfaat bagi orang lain.
2. Ibu Sri Mulyati, S.Kom., M.Kom., selaku pembimbing yang telah memberi masukan, motivasi, dan bimbingan selama proses pengerjaan Tugas Akhir dan penulisan laporan.
3. Kedua Orang Tua saya Bapak Sunardi dan Ibu Dwi Puji Rahayu atas segala doa dan dukungan selama penulis menempuh masa studi di Jurusan Teknik Informatika Universitas Islam Indonesia dan selama penyusunan Tugas Akhir.
4. Adek saya Nuri Fery Prasetyanti yang membantu saya selama pengerjaan Tugas Akhir ini serta semangat selalu yang diberikan kapan pun dan di manapun.
5. Bapak Prof. Fathul Wahid, S.T. M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
6. Bapak Prof. Dr. Ir. Hari Purnomo, selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Teknik Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.

8. Bapak Galang Prihadi Mahardhika, S.Kom., M.Kom., selaku Dosen Pembimbing Akademik di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Yogyakarta, 17 Juni 2021



(Hanung Setyo Sambudi)



SARI

Kabupaten Boyolali merupakan produsen pepaya terbesar di Jawa Tengah dengan 268.043 kuintal di mana varietas unggulan adalah pepaya jenis California dan pepaya jenis Thailand. Salah satu wilayah di Kabupaten Boyolali yaitu Desa Keposong rata-rata warga bekerja sebagai petani dengan sistem lahan milik pribadi ataupun sewa lahan milik bengkok pemerintah desa. Dengan luas lahan berkisar 1000M² memiliki hasil pepaya pasca panen yang cukup banyak dengan berbagai kondisi pepaya yang ada maka pepaya harus dikelompokkan menurut kondisinya yaitu matang dan tidak matang untuk menghindari adanya kerugian baik untuk pihak petani ataupun pihak pengepul dikarenakan pepaya dengan kondisi belum matang tidak memiliki nilai jual. Selain pengelompokan hasil panen, perkiraan harga jual pepaya pasca panen juga diperlukan oleh petani untuk mengetahui potensi hasil tanamnya dalam kurun waktu tertentu. Sistem cerdas klasifikasi pepaya dapat membantu petani dalam mengelompokkan hasil panen pepaya pasca panen sekaligus mengetahui prediksi harga jual untuk setiap buahnya.

Metode klasifikasi *Naive Bayes* digunakan untuk mengklasifikasikan pepaya sesuai dengan kondisinya ke dalam dua kelas yaitu matang dan belum matang. Ciri pepaya matang dan belum matang diperoleh dari ekstraksi ciri nilai *Gray Level Co-occurrence Matrix* (GLCM) untuk setiap data set citra pepaya. Data pakar diperoleh pada saat tahap pengujian dengan menggunakan 70 data set citra untuk pepaya California dan Thailand dengan kondisi matang dan belum matang yang kemudian disimpan dalam *database* bernama Mdl. Data inilah yang menjadi nilai pembandingan untuk klasifikasi kelas baik untuk tahap pengujian ataupun pada tahap program utama. Prediksi nilai jual diperoleh dari ekstraksi luas pepaya dengan menghitung jumlah piksel yang ada di dalam obyek pepaya dimana untuk mengidentifikasi batas antara obyek dengan latar belakang alas potret menggunakan metode deteksi tepi dengan operator *Prewitt*. Hasil akhir dari sistem cerdas ini adalah hasil klasifikasi kematangan serta prediksi nilai jual yang ditampilkan dalam bentuk tabel pada GUI MATLAB dengan nilai akurasi program dihitung menggunakan metode *Confusion Matrix* pada tahap pelatihan sebesar 82% dan tahap uji sebesar 86% serta memperoleh rata-rata nilai AUC dalam 6 kali percobaan sebesar 0.87 untuk pelatihan dan 0.82 untuk pengujian sehingga metode *Naive Bayes* yang digunakan masuk dalam kategori klasifikasi baik.

Kata kunci: pepaya, klasifikasi, *Naive Bayes*, GLCM, *Prewitt*.

GLOSARIUM

AUC	Metode pengukuran performa klasifikasi Naive Bayes
Confusion Matrix	Metode pengukuran akurasi prediksi sistem cerdas
Background	Latar belakang citra.
Flowchart	Langkah-langkah proses dalam bentuk diagram alur
Grayscale	Citra dengan format warna keabuan.
GUI	Tampilan grafis yang menjadi penghubung antara sistem dengan pengguna.
Input	Nilai masukan.
Naive Bayes	Metode klasifikasi data.
Output	Nilai keluaran dari sistem.
Pixel	Titik dalam sebuah citra.
Prewitt	Operator yang digunakan untuk pemrosesan data citra guna memperoleh batas deteksi tepi dari citra.
Resize	proses untuk mengubah berkas citra menjadi ukuran yang lebih kecil dari ukuran aslinya.
RGB	Citra dengan format berwarna.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I LATAR BELAKANG	17
1.1 Latar Belakang	17
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	8
2.1 Tinjauan Pustaka	8
2.2 Landasan Teori.....	12
2.2.1 Pepaya	12
2.2.2 Jenis Pepaya California dan Pepaya Thailand.....	12
2.2.3 Sistem Cerdas	13
2.2.4 MATLAB	14
2.2.5 <i>Graphics User Interface</i> (GUI)	14
2.2.6 Pengolahan Citra Digital	14
2.2.7 Ruang Warna RGB.....	15
2.2.8 <i>Preprocessing</i>	16
2.2.9 Metode Ekstraksi GLCM	17

2.2.10	Metode Klasifikasi <i>Naive Bayes Classification</i>	24
2.2.11	Segmentasi Deteksi Tepi	25
2.2.12	Ekstraksi Ciri Bentuk (Luas Area)	27
2.2.13	Penentuan Harga Jual Pepaya.....	28
2.2.14	Pengujian Sistem <i>Confusion Matrix</i>	30
BAB III METODOLOGI PENELITIAN.....		34
3.1	Pengumpulan Data	34
3.1.1	Karakteristik Pepaya California	34
3.1.2	Karakteristik Pepaya Thailand	35
3.2	Analisa Kebutuhan Sistem	36
3.2.1	Analisa Kebutuhan Input.....	37
3.2.2	Analisis Kebutuhan Proses.....	37
3.2.3	Analisa Kebutuhan <i>Output</i>	39
3.3	Perancangan Antarmuka Sistem	39
3.4	Perancangan Alur Proses Sistem.....	41
3.5	Metode Pengujian Sistem.....	42
BAB IV HASIL DAN PEMBAHASAN.....		43
4.1	Implementasi Sistem	43
4.1.1	Tahap Pelatihan	44
4.1.2	Tahap Pengujian	61
4.1.3	Program Utama.....	65
4.2	Implementasi Antarmuka	77
4.3	Pengujian Sistem.....	79
4.3.1	Pengujian <i>Confusion Matrix</i>	79
4.3.2	Pengujian Metode Klasifikasi <i>Area Under Curve</i>	83
4.4	Penyebab Kegagalan	83
4.5	Perbandingan Kondisi Riil	85
4.6	Kelebihan dan Kekurangan	88
BAB V KESIMPULAN DAN SARAN		90
5.1	Kesimpulan	90
5.2	Saran.....	90
DAFTAR PUSTAKA.....		92
LAMPIRAN		96

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sebelumnya.....	9
Tabel 2.2 <i>Confusion Matrix</i>	32
Tabel 2.3 Nilai <i>Area Under Curve</i> (AUC).....	32
Tabel 4.1 <i>Offset</i> Matriks Kookurensi.....	50
Tabel 4.2 Hasil Ekstraksi Ciri GLCM	54
Tabel 4.3 Nilai <i>Mean</i> Ekstraksi Ciri GLCM.....	59
Tabel 4.4 Nilai Standar Deviasi	60
Tabel 4.5 Hasil Tahap Pengujian	63
Tabel 4.6 <i>Range</i> Ukuran Pepaya.....	73
Tabel 4.7 Rentang Nilai Luas Obyek Pepaya	73
Tabel 4.8 Hasil Akhir Klasifikasi	75
Tabel 4.9 Hasil Percobaan Satu	81
Tabel 4.10 Hasil Percobaan Dua.....	81
Tabel 4.11 Hasil Percobaan Tiga	81
Tabel 4.12 Hasil Percobaan Empat.....	82
Tabel 4.13 Hasil Percobaan Lima.....	82
Tabel 4.14 Hasil Percobaan Enam Data Utama.....	82
Tabel 4.15 Nilai UAC Percobaan Sistem	83
Tabel 4.16 Hasil Perbandingan Kondisi Riil	87

DAFTAR GAMBAR

Gambar 2.1 Arah <i>Offset</i> Dengan Sudut 0°	18
Gambar 2.2 Arah <i>Offset</i> Dengan Sudut 45°	18
Gambar 2.3 Arah <i>Offset</i> Dengan Sudut 90°	18
Gambar 2.4 Arah <i>Offset</i> Dengan Sudut 135°	19
Gambar 2.5 Sudut arah GLCM dengan sudut 0° , 45° , 90° atau 135° (The MathWorks, 2006).	19
Gambar 2.6 Matriks awal citra asli (a) matriks kookurensi hasil penjumlahan (b) (Kusanti et al., 2018).	20
Gambar 2.7 Nilai Hasil Ekstraksi GLCM.....	23
Gambar 2.8 Penggunaan Nilai <i>Mean</i> Hasil Ekstraksi Fitur GLCM	23
Gambar 2.9 Alat Bantu Alas Potret	26
Gambar 2.10 Perbandingan Ukuran Pepaya	27
Gambar 2.11 Piksel Aktif Obyek Pepaya	28
Gambar 2.12 Perbandingan Ukuran Pepaya Thailand.....	29
Gambar 2.13 Perbandingan Ukuran Pepaya California.....	29
Gambar 2.14 Pengukuran Pepaya Keadaan Nyata Menggunakan Pita Ukur	29
Gambar 2.15 Pengukuran Berat Pepaya Thailand	30
Gambar 3.1 Produktivitas Buah Pepaya California	35
Gambar 3.2 Produktivitas Buah Pepaya Thailand	36
Gambar 3.3 <i>Flowchart</i> Analisis <i>Input</i>	37
Gambar 3.4 Tampilan antarmuka <i>guide</i> pada MATLAB	40
Gambar 3.5 <i>Flowchart</i> Program Utama.....	41
Gambar 4.1 Struktur Sistem.....	43
Gambar 4.2 Struktur Folder	44
Gambar 4.3 Diagram Alur Tahap Pelatihan	45
Gambar 4.4 <i>Flowchart</i> Pemilihan Citra.....	45
Gambar 4.5 Kode Program Pilih Citra.....	46
Gambar 4.6 Kode Program Penentuan <i>Folder</i>	46
Gambar 4.7 Kode Program Inisialisasi Variabel	47
Gambar 4.8 Ukuran Citra Awal	47
Gambar 4.9 Diagram Alur <i>Resize</i> Citra	48
Gambar 4.10 Kode Program <i>Resize</i> Citra.....	48

Gambar 4.11 <i>Flowchart</i> Konversi Citra <i>Grayscale</i>	48
Gambar 4.12 Kode Program Konversi <i>Grayscale</i>	49
Gambar 4.13 Perbedaan Struktur Matriks Citra RGB dan <i>Grayscale</i>	49
Gambar 4.14 <i>Flowchart</i> Pembentukan Matriks Kookurensi.....	49
Gambar 4.15 Arah Matriks Kookurensi.....	50
Gambar 4.16 Kode Program Pembentukan Matriks Kookurensi	50
Gambar 4.17 Hasil Perhitungan <i>Matrix</i> GLCM Data Latih	51
Gambar 4.18 Langkah Perhitungan Matriks Kookurensi (Fatihah, 2016).....	52
Gambar 4.19 <i>Flowchart</i> Ekstraksi Ciri GLCM	53
Gambar 4.20 Kode Program Ekstraksi Ciri GLCM.....	54
Gambar 4.21 Struktur Fungsi <i>fitncb</i> Klasifikasi Naive Bayes.....	57
Gambar 4.22 <i>Flowchart</i> Klasifikasi Naive Bayes.....	57
Gambar 4.23 Kode Program Klasifikasi <i>Naive Bayes</i>	58
Gambar 4.24 Struktur <i>Database</i> Mdl.....	58
Gambar 4.25 <i>Flowchart</i> Tahap Pengujian	61
Gambar 4.26 Struktur Variabel Data Tahap Pengujian	62
Gambar 4.27 Kode Program Pemanggilan Data Pakar.....	62
Gambar 4.28 <i>Flowchart</i> Program Utama.....	65
Gambar 4.29 Kode Program Pilih Citra <i>Single File</i>	66
Gambar 4.30 Kode Program <i>Preprocessing</i> Program Utama.....	66
Gambar 4.31 Kode Program Pembentukan Matriks Kookurensi dan Ekstraksi Ciri GLCM..	67
Gambar 4.32 Kode Program Penyusun Variabel.....	68
Gambar 4.33 Kode Program Tampil Tabel Ekstraksi GLCM.....	68
Gambar 4.34 Kode Program Klasifikasi <i>Naive Bayes</i> Program Utama.....	69
Gambar 4.35 Data Sampel Citra Pepaya.....	70
Gambar 4.36 Kode Program Deteksi Tepi Metode <i>Prewitt</i>	71
Gambar 4.37 Representasi Luas Obyek Hasil Operasi Fungsi <i>bwareaopen</i> (a) Representasi Obyek Hasil Deteksi Tepi <i>Prewitt</i> (b)	72
Gambar 4.38 Kode Program Ekstraksi Luas dan Prediksi Harga.....	75
Gambar 4.39 Rancangan Antar Muka Dengan <i>Guide</i>	78
Gambar 4.40 Tampilan <i>User Interface</i> Sistem	78
Gambar 4.41 Model <i>Confusion Matrix</i>	79
Gambar 4.42 Kode Program Pengujian <i>Confusion Matrix</i>	80
Gambar 4.43 Citra Gagal Klasifikasi.....	84

Gambar 4.44 Kegagalan Metode Deteksi Tepi.....	84
Gambar 4.45 Obyek Kertas HVS; (a) Citra RGB (b) Citra Hasil Deteksi Tepi <i>Prewitt</i>	85
Gambar 4.46 Pengukuran Riil Obyek Pepaya	86
Gambar 4.47 Pemodelan Ukuran Riil Pepaya	86
Gambar 4.48 Hasil Pengukuran Luas Pepaya Pada Sistem	87
Gambar 4.49 Pengukuran Luas Pepaya Dalam Obyek Kertas HVS	87
Gambar 4.50 Pengukuran Berat Dengan Timbangan Duduk	88



BAB I

LATAR BELAKANG

1.1 Latar Belakang

Perkembangan digitalisasi industri 4.0 menuju industri 5.0 mendorong semua pemenuhan kebutuhan teknologi terhadap berbagai sektor kehidupan tak terkecuali pada sektor pertanian pada khususnya sektor pertanian pepaya. Penggunaan teknologi digunakan dari mulai riset pembibitan sampai dengan masa panen serta peningkatan FVC (*Food Value Chain*) yang berkaitan dengan alur produk pangan pertanian untuk pemenuhan kebutuhan pepaya yang efektif guna peningkatan mutu dan keuntungan di sektor pangan pertanian khususnya pada sektor pertanian pepaya. Implementasi teknologi dalam sektor pertanian berupa sistem monitor yang dapat memberi informasi seputar pertanian dalam hal berbagai parameter pertanian yang ada sekaligus dapat memberikan saran dalam proses pertanian (Al Faruqi, 2019). Sektor pertanian pepaya dituntut untuk menghasilkan hasil panen buah pepaya dengan kualitas bagus baik dalam segi visual ataupun rasa pepaya, pepaya yang siap petik memiliki ciri citra terdapat warna kuning atau oranye pada badan buah pepaya. Pemetikan buah pada saat panen akan menentukan kualitas jual buah pepaya itu sendiri, masa pemetikan pepaya yang baik adalah 128 hari setelah pembungaan pepaya akan memiliki umur simpan yang optimal selama 6 hari masa simpan (Arifiya et al., 2015). Teknologi pertanian akan membantu mengoptimalkan sumber daya yang ada dengan mengetahui parameter-parameter kualitas buah pepaya, sistem dapat mengolah kemudian memberikan saran kepada petani buah pepaya sehingga dapat diperoleh buah pepaya dengan kualitas matang yang baik.

Kabupaten Boyolali merupakan penghasil komoditas buah pepaya di Provinsi Jawa Tengah, terdapat varietas pepaya terdapat 366.561 pohon pepaya dengan hasil produksi 268.043 kuintal per tahunnya yang berada pada wilayah Kecamatan Teras, Boyolali, Musuk, Mojosongo dan Ampel (Sapti, 2019). Desa Keposong adalah Desa yang berada di Kecamatan Tamansari Kabupaten Boyolali hasil pemekaran wilayah dari Kecamatan Musuk selain warga masyarakat beternak sapi perah ataupun sapi pedaging juga mengandalkan pepaya sebagai komoditas buah untuk meningkatkan pendapatan sehari-hari. Rata-rata warga memiliki lahan dengan luas 1000M² baik yang dimiliki oleh pribadi ataupun sewa lahan. Produksi pepaya di Kabupaten Boyolali pada tahun 2019 dapat dilihat pada Tabel 1.1.

Tabel 1.1 Produksi pepaya di Kabupaten Boyolali Tahun 2019 dalam kuintal (Badan Pusat Statistik Kabupaten Boyolali, 2019).

No	Kecamatan	Jumlah Produksi Pepaya (Kuintal)
1	Ampel	3160
2	Andong	177
3	Banyudono	120
4	Boyolali	77700
5	Cepogo	-
6	Juwangi	4148
7	Karanggede	650
8	Kemusu	-
9	Klego	-
10	Mojosongo	92555
11	Musuk	5827
12	Ngemplak	400
13	Nogosari	-
14	Sambi	-
15	Sawit	38
16	Selo	-
17	Simo	108
18	Teras	29558
19	Wonosegoro	-

Kabupaten Boyolali merupakan daerah produsen pepaya paling banyak di Provinsi Jawa Tengah dimana pada tahun 2015 varietas pepaya yang ada di Kabupaten Boyolali terdapat 366.561 pohon, sedangkan wilayah tetangga yaitu Kabupaten Wonogiri terdapat pohon pepaya sebanyak 156.157 pohon, Kabupaten Kebumen 119.278 pohon dan Kabupaten Karanganyar 113.998 pohon (Sapti, 2019). Terdapat dua jenis varietas tanaman pepaya yang menjadi andalan warga di Desa Keponsong yaitu pepaya Bangkok atau lebih dikenal dengan

pepaya Thailand dan pepaya California dengan daya tahan tanaman yang cukup baik di Desa Keposong.

Efektivitas pasca panen pepaya baik varietas jenis California ataupun Thailand dibutuhkan untuk memaksimalkan harga jual dengan pemilihan jenis pepaya yang baik antara pepaya matang dengan pepaya belum matang sehingga tidak terjadi kesalahan penjualan. Selama ini petani di Desa Keposong saat melakukan pemilahan buah pepaya pasca panen dengan mengidentifikasi tingkat kematangan dengan analisa secara visual sederhana menggunakan indra penglihatan mata. Analisa sederhana ini memiliki beberapa kelemahan seperti tingkat kelelahan manusia dan perbedaan persepsi tingkat kematangan antara tiap orang akan berbeda-beda karena batas antara stadium kematangan buah sulit diketahui jika hanya dengan menggunakan mata telanjang yang mengakibatkan penentuan kematangan buah menjadi bersifat subjektif (Pantastico & Kamariyani, 1989). Tingkat kematangan menjadi penentu penjualan pepaya karena jika tidak matang pepaya tidak laku untuk dijual, nilai jual pepaya ditentukan dengan ukuran besarnya pepaya dengan ukuran kecil, sedang dan besar. Selain menggunakan metode visual dengan melihat tingkat kematangan buah melalui warna juga dapat melalui uji laboratorium untuk mengetahui kandungan Total Padat Terlarutnya (TPT). Namun hal ini selain membutuhkan biaya yang lebih mahal buah yang diperiksa akan menjadi rusak sehingga tidak memiliki nilai jual lagi (Manik & Saragih, 2017).

Beberapa penelitian sebelumnya mengenai teknik pengolahan citra untuk mengetahui tingkat kematangan buah yang menjadi rujukan pada penelitian ini. Metode klasifikasi *Naive Bayes* dan KNN memiliki performa yang cukup dalam mengklasifikasikan daun herbal, dimana metode *Naive Bayes* memiliki performa sebesar 75% sedangkan metode KNN memiliki performa sebesar 70.83% (Liantoni & Nugroho, 2015). Penelitian untuk mengetahui luas sebuah bidang menggunakan MATLAB dan teknik deteksi tepi dengan membandingkan hasil hitungan manual, dari lima pengambilan diperoleh tingkat akurasi pengukuran lebih dari 80% hal ini menunjukkan jika teknik deteksi tepi memiliki perhitungan yang baik (Utami et al., 2015). Metode *Naive Bayes Classifier* akan digunakan dalam penelitian ini mengingat metode ini adalah metode *Bayesian Learning* dengan proses yang cepat serta sederhana dengan tingkat keakuratan yang cukup baik untuk pengklasifikasian kematangan pepaya dan deteksi citra tepi untuk mengetahui besar pepaya guna penentuan harga pepaya hal ini berdasarkan penelitian-penelitian yang sudah dilakukan sebelumnya dengan metode tersebut mendapatkan tingkat akurasi yang cukup baik.

Penggunaan teknologi untuk bidang pertanian terutama pertanian pepaya di Desa Keposong untuk klasifikasi buah pepaya kedalam dua jenis kelas klasifikasi, yaitu yang belum matang dan sudah matang berupa fitur ekstraksi *Gray Level Cooccurrence Matrix* (GLCM) dengan menggunakan fitur intensitas, energi, entropi, standar deviasi, *skewness* dan *smoothness* (Hakiky et al., 2021). Sedangkan untuk klasifikasi harga akan ditentukan berdasarkan besarnya pepaya dengan kategori harga kecil, sedang dan besar menggunakan hasil deteksi tepi untuk mendapatkan luas dari citra pepaya dan menentukan harga sesuai dengan kategorinya. Data yang digunakan merupakan data primer buah pepaya berupa *image* buah pepaya yang diperoleh dari hasil pemotretan buah pepaya sebanyak 100 buah pepaya. Dalam penelitian ini penulis akan menggunakan jarak 1 *pixel* dengan derajat 0, 45, 90 dan 135 pada empat fitur ekstraksi di GLCM seperti *Contrast*, *Correlation*, *Energy* dan *Homogeneity*.

Berdasarkan latar belakang sebelumnya yang telah penulis uraikan penelitian dapat dilakukan suatu penelitian menggunakan sistem cerdas untuk bidang pertanian khususnya pertanian buah pepaya dengan judul “*Sistem Cerdas Klasifikasi Kematangan Dan Harga Buah Pepaya Berdasarkan Ekstraksi Fitur Gray Level Co-occurrence Matrix Dengan Metode Naive Bayes*”.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana mengklasifikasikan buah pepaya jenis Thailand dan California yang sudah matang secara konsisten serta memperoleh perkiraan harga jual dengan memanfaatkan sistem cerdas?

1.3 Batasan Masalah

Penelitian yang baik adalah penelitian yang fokus dan terarah sehingga dapat menghasilkan hasil penelitian yang optimal maka beberapa batasan berikut perlu diperhatikan:

- a. Sistem cerdas ini hanya digunakan untuk klasifikasi buah pepaya Thailand dan California sesuai dengan varietas yang banyak ditanam oleh petani di Desa Keposong.
- b. Sistem cerdas ini hanya untuk identifikasi tingkat kematangan buah pepaya bukan sebagai identifikasi tingkat kemanisan buah pepaya.
- c. Lokasi penelitian berada di Desa Keposong Kecamatan Tamansari Kabupaten Boyolali.
- d. Penelitian ini akan dibangun dengan *software* MATLAB.
- e. Data untuk tahap latih dan data untuk tahap uji yang akan digunakan sebanyak 100 *image* buah pepaya Thailand dan pepaya California

- f. Fitur ekstraksi GLCM yang akan digunakan yaitu: *Contrast*, *Correlation*, *Energy* dan *Homogeneity*.
- g. Klasifikasi kematangan pepaya akan dibagi menjadi pepaya matang dan pepaya belum matang.
- h. Metode deteksi tepi menggunakan operator *Prewitt*.
- i. Penentuan harga jual pepaya pada periode Februari 2021 dengan rincian untuk pepaya Thailand ukuran kecil Rp.2000, sedang Rp.3000 dan besar Rp.5000 dan untuk pepaya California sebesar kecil Rp.2000, sedang Rp.2500 dan besar Rp.3000.

1.4 Tujuan Penelitian

Mengetahui tingkat keakuratan metode klasifikasi *Naive Bayes* dan ekstraksi *Gray Level Co-occurrence Matrix* (GLCM) dalam menentukan tingkat kematangan serta penggunaan metode deteksi tepi operator *Prewitt* untuk memperoleh prediksi harga jual buah pepaya varietas Thailand dan pepaya California.

1.5 Manfaat Penelitian

Kontribusi dari penelitian Tugas Akhir ini adalah:

- a. Mengetahui tingkat efektivitas metode *Naive Bayes* untuk identifikasi kematangan buah pepaya.
- b. Memudahkan klasifikasi buah pepaya ke dalam dua kelas buah dengan kondisi sudah matang dan belum matang.
- c. Meningkatkan konsistensi serta mengurangi salah klasifikasi kematangan buah pepaya dikarenakan kelelahan manusia atau faktor humanis lainnya.
- d. Buah pepaya dapat dikelompokkan sesuai dengan kelas kematangannya pada kondisi riil dengan menggunakan bantuan sistem cerdas yang dibangun.
- e. Mengetahui perkiraan harga jual pepaya jenis varietas Thailand dan California.

1.6 Metodologi Penelitian

Pembuatan Tugas Akhir ini menggunakan tahapan metodologi penelitian sebagai berikut:

- a. Studi Pustaka

Studi pustaka dilakukan dengan mencari sumber-sumber pustaka seperti buku, jurnal ilmiah, bertemu pakar dan literatur lain yang berhubungan dengan perancangan sistem

cerdas dengan ekstraksi GLCM dan metode klasifikasi *naive bayes* yang dibutuhkan dalam penelitian ini.

b. Pengumpulan Data

Untuk memperoleh data uji dan data latih yang diperlukan dalam pengembangan sistem cerdas pengumpulan data dibutuhkan sebagai bahan dasar penelitian. Observasi langsung di lapangan juga dilakukan untuk mengetahui perbedaan buah yang matang dan belum matang berdasarkan pengetahuan petani di dukung dengan referensi yang sudah diperoleh sebelumnya di studi pustaka. Penelitian ini menggunakan 100 citra data yang dibagi menjadi dua kelompok data, yaitu data pelatihan sebanyak 70 citra pepaya dan data pengujian sebanyak 30 citra.

c. Pengembangan Sistem Cerdas

Pengembangan sistem cerdas dibantu dengan menggunakan *software* MATLAB dengan ekstraksi citra pepaya menggunakan metode GLCM setelah didapatkan hasil ekstraksi setiap citra baik data uji ataupun citra data latih selanjutnya akan diklasifikasikan dengan menggunakan metode klasifikasi *Naive Bayes*. Untuk perhitungan harga jual akan ditentukan berdasarkan besarnya pepaya dengan menggunakan deteksi tepi baik untuk pepaya Thailand dan California.

d. Implementasi Sistem Cerdas

Implementasi sistem cerdas dilakukan sesuai hasil pengembangan berupa klasifikasi buah pepaya yang sudah matang dan belum matang. Pengujian hasil klasifikasi buah akan menggunakan tampilan *Graphics User Interface* (GUI) yang dari *software* MATLAB.

e. Evaluasi Hasil Akhir

Evaluasi dan analisis hasil akhir dilakukan untuk mengetahui tingkat keakuratan sistem cerdas yang dikembangkan dalam mengelompokkan buah pepaya yang sudah matang dan belum matang. Tingkat keakuratan sistem ditentukan pada tahap pengujian dengan menggunakan metode *Confusion Matrix* sehingga akan diketahui seberapa tingkat akurasi sistem cerdas yang dikembangkan.

1.7 Sistematika Penulisan

Dalam penulisan laporan Tugas Akhir sistematika penulisan membantu peneliti dalam penyusunan dokumen yang sesuai dengan kaidah penelitian dan penelitian menjadi lebih terarah. Adapun sistematika laporan penulisan Tugas Akhir adalah sebagai berikut:

BAB 1 Pendahuluan

Bab Pendahuluan membahas mengenai awal penulisan laporan penelitian. Bagian pendahuluan memuat Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

BAB 2 Tinjauan Pustaka dan Landasan Teori

Bagian ini memuat berbagai dasar kerangka pemikiran yang digunakan dalam penelitian yang didapatkan dari studi pustaka yang untuk memperoleh solusi dari permasalahan yang ada. Bagian ini berisi mengenai penelitian yang pernah dilakukan sebelumnya yang berkaitan dengan pengenalan citra khususnya yang menggunakan metode klasifikasi *Naive Bayes*, deteksi citra tepi dan ekstraksi GLCM, sedangkan Landasan Teori membahas mengenai berbagai metode yang digunakan.

BAB 3 Metodologi

Bagian metodologi membahas analisis teori yang diterapkan dalam penelitian serta bagaimana mengimplementasikannya ke dalam sistem cerdas yang dikembangkan. Materi yang digunakan dalam penelitian ini akan dijelaskan sesuai dengan alur pengembangan yang dilakukan, serta penguraian simulasi perancangan sistem cerdas.

BAB 4 Hasil dan Pembahasan

Bagian ini membahas secara rinci penelitian yang dilaksanakan oleh penulis. Hasil penelitian akan disajikan dalam bentuk gambaran alur sistem, gambar kode program, tabel, foto ataupun bentuk lainnya yang diuraikan sesuai dengan tahapan penelitian. Sedangkan untuk hasil yang bersifat teoritis akan dibandingkan dengan hasil penelitian terdahulu yang membahas hal sejenis mengenai ekstraksi GLCM dan klasifikasi *Naive Bayes*.

BAB 5 Kesimpulan dan Saran

Kesimpulan dan saran merupakan bagian terakhir dari penulisan laporan yang membahas tentang kesimpulan hasil penelitian yang telah dilakukan dan berisi saran serta wawasan untuk penelitian selanjutnya yang berkaitan dengan topik bahasan pada penelitian ini.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang akan dilakukan memerlukan acuan dari beberapa penelitian sebelumnya untuk mengetahui penelitian terdahulu yang sudah pernah dilakukan memiliki hubungan dengan penelitian yang akan dilakukan, selain untuk menghindari kesamaan topik penelitian serta membimbing peneliti menjadi lebih terarah dan memperbaiki hasil penelitian sebelumnya yang masih belum baik.

Beberapa penelitian sebelumnya yang berkaitan dengan pengenalan pola ataupun pengenalan tekstur sudah pernah dilakukan dengan hasil yang cukup bervariasi mengikuti metode yang digunakan. Penggunaan metode klasifikasi *Naive Bayes* pada kematangan jambu biji merah dengan jenis warna RGB menggunakan data uji dan data pelatihan sebanyak 30 citra buah memperoleh nilai akurasi sebesar 75% dengan hanya menggunakan satu metode *RGB* saja (Arifin, 2016).

Penelitian yang dilakukan oleh (Haba & Pelangi, 2020) dengan judul “Pengelompokan Buah Jeruk menggunakan *Naive Bayes* dan *Gray Level Co-occurrence Matrix*” yang mengkaji tentang klasifikasi kematangan jeruk keprok dengan data latih sebanyak 110 dan data uji sebanyak 20. Penggunaan GLCM digunakan untuk mendapatkan hasil ekstraksi dari citra jeruk yang kemudian diklasifikasikan memakai metode *Naive Bayes*. Fitur yang akan dipakai dalam ekstraksi GLCM yaitu *Contrast*, *Correlation*, *Energi* dan *Homogeneity*. Hasil klasifikasi yang didapatkan berupa jeruk matang dan jeruk belum matang dengan akurasi yang didapatkan sebanyak 82%.

Penelitian kematangan buah pisang kapok dengan metode *Naive Bayes* juga diteliti oleh (Rachmadi et al., n.d.). Penelitian ini menggunakan metode ekstraksi fitur warna HSV serta algoritma *Naive Bayes*. Metode ekstraksi fitur warna yang digunakan untuk identifikasi kematangan buah dengan memanfaatkan ekstraksi fitur warna (R, G, dan B) ke HSV (*Hue Saturation Value*). Dari pengujian yang dilakukan terbukti mampu melakukan klasifikasi citra pada 15 data uji pisang kapok memperoleh hasil akurasi sebesar 73.33% dimana 11 buah pisang kapok dapat diprediksi sesuai dengan tingkat kematangannya.

Penelitian yang juga membahas tentang kematangan pepaya dilakukan oleh (Sugiyanto & Wibowo, 2015) untuk mengklasifikasikan buah California menggunakan ruang warna hsv

menggunakan algoritma *k-nearest neighbors*. Pengujian dengan data citra uji sebanyak 12 data set, diperoleh tingkat akurasi algoritma KNN sebesar 75%, dengan data salah prediksi sebanyak 3 data dan yang sesuai target prediksi sejumlah 9 data. Pengujian dengan jumlah data citra uji sebanyak 12 citra data, didapatkan tingkat akurasi algoritma KNN sebesar 83,34%, dengan data salah prediksi berjumlah 2 data dan yang sesuai target sejumlah 10 data.

Penelitian mengenai pengolahan citra dapat dilakukan untuk berbagai objek untuk mengetahui klasifikasi yang ditentukan pada sebuah objek tersebut layak untuk digunakan atau tidak. Tidak hanya terbatas pada objek pepaya atau buah saja pengolahan citra juga dapat digunakan untuk median lain seperti batik untuk mengetahui jenis batik berdasarkan pola batik maupun untuk pengolahan citra pada median lainnya. Hal ini dapat disimpulkan bahwa pengolahan citra sangat berguna untuk kehidupan sehari-hari.

Berdasarkan referensi penelitian di atas langkah serta metode yang berkaitan dengan klasifikasi *Naive Bayes* akan digunakan peneliti sebagai acuan dalam penelitian ini. Fokus penelitian adalah bagaimana metode ekstraksi fitur GLCM serta metode *Naive Bayes Classification* dapat efektif mengenali kematangan buah pepaya California dan pepaya Thailand. Perbandingan dari setiap penelitian-penelitian yang sudah pernah dilakukan sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

No	Penulis	Metode	Dataset	Kesimpulan
1	(Arifin, 2016)	<i>Naive Bayes</i>	30 citra data jambu biji merah	Dataset dibedakan menjadi data pelatihan dan data pengujian. Dengan proses pengambilan data kemudian dicari nilai RGB (Red Green Blue) didapatkan nilai akurasi sebesar 75%.
2	(Manuel et al., 2020)	<i>Naive Bayes</i>	114 citra bunga	Evaluasi dengan metode <i>Confusion Matrix</i>

			Anggrek dengan 18 data <i>testing</i> dan 96 data <i>training</i> .	memperoleh akurasi 61,1%.
3	(Rachmadi et al., n.d.)	<i>Naive Bayes</i>	50 citra buah pisang kepok.	Pengujian 15 buah pisang kepok mendapat nilai akurasi 73.33% dimana 11 data dapat diprediksi dengan benar.
4	(Haba & Pelangi, 2020)	<i>Naive Bayes</i>	130 citra buah jeruk	Hasil klasifikasi yang didapatkan berupa jeruk matang dan jeruk belum matang dengan akurasi yang didapatkan sebanyak 82%.
5	(Sugiyanto & Wibowo, 2015)	<i>Naive Bayes</i>	31 data citra pepaya <i>California</i> (<i>Callina</i>)	Dengan citra uji sebanyak 12 citra didapat tingkat keakuratan algoritma KNN sebesar 75%.
6	(Nurhikmat, 2018)	<i>Naive Bayes</i>	Citra wayang golek dengan data latih sebanyak 240 dan data uji sebanyak 60	Mendapat tingkat akurasi pelatihan dan pengujian sebesar 95 % dan 90 %.
7	(Huda, 2016)	Deteksi tepi <i>Prewitt</i>	Obyek telur	Citra deteksi tepi dengan operator <i>Prewitt</i> dapat mengetahui volume telur dimana urutan <i>ringing error</i> terkecil sampai

				terbesar dengan urutan metode deteksi tepi operator Prewitt, Sobel, Canny dan Robert
8	(Indriyani et al., 2017)	Deteksi tepi <i>Prewitt</i> , <i>Canny</i> , <i>Robert</i> dan <i>Sobel</i>	30 citra sampel buah jeruk	Penentuan harga jual jeruk didasarkan pada ukuran dan kualitas, jeruk dengan ukuran besar akan berharga lebih mahal dari pada jeruk dengan ukuran kecil atau sedang. Metode deteksi tepi digunakan untuk mengetahui diameter jeruk yang akan menentukan ukuran dan harga jeruk.

Dari penelitian terdahulu dapat ditarik kesimpulan di antaranya :

- a. Metode *Naive Bayes Classification* dapat digunakan untuk proses klasifikasi berbagai macam kelas yang dibutuhkan dengan tingkat akurasi yang cukup baik dengan hasil terendah sebesar 73% pada penelitian Pisang Kepok serta hasil tertinggi sebesar 95% pada penelitian citra Wayang Golek.
- b. Deteksi tepi dapat digunakan untuk mengetahui ukuran sebuah obyek baik volume, diameter ataupun luas obyek dengan tingkat akurasi model deteksi tepi dari yang tertinggi hingga terendah yaitu metode tepi *Prewitt*, *Sobel*, *Canny* dan *Robert*

Dengan kesimpulan yang didapat, penelitian ini akan membuat program yang dapat mengklasifikasikan tingkat kematangan pepaya untuk varietas pepaya Thailand dan pepaya California serta dapat menentukan harga jual pepaya. Fitur yang digunakan berupa metode klasifikasi *Naive Bayes* sebagai penentuan tingkat kematangan buah pepaya yang diklasifikasikan dalam dua kelas yaitu matang dan belum matang, selain itu untuk penentuan harga pepaya akan menggunakan metode deteksi tepi *Prewitt* dengan menggunakan metode

perhitungan luas melihat hasil akurasi penelitian sebelumnya dengan metode luas mendapatkan akurasi sebesar 80% selain itu akurasi metode deteksi tepi *Prewitt* juga memiliki akurasi yang paling tinggi (Huda, 2016). Metode perbandingan luas dapat digunakan sebagai penentuan harga pepaya dengan klasifikasi besaran obyek pepaya berupa kecil, sedang dan besar dengan acuan harga semakin besar pepaya maka harga jual pepaya juga semakin mahal.

2.2 Landasan Teori

2.2.1 Pepaya

Meksiko selatan merupakan asal tanaman pepaya (*Carica Papaya L*), buah wilayah tropis dengan habitat daerah dengan kondisi kering, daerah dataran rendah, basah serta pegunungan dengan ketinggian hingga 1000 meter di atas permukaan air laut dengan beragam bentuk dan jenis mulai dari yang lonjong, bulat dan silindris. Ukuran pepaya pun beraneka ragam mulai dari yang kecil, sedang dan besar dengan daging berwarna merah, kuning hingga oranye serta kulit buah pepaya berwarna hijau muda, hijau tua dan kuning. Keanekaragaman jenis pepaya merupakan bahan dasar untuk membuat varietas pepaya unggul (Sujiprihati & Suketi, n.d.).

Dalam bahasa Indonesia nama pepaya merupakan serapan dari bahasa Belanda yaitu *papaja*, selain itu juga mengadopsi dari bahasa Arawak yaitu *papaya*. Pohon pepaya pada umumnya tidak tumbuh bercabang dengan tinggi pohon lima hingga sepuluh meter dimana pada batang pohon bagian atas terdapat daun dengan bentuk spiral. pepaya adalah varietas *monodieciuous* yaitu berumah tunggal dan ganda serta memiliki tiga kelamin yaitu jantan, betina, dan tumbuhan banci (hermafrodit). Memiliki bentuk buah dengan ujung runcing bulat memanjang, warna buah saat belum matang memiliki tekstur hijau gelap kemudian saat matang berubah menjadi hijau muda hingga kuning (Bonaditya, 2014).

Desa Keposong memiliki kapasitas produksi pepaya pada tahun 2019 sebesar 57 kwintal (BPS Boyolali, 2019). Dengan dominasi tanaman pepaya jenis varietas California dan Thailand, setiap petani di Desa Keposong menanam pepaya pada musim akhir musim kemarau dengan membuat lubang-lubang tanam berjarak 1 meter antar lubang. Kedua jenis pepaya ini dipilih karena dalam mendapatkan bibit relatif mudah serta metode tanamnya yang mudah.

2.2.2 Jenis Pepaya California dan Pepaya Thailand

Pepaya California

Pepaya California Callina (IPB-9) dapat tumbuh pada ketinggian rendah hingga menengah yaitu 100 hingga 500 meter di atas permukaan laut. Pepaya ini dikembangkan oleh

Prof. Dr. Sriani Sujiprihati sebagai pemulia dan Tim dari Pusat Kajian Buah-buahan Tropika IPB. Pepaya California memiliki bobot sekitar 1 Kg per buahnya serta memiliki bentuk silindris. Ciri buah pepaya Callina diantaranya memiliki daging berwarna jingga dengan rasa manis dan kulit buah dengan warna hijau lumut dan menjadi kuning oranye saat buah sudah matang (Pusat Kajian Hortikultura Tropika, 2015). Pepaya jenis ini merupakan pepaya unggulan dengan umur genjah sehingga dapat dipanen dalam waktu singkat saat berumur tujuh bulan, selain itu jika dibanding dengan jenis pepaya lainnya pepaya ini memiliki batang yang lebih pendek (Isnawan, 2015). Petani di Desa Keposong sendiri dalam melakukan penanam pepaya California berdampingan dengan jenis pepaya Thailand sehingga dalam sekali panen akan terdapat dua jenis pepaya yang didapat.

Pepaya Thailand

Masih dalam famili *Caricaceae* sehingga pepaya Thailand ini memiliki tekstur buah yang mirip dengan pepaya California. Sesuai dengan namanya pepaya yang memiliki nama lain pepaya Bangkok ini berasal dari Thailand. Pepaya ini masuk ke Indonesia pada tahun 70-an dengan bentuk yang lebih besar dari California. Pepaya ini dipilih oleh petani karena bobotnya yang berat dan memiliki ukuran besar yang beratnya bisa mencapai 3,5 Kg serta batang pohon yang lebih besar dari pepaya jenis lainnya (Utami Putri, 2016).

Selain memiliki bentuk yang lebih besar dibandingkan pepaya jenis lainnya pepaya Thailand juga memiliki rasa dan ketahanan buah yang baik sehingga akan memudahkan dalam proses penjualan dari petani sampai ke konsumen. Dengan warna daging jingga dengan rasa manis segar menjadi pilihan alternatif selain pepaya California serta dapat ditemukan di pasar modern maupun pasar tradisional.

Petani di Desa Keposong memilih pepaya jenis Thailand karena memiliki kecenderungan harga jual yang stabil dengan berat buah yang besar. Proses budidaya pepaya Thailand pun tidak rumit dengan banyaknya bibit yang dijual ataupun mengembangkan bibit sendiri dengan mengambil benih dari pohon yang sehat dengan memilih pepaya yang sudah dewasa atau matang.

2.2.3 Sistem Cerdas

Sistem cerdas disebut juga sistem pakar merupakan sebuah program yang dapat memberikan saran yang terkomputerisasi yang bertujuan meniru proses pengambilan keputusan dan pengetahuan dari pakar dalam menyelesaikan suatu masalah yang spesifik

(Subakti, 2018). *Knowledge base, working memory dan inference engine* merupakan tiga komponen utama struktur dalam sistem cerdas. Penyimpanan pengetahuan dari setiap domain sistem yang dibangun ada di dalam *knowledge base*, jadi setiap sistem akan memiliki *knowledge base* yang berbeda sesuai dengan bidang kepakaran sistem yang dibangun kemudian *database* menjadi tempat penyimpanan fakta-fakta yang ditemukan selama proses pencarian dengan pakar yang digunakan untuk padanan tentang fakta-fakta yang ada dengan informasi yang terdapat pada *database knowledge base*, peranan pengambil keputusan dari masalah yang diberikan kepada system ditentukan oleh *inference engine* (Irawan, 2007).

2.2.4 MATLAB

MATLAB merupakan aplikasi perangkat lunak dengan bahasa pemrograman mekanisme tingkat tinggi (*high performance*) dalam bidang komputasi teknis penggabungan komputasi, visualisasi, dan pemrograman dengan pengoperasian yang mudah digunakan untuk penyelesaian masalah dengan menggunakan solusi notasi matematik (Wijaya & Prijono, 2007). Elemen dasar dalam MATLAB adalah *array* yang memudahkan penyelesaian masalah komputasi yang berkaitan dengan matriks dan vektor. Nama MATLAB diperoleh dari kependekan *Matrix Laboratory* dengan tujuan awal pembuatan proyek perangkat lunak LINPACK dan EISPACK agar mudah dikembangkan dengan berbasis pada matriks. Sistem pada aplikasi MATLAB mempunyai lima bagian utama yaitu bahasa pemrograman MATLAB, lingkungan kerja MATLAB, penanganan grafik, pustaka fungsi matematis MATLAB (*library*), dan *Application Program Interface* (API) MATLAB (Wijaya & Prijono, 2007).

2.2.5 Graphics User Interface (GUI)

Graphics User Interface merupakan sebuah antarmuka yang memudahkan dalam penggunaan sebuah program ataupun aplikasi lainnya sehingga dapat dengan mudah dioperasikan oleh pengguna awam non-teknis.

2.2.6 Pengolahan Citra Digital

Mengolah sebuah citra menggunakan komputer secara khusus untuk memperoleh citra dengan hasil yang lain adalah prinsip dasar pengolahan citra digital (Fadlisyah, 2007). Citra digital memiliki bentuk umum matriks dua dimensi dengan posisi koordinat x dan y yang mewakili nilai piksel dalam citra. Matriks berukuran N merupakan baris piksel dan M adalah

kolom piksel dalam ruang dua dimensi (Madenda, 2015). Beberapa jenis citra yang ada di antaranya:

Citra Berwarna

Citra berwarna atau juga dapat disebut *true color image* adalah citra dengan informasi warna yang dikandung pada setiap pikselnya dengan representasi komponen *chrominance/saturation*, *luminance* dan *hue* (Madenda, 2015). Citra berwarna ini biasa ditemukan dalam kehidupan keseharian seperti saat menggunakan kamera digital, melihat galeri di gawai ataupun foto berwarna yang dipajang di ruang keluarga. Secara matematis citra warna digambarkan menggunakan fungsi matriks $f(n, m, k)$ tiga dimensi dimana N merupakan banyaknya baris, M adalah jumlah kolom sedangkan K atau dimensi berisi komponen warna merah, hijau dan biru (RGB). Fungsi inilah yang kemudian akan digunakan untuk analisis citra digital berwarna.

Citra Gray-level

Citra skala keabuan atau *gray-level* adalah citra yang hanya memiliki 8 bit kode untuk nilai *luminance* dengan tingkat keabuan yang variatif dimulai dari nilai 0 hingga 255 di mana 0 menggambarkan warna hitam dan 255 menggambarkan warna putih. Citra skala keabuan diperoleh dari citra warna dengan transformasi warna RGB ke warna lain dengan representasi citra tingkat keabuan Y, V atau L untuk ruang warna tersebut.

Citra Biner

Citra biner hanya memiliki dua level keabuan yang bagian dari citra *gray-level*. Nilai 0 akan membentuk warna hitam sedangkan nilai 1 akan membentuk warna putih dimana citra biner memiliki batas yang akan menentukan pembentukan citra. Piksel dengan nilai yang lebih besar dari batas akan diubah menjadi 1 atau warna putih sedangkan piksel yang lebih kecil dari batas akan diubah menjadi 0.

2.2.7 Ruang Warna RGB

Dalam citra RGB terdapat ruang warna yaitu nilai dari angka pembentuk warna tertentu sebagai bentuk perwujudannya yang di modelkan dengan model matematis (Swedia & Cahyanti, 2010). Ruang warna bertujuan untuk memodelkan, melakukan perhitungan dan

visualisasi informasi warna sehingga komputer dapat mengolah informasi warna yang ada serta dapat membedakan warna seperti yang dilakukan manusia dalam membedakan antar warna.

Sinyal analog menghasilkan nilai akuisisi frekuensi warna dengan sensor elektronik ruang warna RGB (*Red, Green, Blue*) dalam ruang warna standar yang kemudian dilakukan proses digitalisasi intensitas amplitudonya dan dikodekan dalam 8 bit untuk setiap warnanya dapat terbentuk 16.777.216 warna lainnya menggunakan tiga warna dasar ini (Madenda, 2015). Bentuk citra warna RGB adalah yang paling banyak digunakan dalam pemrosesan dan penyimpanan data citra digital. Normalisasi RGB direpresentasikan dengan normalisasi sederhana pada persamaan 2.1, 2.2 dan 2.3.

$$r = \frac{R}{R + G + B} \quad (2.1)$$

$$g = \frac{G}{R + G + B} \quad (2.2)$$

$$b = \frac{B}{R + G + B} \quad (2.3)$$

Keterangan :

R : Intensitas warna merah

G : Intensitas warna hijau

B : Intensitas warna biru

2.2.8 Preprocessing

Preprocessing merupakan tahap awal dilakukannya pengecilan ukuran berkas citra agar dalam proses komputasi tidak membutuhkan waktu yang lama yang kemudian diproses pada tahapan selanjutnya (Rohpandi et al., 2015). Beberapa cara dapat dilakukan untuk melakukan *preprocessing* antara lain meningkatkan kontras, menghilangkan *noise*, *resizing image* dan *grayscale image* dengan tujuan membuat citra asli menjadi lebih baik kualitasnya, hal ini akan meningkatkan prosentase keberhasilan pada tahap berikutnya akan meningkat. Pada penelitian ini tahap *preprocessing* dilakukan dengan mengubah warna citra dari berwarna *red green blue* menjadi citra keabuan (*grayscale*) serta melakukan *resize image* menjadi 10% dari ukuran *file* semula yang akan digunakan pada proses selanjutnya.

2.2.9 Metode Ekstraksi GLCM

Pengertian GLCM

Gray Level Co-occurrence Matrix merupakan matriks dengan frekuensi munculnya pasangan dua piksel dalam jarak tertentu, intensitas tertentu serta arah tertentu dalam citra (Tamarena, 2020). Robert Haralick mengembangkan sejumlah metode analisis tekstur salah satunya adalah GLCM suatu metode analisis tekstur yang paling sering digunakan (Madenda, 2015). Selain itu dalam klasifikasi fitur citra beberapa peneliti menyetujui bahwa GLCM memiliki kemampuan terbaik untuk melakukan ekstraksi fitur dari pada menggunakan metode lain (Jusman et al., 2017).

GLCM dapat dihitung dengan arah dari piksel ketetangaan dengan arah sudut 0 derajat, 40 derajat, 90 derajat dan 135 derajat dengan menghitung nilai rata-rata setiap fitur yang digunakan. Terdapat 14 fitur dalam GLCM namun dalam penelitian ini akan menggunakan 4 fitur yang biasa digunakan. Hal ini didasarkan karena keempat fitur ini merupakan fitur utama GLCM sedang sisanya adalah turunan dari empat fitur ini. Keempat fitur yang akan digunakan dalam penelitian ini di antaranya adalah *contrast*, *correlation*, *energi* dan *homogenity*.

Proses ekstraksi GLCM

a. Pembentukan matriks kookurensi

Setiap citra pepaya mempunyai nilai matriks yang berbeda sesuai tingkat intensitas warna yang diperoleh setelah proses segmentasi, nilai ini akan membentuk sebuah nilai matriks awal GLCM dengan arah 0° , 45° , 90° atau 135° menggunakan 2 piksel yang berjajar. Nilai yang berjajar pada matriks dua piksel ini dapat dilakukan operasi perhitungan berikutnya berupa proses normalisasi. Sudut yang digunakan akan menentukan nilai matriks dengan dimensi 8×8 yang terbentuk.

- Sudut 0° : Kookurensi munculnya kejadian ketetangaan antar piksel dengan nilai yang sama pada jarak 1 piksel dengan sudut 0° sehingga perhitungan akan dilakukan ke arah kanan secara horizontal. Arah perhitungan nilai matrik kookurensi dapat dilihat pada Gambar 2.1.

1	6	1
8	6	8
3	4	5

Gambar 2.1 Arah *Offset* Dengan Sudut 0°

- Sudut 45° : Kookurensi munculnya kejadian ketetangaan antar piksel dengan nilai yang sama pada jarak 1 piksel dengan sudut 45° sehingga perhitungan akan dilakukan ke arah menjorok kanan atas. Arah perhitungan nilai matrik kookurensi dapat dilihat pada Gambar 2.2.

1	6	1
8	6	8
3	4	5

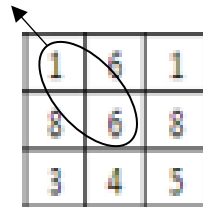
Gambar 2.2 Arah *Offset* Dengan Sudut 45°

- Sudut 90° : Kookurensi munculnya kejadian ketetangaan antar piksel dengan nilai yang sama pada jarak 1 piksel dengan sudut 90° sehingga perhitungan akan dilakukan secara vertikal. Arah perhitungan nilai matrik kookurensi dapat dilihat pada Gambar 2.3.

1	6	1
8	6	8
3	4	5

Gambar 2.3 Arah *Offset* Dengan Sudut 90°

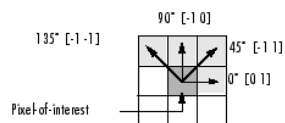
- Sudut 135° : Kookurensi munculnya kejadian ketetangaan antar piksel dengan nilai yang sama pada jarak 1 piksel dengan sudut 135° sehingga perhitungan akan dilakukan menjorok ke arah kiri atas. Arah perhitungan nilai matrik kookurensi dapat dilihat pada Gambar 2.4.

Gambar 2.4 Arah *Offset* Dengan Sudut 135°

Dalam implementasinya di aplikasi MATLAB arah derajat akan ditentukan dengan bentuk *array offset* [0 1; -1 1; -1 0; -1 -1] dengan urutan derajat 0° , 45° , 90° dan 135° . MATLAB menyediakan *library graycomatrix* yang bisa dipakai untuk pembentukan matriks GLCM. Arah sudut dan *array offset* pada MATLAB dapat dilihat pada Gambar 2.5.

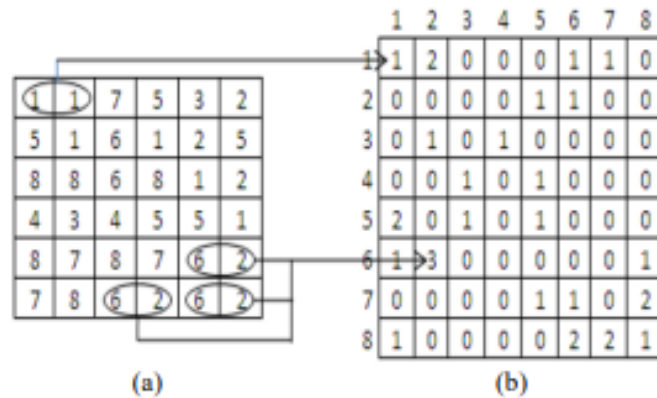
Angle	Offset
0	[0 D]
45	[-D D]
90	[-D 0]
135	[-D -D]

The figure illustrates the array: *offset* = [0 1; -1 1; -1 0; -1 -1]



Gambar 2.5 Sudut arah GLCM dengan sudut 0° , 45° , 90° atau 135° (The MathWorks, 2006).

- b. Pembentukan matriks simetris menggunakan penjumlahan matriks awal dalam GLCM dengan nilai transposnya. Langkah pembentukan nilai matriks kookurensi untuk sudut 0° dapat dilihat pada Gambar 2.6.



Gambar 2.6 Matriks awal citra asli (a) matriks kookurensi hasil penjumlahan (b)
(Kusanti et al., 2018).

c. Proses normalisasi GLCM

Normalisasi dilakukan dengan membagi setiap elemen matriks yang diperoleh pada proses sebelumnya dengan jumlah pasangan piksel.

d. Ekstraksi Fitur

Hubungan antar ketetanggaan piksel dalam citra *grayscale* dapat diidentifikasi dengan menggunakan 14 fitur ekstraksi GLCM antara lain: *Angular Second Moment*, *Contrast*, *Correlation*, *Sum of Squares*, *Variance*, *Inverse Difference Moment*, *Sum Average*, *Sum Variance*, *Sum Entropy*, *Entropy*, *Difference Variance*, *Difference Entropy*, *Information Measures of Correlation type* dan *Maximal Correlation Coefficient* (Haralick et al., 1973). Namun Connors dan Harlow mengkaji bahwa dari 14 fitur yang diusulkan Haralick tersebut hanya 5 fitur yang sering digunakan untuk proses image processing yaitu: *Energy*, *Entropy*, *Correlation*, *Local Homogeneity* dan *Inertia* (Connors & Harlow, 1980). Kemudian dalam penelitian ini untuk memperoleh ekstraksi fitur hasil GLCM besaran yang digunakan yaitu *Angular Second Moment* (ASM) dengan fitur yang diimplementasikan dalam penelitian ini yaitu fitur *Contrast*, *Correlation*, *Energy* dan *Homogeneity* (Auliasari & Kertaningtyas, 2018). Dengan persamaan sebagai berikut:

1. *Contrast*

Tinggi rendahnya piksel yang dihasilkan dengan nilai 0 jika nilai ketetanggaan piksel memiliki nilai yang sama. Persamaan nilai *Contrast* dapat dilihat pada persamaan 2.4.

$$Contrast = \sum_{i,j} |i - j|^2 P(i,j) \quad (2.4)$$

Semakin tinggi nilai *Contrast* menunjukkan semakin tingginya nilai kekontrasannya dimana $P(i,j)$ adalah elemen baris ke- i dan elemen kolom ke- j dari matriks kookurensi (Parker, 2013).

2. *Correlation*

Correlation bertujuan untuk mengukur linearitas dari sejumlah pasangan piksel antar nilai tingkat keabuan atau tingkatan besar hubungan antar piksel ketetanggaan dalam suatu citra (Parker, 2013). . Persamaan nilai *Correlation* dapat dilihat pada persamaan 2.5.

$$Correlation = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)P(i,j)}{\sigma_i \sigma_j} \quad (2.5)$$

Keterangan :

$P(i,j)$: menunjukkan elemen baris ke- i dan kolom ke- j dari nilai matriks kookurensi

μ_i : nilai rata-rata baris ke- i

μ_j : nilai rata-rata kolom ke- j

σ_i : nilai standar deviasi baris ke- i

σ_j : nilai standar deviasi kolom ke- j

3. *Energy*

Fitur *energy* berfungsi menunjukkan konsentrasi intensitas antar pasangan piksel, sebuah nilai energi akan tinggi jika tekstur citra yang diperoleh seragam dalam penyebaran derajat keabuannya. Persamaan nilai *Energy* dapat dilihat pada persamaan 2.6.

$$Energy = \sum_{i,j} P(i,j)^2 \quad (2.6)$$

Dengan $P(i,j)$ adalah elemen baris ke- i dan elemen kolom ke- j nilai matriks kookurensi (Parker, 2013).

4. *Homogeneity*

Fitur *homogeneity* berfungsi untuk menunjukkan sifat homogenitas atau keseragaman dengan derajat keabuan sejenis dimana citra yang memiliki sifat homogenitas akan mempunyai nilai tinggi dari sifat homogenitasnya. . Persamaan nilai *Homogeneity* dapat dilihat pada persamaan 2.7.

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1 + |i - j|} \quad (2.7)$$

Dengan $P(i, j)$ merupakan elemen baris ke- i , kolom ke- j dari matriks kookurensi (Parker, 2013).

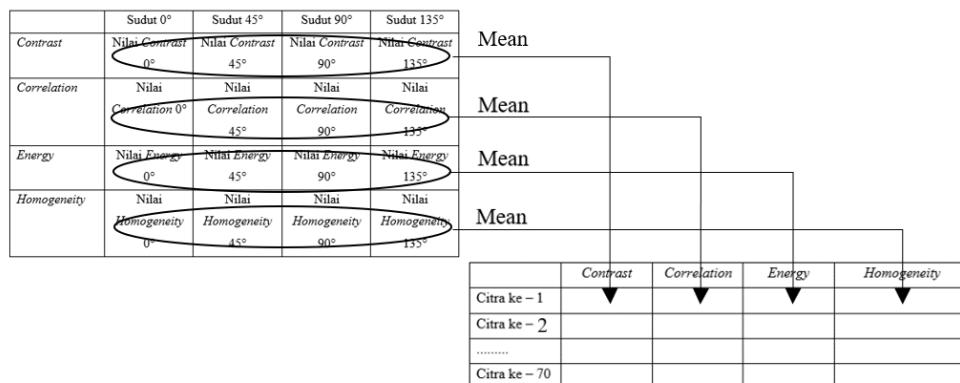
e. Hasil Ekstraksi Fitur GLCM

Penggunaan data citra yang baik akan memudahkan identifikasi ciri citra untuk menyusun *database* citra sebagai nilai rujukan dalam tahap klasifikasi *Naive Bayes*. Terdapat tiga fitur citra yang umum digunakan untuk identifikasi ciri citra yaitu: warna, bentuk obyek, dan tekstur (Ahmad, 2005). Dalam penelitian ini identifikasi ciri tekstur digunakan untuk mengenali ciri citra pepaya sudah matang maupun pepaya belum matang yang dicatat menggunakan matriks GLCM (*Gray Level Co-occurrence Matrix*). Matriks GLCM berisi data kookurensi tingkat keabuan piksel untuk ketetangaan pada arah 0° , 45° , 90° , dan 135° . Pada penelitian ini terdapat enam belas fitur yang dihasilkan pada tahap ekstraksi fitur GLCM yaitu: fitur *Contrast* arah 0° , fitur *Correlation* arah 0° , fitur *Energy* arah 0° , fitur *Homogeneity* arah 0° , fitur *Contrast* arah 45° , fitur *Correlation* arah 45° , fitur *Energy* arah 45° , fitur *Homogeneity* arah 45° , fitur *Contrast* arah 90° , fitur *Correlation* arah 90° , fitur *Energy* arah 90° , fitur *Homogeneity* arah 90° , fitur *Contrast* arah 135° , fitur *Correlation* arah 135° , fitur *Energy* arah 135° dan fitur *Homogeneity* arah 135° . Representasi nilai hasil ekstraksi fitur GLCM dapat dilihat pada Gambar 2.7.

	Sudut 0°	Sudut 45°	Sudut 90°	Sudut 135°
<i>Contrast</i>	Nilai <i>Contrast</i> 0°	Nilai <i>Contrast</i> 45°	Nilai <i>Contrast</i> 90°	Nilai <i>Contrast</i> 135°
<i>Correlation</i>	Nilai <i>Correlation</i> 0°	Nilai <i>Correlation</i> 45°	Nilai <i>Correlation</i> 90°	Nilai <i>Correlation</i> 135°
<i>Energy</i>	Nilai <i>Energy</i> 0°	Nilai <i>Energy</i> 45°	Nilai <i>Energy</i> 90°	Nilai <i>Energy</i> 135°
<i>Homogeneity</i>	Nilai <i>Homogeneity</i> 0°	Nilai <i>Homogeneity</i> 45°	Nilai <i>Homogeneity</i> 90°	Nilai <i>Homogeneity</i> 135°

Gambar 2.7 Nilai Hasil Ekstraksi GLCM

Nilai yang sudah diperoleh pada tahap ekstraksi fitur GLCM kemudian dilanjutkan dengan pencarian nilai *Mean* untuk setiap fitur yang digunakan. Nilai *Mean* inilah yang digunakan untuk data rujukan citra pada klasifikasi *Naive Bayes*. Setiap data citra yang ada akan dilakukan identifikasi nilai fitur GLCM menggunakan fitur *Contrast*, *Correlation*, *Energy* dan *Homogeneity*. Representasi penggunaan nilai *Mean* dapat dilihat pada Gambar 2.8.

Gambar 2.8 Penggunaan Nilai *Mean* Hasil Ekstraksi Fitur GLCM

Dari 16 nilai fitur awal yang diperoleh dari perhitungan fitur untuk setiap sudutnya akan digunakan 4 fitur hasil perhitungan *Mean* sebagai nilai pakar klasifikasi *Naive Bayes*. Nilai *Mean* inilah yang digunakan untuk identifikasi tekstur dari data citra pepaya

matang dan pepaya belum matang. Setiap data citra akan memiliki nilai *Mean* untuk setiap fitur ekstraksi GLCM yang berbeda-beda sesuai dengan kondisi kematangan pepaya. Nilai *Mean* nantinya akan disimpan dalam variabel *ciri_matang* untuk data citra pepaya matang dan variabel *ciri_belum_matang* untuk data citra pepaya belum matang.

2.2.10 Metode Klasifikasi *Naive Bayes Classification*

Naive Bayes Classification adalah metode klasifikasi dengan dasar teorema Bayes yaitu metode yang digunakan untuk memprediksi peluang berdasarkan pengalaman yang ada dengan memanfaatkan metode probabilitas serta statistik ditemukan oleh Thomas Bayes yang berasal dari Inggris (Syukri Mustafa et al., 2017). *Naive Bayes Classification* memiliki hasil akurasi yang lebih baik dibandingkan metode klasifikasi lain dengan proses komputasi yang cepat untuk diaplikasikan dalam *database* yang besar. Teorema Bayes ini memiliki persamaan 2.8 sebagai berikut:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (2.8)$$

Keterangan:

X : data kelas yang belum diketahui nilainya.

H : hipotesis data kelas X yang spesifik.

P(H|X) : nilai probabilitas hipotesis H terhadap kondisi X

P(H) : nilai probabilitas hipotesis H

P(X|H) : nilai probabilitas X terhadap H

P(X) : nilai probabilitas dari X

Naive Bayes Classification menggunakan dua kategori yaitu himpunan latih dan himpunan pengujian. Himpunan latih berguna untuk data pada tahap pelatihan yang diidentifikasi sebagai obyek yang ingin diklasifikasi, dalam penelitian ini memakai 70 data latih dengan rincian 35 data citra untuk pepaya matang dan 35 citra pepaya belum matang. Sedangkan himpunan uji merupakan data yang digunakan sebagai parameter ketepatan dari klasifikasi yang dihasilkan, dalam penelitian ini menggunakan 30 citra data uji dengan rincian 15 citra pepaya matang dan 15 citra pepaya belum matang. Penggunaan data citra yang baik

diperlukan untuk memperoleh hasil penelitian yang akurat dan efektif, citra yang digunakan sebisa mungkin memiliki resolusi yang baik dengan pencahayaan yang mencukupi sehingga akan memudahkan komputer dalam melakukan identifikasi citra yang ada guna mempercepat proses komputasi data.

2.2.11 Segmentasi Deteksi Tepi

Dengan mengetahui perbedaan intensitas warna antar piksel ketetanggannya maka dapat diketahui batas deteksi tepi suatu obyek, dimana nilai intensitas yang tinggi akan semakin terlihat jelas tepi suatu obyek citra (Madenda, 2015). Tepi sebuah citra dapat digunakan sebagai identifikasi batas antara obyek utama dengan obyek yang lain pada sebuah citra seperti *background* citra ataupun obyek lainnya yang tidak dibutuhkan. Dengan mengetahui batas tepi sebuah obyek maka dapat diketahui pula bentuk ataupun besar obyek sebuah citra. Beberapa operator deteksi tepi yang dapat digunakan di antaranya gradien citra, gradien Roberts, operator Sobel, Prewitt, Kirsch, Laplace hingga filter FIR dan IIR seperti filter Gaussian, Laplacian of Gaussian (LoG), Shen-Castan, Canny-Deriche dan Madenda (Madenda, 2015). Secara matematis perbedaan intensitas pada jarak spasial tertentu dikenal sebagai turunan pertama sebuah fungsi yang dapat dikembangkan menjadi sebuah fungsi pendeteksian tepi.

Dalam penelitian ini akan menggunakan operator *Prewitt* untuk mengetahui batas antara obyek pepaya dengan *background* yang ada di belakang obyek. Untuk mendapatkan citra yang konsisten maka dibuat sebuah alat bantu berupa alas potret yang digunakan untuk pengambilan data citra. Untuk membantu pengambilan gambar yang optimal digunakan alat bantu berupa alas potret untuk mengambil citra pepaya. Gambar 2.9 merupakan alas potret yang digunakan dalam pengambilan data citra.



Gambar 2.9 Alat Bantu Alas Potret

Pengambilan citra yang konsisten tingkat ketinggiannya diperlukan untuk mendapatkan perbandingan citra yang bagus antara buah pepaya yang berukuran kecil, sedang ataupun besar. Dengan citra yang konsisten diharapkan didapatkan ukuran citra yang bagus dengan memanfaatkan operator deteksi tepi *Prewitt*.

Metode *Prewitt* dikembangkan oleh Judith MS Prewitt hasil dari pengembangan metode sobel dengan satu angka nol sebagai penyangga dengan memakai prinsip dari fungsi *Gaussian* dan *Laplacian* yang berguna untuk membangkitkan filter *High Pass Filter* (Pranata & Astuti, 2017). Kelebihan dari metode *Prewitt* mempunyai akurasi yang cukup bagus terbukti dengan penelitian yang dilakukan oleh (Huda, 2016) mengetahui ukuran volume telur dengan hasil deteksi tepi terbaik sampai dengan terkecil dengan urutan *Prewitt*, *Sobel*, *Canny* dan *Robert*. Selain itu metode *Prewitt* juga dapat meminimalisir bias sebelum perhitungan deteksi tepi sehingga hasil deteksi yang didapatkan menjadi optimal. Deteksi tepi dengan operator *Prewitt* ini menggunakan dua *kernel* dengan dimensi 3x3 dengan rumus seperti pada persamaan 2.9.

$$\begin{matrix}
 -1 & 0 & 1 \\
 -1 & 0 & 1 \\
 -1 & 0 & 1
 \end{matrix}
 P_x =
 \begin{matrix}
 -1 & -1 & -1 \\
 0 & 0 & 0 \\
 1 & 1 & 1
 \end{matrix}
 P_y =
 \tag{2.9}$$

Untuk mengetahui akurasi pengukuran luas area obyek hasil deteksi tepi *Prewitt* dilakukan pengukuran manual dengan perbandingan obyek pepaya dengan kertas HVS

berukuran 215 x 330 mm untuk memudahkan ukuran perbandingan di kondisi nyata. Perhitungan luas pepaya dilakukan dengan menggunakan alat bantu ukur berupa pita ukur dengan satuan sentimeter. Setelah diketahui ukuran pepaya pada kondisi nyata dalam satuan sentimeter akan dikonversi ke dalam piksel untuk membandingkan dengan hasil yang ada di dalam program. Perbandingan citra yang dimaksud terdapat pada Gambar 2.10 berikut.



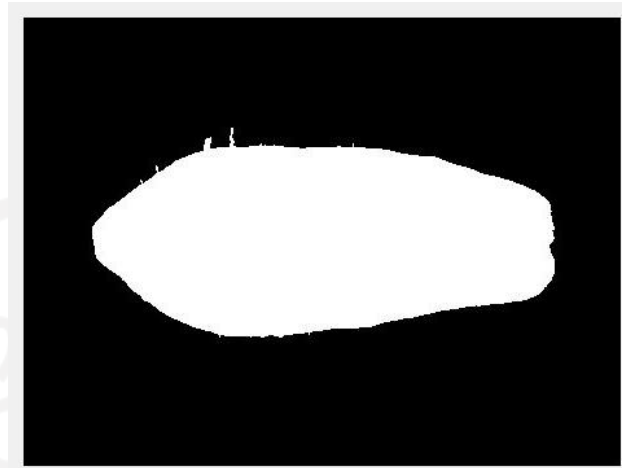
Gambar 2.10 Perbandingan Ukuran Pepaya

2.2.12 Ekstraksi Ciri Bentuk (Luas Area)

Ciri bentuk digunakan untuk membedakan obyek citra yang secara geometri berbeda seperti obyek citra apel yang berbentuk menyerupai lingkaran dan obyek mentimun yang menyerupai persegi dengan ciri bentuk yang biasa digunakan seperti luas area obyek, perimeter, faktor kekompakan (*compactness*) dan faktor kebundaran (*roundness*) (Andono et al., 2017). Setiap piksel yang ada dalam sebuah citra dapat digunakan untuk menghitung keliling ataupun luas sebuah obyek dengan memperhatikan arah pergerakan dari perbatasan luar yang saling menyambung membentuk sebuah batas yang disebut kode rantai.

Ekstraksi ciri luas digunakan untuk mengenali besarnya ukuran setiap citra pepaya yang diidentifikasi sehingga setiap citra yang ada dapat dibedakan menurut jenis pepayanya serta dapat diketahui pula harga jual pepaya yang didapat dari luas tiap jenis pepaya yang ada. Untuk membedakan ukuran suatu obyek dalam citra dalam penelitian ini menggunakan parameter luas melihat ukuran luas citra yang ada antara pepaya varietas jenis California dan Thailand memiliki ukuran yang berbeda secara signifikan sehingga dapat digunakan sebagai acuan pengenalan citra. Luas dalam citra merupakan banyaknya jumlah piksel yang menyusun sebuah obyek. Ekstraksi ciri luas dapat diketahui dengan menjumlah piksel aktif putih didalam yang

ada di dalam obyek pepaya setelah diketahui batas tepinya. Pikel aktif yang dimaksud dapat dilihat pada Gambar 2.11.



Gambar 2.11 Pikel Aktif Obyek Pepaya

2.2.13 Penentuan Harga Jual Pepaya

Harga jual pepaya ditentukan oleh kualitas dan ukuran pepaya dengan harga yang berbanding lurus pada keduanya. Petani dan penjual di Desa Keposong Kecamatan Tamansari Kabupaten Boyolali melakukan transaksi jual beli pepaya dengan berpedoman pada ukuran buah dengan buah pepaya yang sudah mulai matang dengan ciri adanya warna kuning atau oranye pada badan buah pepaya. Untuk pepaya yang belum matang tidak memiliki nilai jual, pepaya ini akan dipilah kemudian digunakan untuk kebutuhan pakan sapi pribadi milik petani setempat. Pepaya yang matang kemudian akan di sortir menurut ukuran pepaya agar memudahkan saat terjadinya proses transaksi. Pada penelitian ini penentuan harga jual pepaya yang digunakan adalah pada periode Februari 2021 dengan rincian untuk pepaya Thailand ukuran kecil berharga Rp.2000, sedang Rp.3000 dan besar Rp.5000 sedangkan untuk pepaya California dengan harga kecil Rp.2000, sedang Rp.2500 dan besar Rp.3000. Untuk mengetahui perbandingan pepaya yang dimaksud dapat terdapat pada Gambar 2.12 serta Gambar 2.13 berikut.

Ukuran perbandingan jenis varietas Pepaya Thailand



Pepaya Thailand Kecil

Pepaya Thailand Sedang

Pepaya Thailand Besar

Gambar 2.12 Perbandingan Ukuran Pepaya Thailand

Ukuran perbandingan jenis varietas Pepaya California



Pepaya California Kecil

Pepaya California Sedang

Pepaya California Besar

Gambar 2.13 Perbandingan Ukuran Pepaya California

Implementasi dalam program menggunakan perbandingan luas yang dapat diketahui dengan deteksi tepi dengan metode *Prewitt*. Obyek citra pepaya akan dipisahkan antara obyek dan latar belakang dengan memanfaatkan deteksi tepi sehingga obyek putih yang ada di belakang pepaya akan dihilangkan atau tidak dihitung, kemudian akan dihitung piksel yang ada dalam obyek pepaya hasil segmentasi untuk diperoleh nilai luasan dari setiap obyek citra pepaya. Nilai luas ini yang nantinya akan digunakan sebagai parameter penentuan ukuran pepaya sehingga dapat ditentukan harga yang sesuai dengan kondisi pepaya di keadaan nyata. Selain perhitungan yang dilakukan oleh komputer juga dilakukan perhitungan secara manual untuk membandingkan hasil perhitungan luas dengan melakukan pengukuran dimensi pepaya menggunakan pita ukur seperti yang ada di Gambar 2.14 berikut.



Gambar 2.14 Pengukuran Pepaya Keadaan Nyata Menggunakan Pita Ukur

Untuk mengetahui harga pepaya dalam satuan kilogram dilakukan perbandingan nilai dengan menggunakan alat bantu timbangan. Dalam perbandingan ini harga pepaya ditentukan dengan persamaan berat pepaya x harga pepaya yang akan ditaksir ke dalam ukuran kecil, sedang dan besar untuk dibandingkan dengan hasil perhitungan harga menggunakan metode luas hasil deteksi tepi operator *Prewitt*. Dengan melakukan perbandingan akan diketahui perkiraan berat pepaya dari hasil klasifikasi sehingga hasil akhir berupa harga diharapkan menjadi lebih akurat. Misalnya, hasil dalam program menunjukkan hasil klasifikasi pepaya Thailand matang dengan ukuran sedang dengan luas 32000 piksel maka berharga Rp.3000 akan dibandingkan dengan satuan kilogram dengan persamaan berat pepaya dikalikan dengan harga pepaya di mana harga pepaya untuk satuan berat mengacu harga di bulan Maret 2021 dengan harga untuk pepaya California Rp.1500/Kg dan Rp.1000/Kg untuk pepaya Thailand. Pengukuran berat pepaya dilakukan dengan timbangan seperti yang ada di Gambar 2.15 berikut.



Gambar 2.15 Pengukuran Berat Pepaya Thailand

2.2.14 Pengujian Sistem *Confusion Matrix*

Pengujian dilakukan untuk mengetahui akurasi sebuah program yang dikembangkan dapat memprediksi sesuai dengan nilai rujukan pakar yang seharusnya. *Confusion Matrix Single Decision Threshold* digunakan untuk mengukur performa hasil klasifikasi yang tersusun dalam sebuah tabel hasil data latih dan uji yang telah diprediksi oleh sistem apakah sesuai dengan nilai yang sebenarnya yaitu pepaya matang dan pepaya belum matang. Dalam metode *Single Decision Threshold* terdapat beberapa istilah yang digunakan diantaranya:

- a. *True Positive* (TP) adalah kondisi di mana sistem dan nilai riil menghasilkan nilai prediksi positif. Jika nilai riil pakar “matang”, maka sistem juga memprediksi nilai “matang”.
- b. *True Negative* (TN) adalah kondisi sistem dan nilai riil rujukan pakar menghasilkan hasil yang negatif. Jika nilai riil pakar “belum matang”, maka sistem juga memprediksi nilai “belum matang”.
- c. *False Positive* (FP) adalah kondisi di mana nilai pakar memiliki hasil negatif namun sistem memprediksi dengan nilai positif. Jika nilai sebenarnya “belum matang” tetapi sistem memprediksi “matang”
- d. *False Negative* (FN) adalah kondisi di mana nilai pakar memiliki hasil positif namun sistem memprediksi dengan nilai negatif. Jika nilai riil rujukan pakar “matang” tetapi sistem memprediksi “belum matang”

Parameter yang digunakan untuk pengujian sistem dengan menggunakan indeks pengukuran berupa sensitivitas, spesifisitas dan akurasi (Muchlis et al., 2018). *Sensitivity* atau sensitivitas berguna untuk mengetahui persentase nilai positif yang diprediksi sudah sesuai (nilai riil dan program menghasilkan nilai matang yang sama). *Specificity* atau spesifisitas berguna untuk memperoleh persentase data negatif yang diprediksi dengan benar (sistem mendeteksi nilai belum matang). Untuk dapat mengetahui performa prediksi klasifikasi data yang tepat oleh sistem dapat menggunakan pengukuran nilai akurasi dengan menggunakan data dengan prediksi benar dibagi dengan keseluruhan data citra yang ada. Untuk perhitungan sensitivitas, spesifisitas dan akurasi terdapat pada persamaan 2.10, 2.11 dan 2.12 berikut.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \cdot 100\% \quad (2.10)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \cdot 100\% \quad (2.11)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\% \quad (2.12)$$

Nilai prediksi adalah keputusan yang diambil oleh system berupa matang dan belum matang, sedangkan nilai riil merupakan nilai yang sesuai kejadian yang ada (nilai pakar). Representasi *Confusion Matrix* dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix*

		Nilai Sebenarnya	
		<i>True</i>	<i>False</i>
Prediksi Nilai	<i>True</i>	TP	FP
	<i>False</i>	FN	TN

Untuk mengetahui kinerja klasifikasi program digunakan alat ukur nilai *Area Under Curve* (AUC). Nilai ini diperlukan untuk menentukan apakah program klasifikasi kematangan dan harga pepaya ini dapat diaplikasikan, semakin baik keakuratan program akan lebih mudah pula untuk diaplikasikan. Nilai dari AUC berkisar antara 0.0 dan 1.0, semakin mendekati nilai 1 menunjukkan kinerja klasifikasi yang digunakan bernilai bagus sebaliknya jika dekat dengan 0 maka kinerja metode klasifikasi yang digunakan kurang baik. Menurut (Gorunescu, 2011) klasifikasi nilai AUC dikategorikan pada Tabel 2.3 berikut.

Tabel 2.3 Nilai *Area Under Curve* (AUC)

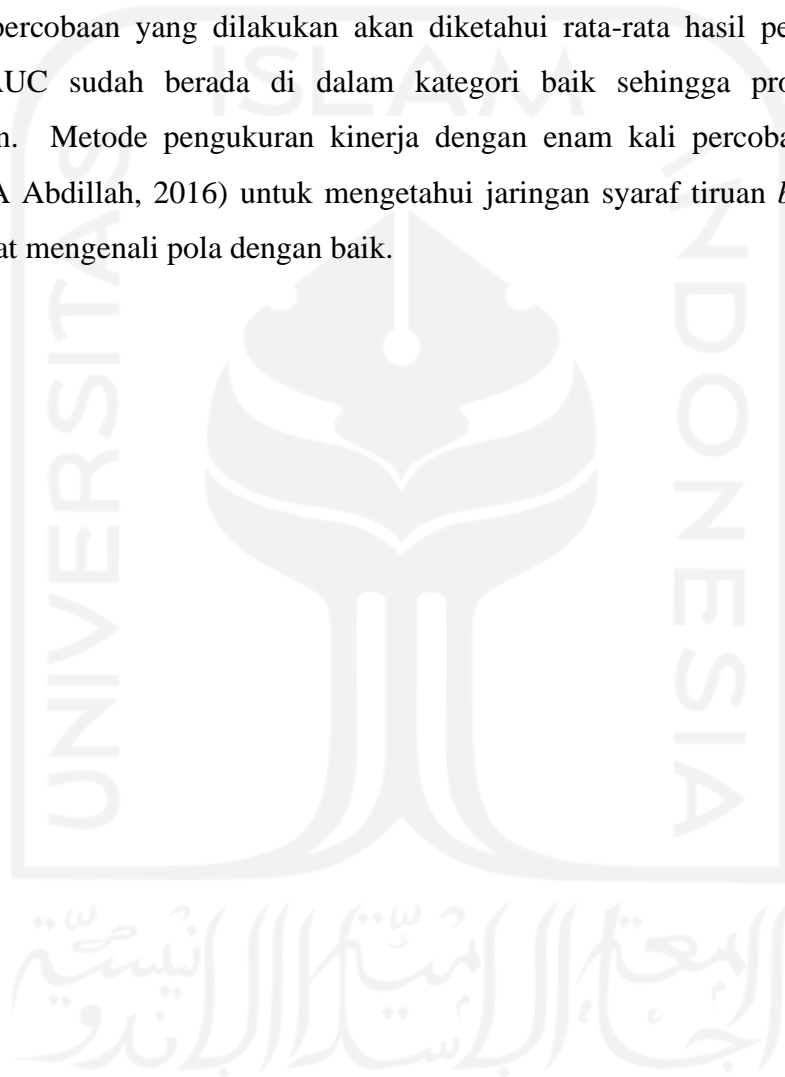
Nilai AUC	Keterangan
0,91 – 1,00	Klasifikasi sangat baik
0,81 – 0,90	Klasifikasi baik
0,71 – 0,80	Klasifikasi Cukup
0,61 – 0,70	Klasifikasi Buruk
$\leq 0,60$	Klasifikasi Salah

Pengukuran kinerja dilakukan dengan melakukan lima kali percobaan dengan *dataset* citra yang berbeda untuk mendapatkan nilai *Accuracy*, *Sensitivity* dan *Specificity*. Untuk perhitungan nilai *Area Under Curve* (AUC) terdapat pada persamaan 2.13 dan *False Positive Rate* (FPR) pada persamaan 2.14.

$$AUC = \frac{1 + Sensitivity - FPR}{2} \quad (2.13)$$

$$FPR = 1 - Specificity \quad (2.14)$$

Keenam percobaan yang dilakukan akan diketahui rata-rata hasil perhitungan AUC apakah nilai AUC sudah berada di dalam kategori baik sehingga program dapat di implementasikan. Metode pengukuran kinerja dengan enam kali percobaan ini merujuk (Suwarno & AA Abdillah, 2016) untuk mengetahui jaringan syaraf tiruan *backpropagation* akurat serta dapat mengenali pola dengan baik.



BAB III

METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Data primer diperoleh dari observasi langsung di lahan pertanian milik petani pepaya di Desa Keposong serta observasi lanjutan di pengepul atau distributor pepaya di Desa Keposong. Observasi langsung di lahan pertanian dibutuhkan untuk mengetahui kondisi pepaya di pohon pada saat belum matang dan sudah matang, sedangkan untuk observasi lanjutan pada distributor pepaya dilakukan untuk mendapat citra pepaya serta informasi harga jual pepaya baik untuk varietas pepaya California dan varietas pepaya Thailand. Data yang dikumpulkan berupa citra pepaya California dan pepaya Thailand dalam kondisi sudah matang dan belum matang. Pengambilan data menggunakan alat bantu berupa alas potret dengan warna latar belakang putih untuk membedakan antara obyek pepaya dengan latar di belakangnya. Data yang dipakai sebanyak 70 citra sebagai data latih yang berisi citra pepaya California dan Thailand. Untuk data pengujian yang dipakai berjumlah 30 citra yang berisi pepaya California dan Thailand guna menentukan kelayakan sistem yang dikembangkan. Selain buahnya yang dapat dijual dan memiliki banyak manfaat kesehatan, pohon pepaya dapat dimanfaatkan seluruh bagian pohonnya untuk kebutuhan pakan ternak. Rata-rata petani di Desa Keposong setelah pohon sudah melewati masa produktifnya yaitu 3 hingga 4 tahun maka akan digantikan pohon baru dan pohon lama yang ditebang dapat dijual ke peternak lain atau digunakan untuk pakan tenaknya sendiri.

3.1.1 Karakteristik Pepaya California

Karakteristik pepaya varietas California menurut (Pusat Kajian Hortikultura Tropika, 2015) antara lain: memiliki umur berbunga 114 hari setelah tanam, memiliki warna kulit hijau terang dengan daging buah berwarna jingga kemerahan, memiliki bentuk tengah buah angular dengan pangkal buah agak ke dalam dan bobot buah berkisar 1.24 kilogram setiap buahnya. Selain itu pepaya California memiliki beberapa keunggulan di antaranya: buahnya berukuran sedang dengan berat berkisar 0,8-1,5 Kg/buah, memiliki kulit buah yang mulus, berbentuk lonjong yang menarik, buah matang yang berwarna jingga, memiliki rasa yang manis serta daging buah kenyal dan tebal (Utami Putri, 2016). Pepaya California termasuk jenis buah unggul dengan kemampuan berbuah sampai dengan umur empat tahun dapat dipanen dalam

empat kali panen dengan rentang waktu satu bulan serta menghasilkan dua hingga tiga buah untuk satu pohonnya (Redo Dwiantara, 2020). Dengan harga di bulan Februari 2021 untuk pepaya California berukuran besar dengan nilai jual Rp.3000 maka setiap pohonnya berpotensi menghasilkan keuntungan kotor Rp.9000 dengan kemampuan berbuah lebat hingga 25-30 buah pepaya California akan sangat menjanjikan keuntungan bagi petani. Bentuk pohon pepaya California dapat dilihat pada Gambar 3.1.



Gambar 3.1 Produktivitas Buah Pepaya California

3.1.2 Karakteristik Pepaya Thailand

Sesuai dengan namanya pepaya Thailand berasal dari negeri Gajah Thailand. Selain memiliki nama Thailand pepaya berukuran besar ini juga memiliki nama lain pepaya Bangkok dimana jenis pepaya ini mulai diketahui masyarakat Indonesia pada tahun 1970-an (Utami Putri, 2016). Karakteristik pepaya Thailand di antaranya: memiliki bentuk buah yang memanjang dan lonjong (elongata), memiliki tinggi pohon yang lebih tinggi dari pada pepaya California, rata-rata memiliki buah dengan ukuran besar bisa mencapai 3,5 kilogram untuk satu buahnya, kulit buah bergelombang dan kurang halus dan memiliki masa tanam yang lebih mudah dibanding pepaya California (Utami Putri, 2016). Meskipun memiliki buah yang besar namun rasa manis buah pepaya Thailand masih kalah dengan pepaya California hal ini yang

membuat penyerapan hasil panen pepaya jenis Thailand tidak sebanyak California. Pepaya Thailand memiliki kemampuan berbuah 7-12 buah yang dapat dipanen beriringan dengan pepaya California dengan menghasilkan buah yang siap petik antara dua hingga tiga buah setiap pohonnya. Dengan mengacu pada harga di bulan Februari untuk pepaya Thailand berukuran besar memiliki nilai jual sebesar Rp.5000 dengan setiap pohonnya memiliki 3 buah yang siap petik maka memiliki potensi keuntungan kotor Rp.15.000. Walaupun buah yang dihasilkan tidak sebanyak pepaya California namun masa tanam yang lebih mudah membuat pepaya Thailand digemari oleh para petani di Desa Keposong. Bentuk pohon pepaya Thailand dapat dilihat pada Gambar 3.2.



Gambar 3.2 Produktivitas Buah Pepaya Thailand

3.2 Analisa Kebutuhan Sistem

Bagian ini kebutuhan dari program yang dikembangkan akan di uraikan setiap langkah dan kebutuhannya agar dapat mencapai tujuan untuk mengklasifikasikan kematangan pepaya

California dan Thailand serta mendapatkan harga jualnya. Peneliti dituntut mengetahui pokok permasalahan agar solusi yang diberikan dapat sesuai dengan kebutuhan yang ada yaitu berupa kebutuhan *input*, proses dan *output*.

3.2.1 Analisa Kebutuhan Input

Masukan data citra yang dipakai adalah citra pepaya dengan warna RGB baik untuk varietas California dan Thailand. Citra yang digunakan memiliki tingkat resolusi yang sama serta konsistensi jarak pengambilan citra dengan jarak dasar obyek sampai dengan kamera *smartphone* sebesar 53 sentimeter. Jarak pengambilan perlu diperhatikan untuk mendapatkan hasil ukur luas pepaya yang baik sehingga sistem dapat membedakan besar pepaya dengan ukuran kecil, sedang dan besar. Pengambilan citra dengan jarak terlalu dekat untuk pepaya yang berukuran kecil akan meningkatkan besaran pepaya tersebut menjadi sedang atau bahkan besar dalam citra sehingga keadaan riil pepaya dan keadaan citra akan berbeda. Kebutuhan *input* citra pepaya menjadi dua kategori yaitu untuk citra latih dan citra uji. Citra latih terdapat 70 data untuk pepaya varietas California dan pepaya Thailand sedangkan citra uji berjumlah 30 citra untuk pepaya California dan Thailand dengan format *file* citra jpg yang diambil menggunakan kamera pada *smartphone*. Untuk proses *input* citra pepaya terdapat pada Gambar 3.3.



Gambar 3.3 Flowchart Analisis Input

3.2.2 Analisis Kebutuhan Proses

Pada tahap ini identifikasi setiap tahap yang akan dilakukan untuk merancang sebuah sistem berdasarkan data citra *input* yang telah diperoleh. Dalam penelitian ini kebutuhan proses adalah sebagai berikut.

a. Pembacaan citra

Tahap ini merupakan tahap awal pengolahan citra digital dengan menentukan berkas citra yang akan digunakan untuk selanjutnya dimuat dalam memori agar dapat di lakukan pemrosesan oleh sistem.

b. Preprocessing

Tahap *preprocessing* dilakukan dengan menurunkan ukuran citra menjadi 10% dari ukuran semula. Hal ini bertujuan untuk mempercepat proses komputasi, selain itu setelah menurunkan ukuran citra dilakukan tahap *preprocessing* lanjutan yaitu mengubah citra berwarna RGB menjadi citra *grayscale* untuk diketahui tingkat derajat keabuannya.

c. Nilai matriks kookurensi

Setelah citra RGB sudah berubah menjadi citra *grayscale* maka dapat diketahui tingkat derajat keabuan dari tiap titik yang ada. Matriks kookurensi diperoleh dengan menghitung seberapa sering titik dengan nilai tingkat keabuan muncul dengan menggunakan operasi morfologis arah tetangga 0, 45, 90 dan 135 derajat.

d. Ekstraksi ciri GLCM

Setelah diketahui nilai matriks kookurensi dari citra pepaya maka dapat dilakukan tahap lanjutan untuk mendapatkan ekstraksi ciri GLCM. Pada penelitian ini menggunakan 4 ciri ekstraksi GLCM yaitu *contrast*, *correlation*, *energi* dan *homogeneity* nilai ekstraksi ini akan digunakan sebagai pengenalan ciri buah pepaya yang sudah matang dan pepaya belum matang.

e. Klasifikasi Naive Bayes

Klasifikasi *Naive Bayes* digunakan untuk mengelompokkan hasil ekstraksi ciri dengan target keluaran berupa buah pepaya matang dan pepaya belum matang. Data latih didapatkan dari hasil ekstraksi ciri GLCM dengan 4 parameter ekstraksi ciri yang akan disusun bersama dengan target latih. Target latih terdapat dua parameter yaitu matang dan belum matang, setiap hasil perhitungan ciri dalam data latih akan diklasifikasikan dengan target latih. Hasil klasifikasi dalam tahap pelatihan akan disimpan dalam *database* yang akan digunakan sebagai parameter di tahap selanjutnya baik pada tahap pengujian ataupun pada tahap implementasi sistem dengan *Graphics User Interface* (GUI).

f. Segmentasi deteksi tepi

Tahap segmentasi deteksi tepi digunakan untuk mengetahui area obyek pepaya sehingga dapat diketahui luasan dari obyek pepaya tersebut. Adanya perbedaan warna antara obyek pepaya dengan latar belakang alas potret berwarna putih dapat digunakan sebagai penanda tepi karena adanya lonjakan intensitas warna yang tinggi menggunakan operator *Prewitt*. Hasil segmentasi deteksi tepi akan ditampilkan dalam sistem berupa citra dengan titik-titik yang membentuk area obyek pepaya.

g. Ekstraksi ciri luas

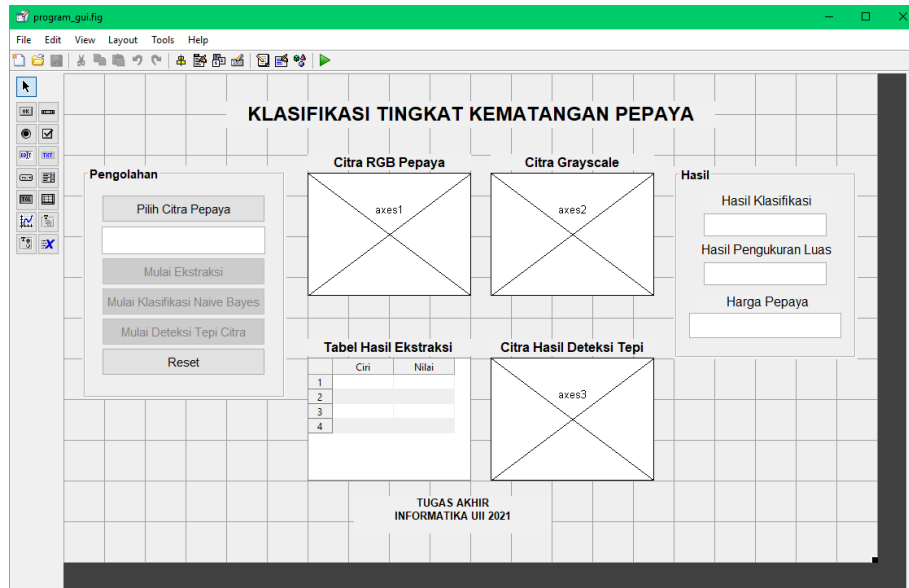
Tahap ekstraksi ciri luas ini memanfaatkan hasil dari tahap sebelumnya deteksi tepi dengan operator *Prewitt*. Area obyek pepaya yang sudah diketahui batasnya berupa titik tepi yang saling berhubungan dapat diketahui luasannya tanpa ditambah oleh area di belakangnya berupa latar belakang putih. Luas yang diperoleh berupa jumlah piksel yang ada di dalam area obyek pepaya. Luasan setiap obyek pepaya yang digunakan akan berbeda menurut ukurannya dibedakan menjadi tiga bagian yaitu kecil, sedang dan besar. Luasan area ini digunakan sebagai penentu harga pepaya semakin luas area piksel obyek pepaya menandakan semakin besar pula obyek pepaya di dalamnya sehingga harganya semakin mahal.

3.2.3 Analisa Kebutuhan Output

Kebutuhan *output* atau keluaran sistem berupa informasi hasil pengolahan citra pada setiap tahap yang ditampilkan dalam program GUI. Tahap awal pemilihan gambar akan ditampilkan citra yang dipilih dalam bentuk *fullcolor* atau RGB. Citra RGB yang sudah melalui tahap ekstraksi akan ditampilkan dalam bentuk *grayscale* serta terdapat informasi nilai dari setiap ciri ekstraksi GLCM. Pada tahap klasifikasi *Naive Bayes* muncul informasi berupa kategori kematangan pepaya apakah masuk ke dalam kategori matang atau belum matang. Tahap terakhir *output* adalah penampilan citra hasil deteksi tepi serta informasi luas pepaya dalam satuan piksel dan harga pepaya dalam rupiah. Hasil keluaran dari sistem diharapkan dapat berguna sebagai parameter rujukan kematangan pepaya dan harga jualnya oleh petani dalam melakukan transaksi jual beli.

3.3 Perancangan Antarmuka Sistem

Perancangan dilakukan untuk memperoleh gambaran perencanaan desain dan fungsi menjadi lebih terstruktur serta memudahkan dalam pengembangan sistem yang akan dibuat. Perancangan antarmuka diperlukan untuk menyesuaikan kebutuhan dan keluaran dari sistem yang dikembangkan, dengan adanya perancangan antarmuka peneliti akan lebih mudah dalam pembuatan sistem karena akan lebih terbimbing. Dalam program MATLAB sudah terdapat *tools guide* yang dapat digunakan untuk membuat rancangan antarmuka, selain dapat digunakan sebagai perancang antarmuka obyek yang disusun dalam tools ini nantinya yang akan digunakan dalam implementasi sistem dengan GUI yang disediakan oleh MATLAB. Gambar 3.4 merupakan tampilan *guide* dari MATLAB.



Gambar 3.4 Tampilan antarmuka *guide* pada MATLAB

Penjelasan dari rancangan tampilan antarmuka dalam *guide* MATLAB adalah sebagai berikut:

a. Panel Pengolahan

Pada panel ini merupakan navigasi pengguna untuk kebutuhan masukan dan pemrosesan data. Tombol pilih citra pepaya digunakan untuk memilih *file* citra. Tombol mulai deteksi, mulai klasifikasi *Naive Bayes* dan mulai deteksi citra digunakan untuk memulai pemrosesan di setiap tahapnya. Tombol akan aktif sesuai dengan urutan tahap yang dijalankan mulai dari tahap deteksi ekstraksi ciri GLCM, klasifikasi *Naive Bayes* dan deteksi tepi citra.

b. Panel citra

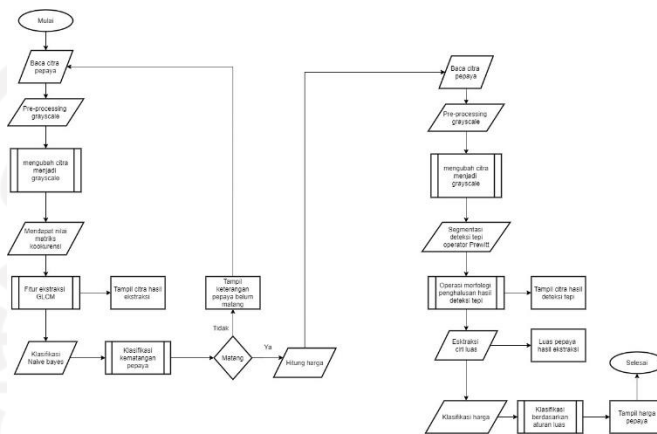
Panel citra akan menampilkan citra hasil pengolahan citra dengan masing-masing tahapnya. Panel citra RGB pepaya akan menampilkan citra awal *fullcolor* pepaya yang dipilih, panel citra *grayscale* menampilkan hasil konversi citra warna ke *grayscale* dan citra hasil deteksi tepi akan memunculkan citra hasil pengolahan pada tahap deteksi tepi *Prewitt*.

c. Panel hasil

Panel hasil akan menampilkan proses final dari berbagai tahap yang telah dilalui. Pada panel ini akan muncul informasi berupa teks *String* berupa hasil klasifikasi, hasil pengukuran luas dan harga pepaya beserta jenis pepaya.

3.4 Perancangan Alur Proses Sistem

Perancangan alur proses sistem merupakan gambaran dari keseluruhan proses yang ada di dalam sistem mulai dari *input*, proses hingga *output* dari program. *Flowchart* digunakan untuk menggambarkan setiap tahapan alur sistem dengan tujuan menguraikan bagaimana gambaran umum sistem akan dibangun. *Flowchart* mengenai program utama terdapat pada Gambar 3.5 berikut.



Gambar 3.5 *Flowchart* Program Utama

Flowchart ini terdapat tujuh proses dengan tahap awal *input* citra pepaya dalam bentuk RGB dan hasil akhir berupa *output* citra hasil deteksi tepi, klasifikasi kematangan pepaya, luas obyek dan harga pepaya. Proses terdiri dari tahap pembacaan citra, *preprocessing*, pembentukan nilai matriks kookurensi, ekstraksi ciri GLCM, klasifikasi *Naive Bayes*, deteksi tepi dan ekstraksi ciri luas. Pembacaan cerita bertujuan untuk memilih *file* citra pepaya yang akan dilakukan analisa, berkas yang dipilih yang dapat diproses adalah berkas dengan ekstensi *file* .jpg dalam bentuk citra *fullcolor* atau RGB.

Setelah *file* citra dipilih tahap berikutnya adalah melakukan *preprocessing* citra dengan mengubah citra warna menjadi citra *grayscale* dimana setiap piksel citra *grayscale* memiliki intensitas keabuan yang berkisar antara 0 sampai 255. Nilai intensitas piksel derajat keabuan akan digunakan dalam pembentukan matriks kookurensi pada tahap berikutnya. Pembuatan matriks kookurensi ditentukan dengan arah tetangga dengan melakukan operasi morfologis dengan arah derajat 0, 45, 90 dan 135 derajat. Dalam MATLAB arah derajat tetangga ini diimplementasikan dalam bentuk *array* [0 1 ; -1 1 ; -1 0 ; -1 -1] dimana [0 1] mewakili arah 0°, [-1 1] mewakili arah 45°, [-1 0] mewakili arah 90° dan [-1 -1] mewakili arah 135°. Operasi morfologis dilakukan dengan menghitung seberapa sering piksel dengan nilai tingkat keabuan muncul sesuai dengan derajat tetangga yang dihitung. Setelah diketahui nilai matrik kookurensi

maka dapat dilakukan ekstraksi ciri GLCM dengan ciri yang digunakan *contrast*, *correlation*, *energy* dan *homogeneity*. Nilai final ekstraksi GLCM dipakai untuk parameter ciri citra buah pepaya matang dan belum matang yang sudah disimpan dalam *database*.

Klasifikasi *Naive Bayes* akan menentukan kelas pepaya yang di analisa apakah masuk dalam kelas matang atau belum matang dengan menelusuri antara ciri citra dan target berupa matang dan belum matang. Hasil klasifikasi akan ditampilkan dalam GUI MATLAB dengan keterangan matang dan tidak matang. Segmentasi deteksi tepi dilakukan untuk mendapatkan luas obyek pepaya dengan menentukan area obyek menggunakan operator *Prewitt* yang akan membentuk titik-titik batas antara obyek pepaya dengan latar belakang putih pada alat bantu potret. Perbandingan intensitas warna yang tinggi memungkinkan untuk diketahui batas obyek dengan mudah, sebaliknya semakin samar batas warna antara obyek dengan latar belakang menjadikan batas menjadi tidak akurat. Setelah diketahui batas obyek maka dapat dilakukan perhitungan luas dengan menjumlah piksel yang ada dalam area obyek. Semakin banyak piksel menandakan semakin besar ukuran pepaya yang akan digunakan sebagai parameter penentuan harga jual dalam satuan rupiah. Hasil akhir dari proses analisa berupa klasifikasi kematangan pepaya, luas area pepaya dan harga jual pepaya.

3.5 Metode Pengujian Sistem

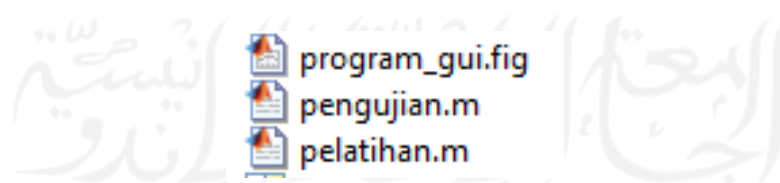
Terdapat dua metode pengujian, yaitu pengukuran performa akurasi sistem dan pengukuran kinerja metode klasifikasi yang digunakan. Pengukuran kinerja sistem dilakukan dengan menggunakan metode *Confusion Matrix* dengan mencari nilai *Accuracy*, *Sensitivity* dan *Specificity*. Pengujian ini dilakukan ke dalam dua kelompok yaitu tahap pelatihan dan pengujian. Tahap pelatihan dilakukan untuk mengetahui kinerja pada tahap pelatihan data latih sedangkan tahap pengujian dilakukan untuk mengetahui kinerja pada tahap pengujian. Nilai *Confusion matrix* berfungsi untuk mengetahui kinerja metode klasifikasi yang digunakan.

Untuk mengetahui tingkat akurasi dari klasifikasi metode *Naive Bayes* dilakukan dengan menggunakan metode *Area Under Curve* (AUC). Metode AUC memiliki nilai keakuratan antara 0 hingga 1 di mana semakin mendekati 1 maka sistem klasifikasi yang digunakan semakin baik pula. Jika hasil yang didapatkan pada pengujian metode AUC mendapatkan skor di bawah 0,6 menandakan jika metode klasifikasi yang digunakan salah dan tidak akurat sama sekali sehingga tidak dianjurkan untuk digunakan.

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

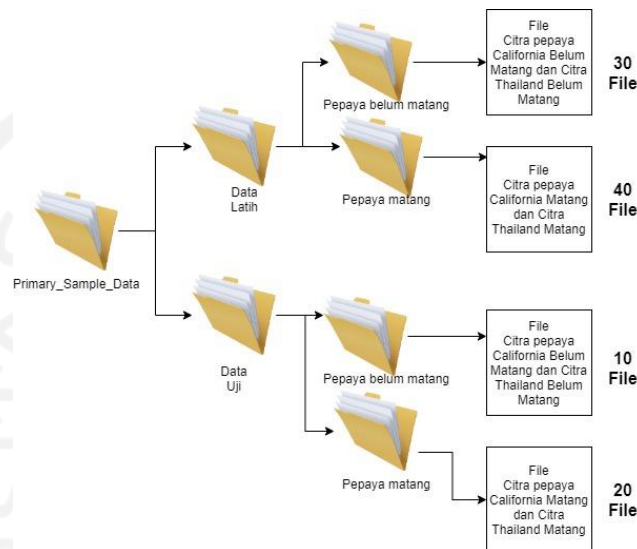
Flowchart pada Gambar 3.2 akan diimplementasikan ke sistem menggunakan *software* MATLAB. Baris *coding* akan dituliskan dengan menggunakan bahasa pemrograman MATLAB dimana bahasa ini memiliki karakteristik yang hampir sama dengan bahasa pemrograman C/C++. Pada BAB ini akan dijelaskan kode program setiap proses yang ada pada sistem serta fungsi dari kode baris program tersebut. Beberapa fungsi yang sudah tersedia di MATLAB juga digunakan untuk efektivitas penulisan kode program. Bagian utama sistem akan terbagi menjadi tiga jenis yaitu pelatihan, pengujian dan program utama. Pada tahap pelatihan digunakan untuk membuat *database* nilai rujukan data latih dan target latih yang digunakan sebagai parameter klasifikasi pepaya ke dalam kelas pepaya matang atau pepaya belum matang. Bagian pengujian akan menguji data latih yang telah didapat dengan menggunakan citra uji untuk mengetahui efektivitas metode yang digunakan serta mengetahui nilai hasil klasifikasi apakah sudah sesuai dan memenuhi syarat untuk diimplementasikan. Program utama merupakan hasil akhir untuk klasifikasi menggunakan GUI di mana pengguna dapat menggunakan dengan mudah hanya dengan memasukkan citra pepaya yang ingin diidentifikasi, program akan memberi informasi berupa klasifikasi kematangan pepaya serta prediksi harga pepaya berdasarkan besaran buah pepaya. Pembagian jenis tahapan terdapat pada Gambar 4.1.



Gambar 4.1 Struktur Sistem

Setiap tahap yang ada akan dijelaskan dalam *flowchart* untuk mengetahui bagaimana alur program dan baris kode bekerja. Akan dijelaskan pula tahap penggunaan sistem yang disertai tampilan antarmuka. Dengan rancangan *flowchart* yang telah dibuat sebelumnya diharapkan dapat membantu peneliti dalam mencapai tujuan dari penelitian ini. Sebelumnya nilai dari pakar ditentukan dengan cara mengklasifikasikan citra secara manual sesuai dengan *folder*

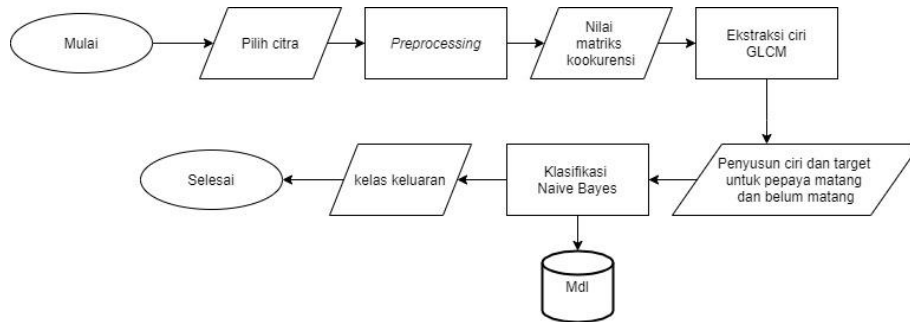
target yaitu dengan membuat *folder* pepaya matang dan belum matang. *Folder* pepaya matang berisi citra pepaya matang yang akan menjadi ciri dan target citra pepaya matang sedangkan *folder* pepaya belum matang akan menjadi ciri dan target pepaya yang belum matang. Struktur folder terdapat pada Gambar 4.2.



Gambar 4.2 Struktur Folder

4.1.1 Tahap Pelatihan

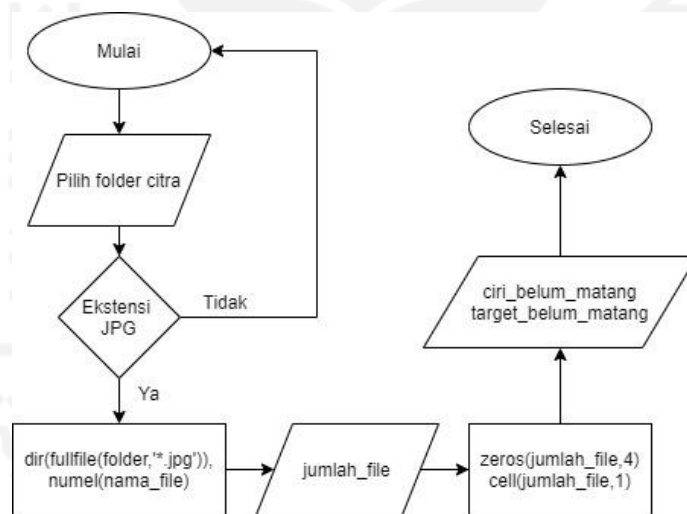
Tahap pelatihan merupakan tahap awal dengan tujuan akhir pembentukan data latih yang digunakan untuk sistem pakar klasifikasi pepaya. Dalam tahap pelatihan akan terbagi menjadi dua proses yaitu pelatihan data pepaya belum matang dan pelatihan data pepaya matang. Kedua proses ini akan membentuk ciri untuk pepaya belum matang dan pepaya matang pada variabel data latih. Selain pelatihan untuk mendapatkan ciri pada data latih juga terdapat variabel target baik untuk pepaya matang dan belum matang yang berisi kondisi riil pepaya yaitu matang dan tidak matang. Kedua parameter data latih dan target latih ini akan diklasifikasikan dengan menggunakan metode *Naive Bayes* sebagai data acuan untuk tahap berikutnya tahap pengujian. Proses pada tahap pelatihan terdapat pada Gambar 4.3.



Gambar 4.3 Diagram Alur Tahap Pelatihan

a. Pemilihan Citra

Pemilihan citra merupakan tahap awal baik untuk tahap pelatihan ataupun tahap pengujian. Pemilihan citra ditujukan untuk menentukan *folder* yang akan digunakan. Dalam *folder* data latih terdapat 70 citra pepaya California dan Thailand dengan kondisi matang dan belum matang yang akan menjadi data latih untuk sistem pakar klasifikasi pepaya. Untuk memudahkan proses identifikasi sebelumnya nama *file* sudah diubah sesuai dengan jenis dan kondisi pepaya. Proses pemilihan *folder* terdapat pada Gambar 4.4.



Gambar 4.4 Flowchart Pemilihan Citra

Proses ini diawali dengan menentukan lokasi *file* citra yang akan digunakan untuk tahap pelatihan. Baris kode untuk pemilihan *file* untuk tahap latih dengan menggunakan perintah folder dari MATLAB. Perlu diperhatikan dalam penentuan lokasi *file* harus berada dalam satu *folder* utama dari M *file* milik MATLAB yang akan dieksekusi agar dapat ditemukan keberadaannya. Sesuai dengan *flowcahart* tahap pemilihan *file* dimulai dengan memilih

folder citra, jika format citra bukan .jpg maka proses tidak dapat dilanjutkan karena sebelumnya sudah ditentukan format citra yang digunakan adalah .jpg hal ini ditujukan untuk menyesuaikan *output* citra yang dihasilkan oleh kamera *smartphone* yang digunakan untuk mendapatkan data primer citra. Jika folder yang diidentifikasi memiliki format citra .jpg maka dapat dilanjutkan dengan membaca jumlah *file* dalam keseluruhan isi satu *folder* untuk *folder* citra latih yang akan disimpan dalam variabel *jumlah_file*. Kode program untuk pemilihan *file* citra terdapat pada Gambar 4.5.

```
% menetapkan nama folder
folder = 'Primary_Sample_Data\Data_Latih\pepaya belum matang';
% membaca berkas dengan ekstensi .jpg
nama_file = dir(fullfile(folder, '*.jpg'));
% membaca jumlah total berkas
jumlah_file = numel(nama_file);
```

Gambar 4.5 Kode Program Pilih Citra

Perbedaan dari baris program antara data latih untuk pepaya matang dan belum matang ada pada penentuan nama *folder* keduanya sesuai dengan citra yang. Untuk analisa pertama pada data latih ditentukan *folder* latih pepaya belum matang dan analisa kedua pada data latih ditentukan folder pepaya matang. Pemilihan *folder* ini akan membentuk variabel *ciri_belum_matang*, *target_belum_matang*, *ciri_matang* dan *target_matang* untuk tahap data latih. Perbedaan kode program untuk data latih antara pepaya belum matang dan matang terdapat pada Gambar 4.6 dan Gambar 4.7.

```
% menetapkan nama folder
folder = 'Primary_Sample_Data\Data_Latih\pepaya belum matang';
→ untuk memilih folder data latih pepaya belum matang
folder = 'Primary_Sample_Data\Data_Latih\pepaya matang';
→ untuk memilih folder data latih pepaya matang
```

Gambar 4.6 Kode Program Penentuan *Folder*

Setelah diketahui jumlah *file* citra kemudian sistem akan membuat variabel ciri untuk pepaya matang dan belum matang serta target pepaya matang dan belum matang. Variabel sementara ini akan bernilai nol yang nantinya akan diisi oleh matriks kookurensi hasil analisa citra. Dengan bentuk matriks *jumlah_file* x 4 untuk ciri pepaya baik matang dan belum matang serta *jumlah_file* x 1 untuk target pepaya baik pepaya matang ataupun pepaya belum matang. Kode baris untuk inialisasi variabel terdapat pada Gambar 4.7.

```
% menginisialisasi variabel ciri_belum_matang & target_belum_matang
ciri_belum_matang = zeros(jumlah_file,4);
target_belum_matang = cell(jumlah_file,1);
```

Gambar 4.7 Kode Program Inisialisasi Variabel

b. Preprocessing

Preprocessing merupakan langkah setelah citra pepaya yang akan digunakan telah dipilih. Dalam penelitian ini tahap pra pengolahan citra dilakukan baik untuk tahap pelatihan, pengujian dan program utama. Pada tahap ini citra yang sudah dipilih akan dilakukan optimalisasi untuk mencapai tujuan penelitian. Tahap *preprocessing* dilakukan dalam dua tahap yaitu *resize* citra dan konversi citra ke dalam bentuk citra *grayscale*. Nilai yang ada dalam citra awal RGB akan disesuaikan dengan kebutuhan menjadi citra dalam bentuk 8 bit di mana setiap piksel akan bernilai antara 0 hingga 255.

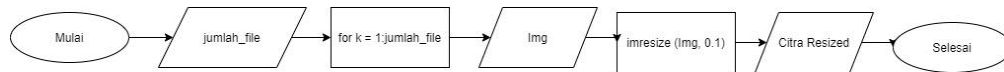
1. *Resize* Citra

Tahap *resize* citra di gunakan untuk mengecilkan ukuran citra dengan satuan piksel. Perubahan ukuran citra ini dilakukan untuk mempercepat komputasi perhitungan nilai citra sehingga program dapat bekerja secara optimal dalam jenis komputer dengan spesifikasi rendah. Ukuran awal citra 3840 x 2160 piksel akan dikecilkan menjadi 10% dari ukuran semula yaitu menjadi 384 x 216 piksel. Ukuran awal citra terdapat pada Gambar 4.8.

Image	
Image ID	
Dimensions	3840 x 2160
Width	3840 pixels
Height	2160 pixels
Horizontal resolution	72 dpi
Vertical resolution	72 dpi
Bit depth	24
Compression	
Resolution unit	2
Color representation	sRGB
Compressed bits/pixel	

Gambar 4.8 Ukuran Citra Awal

Variabel `jumlah_file` yang sudah didapatkan sebelumnya digunakan sebagai parameter perulangan pembacaan citra RGB untuk proses pengecilan ukuran citra. Dengan menggunakan perulangan maka semua *file* citra dalam satu lokasi *folder* dapat dikecilkan ukurannya secara menyeluruh untuk semua *file* yang ada. Dalam citra latih terdapat 70 citra maka akan terjadi pengecilan ukuran terhadap 70 citra latih tersebut. Untuk proses pengecilan ukuran citra terdapat pada Gambar 4.9.

Gambar 4.9 Diagram Alur *Resize* Citra

Variabel `jumlah_file` yang didapatkan dari proses pembacaan citra RGB pada tahap sebelumnya digunakan sebagai masukan dalam awal proses pengecilan citra. Dalam variabel `jumlah_file` terdapat informasi jumlah *file* yang ada dalam folder data latih baik untuk citra pepaya matang dan pepaya belum matang, jumlah *file* ini akan digunakan untuk alur perulangan *resize* citra pada data latih. Citra hasil *resize* kemudian disimpan dalam variabel `Img` untuk digunakan pada tahap berikutnya yaitu konversi citra RGB menjadi *grayscale*. Kode program untuk *resize* citra terdapat pada Gambar 4.10.

```

for k = 1:jumlah_file
    % membaca citra RGB
    Img = imread(fullfile(folder,nama_file(k).name));
    % memperkecil ukuran citra menjadi 10% ukuran semula untuk
    mempercepat komputasi
    Img = imresize(Img,0.1);
    % mengkonversi cira rgb menjadi grayscale
  
```

Gambar 4.10 Kode Program *Resize* Citra

2. Konversi *Grayscale*

Tahap konversi *grayscale* digunakan untuk mendapatkan nilai tingkat keabuan dari setiap piksel citra. Jika sebelumnya pada citra RGB terdapat informasi mengenai tingkat warna dalam tiga komponen yaitu Red, Green dan Blue dalam citra *grayscale* informasi ini akan diubah menjadi informasi tunggal yaitu tingkat keabuan dengan skala 0 hingga 255 pada setiap pikselnya. Proses konversi citra RGB menjadi *grayscale* terdapat pada Gambar 4.11 dan untuk kode konversi citra *grayscale* terdapat pada Gambar 4.12.

Gambar 4.11 *Flowchart* Konversi Citra *Grayscale*


```
% mengkonversi cira rgb menjadi grayscale
Img_gray = rgb2gray(Img);
```

Gambar 4.12 Kode Program Konversi *Grayscale*

Citra RGB menghasilkan dimensi matriks $346 \times 461 \times 3$. Dimensi 346×461 merupakan resolusi citra setelah proses *resize image*, sedangkan $\times 3$ merupakan dimensi intensitas warna RGB. Perbedaan struktur matriks citra RGB dan *grayscale* dapat dilihat pada Gambar 4.13.

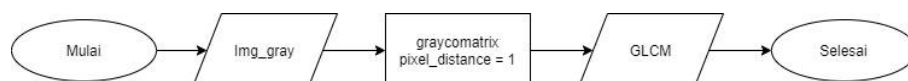


Gambar 4.13 Perbedaan Struktur Matriks Citra RGB dan *Grayscale*

Dari nilai hasil variabel *Img* dan *Img_gray* terlihat perbedaan keduanya di mana pada variabel *Img_gray* tidak terdapat dimensi terakhir berupa 3 unit warna RGB. Warna yang sudah dikonversi ke citra *grayscale* nilainya menjadi hilang dan diganti dengan intensitas aras keabuan pada citra *grayscale*.

c. Pembentukan Matriks Kookurensi

Matriks kookurensi merupakan metode statistik yang digunakan untuk analisis tekstur citra yang terbentuk dari nilai aras keabuan citra dengan hubungan antar piksel yang bertetangga (Syauqi Falasev et al., 2011). Dalam MATLAB sudah disediakan fungsi *Graycomatrix* yang dapat digunakan untuk melakukan perhitungan nilai kookurensi sebuah citra. Nilai kookurensi ini didapat dari citra *grayscale* yang memiliki nilai aras keabuan 0 hingga 255 pada setiap pikselnya. Dalam penelitian ini arah keabuan yang dipakai adalah 0° , 45° , 90° dan 135° . Dalam MATLAB *Gray Level Co-occurrence Matrix* disebut sebagai *Gray Level Spatial Dependence Matrix*. Proses pembentukan nilai matriks kookurensi terdapat pada Gambar 4.14.



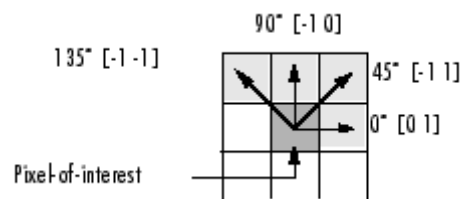
Gambar 4.14 Flowchart Pembentukan Matriks Kookurensi

Dalam penelitian ini *offset* ditentukan dengan nilai 1 atau disebut juga dengan *pixel distance*. MATLAB menggunakan *array* sebagai operator nilai pembangkit matriks kookurensi dengan persamaan pada Tabel 4.1.

Tabel 4.1 *Offset* Matriks Kookurensi

Sudut	<i>Offset</i>
0	[0 D]
45	[-D D]
90	[-D 0]
135	[-D -D]

Dalam implementasi di MATLAB akan ditulis dengan menggunakan *array* [0 1; -1 1; -1 0; -1 -1] dapat juga mendefinisikan sendiri nilai *pixel distance* dengan mengganti nilai D. Untuk arah aras keabuan dalam membangkitkan nilai matriks kookurensi terdapat pada Gambar 4.15.



Gambar 4.15 Arah Matriks Kookurensi

Nilai matriks kookurensi akan menyesuaikan dengan nilai intensitas maksimum sebuah citra L, dalam penelitian ini nilai matriks kookurensi memiliki ukuran $(L+1) \times (L+1)$. Kode program untuk perhitungan nilai kookurensi terdapat pada Gambar 4.16.

```
% pembentukan nilai matriks kookurensi
pixel_dist = 1;
GLCM = graycomatrix(Img_gray, 'Offset', [0 pixel_dist; ...
    -pixel_dist pixel_dist; -pixel_dist 0; ...
    -pixel_dist -pixel_dist]);
```

Gambar 4.16 Kode Program Pembentukan Matriks Kookurensi

Nilai matriks kookurensi hasil perhitungan data latih terdapat pada Gambar 4.17.

```

val(:,:,1) =
    0         0         0         0         0         0         0         0
    0    1607         517         3         1         0         0         0
    0     518    9440         947         30         1         0         0
    0         3     975    9189    1152         34         0         0
    0         0         4    1206    30091    1667         4         0
    0         0         0         8    1453    41292    874         0
    0         0         0         0         6     777    54820    301
    0         0         0         0         0         1     300    1939

val(:,:,2) =
    0         0         0         0         0         0         0         0
    0    1279         825         22         2         0         0         0
    0     836    8502    1488         102         8         0         0
    0         10    1588    7940    1701    113         1         0
    0         1         21    1875    28874    2112         5         0
    0         2         0         28    2014    40252    957         0
    0         0         0         0         18    1184    54346    354
    0         0         0         0         0         21     337    1882

val(:,:,3) =
    0         0         0         0         0         0         0         0
    0    1490         637         1         0         0         0         0
    0     633    9213    1059         31         0         0         0
    0         5    1077    8913    1326         31         1         0
    0         0         9    1372    30078    1436         20         0
    0         0         0         8    1522    41491    456         0
    0         0         0         0         20     807    55006    163
    0         0         0         0         0         4     163    2073

val(:,:,4) =
    0         0         0         0         0         0         0         0
    0    1465         661         2         0         0         0         0
    0     656    9144    1100         36         0         0         0
    0         6    1083    8817    1390         55         2         0
    0         1         43    1398    29511    1634         63         0
    0         0         5         36    1982    40709    665         0
    0         0         0         0         31    1143    54515    307
    0         0         0         0         0         3     308    1929

```

Gambar 4.17 Hasil Perhitungan *Matrix* GLCM Data Latih

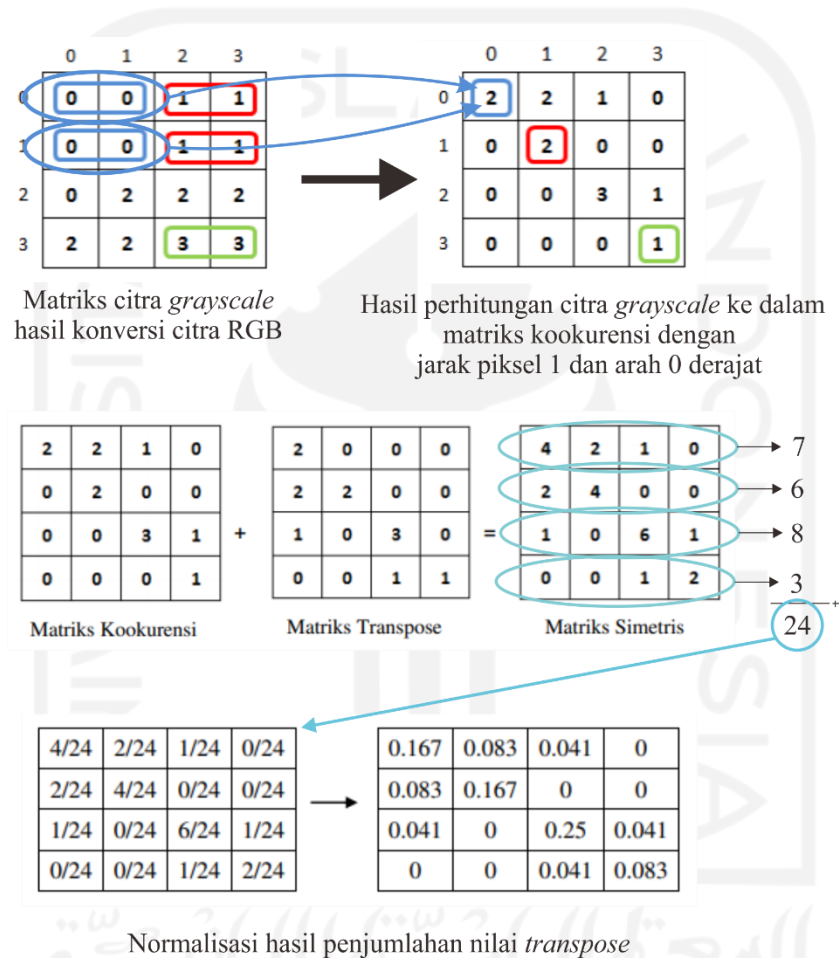
Citra *grayscale* merupakan citra dengan kedalaman 8 bit sehingga dimensi matriks yang dihasilkan akan berukuran $n \times n$ di mana n merupakan tingkat kedalaman citra sehingga dimensi matriks yang dihasilkan akan berukuran 8×8 (Jusman et al., 2017). Dimensi matriks yang dihasilkan pada Gambar 4.17 memiliki ukuran $8 \times 8 \times 4$, di mana dimensi 8×8 diperoleh dari kedalaman citra *grayscale* sedangkan 4 merupakan arah sudut yang digunakan yaitu 0° , 45° , 90° dan 135° .

Dimensi matriks 8×8 pada `val(:,:,1)` menunjukkan hasil perhitungan matriks kookurensi dengan sudut 0° . Untuk memperoleh nilai matriks tersebut berikut langkah-langkah yang dilakukan:

1. Memperoleh nilai matriks awal hasil konversi citra RGB ke citra *grayscale*.
2. Membentuk nilai matriks kookurensi dengan *pixel distance* 1 dan sudut yang dipakai 0° . Perhitungan ini akan dilakukan berulang sesuai dengan arah sudut yang digunakan berikutnya yaitu 45° , 90° dan 135° .
3. Menjumlah nilai matriks kookurensi dengan nilai matriks *tranpose*-nya.

4. Melakukan perhitungan normalisasi hasil penjumlahan nilai matriks kookurensi dengan nilai matriks *transpose*.
5. Hasil normalisasi dapat digunakan untuk tahap selanjutnya sesuai dengan fitur ekstraksi yang digunakan yaitu *Contrast*, *Correlation*, *Energy* dan *Homogeneity*.

Langkah satu hingga lima tersebut dapat dilihat pada contoh perhitungan matriks kookurensi dengan dimensi 4x4 serta arah sudut 0 derajat pada Gambar 4.18.

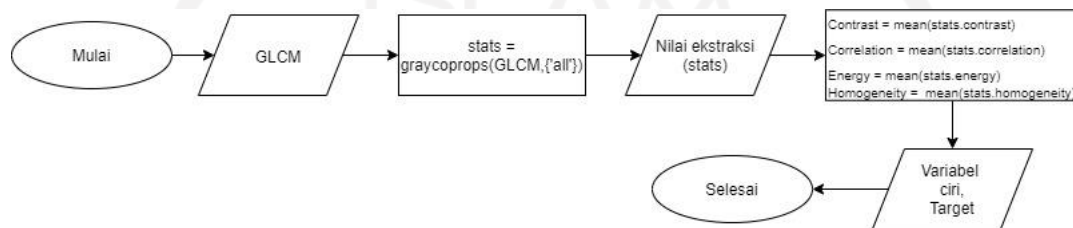


Gambar 4.18 Langkah Perhitungan Matriks Kookurensi (Fatihah, 2016)

Dimensi matriks 8x8 kedua pada `val(:, :, 2)` menunjukkan hasil perhitungan matriks kookurensi dengan sudut 45° menggunakan *array* [-1 1]. Langkah pembentukan matriks kedua ini sama seperti pada sudut 0° hanya saja pembentukan matriks berubah menggunakan arah 45°. Hal ini juga berlaku untuk matriks berikutnya yaitu pembentukan matriks kookurensi menggunakan arah 90° dan 135°. Dengan adanya 4 sudut yang digunakan sehingga akan membentuk matriks dengan dimensi 8x8x4.

d. Ekstraksi Ciri GLCM

Setelah diperoleh nilai matriks kookurensi langkah selanjutnya adalah melakukan ekstraksi ciri GLCM yang bertujuan untuk memperoleh nilai dari setiap ciri yang dipakai yaitu ciri *Contrast*, *Correlation*, *Energy* dan *Homogeneity*. Nilai ciri tersebut dipakai sebagai nilai ciri baik untuk pepaya matang ataupun belum matang. Dalam penelitian ini ciri kematangan buah pepaya California dan Thailand memiliki ciri yang sama yaitu terdapat warna oranye maka dapat menggunakan fitur ekstraksi ciri yang sama. Proses ekstraksi ciri terdapat pada Gambar 4.19.



Gambar 4.19 Flowchart Ekstraksi Ciri GLCM

Nilai ekstraksi ciri GLCM akan disimpan dalam variabel stats dengan mengacu ciri *Contrast*, *Correlation*, *Energy* dan *Homogeneity*. Dalam implementasi pada program MATLAB. Ekstraksi ciri fitur *Contrast* menggunakan *library* yang sudah disediakan dengan persamaan 4.1 berikut.

$$Contrast = \sum_{i,j} |i - j|^2 p(i,j) \quad (4.1)$$

Contrast pada MATLAB akan mengembalikan ukuran kontras intensitas antara piksel dan tetangganya di seluruh gambar dengan $range = [0 (size(GLCM,1)-1)^2]$. Ekstraksi ciri fitur *Correlation* menggunakan *library* yang sudah disediakan dengan persamaan 4.2 berikut.

$$Correlation = \sum_{i,j} \frac{(I - MI)(J - MJ)P(I,J)}{\Sigma i \Sigma j} \quad (4.2)$$

Correlation akan mengembalikan ukuran korelasi piksel dengan tetangganya pada citra dengan $range = [-1 1]$. Ekstraksi ciri fitur *Energy* menggunakan *library* yang sudah disediakan dengan persamaan 4.3 berikut.

$$Energy = \sum_{i,j} p(i,j)^2 \quad (4.3)$$

Energy akan mengembalikan jumlah elemen kuadrat dalam GLCM dengan *range* = [0 1]. Ekstraksi ciri fitur *Homogeneity* menggunakan *library* yang sudah disediakan dengan persamaan 4.4 berikut.

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1 + |i - j|} \quad (4.4)$$

Homogeneity akan mengembalikan nilai yang mengukur kedekatan distribusi elemen dalam GLCM ke diagonal GLCM dengan *range* = [0 1].

Karena dalam variabel *stats* semua ciri dalam citra menjadi satu maka diperlukan nilai pembalik untuk mengembalikan nilai ekstraksi ciri dari setiap citra yang ada sesuai urutan dan jumlah file (nilai *k*). Setiap nilai yang diperoleh untuk setiap *file* citra akan disimpan dalam variabel *ciri_belum_matang* yang akan berbentuk matriks dengan dimensi *k* x 4. Sedangkan untuk target yang diperoleh dari pelatihan pepaya belum matang dan matang akan disimpan dalam variabel *target*, nilai target disesuaikan dengan rujukan yang diberikan oleh pakar dalam hal ini sesuai pengklasifikasian manual dalam *folder* awal pemilihan citra. Kode program ekstraksi ciri GLCM terdapat pada Gambar 4.20.

```

% ekstraksi ciri GLCM
stats = graycoprops(GLCM, {'all'});
Contrast = mean(stats.Contrast);
Correlation = mean(stats.Correlation);
Energy = mean(stats.Energy);
Homogeneity = mean(stats.Homogeneity);
% menyusun variabel ciri_belum_matang
ciri_belum_matang(k,1) = Contrast;
ciri_belum_matang(k,2) = Correlation;
ciri_belum_matang(k,3) = Energy;
ciri_belum_matang(k,4) = Homogeneity;
% menyusun variabel target belum matang
target_belum_matang{k,1} = 'Belum Matang'; → untuk folder belum
matang
target_matang{k,1} = 'Matang'; → untuk folder matang

```

Gambar 4.20 Kode Program Ekstraksi Ciri GLCM

Hasil ekstraksi ciri akan membentuk nilai ciri untuk data latih yang terdapat pada Tabel 4.2.

Tabel 4.2 Hasil Ekstraksi Ciri GLCM

File Citra	Ciri Citra				Kelas
	Contrast	Correlation	Energy	Homogeneity	
1	0.043526252	0.990015251	0.244487546	0.980027431	'Belum Matang'

2	0.035697619	0.984810595	0.519558613	0.983459391	'Belum Matang'
3	0.034777633	0.986582824	0.53373629	0.983448571	'Belum Matang'
4	0.031632749	0.985984264	0.584272008	0.984725754	'Belum Matang'
5	0.030692765	0.988776777	0.509727923	0.985119684	'Belum Matang'
6	0.046838252	0.984756876	0.410349775	0.978595078	'Belum Matang'
7	0.043747741	0.984650132	0.381047529	0.979440359	'Belum Matang'
8	0.044052769	0.987905211	0.245072397	0.979750692	'Belum Matang'
9	0.05288398	0.978919615	0.319463328	0.976652968	'Belum Matang'
10	0.045372894	0.987279523	0.266331395	0.980139406	'Belum Matang'
11	0.041651822	0.987132055	0.288810738	0.982485783	'Belum Matang'
12	0.044239903	0.988264795	0.297958993	0.981260032	'Belum Matang'
13	0.05804999	0.983057742	0.293902513	0.975020462	'Belum Matang'
14	0.052313298	0.98648967	0.287818725	0.977342329	'Belum Matang'
15	0.036137616	0.986053807	0.473304892	0.98325809	'Belum Matang'
16	0.164837534	0.983146914	0.136064037	0.9311905	'Belum Matang'
17	0.060605195	0.986738016	0.206752598	0.972014723	'Belum Matang'
18	0.062554092	0.986915811	0.221536676	0.972391006	'Belum Matang'
19	0.10091924	0.988669159	0.2236718	0.957425194	'Belum Matang'
20	0.066452445	0.988944377	0.256258049	0.972006445	'Belum Matang'
21	0.068355437	0.988049549	0.261514599	0.969301336	'Belum Matang'
22	0.100628179	0.988768311	0.231861353	0.959560315	'Belum Matang'
23	0.047790453	0.987669245	0.337895716	0.978922551	'Belum Matang'
24	0.056545081	0.984614016	0.241753274	0.973036188	'Belum Matang'
25	0.068758992	0.985520874	0.264495953	0.970570223	'Belum Matang'
26	0.06365357	0.986712726	0.234146053	0.971751	'Belum Matang'
27	0.064633809	0.991414786	0.203699516	0.971921681	'Belum Matang'
28	0.06584385	0.988860631	0.210076001	0.969421102	'Belum Matang'
29	0.045649611	0.990764587	0.232661	0.97792519	'Belum Matang'
30	0.078901044	0.98608851	0.203615197	0.965612655	'Belum Matang'
31	0.046638655	0.986462783	0.350233699	0.977469683	'Matang'
32	0.061555195	0.961220028	0.508734951	0.969646941	'Matang'
33	0.069628514	0.979823123	0.283723913	0.966810658	'Matang'
34	0.053840534	0.982746067	0.347219721	0.973898804	'Matang'
35	0.071995217	0.97156936	0.30837656	0.964858228	'Matang'
36	0.039797464	0.9858725	0.595575378	0.981454281	'Matang'
37	0.06833142	0.977867465	0.289615768	0.96699025	'Matang'
38	0.033513353	0.975234896	0.655907725	0.983514295	'Matang'
39	0.069875198	0.976845347	0.314290367	0.96604816	'Matang'
40	0.067892929	0.979062796	0.292044693	0.967048191	'Matang'
41	0.067054734	0.972928523	0.316819342	0.967313126	'Matang'
42	0.057081433	0.979260228	0.362940725	0.972438418	'Matang'
43	0.067763027	0.971002714	0.430956522	0.966877201	'Matang'
44	0.076241296	0.975301718	0.276413851	0.963938018	'Matang'
45	0.049721126	0.986119323	0.33615917	0.975910048	'Matang'

46	0.059116387	0.982883006	0.301497441	0.971346299	'Matang'
47	0.052953022	0.982961777	0.349982701	0.974681867	'Matang'
48	0.062903186	0.977240209	0.357939903	0.969791372	'Matang'
49	0.055407787	0.984975396	0.323327013	0.973158931	'Matang'
50	0.040031958	0.98899209	0.520239172	0.981171972	'Matang'
51	0.078620873	0.982446037	0.192624567	0.963851124	'Matang'
52	0.072868027	0.985268871	0.199565285	0.965678297	'Matang'
53	0.092863472	0.984290287	0.191534449	0.958754633	'Matang'
54	0.082954933	0.98349305	0.196208423	0.966586999	'Matang'
55	0.095429024	0.985873439	0.261929462	0.959682481	'Matang'
56	0.074557245	0.984166596	0.27156746	0.967374983	'Matang'
57	0.081950383	0.986168398	0.281163	0.964237076	'Matang'
58	0.074747037	0.985622736	0.330877541	0.967949256	'Matang'
59	0.067637037	0.984847906	0.320266987	0.969245063	'Matang'
60	0.081154376	0.979693888	0.325834816	0.962851481	'Matang'
61	0.104507588	0.983872603	0.191465405	0.954373434	'Matang'
62	0.072062304	0.980456562	0.207029127	0.965991777	'Matang'
63	0.084072893	0.984002494	0.171149881	0.961234015	'Matang'
64	0.084911177	0.98386069	0.181435816	0.960067543	'Matang'
65	0.082660526	0.976507304	0.190417051	0.961108206	'Matang'
66	0.071656267	0.977005731	0.222355482	0.965734343	'Matang'
67	0.089835603	0.977636788	0.264943824	0.95864676	'Matang'
68	0.063658612	0.985582228	0.228153871	0.974416339	'Matang'
69	0.055035039	0.986589356	0.225492058	0.978335942	'Matang'
70	0.082650823	0.975421057	0.226692153	0.960589977	'Matang'

e. Klasifikasi *Naive Bayes*

Klasifikasi *Naive Bayes* pada penelitian ini digunakan untuk mengklasifikasikan kematangan pepaya baik untuk pepaya California ataupun pepaya Thailand dengan menggunakan parameter nilai ekstraksi ciri GLCM. Untuk menyusun klasifikasi *Naive Bayes* pada MATLAB dapat menggunakan fungsi `fitcnb` dengan menggunakan parameter `data_latih` dan `target_latih` yang sudah didapat pada tahap sebelumnya yaitu tahap ekstraksi ciri GLCM. Fungsi `fitcnb` digunakan pada MATLAB untuk menyusun data yang sudah didapatkan menjadi format yang diketahui oleh MATLAB yang berbentuk *kernel*. Struktur `fitcnb` terdapat pada Gambar 4.21.

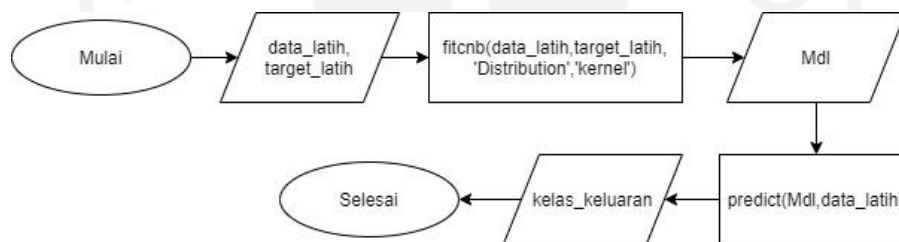

```

Mdl =
  ClassificationNaiveBayes
    ResponseName: 'Y'
    CategoricalPredictors: []
    ClassNames: {'Belum Matang' 'Matang'}
    ScoreTransform: 'none'
    NumObservations: 70
    DistributionNames: {'kernel' 'kernel' 'kernel' 'kernel'}
    DistributionParameters: {2x4 cell}
      Kernel: {'normal' 'normal' 'normal' 'normal'}
      Support: {'unbounded' 'unbounded' 'unbounded' 'unbounded'}
      Width: [2x4 double]

```

Gambar 4.21 Struktur Fungsi fitncb Klasifikasi Naive Bayes

Dalam struktur fitncb tersebut dapat diketahui jika terdapat dua kelas klasifikasi yaitu Belum Matang dan Matang dengan NumObservations 70 menunjukkan jumlah data dalam data latih sebanyak 70 citra latih. Proses klasifikasi *Naive Bayes* terdapat pada Gambar 4.22.



Gambar 4.22 Flowchart Klasifikasi Naive Bayes

Nilai data_latih dan target_latih yang didapat pada tahap ekstraksi ciri digunakan sebagai parameter rujukan pakar. Sehingga penting dalam hal ini untuk menggunakan citra latih yang baik karena akan membantu pada tahap pengujian karena data pakar latih yang akan dijadikan rujukan sudah memiliki nilai yang sesuai dengan target latih. Variabel data_latih berisi nilai ekstraksi ciri GLCM untuk pepaya belum matang dan matang sedangkan target latih berisi klasifikasi kelas pepaya yaitu matang dan belum matang. Fungsi fitncb akan mengelompokkan data sesuai dengan kelasnya yang kemudian disimpan pada variabel Mdl. Untuk mengetahui berapa citra yang dapat diklasifikasikan sesuai dengan kategorinya dapat digunakan fungsi Predict yaitu untuk memprediksi peluang nilai ekstraksi ciri masuk ke dalam sebuah kelas baik kelas pepaya matang maupun pepaya belum matang. Hasil dari prediksi ini akan disimpan dalam variabel kelas_keluaran untuk digunakan pada proses pengujian dengan menggunakan *Confusion Matrix* dan *Area Under Curve* (AUC). Kode program klasifikasi *Naive Bayes* terdapat pada Gambar 4.23.

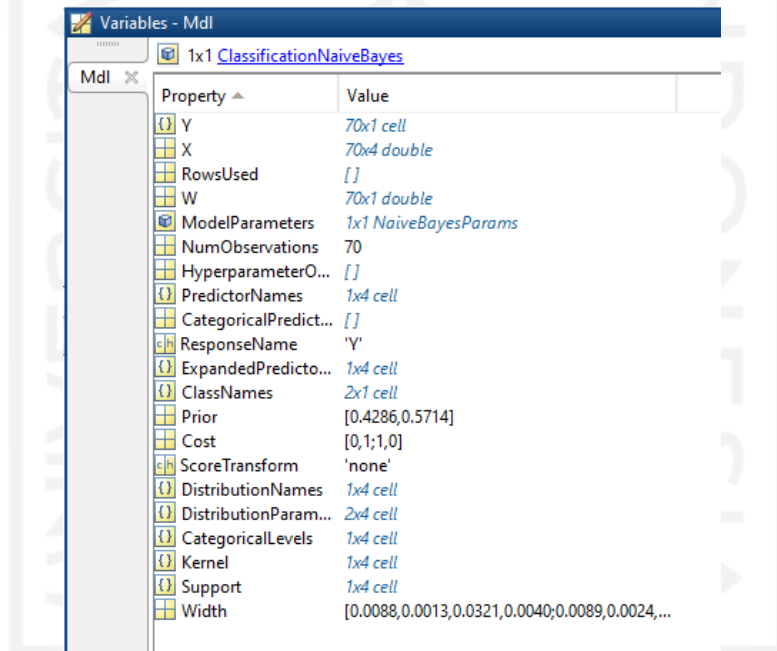
```

% menyusun data latih dan target latih
data_latih = [ciri_belum_matang; ciri_matang];
target_latih = [target_belum_matang; target_matang];
% klasifikasi naive bayes
Mdl = fitcnb(data_latih,target_latih,'Distribution','kernel');
% membaca kelas keluaran klasifikasi
kelas_keluaran = predict(Mdl,data_latih);
% menyimpan model Naive Bayes hasil pelatihan
save Mdl Mdl

```

Gambar 4.23 Kode Program Klasifikasi *Naive Bayes*

Dalam klasifikasi *Naive Bayes* menggunakan program MATLAB akan disimpan dalam bentuk *database* dalam format *.mat*. Dimana terdapat nilai Y untuk target citra berupa matang dan belum matang serta nilai X yang berisi nilai ekstraksi ciri GLCM untuk setiap citra yang dimasukkan. Untuk mengetahui struktur *database* terdapat pada Gambar 4.24.



Property	Value
Y	70x1 cell
X	70x4 double
RowsUsed	[]
W	70x1 double
ModelParameters	1x1 NaiveBayesParams
NumObservations	70
HyperparameterO...	[]
PredictorNames	1x4 cell
CategoricalPredict...	[]
ResponseName	'Y'
ExpandedPredicto...	1x4 cell
ClassNames	2x1 cell
Prior	[0.4286,0.5714]
Cost	[0,1;1,0]
ScoreTransform	'none'
DistributionNames	1x4 cell
DistributionParam...	2x4 cell
CategoricalLevels	1x4 cell
Kernel	1x4 cell
Support	1x4 cell
Width	[0.0088,0.0013,0.0321,0.0040;0.0089,0.0024,...

Gambar 4.24 Struktur *Database Mdl*

Untuk lebih memahami cara kerja klasifikasi *Naive Bayes* berikut adalah urutan proses klasifikasi *Naive Bayes*:

- Mendapatkan nilai ciri yaitu berupa nilai ekstraksi ciri GLCM yang terdiri dari *Contrast*, *Correlation*, *Energy*, dan *Homogeneity*. Sedangkan untuk variabel keduanya dibutuhkan nilai target yaitu belum matang dan matang.
- Menghitung nilai probabilitas total kelas yaitu berapa kali munculnya nilai kelas matang atau belum matang dalam keseluruhan data kelas. Dalam penelitian ini probabilitas untuk data latih pada kelas pepaya belum matang adalah 0.428571429 sedangkan probabilitas

untuk pepaya matang adalah 0.571428571. Nilai probabilitas dapat dilihat pada persamaan 4.5.

$$\text{Probabilitas Kelas Total} = \frac{\text{Munculnya nilai matang atau belum matang}}{\text{jumlah total data set}} \quad (4.5)$$

- c. Menghitung nilai *Mean* dari semua ciri yang didapat dari rata-rata ciri *Contrast*, *Correlation*, *Energy*, dan *Homogeneity*. Nilai *Mean* merupakan rata-rata dari setiap kelas matang dan belum matang pada setiap nilai ciri ekstraksi GLCM. Persamaan perhitungan nilai *Mean* terdapat berikut ini. Nilai *Mean* dapat dilihat pada persamaan 4.6.

$$\text{Mean} = \frac{\text{Jumlah nilai tiap ciri ekstraksi}}{\text{jumlah total data set}} \quad (4.6)$$

Dalam penelitian ini nilai *Mean* terdapat pada Tabel 4.3.

Tabel 4.3 Nilai *Mean* Ekstraksi Ciri GLCM

Kelas	<i>Contrast</i>	<i>Correlation</i>	<i>Energy</i>	<i>Homogeneity</i>
Belum Matang	0.058591461	0.986785222	0.304061483	0.974125871
Matang	0.069129392	0.980779384	0.305067632	0.968026912

- d. Menghitung nilai standar deviasi dari setiap ciri ekstraksi GLCM pada setiap set data latih. Nilai standar deviasi yang didapat masing-masing sesuai dengan ciri *Contrast*, *Correlation*, *Energy*, dan *Homogeneity* akan digunakan sebagai normalisasi nilai *Normal Distance*. Nilai standar deviasi akan terbagi sesuai dengan kelas yang ada, dalam penelitian ini terdapat dua nilai deviasi untuk kelas belum matang dan matang. Hasil perhitungan standar deviasi terdapat berikut ini. Persamaan untuk mencari nilai standar deviasi terdapat berikut ini. Nilai standar Deviasi dapat dilihat pada persamaan 4.7.

$$\text{Standar Deviasi} = \sqrt{\frac{\sum (x_i - \bar{x})^2}{(n - 1)}} \quad (4.7)$$

Keterangan :

n = jumlah sampel

\bar{x} = rata-rata

x_i = nilai x ke i

Tabel 4.4 Nilai Standar Deviasi

Kelas	<i>Contrast</i>	<i>Correlation</i>	<i>Energy</i>	<i>Homogeneity</i>
Belum matang	0.026191618	0.002476594	0.11270125	0.010497523
Matang	0.015520845	0.00552406	0.109023087	0.006687424

- e. Menghitung nilai distribusi normal untuk mengembalikan distribusi normal untuk rata-rata dan simpangan baku terhadap parameter yang dicari sesuai dengan ciri *Contrast*, *Correlation*, *Energy*, *Homogeneity* dengan nilai *Mean* dan nilai standar deviasi. Untuk nilai distribusi normal akan berbeda untuk tiap baris data yang ada menyesuaikan nilai dari tiap ciri ekstraksi GLCM. Persamaan untuk mencari nilai distribusi normal terdapat pada persamaan 4.8.

$$DN = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.8)$$

Keterangan:

σ : standar deviasi

μ : *mean* (rata-rata)

e : konstanta bilangan *euler* (2,178)

π : konstanta pi

x : nilai variabel acak x

- f. Mendapatkan nilai probabilitas tiap citra ciri yang dicari dengan mengalikan nilai probabilitas kemunculan dengan semua nilai hasil perhitungan *Normal Distance* untuk semua kelas yang ada yaitu kelas belum matang dan matang. Untuk nilai probabilitas tiap citra akan berbeda pada tiap baris data yang ada menyesuaikan nilai dari tiap ciri ekstraksi GLCM. Nilai probabilitas yang lebih besar akan menentukan hasil klasifikasi data, jika nilai probabilitas kelas matang lebih besar maka citra akan diklasifikasikan ke kelas

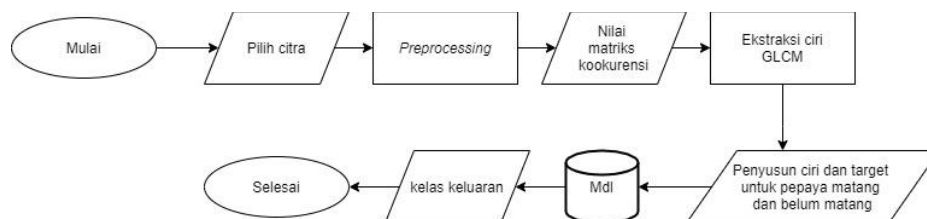
matang begitu pula sebaliknya. Persamaan untuk mencari nilai distribusi normal terdapat pada persamaan 4.9.

$$\begin{aligned} \text{Probabilitas} = & \text{probabilitas kelas total} \times \text{DN Contrast} \\ & \times \text{DN Correlation} \times \text{DN Energy} \times \text{DN Homogeneity} \end{aligned} \quad (4.9)$$

- g. Mendapatkan nilai klasifikasi dengan membandingkan nilai yang lebih besar antara hasil perhitungan probabilitas kelas pepaya belum matang dan matang. Hasil yang didapat paling besar menunjukkan hasil klasifikasi kelas hasil analisis citra.

4.1.2 Tahap Pengujian

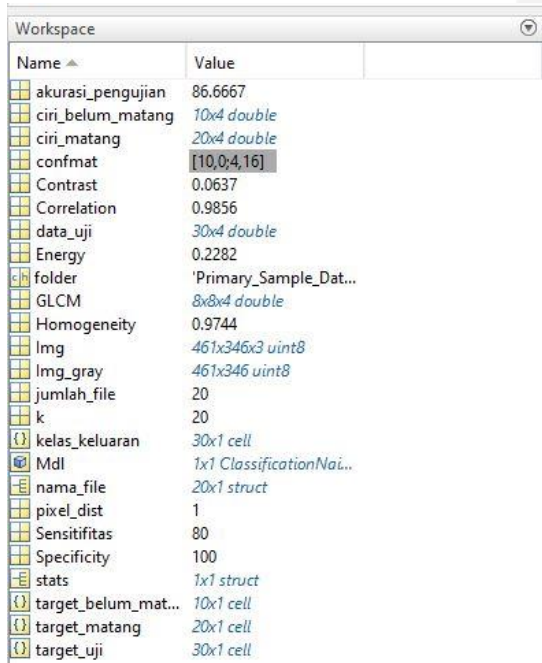
Tahap pengujian merupakan bagian kedua setelah tahap pelatihan di mana data pakar yang sudah ada digunakan sebagai nilai rujukan. Tahap yang dilalui dalam bagian pengujian ini memiliki beberapa persamaan dengan tahap pelatihan yang membedakan adalah jika pada tahap pelatihan membentuk nilai pakar sebagai rujukan pada tahap pengujian hanya akan menggunakan nilai pakar yang sudah ada untuk proses klasifikasi data. Tahap pengujian ini digunakan untuk mengetahui seberapa efektif metode yang digunakan untuk klasifikasi data. Dalam penelitian ini menggunakan 30 citra uji baik untuk pepaya California dan Thailand dengan kondisi matang dan belum matang. Sistem harus bisa mengklasifikasikan data citra sesuai keadaan riil setidaknya dengan persentase 80% sehingga metode yang digunakan dapat diimplementasikan dalam program utama. Beberapa tahapan seperti pilih citra, *preprocessing*, pembentukan matriks kookurensi, ekstraksi ciri GLCM, penyusunan data ciri dan target serta kelas keluaran memiliki tahap dan baris kode yang sama dengan tahap pelatihan sehingga penulis tidak uraikan lagi dalam tahap pengujian ini. Tahap yang berbeda adalah pada penggunaan data latih digunakan untuk proses klasifikasi jika pada tahap pelatihan menggunakan *folder* data latih sedangkan pada tahap pengujian menggunakan *folder* data uji. Proses pada tahap pengujian terdapat pada Gambar 4.25.



Gambar 4.25 Flowchart Tahap Pengujian

a. Penggunaan Data Latih

Perlu diperhatikan dalam penggunaan data latih adalah penempatan lokasi *file database* harus berada dalam *folder* yang sama dengan *file* modul MATLAB agar dapat terdeteksi oleh sistem. Pemanggilan data pakar rujukan menggunakan fungsi load dari MATLAB. Data pakar yang sudah berhasil dimuat dapat digunakan untuk klasifikasi data pengujian dengan mengolah data yang ada antara hasil dari masukan tahap pengujian berupa *data_uji* terdiri dari ciri dan target uji. Ciri uji terdiri dari nilai hasil ekstraksi GLCM sedangkan target terdiri dari nilai kelas pepaya belum matang dan matang. Untuk struktur variabel dari tahap pengujian terdapat pada Gambar 4.26.



Name	Value
akurasi_pengujian	86.6667
ciri_belum_matang	10x4 double
ciri_matang	20x4 double
confmat	[10,0;4,16]
Contrast	0.0637
Correlation	0.9856
data_uji	30x4 double
Energy	0.2282
folder	'Primary_Sample_Dat...
GLCM	8x8x4 double
Homogeneity	0.9744
Img	461x346x3 uint8
Img_gray	461x346 uint8
jumlah_file	20
k	20
kelas_keluaran	30x1 cell
Mdl	1x1 ClassificationNai...
nama_file	20x1 struct
pixel_dist	1
Sensitifitas	80
Specificity	100
stats	1x1 struct
target_belum_mat...	10x1 cell
target_matang	20x1 cell
target_uji	30x1 cell

Gambar 4.26 Struktur Variabel Data Tahap Pengujian

Untuk kode program pemanggilan data pakar (*database*) menggunakan fungsi load MATLAB terdapat pada Gambar 4.27.

```
% memanggil model naive bayes hasil pelatihan
load Mdl
% membaca kelas keluaran klasifikasi
kelas_keluaran = predict(Mdl,data_uji);
```

Gambar 4.27 Kode Program Pemanggilan Data Pakar

h. Hasil Tahap Pengujian

Dalam tahap pengujian diharapkan metode klasifikasi *Naive Bayes* dapat mengklasifikasikan kelas kematangan pepaya sesuai dengan nilai rujukan pakar. Semakin dominan kelas yang terklasifikasi dengan benar semakin baik pula metode klasifikasi yang digunakan untuk dapat diimplementasikan ke dalam sistem. Penentuan kelas didapatkan dari nilai rujukan pakar yang ada dalam *database* Mdl dengan menggunakan nilai parameter ekstraksi ciri GLCM berupa *Contrast*, *Correlation*, *Energi* dan *Homogeneity*. Hasil klasifikasi menggunakan data uji terdapat pada Tabel 4.5.

Tabel 4.5 Hasil Tahap Pengujian

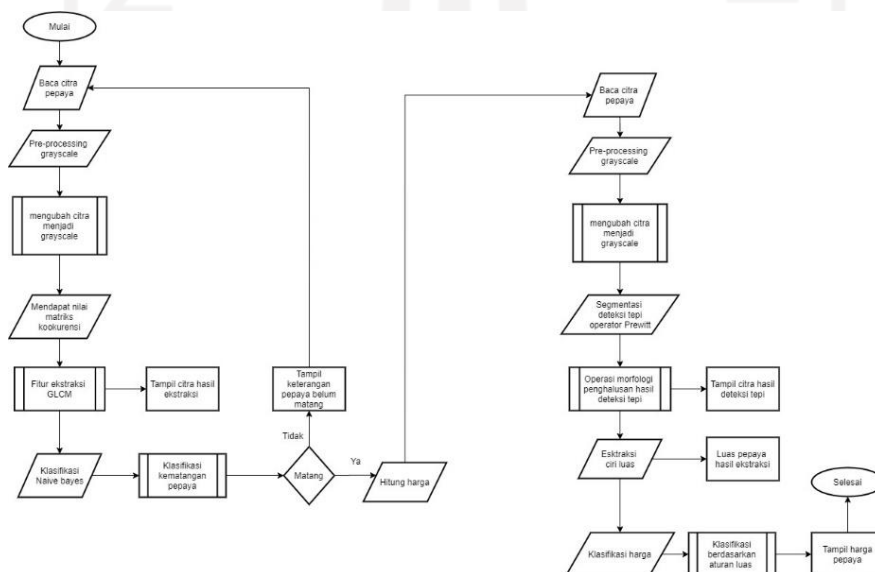
No	Nama Buah	Ekstraksi Ciri GLCM				Hasil Klasifikasi <i>Naive Bayes</i>
		<i>Contrast</i>	<i>Correlation</i>	<i>Energi</i>	<i>Homogeneity</i>	
1	California_mata ng (1).jpg	0.03673	0.985841	0.560189	0.982699	Benar
2	California_mata ng (2).jpg	0.042068	0.981713	0.427064	0.979912	Benar
3	California_mata ng (3).jpg	0.040904	0.983405	0.442569	0.980149	Salah
4	California_mata ng (4).jpg	0.040489	0.982275	0.362722	0.98043	Benar
5	California_mata ng (5).jpg	0.046838	0.984757	0.41035	0.978595	Benar
6	California_mata ng (6).jpg	0.10258	0.98968	0.270431	0.958474	Benar
7	California_mata ng (7).jpg	0.070469	0.987324	0.262636	0.969448	Salah
8	California_mata ng (8).jpg	0.061503	0.989355	0.265606	0.973334	Benar
9	California_mata ng (9).jpg	0.066452	0.988944	0.256258	0.972006	Benar
10	California_mata ng (10).jpg	0.068355	0.98805	0.261515	0.969301	Benar

11	Thailand_matang (1).jpg	0.049305	0.981373	0.361105	0.975911	Benar
12	Thailand_matang (2).jpg	0.060003	0.979652	0.326911	0.970806	Benar
13	Thailand_matang (5).jpg	0.058908	0.982502	0.299286	0.972257	Benar
14	Thailand_matang (6).jpg	0.043715	0.98145	0.488118	0.978867	Benar
15	Thailand_matang (7).jpg	0.049979	0.978113	0.447603	0.975612	Benar
16	Thailand_matang (8).jpg	0.061409	0.980018	0.323614	0.970263	Benar
17	Thailand_matang (9).jpg	0.065185	0.977417	0.322875	0.968217	Salah
18	Thailand_matang (10).jpg	0.046639	0.986463	0.350234	0.97747	Salah
19	Thailand_matang (11).jpg	0.06555	0.973282	0.344296	0.967978	Benar
20	Thailand_matang (12).jpg	0.057081	0.97926	0.362941	0.972438	Benar
21	California_belumatang (1).jpg	0.104508	0.983873	0.191465	0.954373	Benar
22	California_belumatang (2).jpg	0.055035	0.986589	0.225492	0.978336	Benar
23	California_belumatang (3).jpg	0.082651	0.975421	0.226692	0.96059	Benar
24	California_belumatang (4).jpg	0.072868	0.985269	0.199565	0.965678	Benar
25	California_belumatang (5).jpg	0.084911	0.983861	0.181436	0.960068	Benar
26	Thailand_belumatang (1).jpg	0.062069	0.984367	0.217714	0.970459	Benar
27	Thailand_belumatang (2).jpg	0.071656	0.977006	0.222355	0.965734	Benar

28	Thailand_belum matang (3).jpg	0.089836	0.977637	0.264944	0.958647	Benar
29	Thailand_belum matang (4).jpg	0.086131	0.984985	0.275817	0.961032	Benar
30	Thailand_belum matang (5).jpg	0.063659	0.985582	0.228154	0.974416	Benar

4.1.3 Program Utama

Berbeda dengan tahap-tahap sebelumnya, dalam tahap program utama penggunaan data citra sudah tidak menggunakan metode analisis *folder* namun menggunakan metode *single file* yang dipilih oleh pengguna. Dalam tahap program utama akan terlihat lebih jelas bagaimana tahapan klasifikasi berlangsung serta penampilan perubahan citra dari RGB menjadi *grayscale*. Beberapa proses pada tahap program utama memiliki cara kerja yang sama pada tahap pelatihan dan pengujian yang membedakan adalah pada tahap ini menggunakan metode *single file* dengan menggunakan bantuan *handles* pada MATLAB untuk pertukaran data pada tiap prosesnya. Pendefinisian nilai pada *handles* memudahkan pemanggilan data yang diinginkan sesuai dengan lokasi *file* yang benar. Proses perhitungan luas obyek pepaya untuk menentukan prediksi harga jual pepaya akan diimplementasikan dalam tahap ini dengan menggunakan metode deteksi tepi dengan menggunakan operator *Prewitt*. Alur proses pada tahap pengujian terdapat pada Gambar 4.28.



Gambar 4.28 Flowchart Program Utama

a. Pilih Citra

Pemilihan citra jika sebelumnya menggunakan sistem penentuan *folder* untuk citra latih dan citra uji pada tahap program utama citra dipilih dengan mekanisme *single file* dengan memilih satu citra yang akan dianalisis. Citra yang dapat dianalisis merupakan citra dengan format *file* berjenis *.jpg*. Dalam melakukan pembacaan citra pepaya menggunakan fungsi yang disediakan oleh MATLAB yaitu dengan *imread*. Setelah dipilih lokasi *file* tahap berikutnya adalah melakukan *resize* citra, seperti pada tahap pelatihan dan pengujian *resize* citra bertujuan untuk mengecilkan ukuran citra. Dalam menyimpan variabel sementara menggunakan fungsi *handles* yang dapat dipanggil oleh bagian lain. Karakteristik ini menggunakan metode *Object Oriented Programming* di mana setiap fungsi atau bagian dapat dipanggil dan saling terhubung satu sama lain. Setelah deskripsi *file* citra didapat maka citra yang dipilih ditampilkan dengan fungsi *imshow* yang akan menampilkan citra pada petak GUI dengan lokasi *axes1*. Kode program untuk pemilihan citra *file* tunggal terdapat pada Gambar 4.29.

```
function pushbutton1_Callback(hObject, eventdata, handles)
[nama_file, nama_path] = uigetfile('*.jpg');
Img = imread(fullfile(nama_path,nama_file));
Img = imresize(Img,0.1);
% menampilkan nama file pada edit text
set(handles.edit4,'String',nama_file)
% menampilkan citra rgb pada axes
axes(handles.axes1)
imshow(Img)
```

Gambar 4.29 Kode Program Pilih Citra *Single File*

b. Preprocessing

Citra pepaya yang sudah disimpan dalam *handles* dapat dipanggil kembali untuk kemudian dilakukan proses konversi citra dari RGB ke *grayscale* untuk pembuatan matriks kookurensi. Kode program *preprocessing* terdapat pada Gambar 4.30.

```
% memanggil variabel yg ada di lokasi handles
Img = handles.Img;
% mengkonversi cira rgb menjadi grayscale
Img_gray = rgb2gray(Img);
% pembuatan matriks kookurensi
```

Gambar 4.30 Kode Program *Preprocessing* Program Utama

c. Pembentukan Matriks Kookurensi dan Ekstraksi Ciri GLCM

Langkah pembentukan matriks kookurensi sama seperti pada tahap pelatihan dan pengujian, hanya saja pada tahap program utama pembentukan matriks kookurensi hanya untuk satu citra tunggal yang akan langsung dihitung nilai ekstraksi ciri GLCM. Kode program untuk pembentukan nilai matriks kookurensi dan ekstraksi ciri GLCM terdapat pada Gambar 4.31.

```

% memanggil variabel energi ada di lokasi handles
Img = handles.Img;
% mengkonversi cira rgb menjadi grayscale
Img_gray = rgb2gray(Img);
% pembuatan matriks kookurensi
pixel_dist = 1;
GLCM = graycomatrix(Img_gray,'Offset',[0 pixel_dist;...
    -pixel_dist pixel_dist; -pixel_dist 0;...
    -pixel_dist -pixel_dist]);
% ekstraksi ciri GLCM
stats = graycoprops(GLCM,{'all'});
Contrast = mean(stats.Contrast);
Correlation = mean(stats.Correlation);
Energi = mean(stats.Energy);
Homogeneity = mean(stats.Homogeneity);

```

Gambar 4.31 Kode Program Pembentukan Matriks Kookurensi dan Ekstraksi Ciri GLCM

Terdapat dua variabel yang digunakan dalam menyimpan nilai hasil ekstraksi GLCM yaitu *ciri_pepaya* dan *data_pepaya*. Variabel *ciri_pepaya* digunakan sebagai parameter perhitungan pada proses klasifikasi data sedangkan *data_pepaya* digunakan sebagai nilai bantu untuk menampilkan data ke dalam GUI MATLAB. Tabel kosong pada GUI akan menampilkan variabel *data_pepaya* dengan bentuk dua kolom. Kolom pertama akan berisi nama ciri ekstraksi dari variabel *data_pepaya* yang digunakan berupa *Contrast*, *Correlation*, *Energy* dan *Homogeneity* sedangkan kolom kedua akan berisi nilai ekstraksi berupa bilangan numerik. Citra hasil konversi *grayscale* akan ditampilkan dalam GUI MATLAB dengan memanggil *handles* yang ditampilkan dalam petak GUI bagian *axes2* dengan fungsi *imshow*. Kode program penyusunan variabel terdapat pada Gambar 4.32.

```

% menyusun variabel ciri_pepaya
ciri_pepaya = zeros(1,4);
ciri_pepaya(1,1) = Contrast;
ciri_pepaya(1,2) = Correlation;
ciri_pepaya(1,3) = Energi;
ciri_pepaya(1,4) = Homogeneity;
% menyusun variabel data_pepaya
data_pepaya = cell(4,2);
data_pepaya{1,1} = 'Contrast';

```

```

data_pepaya{2,1} = 'Correlation';
data_pepaya{3,1} = 'Energi';
data_pepaya{4,1} = 'Homogeneity';
data_pepaya{1,2} = Contrast;
data_pepaya{2,2} = Correlation;
data_pepaya{3,2} = Energi;
data_pepaya{4,2} = Homogeneity;
% menampilkan citra grayscale pada axes
axes(handles.axes2)
imshow(Img_gray)

```

Gambar 4.32 Kode Program Penyusun Variabel

Untuk memberi informasi kepada pengguna berupa nilai hasil ekstraksi ciri dalam bilangan numerik maka variabel `data_pepaya` ditampilkan dalam GUI dengan bentuk tabel dengan lokasi tabel `uitable1`. Setelah proses perhitungan nilai ekstraksi ciri GLCM maka tombol `pushbutton3` menjadi aktif dan dapat digunakan untuk tahap berikutnya. Kode program untuk menampilkan hasil ekstraksi GLCM terdapat pada Gambar 4.33.

```

% menampilkan data_pepaya pada tabel
set(handles.uitable1,'Data',data_pepaya,'RowName',1:4)
% menyimpan variabel pada lokasi handles agar dapat dipanggil oleh
% pushbutton yg lain
handles.ciri_pepaya = ciri_pepaya;
guidata(hObject, handles)
% mereset GUI
set(handles.pushbutton3,'Enable','on')

```

Gambar 4.33 Kode Program Tampil Tabel Ekstraksi GLCM

d. Penggunaan Data Latih Klasifikasi *Naive Bayes*

Dengan menggunakan mekanisme analisis *single file* citra pepaya yang sudah dipilih akan diklasifikasi dengan menggunakan metode *Naive Bayes*. Dalam melakukan klasifikasi terlebih dahulu dilakukan pemanggilan *database* pakar dengan fungsi `load`. Data pakar yang sudah terpanggil akan digunakan sebagai parameter dengan mengklasifikasikan data antara ciri citra *single file* yang sudah dipilih dengan data pakar yang sudah ada menggunakan fungsi `predict` yang sudah disediakan oleh MATLAB. Hasil klasifikasi *Naive Bayes* akan ditampilkan dalam GUI MATLAB dalam bentuk data *String* pada bagian `edit1` untuk menampilkan teks berupa pepaya matang dan pepaya belum matang. Jika hasil klasifikasi pepaya memiliki hasil pepaya belum matang maka tidak bisa lanjut ke tahap berikutnya karena pepaya yang belum matang tidak memiliki nilai jual, namun jika hasil keluaran klasifikasi adalah pepaya matang dapat melanjutkan ke tahap berikutnya yaitu segmentasi deteksi tepi untuk mendapatkan luas obyek pepaya yang sedang dianalisa. Kode program klasifikasi *Naive Bayes* terdapat pada Gambar 4.34.

```

% memanggil model naive bayes hasil pelatihan
load Mdl

% membaca kelas keluaran klasifikasi
kelas_keluaran = predict(Mdl,ciri_pepaya);

% menampilkan kelas keluaran klasifikasi pada edit text
set(handles.edit1,'String',kelas_keluaran)

```

Gambar 4.34 Kode Program Klasifikasi *Naive Bayes* Program Utama

e. Segmentasi Deteksi Tepi

Segmentasi deteksi tepi digunakan untuk memperoleh nilai luas dari obyek pepaya yang di analisa. Nilai luas yang diperoleh digunakan sebagai parameter penentuan harga jual pepaya yang akan terbagi dalam tiga tingkat yaitu kecil, sedang dan besar. Segmentasi deteksi dilakukan dengan menggunakan citra awal yang disimpan dalam handles yang akan dikonversi menjadi citra *grayscale*. Langkah perhitungan deteksi tepi dengan operator *Prewitt* adalah sebagai berikut (Romindo & Khairina, 2017).

1. Konversi citra RGB ke *grayscale*.
2. Operasi konvolusi citra *grayscale* dengan *kernel* horizontal P_x untuk semua nilai piksel citra yang ada. Operator *Prewitt* untuk *kernel* horizontal dapat dilihat pada persamaan 4.10.

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.10)$$

3. Operasi konvolusi citra *grayscale* dengan *kernel* vertikal P_y untuk semua nilai piksel citra yang ada. Operator *Prewitt* untuk *kernel* vertikal P_y dapat dilihat pada persamaan 4.11.

$$P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.11)$$

4. Perhitungan nilai gradien dimana P_x dan P_y adalah hasil konvolusi *kernel* horizontal dan vertikal sebelumnya. Nilai gradien P_x dan P_y dapat dicari dengan menggunakan persamaan 4.12.

$$G = \sqrt{P_x^2 + P_y^2} \quad (4.12)$$

5. Melakukan proses perhitungan ambang batas *threshold*

Nilai *threshold* dapat ditentukan secara manual atau menggunakan sebuah algoritma *thresholding* yang dapat menghitung nilai secara otomatis, namun secara sederhana metode yang bisa digunakan dengan nilai rata-rata atau nilai tengah dengan asumsi jika piksel obyek lebih terang daripada *background*, piksel tersebut juga lebih terang dari rata-rata *background* tersebut (Bhahri & Rachmat, 2018). Dalam penelitian ini pendekatan nilai rata-rata dipilih dengan pertimbangan *background* dan obyek pepaya memiliki perbedaan yang cukup signifikan sehingga nilai rata-rata sudah bisa membedakan obyek pepaya dengan *background*. Pencarian nilai *threshold* dapat dilihat pada persamaan 4.13.

$$\text{Threshold} = \text{rata} - \text{rata gradien} \quad (4.13)$$

Untuk memudahkan penggunaan nilai ambang batas maka dilakukan inisialisasi nilai *threshold* dengan menggunakan citra sampel, hasil perhitungan nilai *threshold* untuk citra sampel ini akan digunakan untuk nilai mutlak *threshold* dalam sistem. Data citra yang digunakan sebagai sampel dapat dilihat pada Gambar 4.35.



Gambar 4.35 Data Sampel Citra Pepaya

Dari nilai rata-rata hasil perhitungan gradien data sampel citra pepaya tersebut didapatkan nilai 0.109358873 yang kemudian disederhanakan menjadi 0.1, nilai inilah yang menjadi nilai mutlak *threshold*.

6. Normalisasi nilai piksel citra sesuai dengan *threshold* di mana dalam penelitian ini menggunakan *threshold* 0.1, jika nilai dibawah *threshold* maka nilai piksel akan menjadi 0 sedangkan jika nilai piksel citra di atas *threshold* maka akan bernilai 1

Setelah diperoleh area hasil deteksi tepi dilanjutkan dengan operasi morfologi untuk menyempurnakan hasil deteksi tepi dengan fungsi MATLAB `imclearborder`. Fungsi

`imclearborder` digunakan untuk menghilangkan garis yang ada di luar batas citra dikarenakan garis ini tidak akan digunakan. Untuk lebih memaksimalkan hasil deteksi tepi fungsi `bwareopen` untuk menghilangkan objek-objek kecil yang tidak digunakan dengan melakukan proses *noise removal* dengan cara menghilangkan area yang memiliki luasan kurang dari 100 pixel sedangkan fungsi `imfill` berguna untuk merekonstruksi citra untuk menghasilkan citra yang baik. Citra hasil deteksi tepi kemudian ditampilkan dengan menggunakan fungsi `imshow` pada petak GUI `axes3`. Dalam implementasi penulisan kode program pada MATLAB terdapat pada Gambar 4.36.

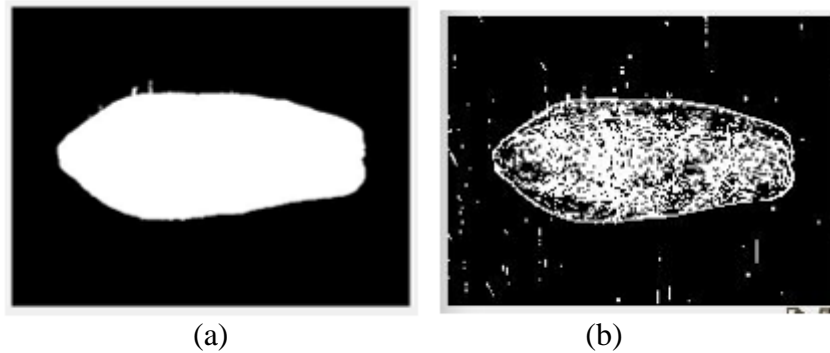
```
% memanggil variabel yg ada di lokasi handles
Img = handles.Img;
kelas_keluaran = handles.kelas_keluaran;
% mengkonversi cira rgb menjadi grayscale
Img_gray = im2double(rgb2gray(Img));
prewitthor = [-1 0 1; -1 0 1; -1 0 1];
prewittver = [-1 -1 -1; 0 0 0; 1 1 1];
Ix = conv2(Img_gray,prewitthor,'same');
Iy = conv2(Img_gray,prewittver,'same');
E = sqrt((Ix.^2)+(Iy.^2));
threshold = 0.1;
edgeImage = (E > threshold);
% operasi morfologi untuk menyempurnakan hasil deteksi tepi
bw = imclearborder(edgeImage);
bw = bwareaopen(bw,100);
bw = imfill(bw,'holes');
% menampilkan citra hasil deteksi tepi pada axes
axes(handles.axes3)
imshow(edgeImage);
```

Gambar 4.36 Kode Program Deteksi Tepi Metode *Prewitt*

f. Ekstraksi Ciri Luas

Dalam tahap deteksi tepi telah diketahui nilai *threshold* serta area obyek yang memiliki nilai 1 dan 0. Piksel dengan nilai 1 menunjukkan bahwa piksel tersebut berada di dalam area obyek pepaya sedangkan piksel 0 merupakan piksel yang berada di luar area obyek pepaya atau *background*. Untuk mengetahui luas obyek pepaya dapat dengan menjumlahkan nilai piksel yang aktif sehingga akan merepresentasikan luas obyek pepaya yang ada. Hasil deteksi tepi dengan operator *Prewitt* belum menghasilkan obyek yang halus karena masih terdapat obyek piksel diluar area yang aktif. Fungsi `imclearborder` digunakan agar struktur garis tipis yang lebih muda warnanya dibandingkan dengan bagian citra *fill* obyek dan garis tersebut terhubung dengan batas luar citra akan dihilangkan. Untuk menyempurnakan hasil identifikasi luas obyek operasi fungsi dengan menggunakan `bwareopen` yang sudah disediakan oleh MATLAB yaitu dengan menghilangkan obyek yang memiliki intensitas piksel di bawah 100 sehingga

titik-titik piksel aktif diluar obyek dapat hilang sehingga diperoleh nilai piksel murni dari obyek pepaya tersebut. Nilai hasil perhitungan fungsi *bwareaopen* akan disempurnakan dengan rekonstruksi citra menggunakan fungsi *imfill*, hasil inilah yang akan digunakan untuk memperoleh luas obyek citra pepaya. Perbedaan citra hasil operasi fungsi *bwareaopen* dengan citra hasil deteksi tepi operator *Prewitt* dapat dilihat pada Gambar 4.37.



Gambar 4.37 Representasi Luas Obyek Hasil Operasi Fungsi *bwareaopen* (a)
Representasi Obyek Hasil Deteksi Tepi *Prewitt* (b)

Dalam representasi di atas diperoleh hasil citra yang baik dimana hanya terdapat sedikit kebocoran piksel, kegunaan fungsi *bwareaopen* akan mengurangi kebocoran piksel pada tepi citra jika citra data yang digunakan kurang baik sehingga masih diperoleh nilai luas yang merepresentasikan kondisi riilnya. Pada penelitian ini luas pepaya akan terbagi menjadi tiga jenis baik untuk pepaya California dan pepaya Thailand yaitu kecil, sedang dan besar. Untuk memperoleh nilai yang padu maka peneliti mencoba menghitung nilai dari sampel obyek pepaya dengan ukuran kecil, sedang dan besar untuk masing-masing jenis pepaya. Dalam setiap jenis pepaya California terdapat 15 sampel obyek dengan ukuran kecil, sedang dan besar sedangkan sampel untuk pepaya Thailand hanya 10 sampel obyek, jadi total terdapat 25 sampel data citra untuk kebutuhan data *range* ukuran pepaya. Pengelompokan kategori ini berdasarkan kondisi berat riil yang merepresentasikan ukuran pepaya. Setiap kategori besar pepaya akan dicari nilai rata-rata luas piksel untuk penentuan rentang nilai yang diimplementasikan ke dalam sistem yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 *Range* Ukuran Pepaya

No	Kategori	Sampel Citra					Rata-Rata
		Citra 1 (piksel)	Citra 2 (piksel)	Citra 3 (piksel)	Citra 4 (piksel)	Citra 5 (piksel)	
1	California Kecil 1,2 Kg – 1,3 Kg	9914	8719	4559	8007	9157	8071.2
2	California Sedang 1,4 Kg – 1,6 Kg	10076	11695	10519	10265	11677	10846.4
3	California Besar 1,6 Kg – 1,8 Kg	12653	13856	12053	14038	13603	13240.6
3	Thailand Kecil 1,6 Kg – 2,5 Kg	14724	15507	16034	15743	18073	16016.2
4	Thailand Sedang 2,5 Kg – 3 Kg	28975	29876	31542	32975	30489	30771.4
5	Thailand Besar >3 Kg	Lebih dari 35000 piksel					

Untuk pepaya dengan kategori Thailand besar tidak dicari nilai *range*-nya dengan asumsi pepaya dengan ukuran lebih dari 35000 akan masuk ke dalam kategori Thailand besar. Untuk pepaya California matang dengan kondisi paling besar memiliki ukuran di bawah pepaya jenis Thailand matang. Untuk perhitungan luas akan fokus pada pepaya dengan kondisi matang mengingat pepaya dengan kondisi belum matang tidak memiliki nilai jual. Pengelompokan kategori besar pepaya ini berdasarkan berat pepaya pada kondisi riil kemudian data citra diambil sebagai data sampel ukuran pepaya. Untuk memudahkan implementasi sistem maka digunakan sistem pembulatan hasil dari rentang nilai luas pepaya untuk ukuran kecil, sedang dan besar terdapat pada Tabel 4.7.

Tabel 4.7 Rentang Nilai Luas Obyek Pepaya

No	Pepaya	Kecil (piksel)	Sedang (piksel)	Besar (piksel)
1	California	< 10000	10000-12000	12000-15000
2	Thailand	15000-30000	30000-35000	>35000

Nilai rerata yang didapat digunakan sebagai parameter penentuan harga jual pepaya dengan menggunakan harga jual pada bulan Februari 2021 dari distributor pepaya di Desa

Kepong. Pepaya California dengan ukuran kecil memiliki harga Rp.2000, sedang Rp.2500 dan besar Rp.3000 sedangkan pepaya jenis Thailand memiliki harga Rp.2000 untuk ukuran kecil, Rp.3000 untuk ukuran sedang dan Rp.5000 untuk ukuran besar. Untuk pepaya dengan hasil klasifikasi belum matang tidak akan muncul prediksi harga jualnya dikarenakan pepaya yang belum matang tidak memiliki nilai jual. Hasil prediksi harga jual akan ditampilkan dalam GUI MATLAB dengan tipe data *String* pada edit3. Kode program untuk ekstraksi luas dan prediksi harga terdapat pada Gambar 4.38.

```
% menghitung luas
luas = sum(sum(bw));
% menampilkan luas pepaya pada edit text
set(handles.edit2,'String',[num2str(luas),' px'])
% klasifikasi ukuran
if isequal(kelas_keluaran{1},'Matang')
    if luas < 4000
        harga = 'Rp 2.000,-'; % cali kecil
    elseif (luas < 5000 && luas >=4001)
        harga = 'Rp 2.100,-'; % cali kecil
    elseif (luas < 6000 && luas >=5001)
        harga = 'Rp 2.200,-'; % cali kecil
    elseif (luas < 7000 && luas >=6001)
        harga = 'Rp 2.300,-'; % cali kecil
    elseif (luas < 8000 && luas >=7001)
        harga = 'Rp 2.400,-'; % cali sedang
    elseif (luas < 9000 && luas >=8001)
        harga = 'Rp 2.500,-'; % cali sedang
    elseif (luas < 10000 && luas >=9001)
        harga = 'Rp 2.600,-'; % cali sedang
    elseif (luas < 11000 && luas >=10001)
        harga = 'Rp 2.700,-'; % cali sedang
    elseif (luas < 12000 && luas >=11001)
        harga = 'Rp 2.800,-'; % cali besar
    elseif (luas < 13000 && luas >=12001)
        harga = 'Rp 2.900,-'; % cali besar
    elseif (luas < 14000 && luas >=13001)
        harga = 'Rp 3,000,-'; % cali besar
    elseif (luas < 15000 && luas >=14001)
        harga = 'Rp 3,100,-'; % cali besar
    %=====
    elseif (luas < 17142 && luas >= 15001)
        harga = 'Rp 2.000,-'; % thai kecil
    elseif (luas < 19284 && luas >= 17143)
        harga = 'Rp 2.250,-'; % thai kecil
    elseif (luas < 21426 && luas >= 19285)
        harga = 'Rp 2.500,-'; % thai kecil
    elseif (luas < 23568 && luas >= 21427)
        harga = 'Rp 2.750,-'; % thai kecil
    elseif (luas < 25711 && luas >= 23569)
        harga = 'Rp 3.000,-'; % thai sedang
    elseif (luas < 27854 && luas >= 25712)
        harga = 'Rp 3.250,-'; % thai sedang
    elseif (luas < 30000 && luas >= 27855)
        harga = 'Rp 3.500,-'; % thai sedang
    elseif (luas < 32000 && luas >= 30001)
        harga = 'Rp 3.750,-'; % thai sedang
    elseif (luas < 34000 && luas >= 32001)
        harga = 'Rp 4.000,-'; % thai sedang
    elseif (luas < 36000 && luas >= 34001)
        harga = 'Rp 4.250,-'; % thai besar
    elseif (luas < 38000 && luas >= 36001)
```

```

        harga = 'Rp 4.500,-'; % thai besar
    elseif (luas < 40000 && luas >= 38001)
        harga = 'Rp 4.750,-'; % thai besar
    else
        harga = 'Rp 5.000,-'; % thai besar
    end
else
    harga = 'Rp -';
end
% menampilkan harga pada edit text
set(handles.edit3,'String',harga)

```

Gambar 4.38 Kode Program Ekstraksi Luas dan Prediksi Harga

Penentuan nilai harga menggunakan rentang hasil pembulatan nilai luas sampel obyek pepaya pada Tabel 4.7. Agar menghasilkan variasi harga yang mendekati nilai riil maka digunakan rentang piksel untuk pepaya California sebesar 1000 piksel untuk setiap kenaikan harga Rp.100. Untuk pepaya Thailand setiap kenaikan 2141 piksel maka harga pepaya akan naik sebesar Rp.250. Nilai ini didapatkan dengan membagi rentang nilai masing-masing untuk pepaya Thailand dan pepaya California agar didapat kenaikan yang sama besar dan mendapatkan harga yang bervariasi. dari Hasil akhir dari klasifikasi pepaya akan menampilkan kelas kematangan buah pepaya baik untuk pepaya California maupun Thailand. Pepaya yang masuk ke dalam kelas matang akan memiliki harga jual yang ditampilkan pada GUI MATLAB sedangkan untuk pepaya yang memiliki hasil klasifikasi belum matang tidak akan dikeluarkan prediksi harga jualnya pada sistem karena pada kondisi riilnya pepaya yang belum matang tidak memiliki nilai jual. Rata-rata pepaya California memiliki bentuk serta bobot antar buah yang hampir mirip di mana pepaya California memiliki berat di rentang 1,2 Kg hingga 1,8 Kg dengan harga Rp.1500 per kilogramnya. Walaupun pepaya Thailand memiliki bentuk yang lebih besar dari jenis California namun memiliki harga jual yang lebih rendah yaitu Rp.1000 per kilogramnya. Pepaya Thailand memiliki berat buah di rentang 1,6 Kg hingga 3,3 Kg. Penggunaan data citra pada hasil akhir ini menggunakan citra pada *folder* data uji dengan jumlah citra sebanyak 30 citra baik untuk pepaya California ataupun pepaya Thailand. Hasil akhir klasifikasi kematangan dan harga pepaya terdapat pada Tabel 4.8.

Tabel 4.8 Hasil Akhir Klasifikasi

No	Nama Buah	Ukuran	Harga Sistem (Rupiah)	Harga Riil (Rupiah)	Klasifikasi <i>Naive Bayes</i>
1	California_matang (1).jpg	3810	2000	2250	Matang

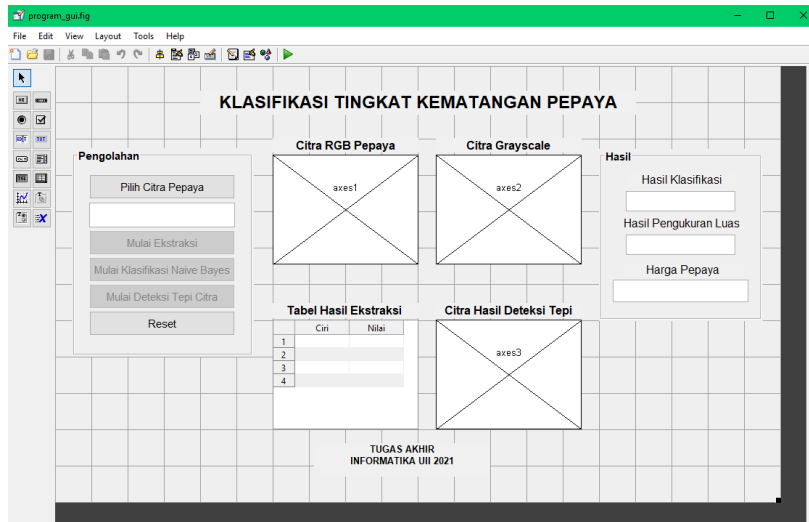
2	California_matang (2).jpg	7690	2400	2400	Matang
3	California_matang (3).jpg	4013	-	1950	Belum Matang
4	California_matang (4).jpg	3466	2000	2100	Matang
5	California_matang (5).jpg	4343	2100	1950	Matang
6	California_matang (6).jpg	4695	2100	2550	Matang
7	California_matang (7).jpg	10137	-	2700	Belum Matang
8	California_matang (8).jpg	9148	2600	1800	Matang
9	California_matang (9).jpg	8636	2500	1950	Matang
10	California_matang (10).jpg	4189	2100	1950	Matang
11	Thailand_matang (1).jpg	28783	3500	2300	Matang
12	Thailand_matang (2).jpg	10597	2700	2400	Matang
13	Thailand_matang (5).jpg	25665	3000	2300	Matang
14	Thailand_matang (6).jpg	24935	3000	2200	Matang
15	Thailand_matang (7).jpg	20975	2500	1800	Matang
16	Thailand_matang (8).jpg	21572	2750	1700	Matang
17	Thailand_matang (9).jpg	36199	-	1600	Belum Matang
18	Thailand_matang (10).jpg	35662	-	1700	Belum Matang
19	Thailand_matang (11).jpg	41527	5000	3300	Matang
20	Thailand_matang (12).jpg	43971	5000	3200	Matang
21	California_belummatang (1).jpg	11070	-	-	Belum Matang
22	California_belummatang (2).jpg	10978	-	-	Belum Matang
23	California_belummatang (3).jpg	10996	-	-	Belum Matang
24	California_belummatang (4).jpg	3104	-	-	Belum Matang
25	California_belummatang (5).jpg	2727	-	-	Belum Matang

26	Thailand_belummatang (1).jpg	27077	-	-	Belum Matang
27	Thailand_belummatang (2).jpg	18169	-	-	Belum Matang
28	Thailand_belummatang (3).jpg	8494	-	-	Belum Matang
29	Thailand_belummatang (4).jpg	6019	-	-	Belum Matang
30	Thailand_belummatang (5).jpg	21863	-	-	Belum Matang

Harga riil pada Tabel 4.8 diperoleh dari harga pepaya per kilogramnya dikalikan dengan berat pepaya pada saat ditimbang, sebagai pembanding harga sistem dengan harga riil. Dimana untuk harga pepaya California per kilogramnya adalah Rp.1500 sedangkan untuk pepaya Thailand adalah Rp.1000.

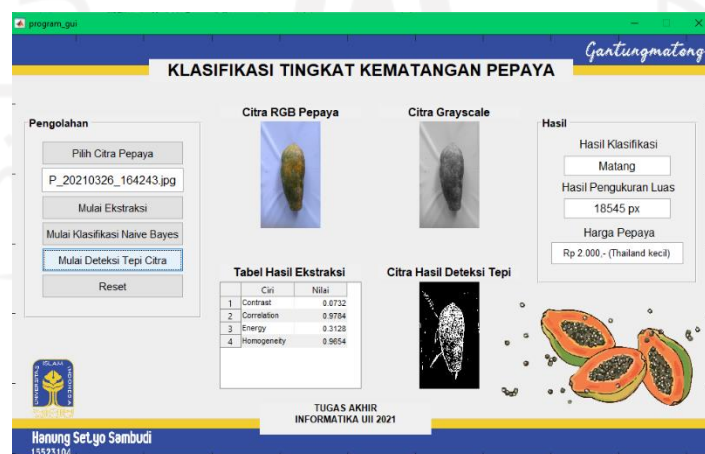
4.2 Implementasi Antarmuka

Graphics User Interface (GUI) berfungsi untuk menghubungkan program dengan pengguna untuk berkomunikasi. Implementasi antar muka dalam program akan merujuk pada rancangan antar muka yang sudah dibuat sebelumnya agar sistem yang dikembangkan lebih terstruktur dimulai dengan pemilihan bentuk dan fitur yang digunakan. Rancangan dalam sistem ini menggunakan *guide* bawaan dari MATLAB sehingga selain lebih efisien dapat diterapkan langsung ke dalam sistem tanpa harus merancang ulang *interface* yang sudah ada. Untuk dapat menampilkan *guide* dapat dengan mengetikkan perintah *guide* pada *command windows* MATLAB. Tampilan *guide* yang dimaksud terdapat pada Gambar 4.39.



Gambar 4.39 Rancangan Antar Muka Dengan *Guide*

Tampilan *user interface* terdiri dari empat panel utama untuk menampilkan informasi dan citra hasil pengolahan. Untuk proses navigasi antar tahap pengolahan citra terdapat panel navigasi pada sebelah kiri antarmuka dengan tombol yang akan aktif secara otomatis jika setiap tahap yang dibutuhkan sudah dipenuhi. Pada sisi kanan antarmuka terdapat panel hasil untuk menampilkan informasi hasil klasifikasi, luas area obyek serta prediksi harga pepaya. Dalam implementasi antarmuka akan menyesuaikan rancangannya yaitu hanya akan terdapat satu halaman saja yaitu halaman utama dengan panel utama, panel navigasi dan panel hasil. Implementasi final rancangan antarmuka pada sistem terdapat pada Gambar 4.40.



Gambar 4.40 Tampilan *User Interface* Sistem

4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui seberapa efektif sistem bekerja dan berhasil mencapai tujuan penelitian. Dalam penelitian ini terdapat dua bagian pengujian yaitu pengujian dengan metode *Confusion Matrix* dan pengujian dengan metode *Area Under Curve* (AUC). Pengujian *Confusion Matrix* bertujuan untuk mengetahui berapa akurasi sistem dalam mengklasifikasikan pepaya dengan benar, semakin tinggi nilai akurasi menunjukkan klasifikasi sistem semakin baik. Sedangkan pengujian *Area Under Curve* (AUC) bertujuan untuk mengetahui performa dari metode klasifikasi yang digunakan dalam penelitian ini adalah metode *Naive Bayes*, AUC memiliki nilai antara 0 hingga 1 di mana jika nilai semakin dekat dengan 1 maka metode yang digunakan memiliki performa yang baik.

4.3.1 Pengujian *Confusion Matrix*

Pengujian *Confusion Matrix* merupakan metode pengujian yang lazim digunakan untuk mengetahui efektivitas sebuah sistem. Bentuk *Confusion Matrix* terdapat pada Gambar 4.41.

		Nilai sebenarnya	
		True	False
Nilai prediksi	True	TP	FP
	False	FN	TN

Gambar 4.41 Model *Confusion Matrix*

Keterangan:

1. **True Positive (TP)** : Jumlah pepaya matang yang terklasifikasi dengan benar oleh sistem
2. **False Positive (FP)** : Jumlah pepaya dengan nilai target pakar belum matang yang terklasifikasi oleh sistem menjadi matang
3. **False Negative (FN)** : Jumlah pepaya dengan nilai target pakar matang yang terklasifikasi oleh sistem menjadi belum matang
4. **True Negative (TN)** : Jumlah pepaya belum matang yang terklasifikasi benar oleh sistem

Nilai *Confusion Matrix* digunakan untuk mengetahui nilai *Accuracy*, *Sensitivity* dan *Specificity*. *Accuracy* merupakan metode untuk mengetahui performa sistem dalam memprediksi data dengan benar (data yang diprediksi benar oleh sistem dibagi dengan seluruh

total data citra yang digunakan). Persamaan untuk mencari nilai *Accuracy*, *Sensitivity* dan *Specificity* dapat dilihat pada persamaan 4.13, 4.14 dan 4.15.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.13)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4.14)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (4.15)$$

Sensitivity digunakan untuk menghitung nilai persentase data positif yang teridentifikasi dengan benar (kematangan yang terdeteksi sistem sama dengan nilai rujukan pakar). Sedangkan *Specificity* merupakan nilai untuk mengukur persentase data negatif yang teridentifikasi dengan benar oleh sistem. Implementasi perhitungan *Confusion Matrix* pada MATLAB dapat menggunakan fungsi `confusionmat` dengan variabel yang digunakan adalah `target_latih` dan `kelas_keluaran`. Pada variabel `target_latih` berisi nilai rujukan pakar sedangkan `kelas_keluaran` adalah hasil klasifikasi oleh sistem. Kode program *Confusion Matrix* terdapat pada Gambar 4.42.

```
% membaca confusion marix
confmat = confusionmat(target_latih,kelas_keluaran)
% membaca akurasi pelatihan
akurasi_pelatihan = (confmat(1,1)+confmat(2,2))/...
    (confmat(1,1)+confmat(1,2)+confmat(2,1)+confmat(2,2))*100
Sensitifitas = confmat(2,2)/...
    (confmat(2,2)+confmat(2,1))*100
Specificity = confmat(1,1)/...
    (confmat(1,1)+confmat(1,2))*100
```

Gambar 4.42 Kode Program Pengujian *Confusion Matrix*

Untuk mengetahui konsistensi sistem dalam mengklasifikasikan data maka peneliti melakukan lima kali percobaan dengan *dataset* yang berbeda dan satu eksekusi data primer. Hasil dari enam kali percobaan terdapat sebagai berikut.

1. Pengujian 1 dengan menggunakan 80 citra latih dan 20 citra uji baik untuk citra pepaya California dan pepaya Thailand. Tabel 4.9 menunjukkan hasil percobaan satu.

Tabel 4.9 Hasil Percobaan Satu

Pengujian 1	Training	Testing	Confusion Matrix			
			Training		Testing	
Accuracy (%)	93.7500	75	9	1	7	3
Sensitivity (%)	94.2857	80	4	66	2	8
Specificity (%)	90	70				

2. Pengujian 2 dengan menggunakan 85 citra latih dan 15 citra uji baik untuk citra pepaya California dan pepaya Thailand. Tabel 4.10 menunjukkan hasil percobaan dua.

Tabel 4.10 Hasil Percobaan Dua

Pengujian 2	Training	Testing	Confusion Matrix			
			Training		Testing	
Accuracy (%)	78.8235	66.667	14	1	4	1
Sensitivity (%)	75.7143	60	17	53	4	6
Specificity (%)	93.333	80				

3. Pengujian 3 dengan menggunakan 60 citra latih dan 40 citra uji baik untuk citra pepaya California dan pepaya Thailand. Tabel 4.11 menunjukkan hasil percobaan tiga.

Tabel 4.11 Hasil Percobaan Tiga

Pengujian 3	Training	Testing	Confusion Matrix			
			Training		Testing	
Accuracy (%)	85.7143	76.6667	23	7	11	4
Sensitivity (%)	92.5000	80	3	37	3	12
Specificity (%)	76.6667	73.3333				

4. Pengujian 4 dengan menggunakan 80 citra latih dan 20 citra uji dengan keseluruhan citra pepaya jenis California. Tabel 4.12 menunjukkan hasil percobaan empat.

Tabel 4.12 Hasil Percobaan Empat

Pengujian 4	Training	Testing	<i>Confusion Matrix</i>			
			Training		Testing	
<i>Accuracy (%)</i>	86.667	82.5000	18	2	16	4
<i>Sensitivity (%)</i>	85	85	6	34	3	17
<i>Specificity (%)</i>	90	80				

5. Pengujian 5 dengan menggunakan 80 citra latih dan 20 citra uji dengan keseluruhan citra pepaya jenis Thailand. Tabel 4.13 menunjukkan hasil percobaan lima.

Tabel 4.13 Hasil Percobaan Lima

Pengujian 5	Training	Testing	<i>Confusion Matrix</i>			
			Training		Testing	
<i>Accuracy (%)</i>	91.2500	100	18	2	10	0
<i>Sensitivity (%)</i>	91.6667	100	5	55	0	10
<i>Specificity (%)</i>	90	100				

6. Pengujian dengan menggunakan data primer untuk penelitian dengan 70 citra latih dan 30 citra uji baik untuk citra pepaya California dan pepaya Thailand. Tabel 4.14 menunjukkan hasil percobaan enam.

Tabel 4.14 Hasil Percobaan Enam Data Utama

Pengujian 6	Training	Testing	<i>Confusion Matrix</i>			
			Training		Testing	
<i>Accuracy (%)</i>	82.8571	86.6667	25	5	10	0
<i>Sensitivity (%)</i>	82.5000	80	7	33	4	16
<i>Specificity (%)</i>	83.3333	100				

4.3.2 Pengujian Metode Klasifikasi Area Under Curve

Nilai AUC secara teoritis berada di antara 0 dan 1. Nilai AUC memberikan gambaran tentang keseluruhan pengukuran atas kesesuaian dari model yang digunakan. Semakin besar area under curve maka semakin baik variabel yang diteliti dalam memprediksi kejadian. Dalam penelitian ini nilai *Accuracy*, *Sensitivity* dan *Specificity* sebelumnya dalam satuan persen dikembalikan lagi ke dalam nilai normal dengan membagi setiap nilai dengan 100 untuk dapat diketahui nilai AUC pada setiap percobaan. Dalam menghitung nilai AUC dapat dengan menggunakan persamaan 4.16.

$$AUC = \frac{(1 + Sensitivity - FPR)}{2} \quad (4.16)$$

Nilai *False Positive Rate* (FPR) dapat dilihat pada persamaan 4.17.

$$FPR = 1 - Spesivicity \quad (4.17)$$

Dari keseluruhan percobaan yang telah dilakukan dapat terbentuk nilai UAC yang terdapat pada Tabel 4.15.

Tabel 4.15 Nilai UAC Percobaan Sistem

Percobaan 1		Percobaan 2		Percobaan 3		Percobaan 4	
Training	Testing	Training	Testing	Training	Testing	Training	Testing
0.92145	0.75	0.8452	0.7	0.84585	0.76665	0.875	0.825
Percobaan 5		Percobaan 6		Rata-rata			
Training	Testing	Training	Testing	Training	Testing		
0.9083	1	0.82915	0.91	0.87082	0.8236		

4.4 Penyebab Kegagalan

Dalam enam kali percobaan yang sudah dilakukan terdapat beberapa citra yang gagal diklasifikasikan sesuai dengan nilai rujukan pakar. Dengan nilai akurasi terendah yaitu 78.8235% serta konsistensi nilai yang didapat pada pengujian AUC berada di rentang nilai 0.8 sebenarnya sistem sudah dapat dikategorikan dalam kategori baik. Peneliti mencoba melakukan analisa terhadap citra yang gagal diklasifikasikan dengan benar didapat beberapa

kesimpulan penyebab kegagalan. Penyebab kegagalan untuk klasifikasi kematangan terjadi karena pengambilan citra dengan kondisi cahaya yang buruk mengakibatkan citra menjadi dominan memiliki filter berwarna biru hal ini membuat konversi tingkat keabuan sebuah citra menjadi tidak baik. Citra dominan memiliki filter berwarna biru terdapat pada Gambar 4.43.



Gambar 4.43 Citra Gagal Klasifikasi

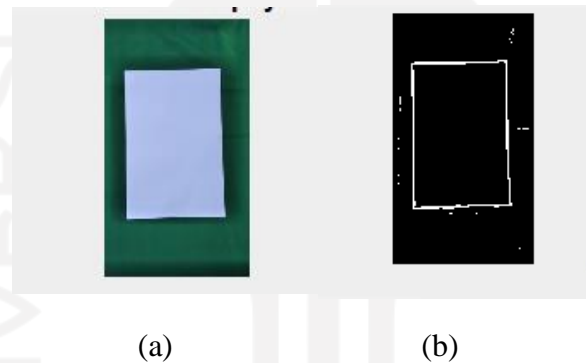
Selain kegagalan yang diakibatkan oleh gagal klasifikasi, kegagalan lain juga terjadi pada saat deteksi tepi di mana hal ini membuat luasan obyek pepaya bertambah bahkan jika citra memiliki kualitas yang terlalu buruk tepi tidak dapat terdeteksi. Deteksi tepi kurang presisi diakibatkan oleh adanya tekstur tambahan pada latar belakang obyek sehingga sistem mendeteksi bahwa tekstur tersebut merupakan bagian dari obyek pepaya, selain itu obyek yang terdapat bayangan gelap juga tidak terdeteksi menjadi bagian dari obyek pepaya. Kegagalan karena tekstur latar belakang yang tidak rata terdapat pada Gambar 4.44.



Gambar 4.44 Kegagalan Metode Deteksi Tepi

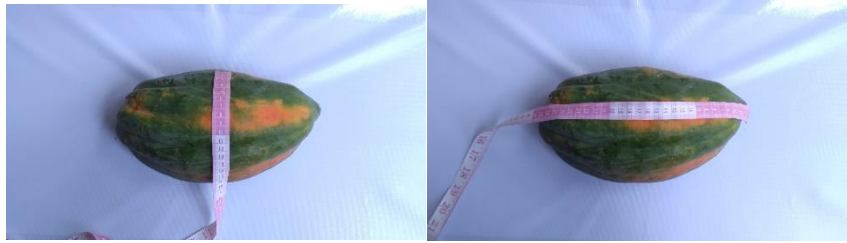
4.5 Perbandingan Kondisi Riil

Untuk mengetahui perbandingan ukuran dalam sebuah citra dengan kondisi riil maka dilakukan sebuah percobaan dengan menggunakan media kertas HVS ukuran 215 x 330 mm. Selain membandingkan ukuran luas juga dilakukan perbandingan ukuran dengan menggunakan timbangan. Berat pepaya yang didapat dengan ukuran Kilogram dikalikan dengan harga pepaya dalam satuan Kilogram apakah akan sesuai dengan harga prediksi dalam sistem. Untuk memudahkan deteksi area HVS digunakan obyek berlatar belakang hijau sehingga kertas putih dapat terdeteksi sebagai obyek. Perbandingan ukuran kertas yang dimaksud terdapat pada Gambar 4.45.



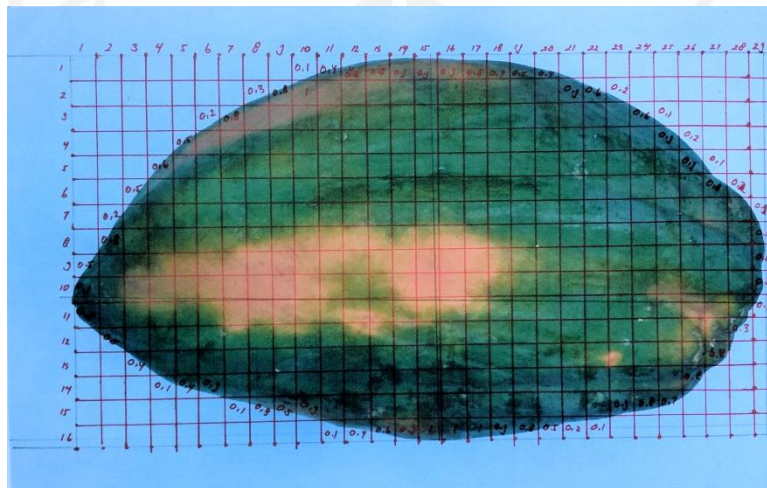
Gambar 4.45 Obyek Kertas HVS; (a) Citra RGB (b) Citra Hasil Deteksi Tepi *Prewitt*

Hasil pengukuran luas dalam satuan milimeter dalam keadaan riil kertas HVS didapat luas sebesar 70950 milimeter, dengan konversi milimeter ke piksel maka menjadi 268157.5 piksel dimana untuk 1 mm dalam piksel menjadi 3.7795275591. Sedangkan hasil pengukuran luas oleh sistem didapatkan hasil 32994 piksel dengan jarak potret dari alas potret 502 milimeter maka didapatkan perbandingan luas 8:1 antara keadaan riil dengan citra hasil potret. Setelah didapatkan nilai perbandingan ukuran luas maka dapat dilakukan perhitungan nilai luas riil pepaya dengan mengukur panjang bidang pepaya dengan pita ukur. Setelah dilakukan perhitungan didapat luas area obyek pepaya dalam kondisi riil 336.9 centimeter atau 33690 milimeter sehingga dalam satuan piksel menjadi 127332.2 piksel. Pengukuran obyek terdapat pada Gambar 4.46.



Gambar 4.46 Pengukuran Riil Obyek Pepaya

Untuk memudahkan dalam perhitungan luas obyek pepaya dibuat model dua dimensi dari ukuran pepaya dalam kertas HVS untuk dihitung luasnya dalam satuan sentimeter. Pemodelan gambar obyek pepaya terdapat pada Gambar 4.47.



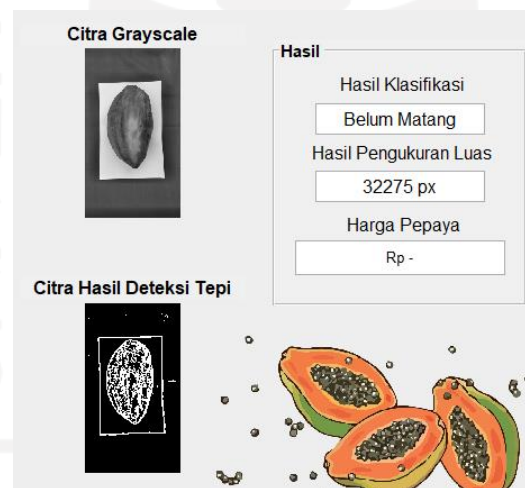
Gambar 4.47 Pemodelan Ukuran Riil Pepaya

Kondisi luas pepaya secara riil sudah diketahui dilanjutkan dengan pengukuran pepaya menggunakan sistem didapat luas obyek sebesar 16251 piksel. Dengan pengukuran kondisi riil mendapat luas 127332.2 piksel maka mendapat perbandingan 7.8:1 antara keadaan riil dengan pengukuran pada sistem. Hasil pengukuran oleh sistem terdapat pada Gambar 4.48.



Gambar 4.48 Hasil Pengukuran Luas Pepaya Pada Sistem

Sedangkan jika obyek berada di dalam kertas HVS maka didapatkan luas 32275 piksel dimana ketika kondisi pengukuran kertas HVS tanpa obyek pepaya sebelumnya didapat ukuran luas sebesar 32994 piksel hal ini menunjukkan bahwa pengukuran luas obyek cukup stabil dengan menggunakan acuan garis terluar dari obyek. Pengukuran obyek pepaya dalam media kertas HVS terdapat pada Gambar 4.49.



Gambar 4.49 Pengukuran Luas Pepaya Dalam Obyek Kertas HVS

Dari kedua perbandingan yang sudah dilakukan didapat data pada Tabel 4.16.

Tabel 4.16 Hasil Perbandingan Kondisi Riil

Obyek	Ukuran (mm)	Riil (mm)	Piksel	Sistem (piksel)	Perbandingan
HVS	215 x 330	70950	268157.5	32994	8.1 : 1
Pepaya	±160 x 290	33690	127332.2	16251	7.8 : 1

Nilai perbandingan yang didapat dari kedua perhitungan adalah 8.1 dan 7.8 sehingga antara perhitungan sistem dengan keadaan nyata memiliki pengukuran yang hampir sama. Selain pengukuran dengan perbandingan luas obyek pada pengukuran keadaan nyata kedua dengan menggunakan parameter berat dengan satuan kilogram. Berat pepaya didapatkan dengan mengukur menggunakan timbangan duduk, pengukuran dengan timbangan duduk terdapat pada Gambar 4.50.



Gambar 4.50 Pengukuran Berat Dengan Timbangan Duduk

Perbandingan ukuran pepaya dengan ukuran berat didapatkan berat pepaya sebesar 1.6 Kilogram dengan harga perkilogram untuk pepaya Thailand sebesar Rp.1000/Kg maka harga pepaya dalam sampel citra tersebut adalah Rp.1600. Sedangkan dalam sistem pepaya thailand dengan ukuran 16251 piksel memiliki harga sebesar Rp.2000 karena masuk dalam kategori pepaya Thailand kecil. Hasil perhitungan harga memiliki nilai yang berdekatan dengan selisih harga Rp.400 sehingga untuk sistem prediksi harga dapat disimpulkan memiliki hasil yang cukup baik.

4.6 Kelebihan dan Kekurangan

b. Kelebihan

1. Dapat mengklasifikasikan kematangan pepaya dengan performa akurasi yang cukup baik.
2. Memberikan informasi mengenai ekstraksi ciri GLCM obyek pepaya yang diamati.
3. Dapat menampilkan luas obyek pepaya dengan cukup baik.
4. Perhitungan nilai akhir harga cukup baik dengan selisih tidak terlalu banyak.

c. Kekurangan

1. Terdapat citra pepaya yang tidak terklasifikasi dengan benar.
2. Latar belakang obyek yang digunakan tidak rata menyebabkan kebocoran tepi sehingga menyebabkan adanya tambahan area obyek yang tidak diperlukan.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapatkan dari penelitian ini adalah sebagai berikut.

- a. Ekstraksi ciri GLCM *Contrast, Correlation, Energy* dan *Homogeneity* citra pepaya dapat digunakan untuk mengklasifikasikan kematangan pepaya dengan menggunakan metode pengenalan tekstur menghasilkan data rujukan pakar yang konsisten untuk digunakan pada tahap pengujian dan program utama. Hal ini dibuktikan dengan hasil tahap pengujian memperoleh nilai akurasi sebesar 81,25% menunjukkan data rujukan pakar yang digunakan sudah baik dan cukup konsisten, hal ini juga didukung dengan penggunaan data citra yang baik saat pengumpulan data.
- b. Deteksi tepi dapat digunakan untuk mengetahui luas obyek pepaya menggunakan operator *Prewitt* didapatkan hasil yang cukup baik dengan melakukan pengujian menggunakan perbandingan luas pada kertas HVS diperoleh nilai yang mendekati nilai riil dengan perbandingan nilai riil 8.1 berbanding 1 sedangkan pada sistem diperoleh nilai 7.8 berbanding 1. Selain pengujian dengan parameter luas obyek pengujian dengan parameter berat pepaya juga dilakukan, pengujian berat menggunakan timbangan diperoleh berat 1.6 Kg dengan harga pepaya Rp1600 sedangkan pada perhitungan sistem diperoleh harga pepaya Rp.2000 yang masuk dalam kategori pepaya Thailand kecil.
- c. Secara keseluruhan klasifikasi menggunakan metode *Naive Bayes* dapat digunakan untuk klasifikasi pepaya hal ini dibuktikan dengan 5 kali percobaan yang mendapat nilai rata-rata *Accuracy* sebesar 86.5% untuk tahap pelatihan dan 81.25% untuk tahap pengujian dengan menggunakan metode *Confusion Matrix*. Pengujian lanjutan dilakukan dengan metode berbeda yaitu *Area Under Curve (AUC)* dengan kategori nilai 0 hingga 1 di mana semakin dekat dengan nilai 1 menunjukkan metode klasifikasi yang digunakan semakin bagus. Pada penelitian ini didapat nilai AUC untuk tahap pelatihan sebesar 0.87082 dan 0.8236 untuk tahap pengujian.

5.2 Saran

Dalam penelitian ini masih terdapat beberapa kekurangan yang perlu diperbaiki sehingga membutuhkan saran untuk pengembangan penelitian selanjutnya. Berikut saran yang dapat dipertimbangkan untuk penelitian selanjutnya.

- a. Penggunaan data citra yang baik dengan kualitas pencahayaan yang baik akan meningkatkan nilai ketepatan klasifikasi *Naive Bayes*. Hal ini dapat dilakukan dengan menggunakan alat tambahan berupa lampu untuk pengambilan citra atau dengan melakukan pengambilan data di luar ruangan dengan intensitas cahaya yang baik.
- b. Latar belakang obyek dapat menggunakan media datar yang tidak bergelombang untuk menghindari adanya tambahan nilai deteksi tepi di luar obyek pepaya dikarenakan sistem mendeteksi obyek yang tidak rata sebagai bagian dari obyek pepaya itu sendiri.
- c. Sistem ini hanya dapat mengolah data citra *single file* saja sehingga dapat dikembangkan lagi untuk pengolahan citra dengan *input multifile* agar dapat diimplementasikan secara langsung pada pohon pepaya.
- d. Penggunaan nilai rujukan harga yang lebih bervariasi lagi menyesuaikan berat pepaya, sehingga hasil prediksi harga antara harga jual per biji dengan harga jual menggunakan sistem timbangan dapat lebih akurat lagi.



DAFTAR PUSTAKA

- Ahmad, U. (2005). *Pengolahan citra digital dan teknik pemrogramannya* (1st ed.). Graha Ilmu.
- Al Faruqi, U. (2019). *Survey Paper : Future Service in Industry 5.0*. 02(01), 67–79.
- Andono, P. N., T, S., & Muljono. (2017). *Pengolahan citra digital* (A. Pramesta (ed.); D. Penerbit Andi).
- Arifin, Z. (2016). *MENGGUNAKAN NAÏVE BAYES BERDASARKAN WARNA*. Universitas Dian Nuswantoro.
- Arifiya, N., Purwanto, Y., & Budiastara, I. (2015). Analisis Perubahan Kualitas Pascapanen Pepaya Varietas Ipb9 Pada Umur Petik Yang Berbeda. *Jurnal Keteknik Pertanian*, 3(1), 21689.
- Auliasari, K., & Kertaningtyas, M. (2018). Studi Komparasi Klasifikasi Pola Tekstur Citra Digital Menggunakan Metode K-Means Dan Naïve Bayes. *Jurnal Informatika*, 18(2), 175–185.
- Badan Pusat Statistik Kabupaten Boyolali. (2019). *Kabupaten Boyolali Dalam Angka Tahun 2019* (B. K. Boyolali (ed.)). Badan Pusat Statistik Kabupaten Boyolali.
- Bhahri, S., & Rachmat. (2018). Transformasi Citra Biner Menggunakan Metode Thresholding Dan Otsu Thresholding. *Jurnal Sistem Informasi Dan Teknologi Informasi*, 7(2), 195–203.
- Bonaditya. (2014). *Pepaya Carica*. Penebar Swadaya.
- BPS Boyolali. (2019). *Kecamatan Tamansari Dalam Angka*.
- Connors, R. W., & Harlow, C. A. (1980). A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(3), 204–222. <https://doi.org/10.1109/TPAMI.1980.4767008>
- Fadlisyah. (2007). *Computer Vision dan Pengolahan Citra* (1st ed.).
- Fatihah, N. (2016). *Identifikasi Jenis Kayu tropis Menggunakan Backpropagation Neural Network*. Universitas Sumatera Utara.
- Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques* (J. Kacprzyk & L. C. Jain (eds.); 12th ed.). Springer-Verlag. <https://doi.org/10.1007/978-3-642-19721-5>
- Haba, R. K., & Pelangi, K. C. (2020). Pengelompokan Buah Jeruk menggunakan Naïve Bayes dan Gray Level Co-occurrence Matrix. *ILKOM Jurnal Ilmiah*, 12(1), 17–24. <https://doi.org/10.33096/ilkom.v12i1.494.17-24>

- Hakiky, R. M., Hikmah, N., & Ariyanti, D. (2021). Klasifikasi Jenis Pohon Mangga Berdasarkan Bentuk dan Tekstur Daun Menggunakan Metode Backpropagation. *Jurnal Informatika Upgris*, 6(2), 247–253. <https://doi.org/10.26877/jiu.v6i2.6645>
- Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions On Systems, Man and Cybernetics*, SMC-3, 610–621.
- Huda, M. (2016). DETEKSI TEPI CITRA TELUR DENGAN ALGORITMA PREWITT UNTUK PERHITUNGAN VOLUME. *PROSIDING SEMINAR NASIONAL IPTEK TERAPAN 2016*, 1, 18–23.
- Indriyani, L., Susanto, W., & Riana, D. (2017). Teknik pengolahan citra menggunakan aplikasi MATLAB pada pengukuran diameter buah Jeruk Keprok. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 2(1), 46–52.
- Irawan, J. (2007). Buku Pegangan Kuliah Sistem Pakar. *Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya (STIKOM)*, July. <http://prints.upnyk.ac.id/774/3/6> Abstrak.pdf
- Isnawan. (2015, October 14). *Cara Budidaya Pepaya California*. E-Petani. <http://epetani.pertanian.go.id/berita/detail/27>
- Jusman, Y., Ng, S. C., Hasikin, K., Kurnia, R., Abu Osman, N. A., & Teoh, K. H. (2017). A system for detection of cervical precancerous in field emission scanning electron microscope images using texture features. *Journal of Innovative Optical Health Sciences*, 10(2), 1–12. <https://doi.org/10.1142/S1793545816500450>
- Kusanti, J., Penyakit, K., Padi, D., & Haris, A. (2018). Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(01), 1–6.
- Liantoni, F., & Nugroho, H. (2015). Klasifikasi Daun Herbal Menggunakan Metode Naïve Bayes Classifier Dan Knearest Neighbor. *Jurnal Simantec*, 5(1), 9–16.
- Madenda, S. (2015). *Pengolahan Citra dan Video Digital : teori, aplikasi dan pemrograman menggunakan MATLAB*. Penerbit Erlangga.
- Manik, F. Y., & Saragih, K. S. (2017). Klasifikasi Belimbing Menggunakan Naïve Bayes Berdasarkan Fitur Warna RGB. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 11(1), 99. <https://doi.org/10.22146/ijccs.17838>
- Manuel, I. S., Ernawati, I., Pembangunan, U., Veteran, N., Labu, P., & Selatan, J. (2020). *Implementasi Glcm Dan Algoritma Naïve Bayes Dalam Klasifikasi Jenis Bunga Anggrek*. 99–109.

- Muchlis, N. F., Muhimmah, I., & Kurniawardhani, A. (2018). Deteksi kemerahan pada kulit wajah dengan teknik pengolahan citra. In *Deteksi Kemerahan Pada Kulit Wajah Dengan Teknik Pengolahan Citra*. Universitas Islam Indonesia.
- Nurhikmat, T. (2018). *Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek* (Issue 4) [Universitas Islam Indonesia].
https://www.jstage.jst.go.jp/article/amr/1/5/1_010501/_article/-char/ja/
- Pantastico, E. ., & Kamariyani. (1989). *Fisiologi pasca panen, penanganan dan pemanfaatan buah - buahan dan sayur - sayuran tropika dan sub tropika* (G. Tjitrosoepomo (ed.)). Gadjah Mada University Press.
- Parker, J. R. (2013). *Algorithms for Image Processing and Computer Vision*. (2nd ed.). John Wiley Publishing. <https://doi.org/https://doi.org/10.1017/CBO9781107415324.004>
- Pranata, A., & Astuti, E. Z. (2017). Pengolahan Citra Berbasis Deteksi Tepi Prewitt Pada Gambar Gigi Manusia. *Eksplora Informatika*, 6(2), 98–105. <https://eksplora.stikom-bali.ac.id/index.php/eksplora/article/view/110/91>
- Pusat Kajian Hortikultura Tropika. (2015). *Pepaya Callina (IPB-9)*. Institut Pertanian Bogor. <https://pkht.ipb.ac.id/index.php/2015/11/03/pepaya-ipb-9/>
- Rachmadi, M. H., Hayaty, N., & Uperiati, A. (n.d.). *Klasifikasi tingkat kematangan buah pisang kepok (musa paradisiaca formatypica) dengan ekstraksi fitur warna dengan menggunakan alogritma naive bayes*.
- Redo Dwiantara, M. (2020). *Preferensi Konsumen Terhadap Buah Pepaya California Dan Thailand Di Kecamatan Gamping Kabupaten Sleman* (Vol. 21, Issue 1). Universitas Muhammadiyah Yogyakarta.
- Rohpandi, D., Sugiharto, A., & Winara, G. A. (2015). Aplikasi Pengolahan Citra Dalam Pengenalan Pola Huruf Ngalagena Menggunakan MATLAB. *Konferensi Nasional Sistem & Informatika 2015*.
- Romindo, & Khairina, N. (2017). Analisa Perbandingan Metode Edge Detection Roberts Dan Prewitt. *Seminar Nasional Teknologi Informatika*, 244–251.
- Sapti, M. (2019). DIVERSIFIKASI RAGAM PRODUK PEPAYA SEBAGAI PRODUK UNGGULAN BOYOLALI Nurul. *Kemampuan Koneksi Matematis (Tinjauan Terhadap Pendekatan Pembelajaran Savi)*, 53(9), 1689–1699.
- Subakti, I. (2018). *Sistem Cerdas* (1st ed.). Rusindo.
- Sugiyanto, S., & Wibowo, F. (2015). *Klasifikasi Tingkat Kematangan Buah Pepaya* (Carica

- Papaya L) California (Callina-Ipb 9) Dalam Ruang Warna Hsv. *Prosiding SENATEK*, 1(November), 335–341.
- Sujiprihati, S., & Suketi, K. (n.d.). *Budi Daya PEPAYA UNGGUL*. Penebar Swadaya. Retrieved February 17, 2021, from https://books.google.co.id/books?hl=id&lr=&id=PO2gCgAAQBAJ&oi=fnd&pg=PP1&dq=pepaya+adalah&ots=8nD_SChXKx&sig=XhAB-8JwKBs8nzZeZc3IkYrtBKE&redir_esc=y#v=onepage&q=pepaya+adalah&f=false
- Suwarno, & AA Abdillah. (2016). PENERAPAN ALGORITMA BAYESIAN REGULARIZATION BACKPROPAGATION UNTUK MEMPREDIKSI PENYAKIT DIABETES. *Jurnal MIPA*, 39(2), 98–106.
- Swedia, E. R., & Cahyanti, M. (2010). Algoritma Transformasi Ruang Warna. *Universitas Gunadharma, Depok, Jakarta*.
- Syauqi Falasev, R., Hidayatno, A., & Rizal, I. R. (2011). *Pengenalan Sidik Jari Manusia Dengan Matriks Kookurensi Aras Keabuan (Gray Level Co-Ocurrence Matrix)*. Universitas Diponegoro.
- Syukri Mustafa, M., Rizky Ramadhan, M., Thenata, A. P., Kunci -Algoritma Naive Bayes Classifier, K., & Akademik Mahasiswa, K. (2017). Implementasi Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Citec Journal*, 4(2).
- Tamarena, R. I. (2020). *Sistem Klasifikasi Citra Karies Gigi Menggunakan Metode Gray Level Co-Ocurrence Matrix dan K-Nearest Neighbor* (Issue 2020) [Universitas Muhammadiyah Yogyakarta]. <http://repository.umy.ac.id/handle/123456789/31601>
- The MathWorks, I. (2006). *Create gray-level co-occurrence matrix from image - MATLAB graycomatrix*. <https://www.mathworks.com/help/images/ref/graycomatrix.html>
- Utami Putri, U. (2016). *Untung Besar Dari Berkebun Pepaya*. Akar Publishing. <https://inlis.madiunkota.go.id/opac/detail-opac?id=13767>
- Utami, R. Z., Suksmadana, I. M. B., & Kanata, B. (2015). Menentukan Luas Objek Citra dengan Teknik Deteksi Tepi. *Dielektrika*, 2(1), 11–17.
- Wijaya, M. C., & Prijono, A. (2007). *Pengolahan citra digital menggunakan MATLAB*. Informatika Bandung.

LAMPIRAN

FORM-TA/TF-A3



UNIVERSITAS ISLAM INDONESIA
Program Studi Informatika FTI

SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Hanung Setyo Sambudi

No. Mhs. : 15523104


Judul TA : Sistem Cerdas Klasifikasi Kematangan Dan Harga Buah Pepaya Berdasarkan Ekstraksi Fitur Gray Level Co-Occurence Matrix Dengan Metode Naive Bayes

Latar belakang sedikit kurang nyambung dengan metode yang dipropose.
Tuliskan di laporan dengan jelas pembagian data training dan data uji

Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 20 April 2021

Dosen,


Shella Nurul Huda
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran