

**PEMANFAATAN CIRI *GRAY LEVEL CO-OCCURRENCE MATRIX*
(GLCM) CITRA BUAH JERUK KEPROK (*CITRUS RETICULATA*
BLANCO) UNTUK KLASIFIKASI MUTU**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Restu Widodo
NIM: 145150207111048



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PEMANFAATAN CIRI *GRAY LEVEL CO-OCCURRENCE MATRIX* (GLCM) CITRA BUAH
JERUK KEPROK (*CITRUS RETICULATA BLANCO*) UNTUK KLASIFIKASI MUTU

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Restu Widodo
NIM: 145150207111048

Skripsi ini telah diuji dan dinyatakan lulus pada
9 Juli 2018

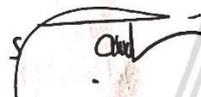
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Agus Wahyu Widodo, S.T, M.Cs
NIP. 19740805 200112 1 001



Ir. Arry Supriyanto, MS
NIP. 19540324 198101 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Ni Astoro Kurniawan, S.T., M.T., Ph.D

NIP. 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Juli 2018



Restu Widodo

NIM: 145150207111048

KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Puji dan syukur tiada hentinya penulis panjatkan kehadiran Allah SWT atas limpahan nikmat serta rahmat yang diberikan olehNya sehingga penulis dapat menyelesaikan tugas akhir skripsi dengan judul “Pemanfaatan Ciri *Gray Level Co-occurrence Matrix* (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu” ini. Skripsi ini disusun sebagai syarat memperoleh gelar sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam penyelesaian skripsi ini penulis banyak mendapatkan dukungan dan bantuan dari berbagai pihak, baik dukungan moral maupun materiil. Oleh karena itu, penulis menyampaikan rasa terimakasih kepada:

1. Allah SWT yang telah memberikan kemudahan dan kelancaran selama proses penulisan skripsi ini.
2. Kedua Orang Tua yaitu Bapak Suranto dan Ibu Nuraini serta adik saya Adriyanto Wicaksono dan keluarga besar yang selalu memberikan do’a, kasih sayang, masukan, motivasi, serta materiil untuk keberhasilan penulis.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Kepala Program Studi Teknik Informatika dan selaku Pembimbing I yang telah banyak memberikan arahan, ilmu, masukan serta saran dalam penyusunan skripsi ini.
4. Bapak Ir. Arry Supriyanto, MS selaku Pembimbing II yang telah banyak memberikan arahan, ilmu, masukan serta saran dalam penyusunan skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
8. Teman-teman kuliah terdekat di Malang khususnya Davia Werdiastu, Anggi Mahadika, Farah Shofiatul dan Harits Abdurrohman yang selalu mendukung dan membantu selama penulisan skripsi.
9. Mega Bela Anendia yang selalu memberikan dukungan dan semangat selama penulisan skripsi.
10. Semua pihak yang tidak dapat disebutkan satu per satu, yang terlibat baik secara langsung maupun tidak langsung dalam proses penulisan skripsi.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan tidak lepas dari kesalahan. Oleh karena itu, penulis mengharapkan kritik dan saran untuk memperbaiki skripsi ini. Penulis berharap semoga skripsi ini dapat memberikan manfaat untuk berbagai pihak.

Malang, 9 Juli 2018

Penulis

restu.widodo24@student.ub.ac.id



ABSTRAK

Mutu buah merupakan hal yang sangat penting dalam hasil produksi buah-buahan. Khususnya pada buah jeruk, sebagai buah dengan produksi terbanyak ke-2 di Indonesia, mutu sangat diperhatikan karena terkait dengan nilai penjualan. Terlebih untuk jenis jeruk keprok yang menguasai 92% produksi buah jeruk. Namun saat ini proses klasifikasi mutu pada industri buah jeruk masih dilakukan secara manual, sehingga menimbulkan subjektivitas dalam penentuan mutu jeruk. Teknologi informasi diperlukan untuk mempercepat proses klasifikasi mutu dan membuat identifikasi mutu jeruk menjadi objektif sesuai dengan standar klasifikasi mutu jeruk. Penelitian ini memanfaatkan hasil ekstraksi ciri *gray level co-occurrence matrix* (GLCM) citra jeruk keprok (*Citrus reticulata* Blanco) untuk klasifikasi mutu. Tahapan yang dilakukan pada awal penelitian adalah pengambilan data citra jeruk keprok. Dari 100 data citra jeruk keprok, 60 data digunakan sebagai data latih dan 40 data digunakan sebagai data uji. Dari tiap-tiap data latih, diambil masing-masing satu citra baik dan citra buruk berukuran 64x64 piksel. Kemudian dilakukan *pre-processing* pada citra agar dapat diolah pada tahap selanjutnya, yaitu perhitungan GLCM. Pada tahap perhitungan GLCM dilakukan pembentukan matriks GLCM pada arah 0°, 45°, 90° dan 135°. Selanjutnya dilakukan proses ekstraksi ciri dengan *gray level co-occurrence matrix* (GLCM). Nilai ciri yang diekstraksi adalah *contrast*, *homogeneity*, *energy* dan *entropy*. Metode *support vector machine* (SVM) digunakan untuk identifikasi citra baik dan citra buruk berdasarkan ciri yang telah di ekstraksi, sehingga didapatkan persentase cacat buah. Klasifikasi mutu jeruk dilakukan berdasarkan persentase cacat buah terhadap luas permukaan jeruk. Mutu jeruk dibagi kedalam 3 kelas yaitu, *Grade Super*, *Grade A* dan *Grade B*. Pengujian dilakukan untuk mengetahui jumlah data latih, nilai *distance* dan arah GLCM terbaik terhadap hasil klasifikasi. Penelitian ini menghasilkan tingkat akurasi terbaik sebesar 82.5% dengan jumlah data latih sebanyak 20, nilai *distance*=2 pada arah GLCM 45°.

Kata kunci: *gray level co-occurrence matrix*, ekstraksi ciri, SVM, jeruk keprok

ABSTRACT

The most important thing of production result of fruit is quality. Especially in citrus, as the fruit with the 2nd largest production value in Indonesia, the quality is important because it is related to selling value. 92% production of citrus is "Keprok". But now the process of quality classification in the fruits industry is still done manually, so it becomes subjective. Information technology is needed to speed up the process of quality classification and making the identification of the quality of citrus to be objective in accordance with the standard of citrus quality classification. This research utilizes the extraction feature of gray level co-occurrence matrix (GLCM) citrus image for quality classification. This research begins with collecting data of citrus. There are 100 citrus image data, 60 data are used as training data and 40 data are used as test data. Of each training data, obtained one 64x64 pixels good and bad data image. Do pre-processing on the image to be processed in the next stage, calculation of GLCM. In GLCM calculation stage, GLCM matrix is formed in direction 0°, 45°, 90° and 135°. Next stage is a feature extraction process with gray level co-occurrence matrix (GLCM). The extracted feature values are contrast, homogeneity, energy and entropy. Support vector machine (SVM) is used for good image and bad image identification based on the feature extraction, to get the percentage of fruit defects. The quality classification is based on the percentage of fruit defects on the citrus surface area. Citrus quality is divided into 3 classes, Super Grade, Grade A and Grade B. Performed testing for finding the best amount of trainee data, distance value and GLCM direction to the classification result. The result of this research shows that the best classification accuracy is 82.5%, with the amount of training data is 20, distance=2 at 45° GLCM.

Keywords: gray level co-occurrence matrix, feature extraction, SVM, citrus

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR SOURCECODE	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Buah Jeruk.....	8
2.3 Jeruk Keprok (<i>Citrus reticulata</i> Blanco)	9
2.4 Mutu Buah Jeruk Keprok	9
2.5 Pengolahan Citra Digital	10
2.6 <i>Gray Level Co-occurrence Matrix</i> (GLCM).....	11
2.7 <i>Support Vector Machine</i> (SVM)	14
2.7.1 <i>SVM Linear</i>	15
2.7.2 <i>SVM Non-linear</i>	15
BAB 3 METODOLOGI	23
3.1 Studi Literatur	23
3.2 Pengumpulan Data	23



3.3 Perancangan	24
3.4 Implementasi	26
3.5 Pengujian & Analisis.....	26
3.6 Kesimpulan dan Saran	27
BAB 4 PERANCANGAN.....	28
4.1 Perancangan Algoritme	28
4.1.1 <i>Pre-Processing</i>	29
4.1.2 Perhitungan GLCM	33
4.1.3 Identifikasi Cacat Buah	42
4.2 Perhitungan Manual	43
4.2.1 <i>Pre-Processing</i>	43
4.2.2 Perhitungan GLCM	45
4.2.3 Klasifikasi Mutu Jeruk.....	55
4.3 Perancangan Skenario Pengujian	55
4.3.1 Pengujian Jumlah Data Latih.....	56
4.3.2 Pengujian Nilai <i>Distance (d)</i>	56
4.3.3 Pengujian Arah GLCM	56
BAB 5 IMPLEMENTASI	58
5.1 Lingkungan Implementasi.....	58
5.1.1 Lingkungan <i>Hardware</i>	58
5.1.2 Lingkungan <i>Software</i>	58
5.2 Batasan Implementasi	59
5.3 Implementasi Aplikasi.....	59
5.3.1 <i>Pre-Processing</i>	59
5.3.2 Perhitungan GLCM	63
5.3.3 Identifikasi Cacat Buah	69
BAB 6 PENGUJIAN DAN ANALISIS.....	73
6.1 Pengujian	73
6.1.1 Pengujian Jumlah Data Latih	73
6.1.2 Pengujian Nilai <i>Distance (d)</i>	74
6.1.3 Pengujian Arah GLCM	74
6.2 Hasil Pengujian.....	75



6.3 Analisis Pengujian	92
BAB 7 PENUTUP	97
7.1 Kesimpulan.....	97
7.2 Saran	97
DAFTAR PUSTAKA.....	98



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	18
Tabel 4.1 Nilai <i>Red</i> dari Gambar 4.8.....	43
Tabel 4.2 Nilai <i>Green</i> dari Gambar 4.8.....	43
Tabel 4.3 Nilai <i>Blue</i> dari Gambar 4.8.....	44
Tabel 4.4 Hasil Perhitungan <i>Grayscale</i>	44
Tabel 4.5 Matriks GLCM Sudut 0°	45
Tabel 4.6 Matriks GLCM Sudut 45°	46
Tabel 4.7 Matriks GLCM Sudut 90°	47
Tabel 4.8 Matriks GLCM Sudut 135°	48
Tabel 4.9 Rancangan Pengujian	56
Tabel 4.10 Rancangan Pengujian Data Latih, <i>Distance</i> dan Arah GLCM.....	57
Tabel 6.1 Hasil Pengujian 10 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	75
Tabel 6.2 Hasil Pengujian 10 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	76
Tabel 6.3 Hasil Pengujian 10 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	77
Tabel 6.4 Hasil Pengujian 20 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	78
Tabel 6.5 Hasil Pengujian 20 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	78
Tabel 6.6 Hasil Pengujian 20 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	79
Tabel 6.7 Hasil Pengujian 30 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	80
Tabel 6.8 Hasil Pengujian 30 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	81
Tabel 6.9 Hasil Pengujian 30 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	82
Tabel 6.10 Hasil Pengujian 40 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	83
Tabel 6.11 Hasil Pengujian 40 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	84
Tabel 6.12 Hasil Pengujian 40 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	85
Tabel 6.13 Hasil Pengujian 50 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	85
Tabel 6.14 Hasil Pengujian 50 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	86
Tabel 6.15 Hasil Pengujian 50 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	87
Tabel 6.16 Hasil Pengujian 60 Data Latih dengan $d=1$ pada 4 Arah GLCM.....	88
Tabel 6.17 Hasil Pengujian 60 Data Latih dengan $d=2$ pada 4 Arah GLCM.....	89
Tabel 6.18 Hasil Pengujian 60 Data Latih dengan $d=3$ pada 4 Arah GLCM.....	90
Tabel 6.19 Rangkuman Pengujian Data Latih, <i>Distance</i> dan Arah GLCM	91

DAFTAR GAMBAR

Gambar 2.1 Buah Jeruk	8
Gambar 2.2 Jeruk Keprok.....	9
Gambar 2.3 Koordinat Citra Digital	11
Gambar 2.4 Arah Kookurensi GLCM	12
Gambar 2.5 Penentuan Nilai Awal Matriks GLCM	12
Gambar 2.6 Pembentukan Matriks GLCM yang Simetris	12
Gambar 2.7 Normalisasi Matriks GLCM.....	13
Gambar 2.8 <i>Hyperplane</i> yang Mungkin untuk Set Data	14
Gambar 2.9 Ilustrasi SVM <i>Linear</i>	15
Gambar 2.10 Ilustrasi SVM <i>Non-linear</i>	16
Gambar 2.11 Ilustrasi Dataset SVM <i>Non-linear</i>	17
Gambar 3.1 Blok Diagram Metodologi Penelitian	23
Gambar 3.2 Arsitektur Sistem	24
Gambar 4.1 Diagram Alir Sistem.....	28
Gambar 4.2 Diagram Alir <i>Pre-Processing</i>	29
Gambar 4.3 Diagram Alir <i>Grayscaleing</i>	30
Gambar 4.4 Diagram Alir Segmentasi	32
Gambar 4.5 Diagram Alir Perhitungan GLCM	33
Gambar 4.6 Diagram Alir Perhitungan Matriks GLCM	38
Gambar 4.7 Diagram Alir Perhitungan Ekstraksi Ciri GLCM.....	41
Gambar 4.8 Potongan Citra Jeruk Keprok dengan Ukuran 8x8 Pixel	43
Gambar 4.9 Perhitungan Pasangan Pixel Pada Sudut 0°	45
Gambar 4.10 Perhitungan Pasangan Pixel Pada Sudut 45°	46
Gambar 4.11 Perhitungan Pasangan Pixel Pada Sudut 90°	47
Gambar 4.12 Perhitungan Pasangan Pixel Pada Sudut 135°	47
Gambar 6.1 Grafik Hasil Pengujian Data Latih 10 Pada 3 <i>Distance</i> dan 4 Arah GLCM	92
Gambar 6.2 Grafik Hasil Pengujian Data Latih 20 Pada 3 <i>Distance</i> dan 4 Arah GLCM	93
Gambar 6.3 Grafik Hasil Pengujian Data Latih 30 Pada 3 <i>Distance</i> dan 4 Arah GLCM	94



Gambar 6.4 Grafik Hasil Pengujian Data Latih 40 Pada 3 *Distance* dan 4 Arah GLCM 94

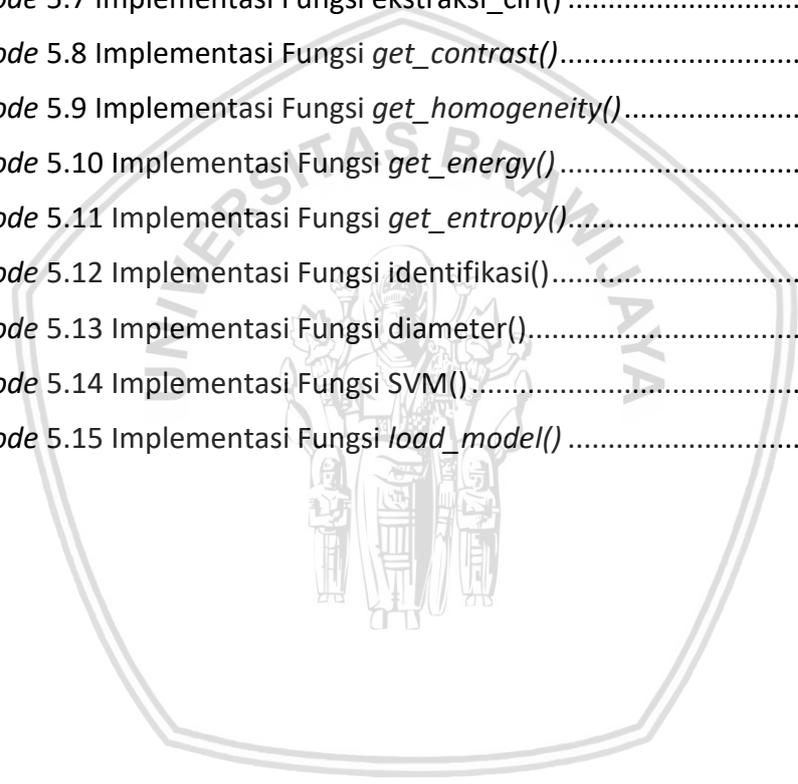
Gambar 6.5 Grafik Hasil Pengujian Data Latih 50 Pada 3 *Distance* dan 4 Arah GLCM 95

Gambar 6.6 Grafik Hasil Pengujian Data Latih 60 Pada 3 *Distance* dan 4 Arah GLCM 95



DAFTAR SOURCECODE

<i>Sourcecode</i> 5.1 Implementasi Fungsi <i>grayscaleing()</i>	60
<i>Sourcecode</i> 5.2 Implementasi Fungsi segmentasi()	61
<i>Sourcecode</i> 5.3 Implementasi Fungsi <i>get_position()</i>	62
<i>Sourcecode</i> 5.4 Implementasi Fungsi titik_baru().....	63
<i>Sourcecode</i> 5.5 Implementasi Fungsi matriks_glcm().....	65
<i>Sourcecode</i> 5.6 Implementasi Fungsi <i>get_total_pasangan()</i>	66
<i>Sourcecode</i> 5.7 Implementasi Fungsi ekstraksi_ciri()	66
<i>Sourcecode</i> 5.8 Implementasi Fungsi <i>get_contrast()</i>	67
<i>Sourcecode</i> 5.9 Implementasi Fungsi <i>get_homogeneity()</i>	67
<i>Sourcecode</i> 5.10 Implementasi Fungsi <i>get_energy()</i>	68
<i>Sourcecode</i> 5.11 Implementasi Fungsi <i>get_entropy()</i>	68
<i>Sourcecode</i> 5.12 Implementasi Fungsi identifikasi().....	69
<i>Sourcecode</i> 5.13 Implementasi Fungsi diameter().....	71
<i>Sourcecode</i> 5.14 Implementasi Fungsi SVM().....	72
<i>Sourcecode</i> 5.15 Implementasi Fungsi <i>load_model()</i>	72



DAFTAR LAMPIRAN

LAMPIRAN A PENGUJIAN 10 DATA LATIH <i>DISTANCE</i> 1	100
LAMPIRAN B PENGUJIAN 10 DATA LATIH <i>DISTANCE</i> 2	102
LAMPIRAN C PENGUJIAN 10 DATA LATIH <i>DISTANCE</i> 3	104
LAMPIRAN D PENGUJIAN 20 DATA LATIH <i>DISTANCE</i> 1	106
LAMPIRAN E PENGUJIAN 20 DATA LATIH <i>DISTANCE</i> 2	108
LAMPIRAN F PENGUJIAN 20 DATA LATIH <i>DISTANCE</i> 3	110
LAMPIRAN G PENGUJIAN 30 DATA LATIH <i>DISTANCE</i> 1	112
LAMPIRAN H PENGUJIAN 30 DATA LATIH <i>DISTANCE</i> 2	114
LAMPIRAN I PENGUJIAN 30 DATA LATIH <i>DISTANCE</i> 3	116
LAMPIRAN J PENGUJIAN 40 DATA LATIH <i>DISTANCE</i> 1	118
LAMPIRAN K PENGUJIAN 40 DATA LATIH <i>DISTANCE</i> 2	120
LAMPIRAN L PENGUJIAN 40 DATA LATIH <i>DISTANCE</i> 3	122
LAMPIRAN M PENGUJIAN 50 DATA LATIH <i>DISTANCE</i> 1	124
LAMPIRAN N PENGUJIAN 50 DATA LATIH <i>DISTANCE</i> 2	126
LAMPIRAN O PENGUJIAN 50 DATA LATIH <i>DISTANCE</i> 3	128
LAMPIRAN P PENGUJIAN 60 DATA LATIH <i>DISTANCE</i> 1	130
LAMPIRAN Q PENGUJIAN 60 DATA LATIH <i>DISTANCE</i> 2	132
LAMPIRAN R PENGUJIAN 60 DATA LATIH <i>DISTANCE</i> 3	134
LAMPIRAN S MUTU JERUK DATA UJI	136
LAMPIRAN T DATASET	137
T.1 Data Latih	137
T.2 Data Uji	139

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Buah-buahan merupakan satu dari sekian banyak komoditas hortikultura yang memegang peran penting dalam pembangunan ekonomi nasional. Khususnya buah jeruk, menjadi buah dengan produksi terbanyak ke-2 setelah pisang. Pada tahun 2016, menurut data dari Badan Pusat Statistik dan Direktorat Jenderal Hortikultura, buah jeruk keseluruhan berhasil mencatatkan jumlah produksi sebesar 20.142.061 Kg dari jumlah tanaman hasil sebanyak 24.135.162 Pohon dengan produktivitas sebesar 83,46 Kg/Pohon (Kementrian Pertanian RI, 2017). Khusus jenis jeruk Keprok/Siam, data dari Direktorat Jenderal Hortikultura Kementerian Pertanian pada tahun 2014 menunjukkan bahwa hasil produksi buah sebesar 1.785.256 Ton dari total produksi buah jeruk sebesar 1.926.544 Ton (Taufik, 2015). Data tersebut menunjukkan bahwa 92% dari keseluruhan hasil produksi buah jeruk merupakan jenis jeruk Keprok. Buah jeruk memiliki kandungan gizi yang sangat baik, dimana kandungan vitamin C yang dibutuhkan oleh manusia dalam menunjang kesehatan tubuh sangat banyak pada buah ini.

Mutu buah merupakan hal utama yang sangat penting dalam hasil produksi buah-buahan. Setiap buah memiliki ciri yang digunakan dalam menentukan mutu/kualitasnya, misalnya dilihat dari ukuran, warna kulit serta kerusakan/cacat pada kulit. Mutu pada buah jeruk dibedakan menjadi 3 kelas yaitu *Grade Super*, *Grade A* dan *Grade B*. Saat ini, klasifikasi mutu buah jeruk masih dilakukan secara manual oleh petani, penjual dan juga konsumen. Kemudian pada suatu industri buah jeruk, proses *packing* (pengemasan) jeruk tersebut masih dilakukan penyortiran jeruk berdasar mutu/kualitas secara manual sehingga membutuhkan waktu yang lama. Dengan proses klasifikasi secara manual ini menimbulkan subjektifitas dalam penentuan mutu jeruk. Hasil klasifikasi mutu oleh seorang petani dengan petani lainnya dapat berbeda-beda, sehingga diperlukan suatu peralatan untuk menentukan standar baku klasifikasi mutu. Dalam hal ini dilakukan penerapan pada komputer agar dapat mengenali mutu buah jeruk secara objektif sesuai dengan standar klasifikasi mutu buah jeruk.

Kemajuan teknologi komputer termasuk interaksi antara manusia dengan komputer pada saat ini sudah berkembang sangat pesat. Bahkan dibidang pertanian penggunaan teknologi komputer sudah banyak digunakan. Seperti deteksi penyakit, klasifikasi mutu, penentuan berat dan identifikasi jenis buah-buahan maupun sayuran. Dengan perkembangan teknologi tersebut, banyak gagasan tentang klasifikasi mutu buah berbasis pengolahan citra digital. Teknologi tersebut dapat menghasilkan suatu sistem yang dapat menggantikan peran manusia. Penggunaan kamera dapat menggantikan peran indra penglihatan manusia. Informasi bisa didapatkan dari objek yang ditangkap oleh kamera menggunakan pengolahan citra. Berdasarkan permasalahan diatas, dapat dibuat suatu sistem klasifikasi mutu buah jeruk.

Klasifikasi merupakan cara mengelompokkan objek pada variabel tertentu. Terdapat banyak algoritme/metode klasifikasi seperti *Artificial Neural Network (ANN)*, *Decision Tree*, *Support Vector Machine (SVM)*, *K-Nearest Neighbor (KNN)*, *Fuzzy* dan lain sebagainya. Dalam proses pengklasifikasian mutu buah jeruk diperlukan pengolahan dari citra digital buah jeruk. Data citra buah jeruk digunakan dalam proses pelatihan sistem agar dapat mengklasifikasikan mutu. Maka diperlukan metode ekstraksi ciri yang baik agar mendapatkan nilai tekstur yang bagus. Dengan memanfaatkan hasil dari metode ekstraksi ciri maka dapat digunakan untuk pelatihan dan dapat mengklasifikasikan mutu buah jeruk pada suatu data uji.

Pada penelitian sebelumnya, dilakukan klasifikasi mutu buah pisang menggunakan tekstur warna dan tekstur buah (Wiharja & Harjoko, 2014). Penelitian ini menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)* untuk analisis tekstur. Klasifikasi menggunakan metode *backpropagation* berdasarkan warna, tekstur dan cacat pada citra buah pisang. Pada penelitian tersebut, mutu buah pisang dibagi menjadi 5 kelas dengan menggunakan 125 citra pisang sebagai data latih dan 100 citra pisang sebagai data uji dan berhasil dilakukan dengan akurasi sebesar 94%. Penelitian lain tentang identifikasi penyakit pada daun tebu (Dewi & Ginardi, 2014). Identifikasi dilakukan dengan ciri tekstur dan warna. Ekstraksi tekstur menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)* dan ekstraksi warna menggunakan metode *color moments* dengan metode klasifikasi *Support Vector Machine (SVM)*. Dari citra daun tebu dilakukan identifikasi terhadap penyakit karat, penyakit mosaic atau normal. Hasil akurasi didapatkan nilai sebesar 97% pada identifikasi penyakit daun tebu.

Penelitian lainnya yang dilakukan yaitu klasifikasi citra jenis daging berdasarkan tekstur (Neneng, et al., 2016). Digunakan metode *Gray Level Co-occurrence Matrix (GLCM)* untuk ekstraksi ciri tekstur dan menggunakan metode klasifikasi yaitu *Support Vector Machine (SVM)*. Sistem ini melakukan klasifikasi dengan data berupa citra daging kambing, daging kerbau, daging kuda dan daging sapi dan berhasil dilakukan dengan akurasi sebesar 87,5% pada jarak pengambilan gambar 20 cm, kemudian jarak ketetanggaan piksel 2 pada arah GLCM 135°. Kemudian penelitian lain yaitu deteksi cacat pada kain yang dikenal dengan *fabric defect detection system (FDSS)* menggunakan metode *gray level co-occurrence matrix (GLCM)* dan *Gabor filter* (Siqueira, et al., 2013). Ekstraksi tekstur GLCM diplot terhadap jarak antar-pixel GLCM sebagai grafik sinyal. Informasi dari citra kain tidak cacat dibandingkan dengan data uji. Pada metode *Gabor filter*, menghasilkan kumpulan dari *Gabor filter* dengan skala dan orientasi berbeda dan memfilter citra kain dengan *convolution mask*. Metode GLCM menghasilkan nilai akurasi deteksi cacat (*defect detection*) yang lebih tinggi dibandingkan dengan *Gabor filter*, juga GLCM unggul dalam efisiensi komputasi.

Pada penelitian lain dilakukan identifikasi jenis buah jeruk berdasarkan tekstur (Resita, et al., 2016). Penelitian ini menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)* dan metode klasifikasi *backpropagation*. Data yang

digunakan adalah citra buah jeruk jeruk lemon, jeruk mandarin, jeruk medan, jeruk nipis dan jeruk sunkist. Hasil akurasi menghasilkan nilai 60% untuk jarak foto antara kamera dengan objek sebesar 8cm, 56% untuk jarak 10cm dan pada jarak 6cm tidak memberikan hasil. Penelitian lainnya yaitu identifikasi jenis buah jeruk menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi ciri dan metode klasifikasi Naïve Bayes (Agustian, et al., 2017). Jenis-jenis buah yang diidentifikasi adalah jeruk Pontianak, jeruk baby, jeruk mandarin dan jeruk orange. Hasil akurasi mendapatkan nilai 91,6% pada identifikasi jenis buah jeruk tersebut.

Berdasarkan paparan penelitian diatas, didapatkan bahwa metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Support Vector Machine* (SVM) menunjukkan nilai akurasi yang tinggi. Metode *Gray Level Co-occurrence Matrix* (GLCM) merupakan metode yang telah terbukti menjadi descriptor tekstur yang efektif (Siqueira, et al., 2013) serta memiliki akurasi dan waktu komputasi yang lebih baik dari metode ekstraksi tekstur lainnya. *Support Vector Machine* (SVM) merupakan metode klasifikasi dengan kemampuan generalisasi yang tinggi juga dengan dimensi dari ruang input yang tinggi (Neneng, et al., 2016). Oleh karena itu penulis mengambil penelitian yang berjudul “Pemanfaatan Ciri *Gray Level Co-occurrence Matrix* (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu”. Ekstraksi ciri dengan metode *Gray Level Co-occurrence Matrix* (GLCM) yang digunakan adalah *contrast*, *homogeneity*, *energy* dan *entropy*. GLCM memiliki empat arah sudut yaitu, 0°, 45°, 90° dan 135°. Pada penelitian ini dilakukan pengujian terhadap empat arah GLCM tersebut. Kemudian menggunakan metode *Support Vector Machine* (SVM) untuk mengidentifikasi cacat atau tidaknya suatu area *windowing* citra. Klasifikasi dilakukan kedalam tiga kelas yaitu *Grade Super*, *Grade A* dan *Grade B* berdasarkan persentase cacat buah. Hasil dari penelitian ini adalah arah GLCM terbaik berdasarkan pengaruh empat arah GLCM tersebut terhadap hasil klasifikasi mutu. Harapannya kedepan sistem ini dapat diimplementasikan dalam pembuatan alat klasifikasi mutu buah jeruk secara otomatis oleh Balitjestro, sehingga nantinya dapat membantu suatu sentra produksi jeruk dalam melakukan klasifikasi mutu buah jeruk yang objektif, serta dapat mempermudah dan mempercepat proses klasifikasi mutu buah jeruk.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana memanfaatkan ciri *Gray Level Co-occurrence Matrix* (GLCM) citra buah jeruk keprok (*Citrus reticulata* Blanco) untuk klasifikasi mutu?
2. Bagaimana tingkat akurasi dari pemanfaatan ciri *Gray Level Co-occurrence Matrix* (GLCM) untuk klasifikasi mutu buah jeruk keprok (*Citrus reticulata* Blanco)?

1.3 Tujuan

Dari rumusan masalah diatas, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Memanfaatkan ciri *Gray Level Co-occurrence Matrix* (GLCM) citra buah jeruk keprok (*Citrus reticulata* Blanco) untuk klasifikasi mutu.
2. Menguji tingkat akurasi dari pemanfaatan ciri *Gray Level Co-occurrence Matrix* (GLCM) untuk klasifikasi mutu buah jeruk keprok (*Citrus reticulata* Blanco).

1.4 Manfaat

Manfaat yang didapatkan dari penelitian ini adalah sebagai berikut:

1. Algoritme dapat diimplementasikan pada pembuatan alat klasifikasi mutu secara otomatis oleh Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro).
2. Dapat mengklasifikasi mutu buah jeruk secara objektif sesuai dengan standar yang telah ditetapkan.
3. Dapat mempersingkat waktu klasifikasi mutu dari buah jeruk.

1.5 Batasan Masalah

Berdasarkan rumusan masalah diatas, maka penulis membatasi pembahasan pada penelitian ini sebagai berikut:

1. Sistem ini hanya melakukan pengujian metode *Gray Level Co-occurrence Matrix* (GLCM) terhadap hasil klasifikasi mutu buah jeruk keprok (*Citrus reticulata* Blanco) untuk mendapatkan arah GLCM yang menghasilkan akurasi terbaik.
2. Citra yang menjadi masukan adalah buah jeruk Keprok Batu 55 yang di-*capture* menggunakan kamera *iPhone 6s Plus* dengan resolusi 4032x3024 piksel dalam format gambar JPEG.
3. Pengambilan citra buah jeruk dengan jarak kamera dengan objek adalah 15 *centimeter* (cm) dari sisi atas buah jeruk.
4. Proses pengerjaan menggunakan bahasa pemrograman *Python 3* dengan *library OpenCV* untuk pengolahan citra digital, *NumPy* untuk pengolahan matriks dan *scikit-learn* untuk klasifikasi.
5. Pengklasifikasian mutu buah jeruk berdasarkan persentase cacat buah dan dibagi menjadi 3 kelas mutu yaitu, *Grade Super*, *Grade A* dan *Grade B*.

1.6 Sistematika Pembahasan

Sistematika dalam penulisan pada setiap bab adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang penulisan, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

BAB II LANDASAN KEPUSTAKAAN

Membahas tentang landasan teori/kajian pustaka yang terkait dengan penelitian. Menjelaskan teori yang berhubungan dengan permasalahan serta teori dasar dalam Pengolahan Citra Digital dan teori tentang mutu buah jeruk, metode *Gray Level Co-occurrence Matrix* (GLCM) dan *Support Vector Machine* (SVM).

BAB III METODOLOGI

Menjelaskan tentang alur kerja yang dilakukan dalam penulisan yaitu studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi dan pengujian & analisis.

BAB IV PERANCANGAN

Menjelaskan tentang perancangan algoritme, perancangan skenario pengujian dan manualisasi dari sistem pemanfaatan ciri *Gray Level Co-occurrence Matrix* (GLCM) citra buah jeruk keprok (*Citrus reticulata* Blanco) untuk klasifikasi mutu.

BAB V IMPLEMENTASI

Menjelaskan tentang implementasi sistem berdasarkan perancangan yang telah diuraikan pada bab sebelumnya.

BAB VI PENGUJIAN

Menjelaskan tentang hasil pengujian serta analisis hasil yang diperoleh dari pengujian tersebut.

BAB VII PENUTUP

Berisi kesimpulan dari keseluruhan uraian bab-bab sebelumnya dan saran-saran yang diharapkan dapat bermanfaat untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab 2 ini dijelaskan mengenai kajian pustaka dari penelitian terdahulu yang sudah pernah dilakukan, serta dasar teori dari berbagai sumber pustaka yang berkaitan dan menunjang penelitian ini dengan tujuan untuk digunakan sebagai referensi dalam proses pengembangan penelitian ini.

2.1 Kajian Pustaka

Penelitian dengan menggunakan metode GLCM yang sudah berhasil dilakukan sebelumnya yaitu oleh Yanuar Putu Wiharja dan Agus Harjoko. Penelitian ini mengklasifikasikan mutu buah pisang kedalam 5 kelas, yaitu super, kelas A, kelas B, luar mutu I dan luar mutu II. Pengklasifikasian berdasarkan warna dan tekstur buah pisang yang diekstrak dari citra buah pisang. Menggunakan 125 pisang sebagai data latih dan 100 pisang sebagai data uji. Data latih yang akan digunakan terlebih dahulu dilakukan proses sortasi mutu dan pengukuran panjang buah pisang. Lalu, pisang yang telah di sortir dan diukur panjangnya tersebut diambil citranya dengan menggunakan kotak pengambilan citra. Didalam kotak yang terbuat dari kayu tersebut terdapat lampu, objek buah pisang dan juga ditempatkan sebuah kamera untuk mengambil citra. Dengan menggunakan kotak diharapkan dapat mengambil citra dengan pencahayaan dan jarak kamera dengan objek yang sama. Citra yang didapat diubah menjadi citra grayscale. Selanjutnya dilakukan deteksi cacat dan analisis tekstur dengan GLCM yang menghasilkan nilai energy, homogeneity dan contrast. Kemudian untuk klasifikasi mutu pisang menggunakan jaringan syaraf tiruan yaitu backpropagation. Sebagai masukan untuk jaringan syaraf tiruan digunakan 7 parameter, yaitu nilai red, green dan blue, persentase luas area cacat, energy, homogeneity dan contrast. Keluarannya berupa 5 kelas mutu. Penelitian ini menghasilkan nilai akurasi sebesar 94% dari 100 data uji citra pisang. Dimana berdasar hasil tersebut penelitian ini berhasil dilakukan dengan baik (Wiharja & Harjoko, 2014).

Dalam penelitian lain yang dilakukan oleh Neneng, Kusworo Adi dan R. Rizal Isnanto berhasil menggunakan metode GLCM dan SVM. Dilakukan klasifikasi citra jenis daging dimana sistem pengenalan ini menggunakan data berupa citra daging kambing, daging kerbau, daging kuda dan daging sapi. Data tersebut diambil dengan jarak antara kamera dan objek sebesar 20 cm, 30 cm dan 40 cm. Didapatkan data untuk masing-masing citra daging sebanyak 150 citra. Sehingga total keseluruhan adalah 1800 citra. Kemudian dilakukan ekstraksi ciri ciri dengan metode GLCM dengan menggunakan empat arah GLCM yaitu 0° , 45° , 90° dan 135° . Proses klasifikasi menggunakan metode SVM. Berdasarkan ketiga jarak pengambilan citra tersebut dilakukan pencarian pengaruh jarak ketetanggaan piksel terhadap hasil klasifikasi. Hasilnya diperoleh akurasi sebesar 87,5% pada jarak pengambilan gambar 20 cm, kemudian jarak ketetanggaan piksel 2 pada arah GLCM 135° dan berada pada parameter sigma 0,4 (Neneng, et al., 2016).

Penelitian lain yang berhasil dilakukan yaitu oleh Jagdish Lal Raheja, Sunil Kumar, dan Ankit Chaudhary. Penelitian ini mendeteksi cacat pada kain yang

dikenal dengan *fabric defect detection system* (FDSS). Menggunakan metode *gray level co-occurrence matrix* (GLCM) dan *Gabor filter*. Pada penelitian ini menyuguhkan skema baru untuk FDSS otomatis dengan menggunakan GLCM dan dibandingkan dengan *Gabor filter*. Ekstraksi tekstur GLCM diplot terhadap jarak antar-pixel GLCM sebagai grafik sinyal. Informasi dari citra kain tidak cacat dibandingkan dengan data uji. Pada metode *Gabor filter*, menghasilkan kumpulan dari *Gabor filter* dengan skala dan orientasi berbeda dan memfilter citra kain dengan *convolution mask*. Metode GLCM menghasilkan nilai akurasi deteksi cacat (*defect detection*) yang lebih tinggi dibandingkan dengan *Gabor filter*, juga GLCM unggul dalam efisiensi komputasi (Siqueira, et al., 2013).

Penelitian lainnya mengenai GLCM yang sudah berhasil dilakukan yaitu oleh Reni Resita, Juratminingsih dan Gasim. Pada penelitian ini dilakukan identifikasi jenis buah jeruk berdasarkan tekstur dari buah jeruk tersebut. Menggunakan 5 jenis buah jeruk yaitu jeruk lemon, jeruk mandarin, jeruk medan, jeruk nipis dan jeruk sunkist. Pengambilan citra buah jeruk menggunakan kamera handphone dengan jarak antara objek dan kamera sebesar 6 cm, 8 cm dan 10 cm. Diambil 25 citra dari masing-masing jenis jeruk. Sehingga total data yang dikumpulkan adalah 375 citra. Citra yang akan digunakan diubah menjadi citra grayscale. Kemudian dilakukan ekstraksi ciri dengan metode GLCM dan didapatkan nilai ciri yaitu energi, entropi, homogenitas dan kontras. Nilai tersebut menjadi masukan untuk proses klasifikasi yang menggunakan metode Jaringan Syaraf Tiruan (JST) Backpropagation. Dari 4 masukan tersebut dilakukan proses klasifikasi yang menghasilkan keluaran berupa 5 jenis jeruk. Hasilnya diperoleh akurasi sebesar 60% untuk JST jarak 8 cm, 56% untuk JST jarak 10 cm dan JST jarak 6 cm tidak dapat memberikan hasil (Resita, et al., 2016).

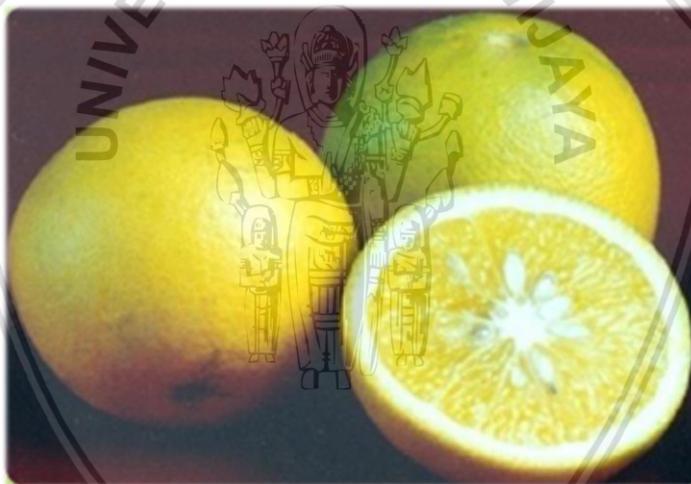
Dalam penelitian oleh Wildan Agustian, Dra. Sri Setyaningsih, M.Si dan Arie Qur'ania, M.Kom berhasil menggunakan metode GLCM. Penelitian ini melakukan klasifikasi 4 jenis buah jeruk. Jenis-jenis buah jeruk yang digunakan adalah jeruk pontianak, jeruk baby, jeruk mandarin dan jeruk orange. Data diambil menggunakan kamera dengan jumlah 52 citra dari keempat jenis jeruk. Data tersebut dibagi menjadi 40 citra data latih dan 12 citra data uji. Proses ekstraksi ciri menggunakan metode GLCM dilakukan untuk menghasilkan nilai ciri entropi, energi, kontras dan homogenitas. Kemudian dilakukan normalisasi warna untuk mendapatkan nilai setiap komponen warna RGB tanpa pengaruh penerangan. Didapatkan persentase nilai red, green, blue dari setiap citra jeruk. Setelah itu, dengan menggunakan metode Naïve Bayes didapatkan nilai rata-rata dan standar deviasi dari parameter setiap jenis jeruk, probabilitas kemunculan tiap parameter, dan likelihood klasifikasi. Hasilnya diperoleh akurasi sebesar 91,6% (Agustian, et al., 2017).

Penelitian lain yang telah berhasil dilakukan oleh Ratih Kartika Dewi dan R.V. Hari Ginardi juga menggunakan metode GLCM dan SVM. Penelitian ini mengidentifikasi penyakit pada daun tebu. Pengambilan data citra daun tebu menggunakan kamera digital yang kemudian pada tiap citra daun tebu diberi label sesuai dengan kelas penyakit, yaitu kelas 1 penyakit karat, kelas 2 penyakit mosaik

dan kelas 3 tebu normal. Data yang berhasil dikumpulkan tersebut dibagi menjadi data latih dan data uji. Kemudian dilakukan preprocessing yaitu proses transformasi citra RGB menjadi citra keabuan dan citra LAB. Selanjutnya ekstraksi ciri tekstur menggunakan metode GLCM dan ekstraksi ciri warna menggunakan color moments. Dengan GLCM menghasilkan nilai ciri energy, homogeneity, correlation dan variance. Dikombinasikan dengan ciri warna color moments 1, 2 dan 3. Nilai-nilai tersebut menjadi masukan untuk klasifikasi, digunakan metode SVM dan hasilnya mendapatkan nilai akurasi sebesar 97% (Dewi & Ginardi, 2014).

Dari paparan diatas, penulis memutuskan untuk menggunakan metode gray level co-occurrence matrix (GLCM) sebagai metode ekstraksi ciri tekstur, karena telah terbukti dapat menjadi deskriptor tekstur yang baik dan cocok untuk diimplementasikan pada kasus ekstraksi tekstur pada buah jeruk. Selain itu pada penelitian sebelumnya juga didapatkan bahwa metode GLCM menghasilkan ekstraksi tekstur dan akurasi yang lebih baik dibandingkan dengan *Gabor filter*. Penjelasan mengenai objek, metode dan hasil dari literatur dalam bentuk tabel dapat dilihat pada Tabel 2.1.

2.2 Buah Jeruk



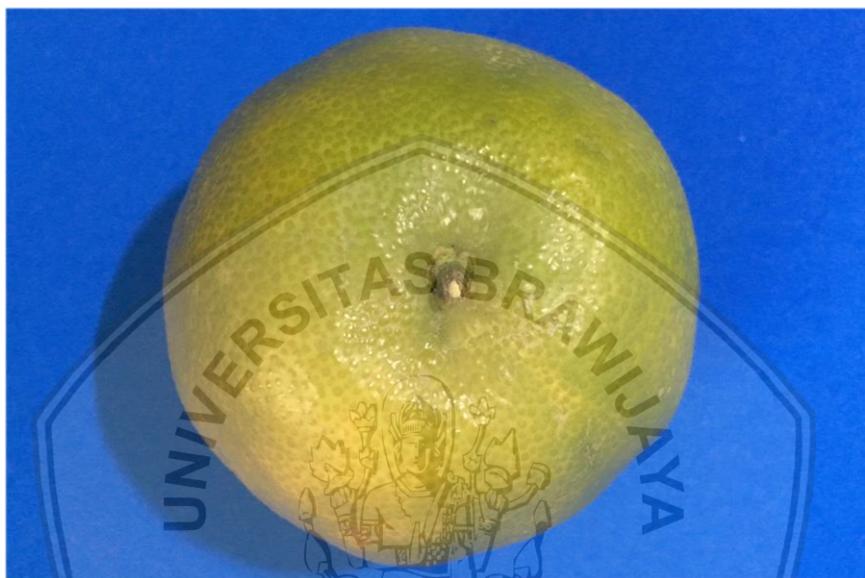
Gambar 2.1 Buah Jeruk

Sumber: (Endarto & Martini, 2016)

Buah Jeruk seperti yang terlihat pada Gambar 2.1 yang memiliki nama latin *Citrus L.* ini adalah tanaman yang berasal dari Asia. Tanaman jeruk tersebar dan banyak terdapat pada Kawasan India sampai dengan China. Di Indonesia sendiri terdapat beberapa jenis jeruk yang dikembangkan, seperti jeruk yang berasal dari Asia Timur yaitu jeruk manis dan jeruk sitrun, dan dari Asia Tenggara yaitu jeruk nipis, jeruk purut dan jeruk pamento. Buah Jeruk memiliki keunggulan karena dapat dikembangkan pada lahan subur, lahan kering, rawa (pasang surut) maupun sawah. Pengembangan buah jeruk dapat berhasil dilakukan apabila dengan teknologi budi daya yang benar. Untuk dapat mengembangkan tanaman jeruk dengan baik, terdapat beberapa syarat untuk tempat pengembangan, yaitu

ketinggian tempat mulai 0-1000 mdpl, suhu 13°-35° C dengan suhu optimum pada 22°-23° C, curah hujan 1000-3000 mm/th dengan curah hujan optimum pada 1500-2500 mm/th, 2-6 bulan kering dengan optimum pada 3-4 bulan, dan tanah dengan pH 5-8 atau optimum pada pH 6. Buah jeruk selain kaya akan kandungan vitamin C, juga memiliki manfaat sebagai wewangian parfum dan daunnya memiliki aroma khas dari kandungan *flavonoid* dan *terpenoid* yang dapat digunakan sebagai rempah-rempah (Endarto & Martini, 2016).

2.3 Jeruk Keprok (*Citrus reticulata* Blanco)



Gambar 2.2 Jeruk Keprok

Jeruk keprok seperti yang terlihat pada Gambar 2.2 memiliki nama latin *Citrus reticulata* Blanco. Jenis jeruk ini merupakan salah satu dari tiga jeruk komersial unggulan di Indonesia, yaitu jeruk siam dan jeruk pamelon. Jeruk ini unggul karena memiliki rasa manis, tekstur daging buah yang lunak dan berair banyak. Jeruk keprok juga memiliki ciri khas yaitu permukaannya yang halus, mudah dikupas karena terdapat rongga antara kulit dan daging buah, serta memiliki aroma khas yang segar ketika dikupas. Jeruk keprok mempunyai kandungan gizi yang sangat baik, yaitu 420 IU vitamin A, 0,07 mg vitamin B, 31 mg vitamin C, 0,1 gram lemak, 0,5 gram protein, asam sitrit, asam maleik, karotin dan glukosida serta dapat digunakan sebagai obat pembersih dahak (Astuti, 2015). Terdapat beberapa sentra produksi buah jeruk keprok di Indonesia, yaitu Batu, Banyuwangi, Jember, Garut, Bali, dan Sulawesi Selatan (Endarto & Martini, 2016).

2.4 Mutu Buah Jeruk Keprok

Sebagai jeruk lokal yang tidak hanya diminati di dalam negeri namun juga di luar negeri, jeruk keprok harus bermutu baik agar dapat bersaing dengan jeruk-jeruk lainnya dan dapat diterima oleh pasar negara lain. Standar Nasional Indonesia (SNI) telah menetapkan untuk standar kualitas jeruk keprok pada SNI

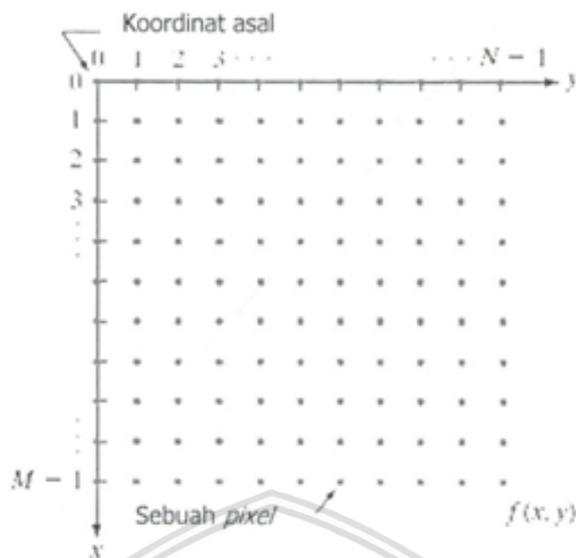
01-3165-1992 yang sekarang telah diperbaiki pada SNI 3165-2009. Standar ini menetapkan beberapa ketentuan meliputi ketentuan minimum, kematangan, pengkelasan mutu, ukuran, pengemasan, residu pestisida, cemaran logam berat dan higienis. Berdasarkan standar tersebut, jeruk keprok dapat diklasifikasikan mutunya dengan batasan toleransi pada tiap mutu. Jeruk keprok dibagi kedalam 3 kelas, yaitu Grade Super, Grade A dan Grade B. Jeruk keprok *Grade Super* merupakan jeruk yang bebas dari kerusakan kecuali kerusakannya sangat kecil. Kemudian untuk Grade A dan Grade B memiliki toleransi kerusakan kecil, yaitu sedikit penyimpangan pada bentuk, warna kulit dan pada kulit terkait dengan pembentukan buah, serta sedikit cacat pada kulit. *Grade A* memiliki ketentuan cacat maksimum sebesar 10% dan *Grade B* memiliki ketentuan cacat maksimum sebesar 15% (Standar Nasional Indonesia, 2009).

Selain ketentuan diatas, terdapat juga ketentuan minimum sebagai jeruk konsumsi yang harus dipenuhi, yaitu utuh (buah sempurna), padat, berpenampilan segar, layak untuk dikonsumsi, bersih dan bebas dari benda asing, bebas hama dan penyakit buah, serta bebas dari aroma dan rasa asing. Kemudian terdapat ketentuan ukuran buah yang dibedakan menjadi 4 kode ukuran, yaitu untuk kode ukuran 1 dengan >70mm, kode ukuran 2 dengan 61-70mm, kode ukuran 3 dengan 51-60mm dan kode ukuran 4 dengan 40-50mm. Ukuran tersebut diukur berdasarkan diameter maksimum buah (Standar Nasional Indonesia, 2009).

2.5 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) merupakan disiplin ilmu yang mempelajari hal-hal mengenai perbaikan kualitas gambar, transformasi gambar dan memilih ciri citra (*feature images*) yang optimal (Sutoyo, et al., 2009). Perbaikan kualitas gambar yang dilakukan meliputi peningkatan kontras, transformasi warna dan restorasi citra. Transformasi gambar meliputi rotasi, translasi, skala dan transformasi geometrik. Tujuan dari pengolahan citra digital ini adalah untuk analisis, melakukan pengambilan informasi atau deskripsi dari objek, pengenalan objek pada citra, melakukan reduksi data atau kompresi data dengan tujuan penyimpanan data, transmisi data dan waktu pemrosesan data (Sutoyo, et al., 2009).

Sebuah gambar atau citra didefinisikan dengan fungsi dua dimensi $f(x, y)$ yang berukuran M baris dan N kolom, dimana x dan y merupakan koordinat spasial serta nilai fungsi f pada titik koordinat (x, y) dinamakan intensitas/tingkat keabuan dari citra di titik tersebut. Suatu gambar atau citra dapat disebut sebagai citra digital apabila keseluruhan nilai x , y , dan f berhingga (*finite*) dan diskrit (Putra, 2010). Nilai pada posisi (x, y) (persilangan antara baris dan kolom) disebut dengan piksel. Ilustrasi posisi koordinat citra digital ditunjukkan pada Gambar 2.3.



Gambar 2.3 Koordinat Citra Digital

Sumber: (Putra, 2010)

2.6 Gray Level Co-occurrence Matrix (GLCM)

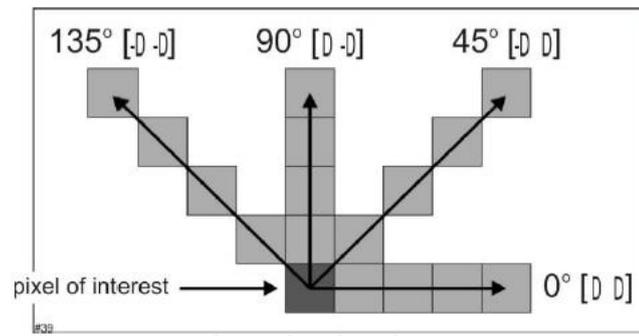
Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra (Prasetyo, 2011).

GLCM adalah matriks yang menggunakan perhitungan tekstur histogram tingkat kedua/pada orde kedua yang memperhitungkan hubungan antar pasangan dua piksel citra asli, sedangkan pada orde pertama tidak memperhitungkan hubungan ketetanggaan piksel serta menggunakan perhitungan statistik hanya berdasarkan nilai piksel citra asli, seperti varians (Kadir & Susanto, 2013).

GLCM berukuran $L \times L$ dimana L menyatakan tingkat keabuan, dengan elemen $P(x_1, x_2)$ yang merupakan distribusi probabilitas (*joint probability distribution*) dari pasangan piksel yang memiliki tingkat keabuan x_1 pada koordinat (j, k) dan x_2 pada koordinat (m, n) (Putra, 2010). $P(x_1, x_2)$ dihitung dengan persamaan 2.1.

$$P(x_1, x_2) = \frac{\text{Banyaknya pasangan titik - titik dengan tingkat keabuan } x_1 \text{ dan } x_2}{\text{Banyaknya titik pada daerah suatu citra}} \quad (2.1)$$

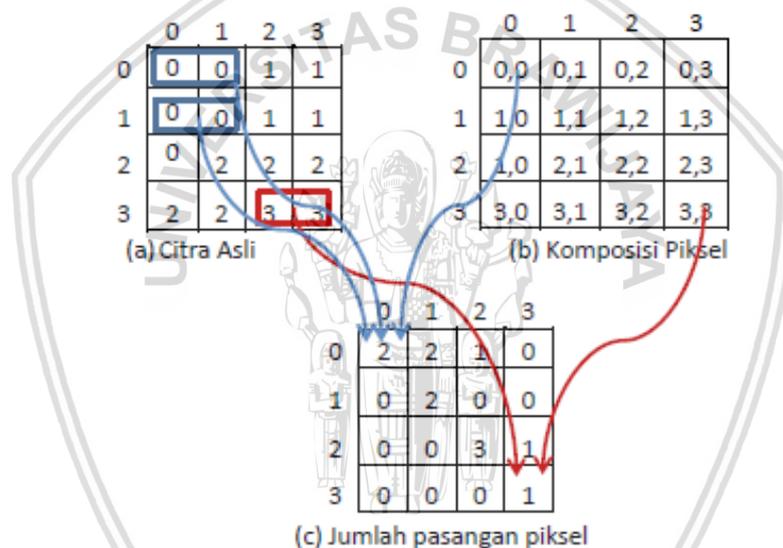
Koordinat pasangan piksel tersebut memiliki jarak d dan orientasi sudut θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah sudut yaitu, 0° , 45° , 90° dan 135° , dan jarak antar piksel sebesar 1 piksel (Surya, et al., 2017). Ilustrasi empat arah tersebut dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arah Kookurensi GLCM

Sumber: (Surya, et al., 2017)

Nilai awal matriks GLCM didapatkan dari pasangan dua piksel yang berjajar horizontal. Ilustrasi perhitungan penentuan awal matriks GLCM dapat dilihat pada Gambar 2.5.



Gambar 2.5 Penentuan Nilai Awal Matriks GLCM

Sumber: (Kadir & Susanto, 2013)

Matriks pada Gambar 2.5(c) adalah matriks *framework* yang perlu diolah agar menjadi matriks simetris dengan cara menjumlahkan dengan nilai matriks transposnya. Pembentukan matriks GLCM yang simetris ditunjukkan pada Gambar 2.6.

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

↑ Transpos ↑ GLCM sebelum dinormalisasi

Gambar 2.6 Pembentukan Matriks GLCM yang Simetris

Sumber: (Kadir & Susanto, 2013)



Kemudian matriks tersebut perlu dinormalisasikan agar jumlah seluruh elemen matriks bernilai 1. Cara yang dilakukan adalah membagi setiap elemen dengan bilangan yang merupakan jumlah total dari pasangan piksel. Contoh normalisasi dapat dilihat pada Gambar 2.7.

$$\begin{pmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{24}{1} & \frac{24}{0} & \frac{24}{6} & \frac{24}{1} \\ \frac{24}{0} & \frac{24}{0} & \frac{24}{1} & \frac{24}{2} \\ \frac{24}{24} & \frac{24}{24} & \frac{24}{24} & \frac{24}{24} \end{pmatrix}$$

Gambar 2.7 Normalisasi Matriks GLCM

Sumber: (Kadir & Susanto, 2013)

Terdapat 4 ciri GLCM yang akan digunakan yang nilainya didapatkan dari matriks GLCM, (Prasetyo, 2011) yaitu:

1. *Contrast*

Contrast merupakan suatu ciri untuk mengukur kekuatan perbedaan intensitas dalam citra. Nilai *contrast* semakin besar apabila variasi intensitas dalam citra tinggi dan sebaliknya nilai *contrast* semakin kecil apabila variasi intensitas dalam citra rendah. *Contrast* didefinisikan pada Persamaan 2.2.

$$Contrast = \sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2) \tag{2.2}$$

2. *Homogeneity*

Homogeneity merupakan suatu ciri untuk mengukur kehomogenan variasi intensitas dalam citra dan merupakan kebalikan dari *contrast*. Nilai *homogeneity* semakin besar apabila variasi intensitas dalam citra nilainya mengecil dan sebaliknya nilai *homogeneity* semakin kecil apabila variasinya membesar. *Homogeneity* didefinisikan pada Persamaan 2.3.

$$Homogeneity = \sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \tag{2.3}$$

3. *Energy*

Energy merupakan suatu ciri untuk mengukur konsentrasi pasangan intensitas pada matriks GLCM. Nilai *Energy* semakin besar apabila pasangan piksel yang telah memenuhi syarat matriks *co-occurrence* terfokus pada beberapa koordinat dan sebaliknya nilai *energy* semakin kecil apabila letaknya menyebar. *Energy* didefinisikan pada Persamaan 2.4.

$$Energy = \sum_{i_1} \sum_{i_2} p^2(i_1, i_2) \tag{2.4}$$

4. Entropy

Entropy merupakan suatu ciri untuk mengukur keteracakan dari distribusi intensitas. Nilai *entropy* maksimum apabila semua elemen matriks sama. Entropy didefinisikan pada Persamaan 2.5.

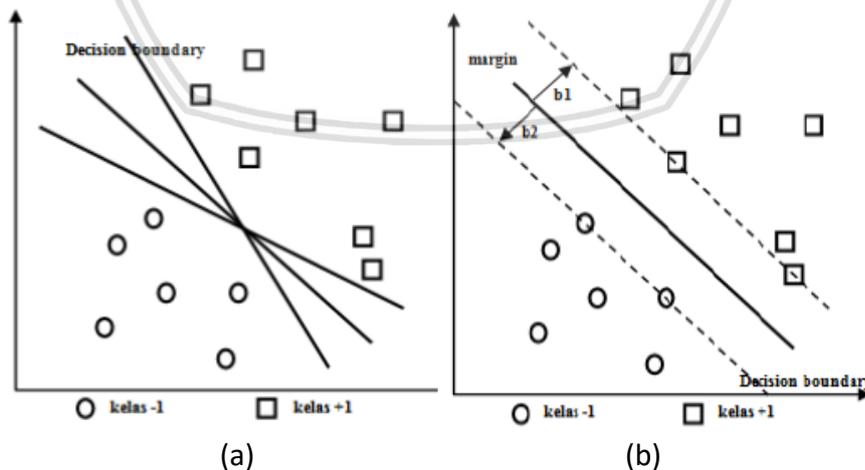
$$Entropy = - \sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2) \tag{2.5}$$

Pada empat persamaan diatas, notasi p adalah probabilitas (0-1) yaitu elemen dalam matriks GLCM, kemudian i_1 dan i_2 adalah pasangan intensitas yang berdekatan, nomor baris dan nomor kolom pada matriks GLCM.

2.7 Support Vector Machine (SVM)

Support Vector Machine adalah metode yang berasal dari teori pembelajaran statistik yang memiliki hasil menjanjikan dan berpotensi memberikan hasil yang lebih baik dari metode klasifikasi lain (Prasetyo, 2014). SVM tidak akan mempelajari seluruh data latih pada proses pelatihan, melainkan hanya sejumlah data terpilih yang akan digunakan untuk membentuk model pada klasifikasi. Berbeda dengan *Nearest Neighbor*, SVM tidak menyimpan seluruh data latih melainkan hanya menyimpan sebagian kecil data latih untuk digunakan dalam prediksi. Hal ini juga menjadi kelebihan dari metode SVM, karena tidak semua data latih dilibatkan pada setiap iterasi pelatihan.

Hal yang dilakukan pada metode SVM adalah memaksimalkan batas *hyperplane* (batas keputusan) ataupun mencari *hyperplane* terbaik sebagai pemisah dua buah kelas data yang diilustrasikan pada Gambar 2.8.



Gambar 2.8 Hyperplane yang Mungkin untuk Set Data

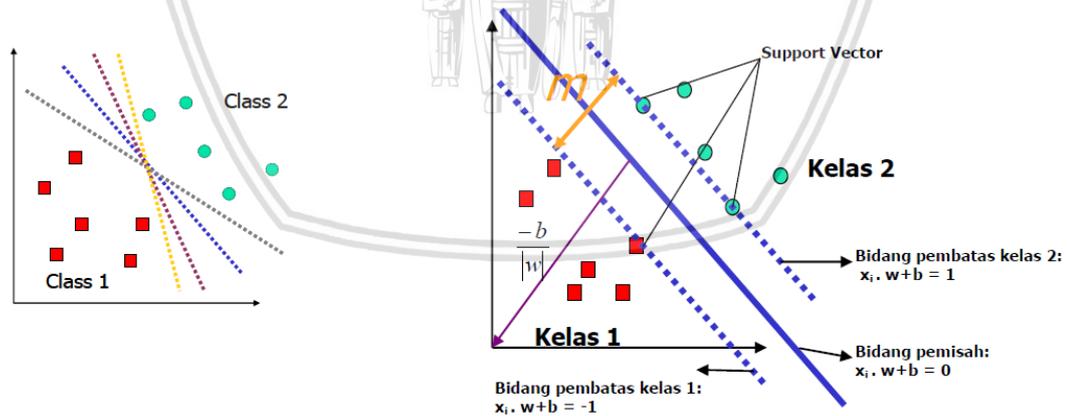
Sumber: (Prasetyo, 2014)



Pada Gambar 2.8(a) terdapat pilihan *hyperplane* yang mungkin untuk set data, kemudian pada Gambar 2.8(b) terdapat *hyperplane* dengan margin yang paling maksimal. Margin adalah jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas. Data terdekat inilah yang disebut dengan *support vector*. *Hyperplane* dengan margin paling maksimal akan memberikan generalisasi lebih baik pada proses klasifikasi. Dapat dilihat pada Gambar 2.8(b), garis solid menunjukkan *hyperplane* terbaik dengan letak pada tengah dari kedua kelas, sedangkan garis putus-putus yang melewati data lingkaran dan bujur sangkar adalah *support vector*. Inti dari pelatihan pada metode SVM adalah usaha dalam mencari lokasi *hyperplane* tersebut (Prasetyo, 2014).

2.7.1 SVM Linear

Data pada SVM *linear* disebut dengan istilah *linearly separable data*, dimana data tersebut adalah data yang dapat dipisahkan secara linier. Dimisalkan terdapat suatu dataset $\{x_1, \dots, x_n\}$ dan $y_i \in \{+1, -1\}$ merupakan label kelas dari data x_i . Pada Gambar 2.9 sebelah kiri terdapat beberapa alternative bidang pemisah yang dapat digunakan sebagai pemisah data set sesuai dengan kelasnya. Dan pada gambar sebelah kanan terdapat ilustrasi dimana dua kelas dapat dipisahkan oleh sepasang bidang pembatas sejajar dan terdapat bidang pemisah diantaranya. Bidang pemisah ini tidak hanya memisahkan data namun juga memiliki margin yang paling besar. Kemudian terdapat bidang pembatas pertama dimana bidang pembatas tersebut membatasi kelas 1 dan bidang pembatas kedua yang membatasi kelas 2. Dapat dilihat bahwa terdapat data baik dari kelas 1 dan kelas 2 yang berada tepat pada bidang pembatas, hal ini disebut *support vector* (Sembiring, 2007).



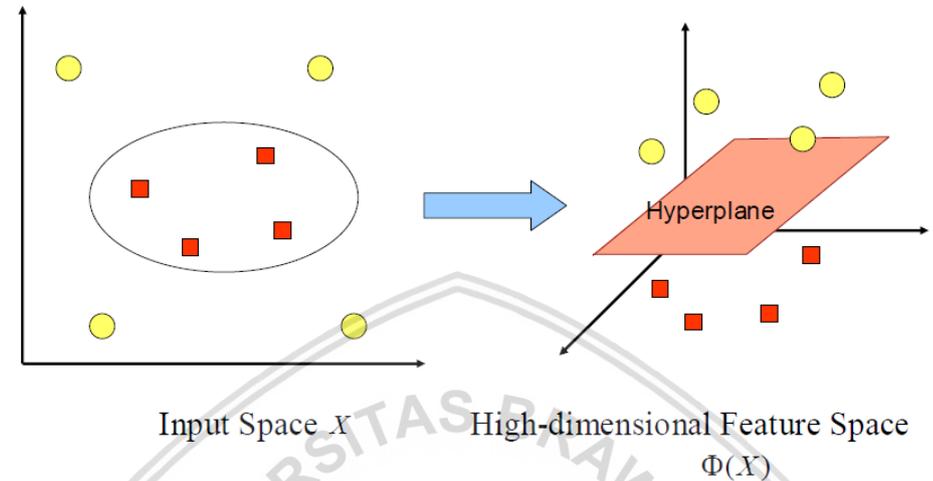
Gambar 2.9 Ilustrasi SVM Linear

Sumber: (Sembiring, 2007)

2.7.2 SVM Non-linear

Dalam SVM *non-linear* untuk menyelesaikan problem-problem yang bersifat *non-linear*, dilakukan modifikasi SVM dengan memasukkan fungsi kernel. Pada Gambar 2.10 bagian kiri terdapat 2 buah kelas data yaitu data kelas kuning dan data kelas merah yang berada pada *input space* dua dimensi dan tidak dapat

dipisahkan secara *linear*. Dan pada gambar bagian kanan menunjukkan *feature space* yang merupakan pemetaan dari sebelumnya yaitu *input space* kedalam ruang vector baru dengan dimensi yang lebih tinggi (tiga dimensi). Ditunjukkan pula bahwa kedua kelas tersebut dapat dipisahkan secara linier oleh *hyperplane* (Nugroho, et al., 2003).



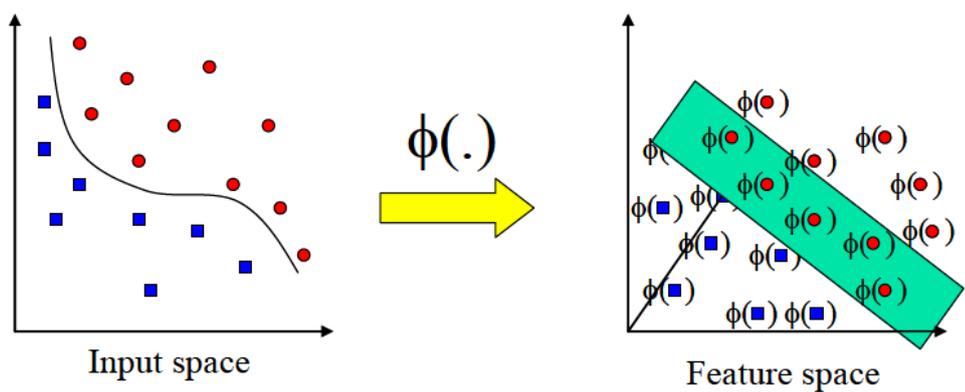
Gambar 2.10 Ilustrasi SVM Non-linear

Sumber: (Nugroho, et al., 2003)

Apabila dimisalkan dengan suatu dataset yang memiliki dua atribut dan dua kelas yaitu positif dan negatif. Data kelas positif $\{(2,2), (2,-2), (-2,2), (-2,-2)\}$ dan data kelas negative $\{(1,1), (1,-1), (-1,1), (-1,-1)\}$. Jika data-data tersebut dipetakan kedalam *input space* dua dimensi seperti yang terlihat pada Gambar 2.11 bagian kiri, maka data tersebut tidak dapat dipisahkan secara linier. Agar data tersebut dapat terpisah sesuai dengan kelasnya oleh bidang pemisah (*hyperplane*) seperti Gambar 2.11 bagian kanan, maka perlu dipetakan *feature space* menggunakan suatu fungsi pemetaan (transformasi) $x_k \rightarrow \Phi(x_k)$. Fungsi transformasinya ditunjukkan pada Persamaan 2.6 berikut ini.

$$\Phi(x_1, x_2) = \begin{cases} \sqrt{x_1^2 + x_2^2} > 2 \rightarrow (4 - x_2 + |x_1 - x_2|, 4 - x_1 + |x_1 - x_2|) \\ \sqrt{x_1^2 + x_2^2} \leq 2 \rightarrow (x_1, x_2) \end{cases} \quad (2.6)$$

Hasil dari transformasi tersebut pada data kelas positif $\{(1,1), (1,-1), (-1,1), (-1,-1)\}$ dan data kelas negative $\{(2,2), (6,2), (6,6), (2,6)\}$. Kemudian dilakukan pencarian *hyperplane* terbaik (Sembiring, 2007).



Gambar 2.11 Ilustrasi Dataset SVM Non-linear

Sumber: (Sembiring, 2007)

Feature space memiliki dimensi yang lebih tinggi sehingga membuat komputasi yang sangat besar, karena terdapat kemungkinan dapat memiliki jumlah *feature* yang tidak terhingga dan sulit mengetahui fungsi transformasi yang tepat untuk digunakan. Solusinya adalah dengan menggunakan *kernel trick*, dimana jika terdapat suatu fungsi kernel $K(x_i, x_d) = \phi(x_i) \cdot \phi(x_d)$, maka fungsi *non-linear* $\phi(x_k)$ tidak perlu diketahui secara persis. Dan fungsi yang dihasilkan terdapat pada Persamaan 2.7 (Sembiring, 2007).

$$f(x_d) = \sum_{i=1}^{n_s} \alpha_i y_i K(x_i, x_d) + b \tag{2.7}$$

Pada penelitian ini diterapkan metode *Support Vector Machine* (SVM) dengan menggunakan *library scikit-learn* dengan kernel *linear*.

Tabel 2.1 Kajian Pustaka

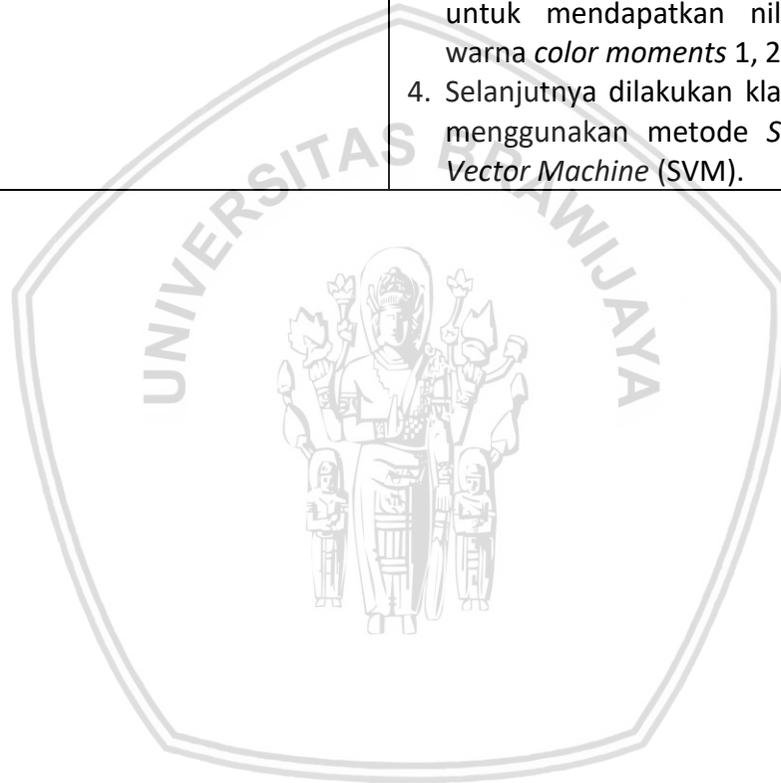
No	Judul	Objek (<i>input</i>)	Metode (<i>proses</i>)	Hasil
1	Pemrosesan Citra Digital untuk Klasifikasi Mutu Buah Pisang Menggunakan Jaringan Saraf Tiruan (Wiharja & Harjoko, 2014).	<i>Input</i> berupa citra buah pisang sebanyak 225 citra, dimana 125 pisang digunakan sebagai data latih dan 100 pisang sebagai data uji. Pisang yang akan digunakan sebagai data penelitian dilakukan sortasi mutu dan pengukuran Panjang terlebih dahulu kemudian dimasukkan kedalam kotak kayu dengan lampu dan dilakukan pengambilan citra menggunakan kamera.	<ol style="list-style-type: none"> 1. Citra awal diubah menjadi citra <i>grayscale</i> dan kemudian dilakukan pengaturan nilai intensitas untuk menonjolkan area cacat. 2. Kemudian dilakukan proses segmentasi yang menghasilkan citra biner dimana warna putih menunjukkan area cacat dan warna hitam bukan area cacat sehingga didapatkan persentase luas area cacat. 3. Selanjutnya menggunakan metode <i>Gray Level Co-occurrence Matrix</i> (GLCM) digunakan untuk analisis tekstur. 4. Nilai <i>energy, homogeneity, contrast, red, green, blue</i> dan persentase luas area cacat didapatkan sebagai masukan untuk proses klasifikasi. 5. Kemudian dilakukan klasifikasi menggunakan metode <i>backpropagation</i>. 	Hasil klasifikasi kedalam 5 kelas mutu yaitu super, kelas A, kelas B, luar mutu I dan luar mutu II berhasil mendapatkan nilai akurasi sebesar 94%.

2	Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri <i>Gray Level Co-Occurrence Matrices</i> (GLCM) (Neneng, et al., 2016).	<i>Input</i> data berupa citra daging yang terdiri dari daging kambing, daging kerbau, daging kuda dan daging sapi. Dengan jumlah 150 citra untuk masing-masing citra daging dan diambil dengan jarak 20 cm, 30 cm dan 40 cm.	<ol style="list-style-type: none"> 1. Dari citra masukan dilakukan pemotongan citra, perubahan ukuran citra dan mengkonversi RGB menjadi <i>grayscale</i>. 2. Dilakukan ekstraksi ciri dengan metode <i>Gray Level Co-occurrence Matrix</i> (GLCM) dengan 4 arah yaitu 0°, 45°, 90° dan 135°. 3. Kemudian dilakukan pelatihan dan pengujian klasifikasi menggunakan metode <i>Support Vector Machine</i> (SVM). 4. Dilakukan pengujian pengaruh jarak ketetangaan piksel dan arah GLCM pada setiap data uji. 	Hasil dari penelitian ini didapatkan nilai akurasi terbaik sebesar 87,5% pada jarak pengambilan citra 20 cm, jarak ketetangaan piksel sebesar 2 pada arah GLCM 135° dan parameter sigma 0,4.
3	Fabric defect detection based on GLCM and Gabor filter: A comparison (Raheja, et al., 2013).	<i>Input</i> berupa citra kain (<i>fabric image</i>).	<ol style="list-style-type: none"> 1. Pada deteksi cacat kain menggunakan GLCM, dilakukan <i>grayscale</i> pada citra uji berukuran 512x512 piksel, kemudian dilakukan pembentukan matriks GLCM yang ternormalisasi dan ekstraksi fitur energi, homogenitas, kontras dan korelasi. 	Hasil yang berhasil didapatkan adalah dibandingkan dengan metode <i>Gabor filter</i> , GLCM dapat menghasilkan akurasi deteksi cacat lebih tinggi dan memiliki waktu komputasi yang lebih efisien.

			<p>2. Pada deteksi cacat kain menggunakan <i>Gabor filter</i>, dari citra masukan diambil fitur <i>Gabor</i>, lalu dikonvolusi menggunakan <i>M*M mask</i> lalu menghasilkan <i>filtered image</i> dan setelah itu hasil deteksi cacat.</p>	
4	<p>Identifikasi Jenis Buah Jeruk Menggunakan Metode Jaringan Syaraf Tiruan Berdasarkan Tekstur Kulit (Resita, et al., 2016).</p>	<p><i>Input</i> berupa data citra jeruk yang terdiri dari 5 jenis buah jeruk, yaitu jeruk lemon, jeruk mandarin, jeruk medan, jeruk nipis dan jeruk sunkist. Data citra diambil dengan jarak 6 cm, 8 cm, dan 10 cm dengan jumlah 25 citra dari masing-masing jenis jeruk sehingga total 375 citra.</p>	<p>1. Citra masukan dikonversi menjadi <i>grayscale</i>. 2. Dilakukan ekstraksi ciri menggunakan metode <i>Gray Level Co-occurrence Matrix (GLCM)</i> dengan jarak ketetanggaan piksel sebesar 1 dan sudut 0° untuk mendapatkan energi, entropi, homogenitas dan kontras. 3. Kemudian dilakukan klasifikasi menggunakan metode Jaringan Syaraf Tiruan (JST) <i>Backpropagation</i>.</p>	<p>Hasil penelitian ini didapatkan nilai akurasi sebesar 60% untuk JST jarak 8 cm, 56% untuk JST jarak 10 cm dan pada JST jarak 6 cm tidak dapat memberikan hasil.</p>
5	<p>Klasifikasi Buah Jeruk Menggunakan Metode <i>Naïve Bayes</i> Berdasarkan Analisis Tekstur dan</p>	<p><i>Input</i> berupa data citra buah jeruk yang terdiri dari jeruk pontianak, jeruk baby, jeruk mandarin dan jeruk orange dengan total 52 citra.</p>	<p>1. Dilakukan ekstraksi ciri menggunakan metode <i>Gray Level Co-occurrence Matrix (GLCM)</i> untuk mendapatkan nilai</p>	<p>Hasil dari penelitian ini didapatkan nilai akurasi sebesar 91,6%.</p>

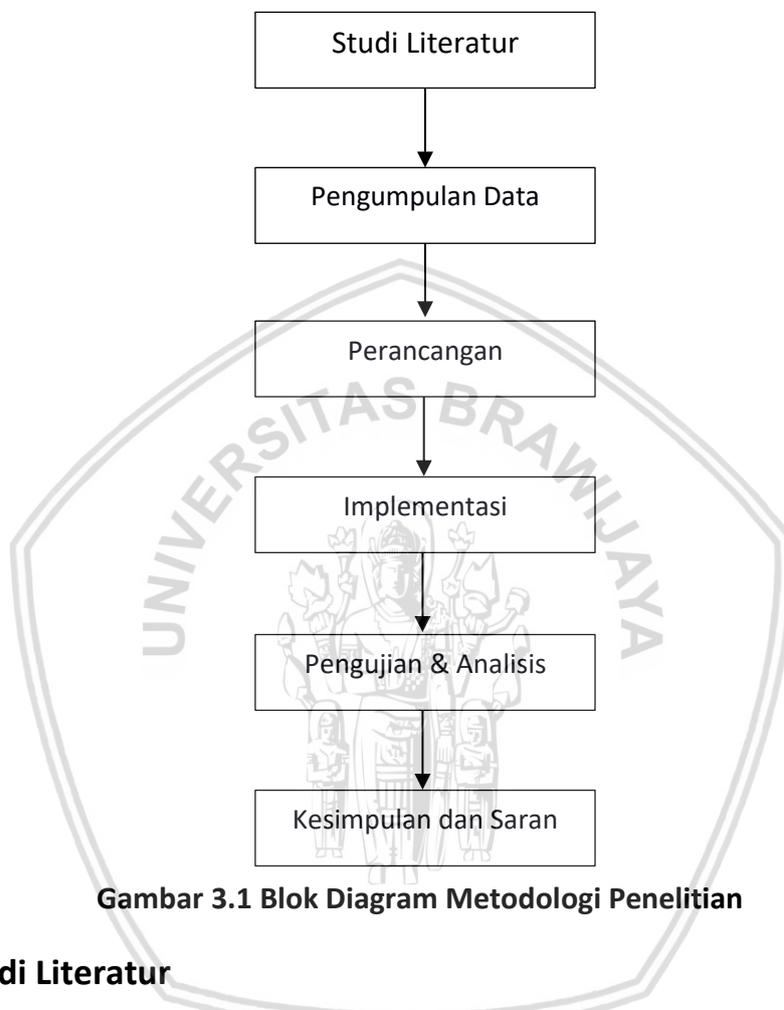
	Normalisasi Warna (Agustian, et al., 2017).		<p>entropi, energi, kontras dan homogenitas.</p> <ol style="list-style-type: none"> 2. Kemudian dilakukan normalisasi warna untuk mendapatkan persentase nilai <i>red, green, blue</i> dari setiap citra jeruk. 3. Selanjutnya dilakukan klasifikasi menggunakan metode Naïve Bayes untuk mendapatkan nilai rata-rata dan standar deviasi dari parameter setiap jenis jeruk, probabilitas kemunculan tiap parameter, dan likelihood klasifikasi 	
6	Identifikasi Penyakit Pada Daun Tebu dengan <i>Gray Level Co-occurrence Matrix</i> dan <i>Color Moments</i> (Dewi & Ginardi, 2014).	<i>Input</i> berupa data citra daun tebu dimana daun tebu tersebut dibagi kedalam tiga kelas penyakit, yaitu kelas 1 penyakit karat, kelas 2 penyakit mosaik dan kelas 3 tebu normal.	<ol style="list-style-type: none"> 1. Dari citra masukan dilakukan <i>pre-processing</i> untuk transformasi RGB menjadi citra keabuan dan citra dalam ruang warna CIELAB. 2. Kemudian dilakukan ekstraksi ciri menggunakan metode <i>Gray Level Co-occurrence Matrix (GLCM)</i> untuk menghasilkan nilai <i>energy, homogeneity, correlation</i> dan <i>variance</i>. 	Hasil dari penelitian ini didapatkan akurasi sebesar 97%.

			<ol style="list-style-type: none">3. Dilakukan ekstraksi ciri warna menggunakan <i>color moments</i> untuk mendapatkan nilai ciri warna <i>color moments</i> 1, 2 dan 3.4. Selanjutnya dilakukan klasifikasi menggunakan metode <i>Support Vector Machine</i> (SVM).	
--	--	--	---	--



BAB 3 METODOLOGI

Pada Bab 3 ini membahas mengenai metodologi penelitian yang digunakan pada penelitian ini. Tahapan-tahapan metodologi yang digunakan digambarkan dengan diagram blok metodologi penelitian seperti pada Gambar 3.1.



Gambar 3.1 Blok Diagram Metodologi Penelitian

3.1 Studi Literatur

Studi literatur bertujuan untuk mendapatkan informasi dengan cara mempelajari beberapa literatur dari bidang ilmu yang terkait dengan penelitian dan atau beberapa penelitian sebelumnya yang terkait dengan klasifikasi objek menggunakan pengolahan citra digital dengan metode *Gray Level Co-occurrence Matrix (GLCM)* dan *Support Vector Machine (SVM)* serta terkait dengan buah jeruk dan mutu buah jeruk keprok melalui buku, skripsi dan jurnal.

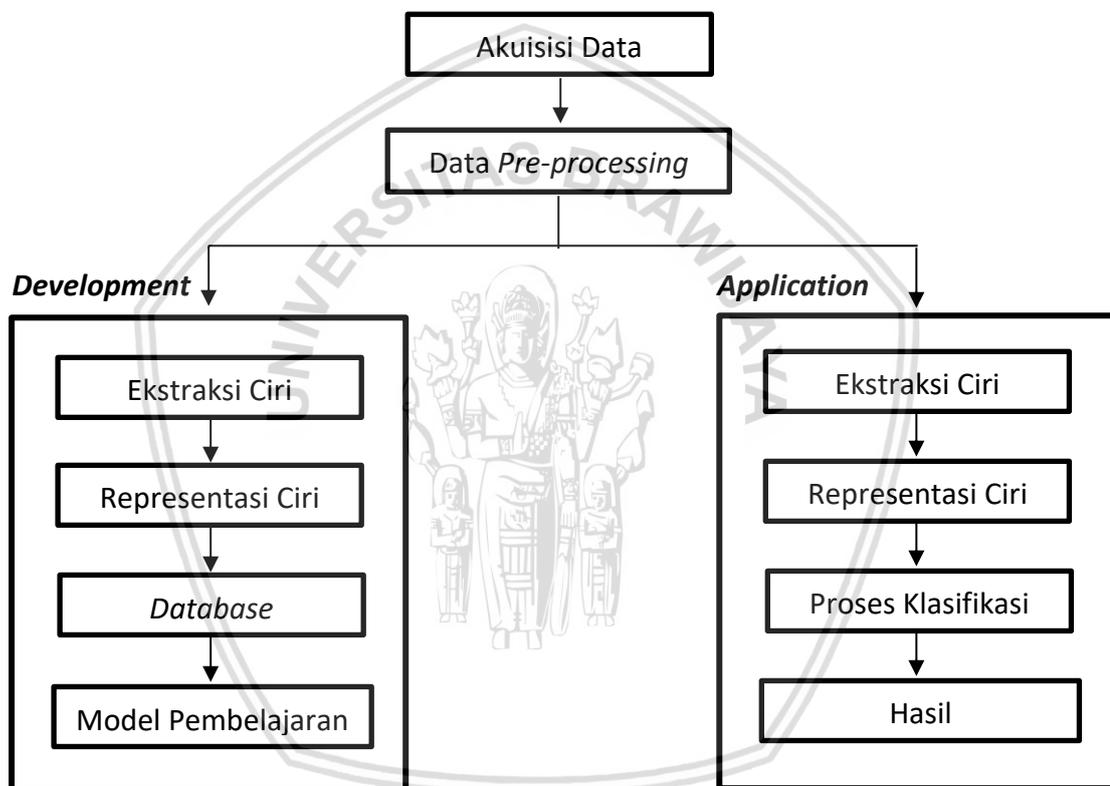
3.2 Pengumpulan Data

Data pada penelitian ini didapatkan dari hasil panen Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro). Data yang diambil adalah sampel buah jeruk keprok batu 55 dengan kualitas tinggi maupun jeruk dengan kualitas rendah. Pengumpulan data dengan cara mengambil foto dari jeruk-jeruk tersebut. Foto jeruk diambil menggunakan kamera *handphone iPhone 6s Plus* dengan jarak

antara kamera dengan buah jeruk sebesar 15cm. Buah jeruk diletakkan diatas kertas karton berwarna biru sebagai warna dasarnya/*background*. Foto diambil dari sisi atas buah jeruk. Sisi atas dipilih berdasarkan anjuran dari pakar karena busuk/cacat buah jeruk tersebut disebabkan oleh penyakit burik kusam (*trips*), dimana penyakit ini menyerang bagian atas jeruk sekitar tangkai buah. Sejumlah 100 data citra buah jeruk berhasil dikumpulkan.

3.3 Perancangan

Sistem pada penelitian ini dibuat dengan tujuan utama mengklasifikasi mutu buah jeruk berdasarkan gambar/citra. Arsitektur sistem dalam penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur Sistem

Alur kerja sistem ini adalah sebagai berikut:

1. Akuisisi Data

Pengumpulan data dilakukan dengan cara mengambil foto dari jenis buah jeruk keprok batu 55. Jeruk didapatkan dari hasil panen di Kebun Banaran Balai Penelitian Tanaman Jeruk dan Buah Subtropika (Balitjestro). Pengambilan foto dilakukan dengan menggunakan kamera *handphone iPhone 6s Plus* dari sisi atas buah jeruk dengan jarak antara kamera dengan objek jeruk keprok sebesar 15 cm. Sisi atas dipilih berdasarkan anjuran dari pakar karena busuk/cacat buah jeruk tersebut disebabkan oleh penyakit burik kusam (*trips*), dimana penyakit ini menyerang bagian atas jeruk sekitar tangkai buah. Buah jeruk keprok diletakkan diatas kertas karton berwarna biru sehingga

warna latar belakang (*background*) citra jeruk adalah warna biru. Warna biru dipilih karena memiliki kandungan nilai *blue* yang cukup kontras dengan jeruk sehingga dapat memudahkan proses segmentasi. Terdapat data latih dan data uji, dimana data latih adalah citra buah jeruk keprok yang dilakukan *cropping* dengan ukuran 64x64 piksel yang mewakili citra baik dan citra buruk dimana jumlahnya masing-masing 1 pada tiap-tiap data latih, sedangkan data uji adalah citra buah jeruk utuh. Data citra tersebut dilakukan *resize* dari ukuran awal 4032x3024 menjadi 2000x1500 piksel. Kemudian dilakukan dua kali proses *cropping*, pertama adalah *cropping* pada area objek buah jeruk saja, dan kemudian setelah itu *cropping* kedua pada bagian tengah jeruk atau sekitar tangkai buah untuk mendapatkan area cacat pada bagian atas buah jeruk dengan ukuran 640x640 piksel.

2. Data *Pre-processing*

Dari citra masukan awal yang berukuran 4032x3024 piksel dilakukan perubahan ukuran citra menjadi 2000x1500 piksel, dilakukan *resizing* karena ukuran citra awal terlalu besar dan juga membutuhkan waktu komputasi yang lebih lama. Kemudian citra hasil *resize* inilah yang digunakan dalam proses *pre-processing*. Selanjutnya dilakukan proses *grayscale* untuk merubah menjadi citra keabuan dan dilakukan segmentasi untuk mendapatkan area letak cacat buah pada bagian atas jeruk dengan ukuran 640x640 piksel. Ukuran 640x640 piksel ini merupakan ukuran yang ideal menggambarkan letak cacat pada bagian atas sekitar tangkai buah jeruk keprok.

(*Development*)

3. Ekstraksi Ciri

Menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk mendapatkan nilai ciri entropi, energi, kontras dan homogenitas pada setiap citra baik cacat (buruk) maupun tidak cacat (baik).

4. Representasi Ciri

Setiap nilai ciri yang telah didapatkan diatas, dikelompokkan sesuai dengan kondisinya, yaitu cacat (buruk) dan tidak cacat (baik).

5. Database

Nilai ciri yang telah direpresentasikan pada suatu data disimpan kedalam database untuk digunakan pada proses selanjutnya.

6. Model Pembelajaran

Dilakukan proses pembelajaran metode *Support Vector Machine* (SVM) berdasarkan database.

(*Application*)

7. Ekstraksi Ciri

Pada data uji dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk mendapatkan nilai ciri entropi, energi, kontras dan homogenitas pada setiap citra baik cacat maupun tidak cacat.

8. Representasi Ciri

Setiap nilai ciri yang telah didapatkan diatas, dikelompokkan sesuai dengan kondisinya, yaitu cacat dan tidak cacat.

9. Proses Klasifikasi

Nilai-nilai dari model pembelajaran yang berhasil dilakukan, diambil untuk dilakukan proses klasifikasi.

10. Hasil

Hasil berupa jumlah data latih dengan akurasi terbaik, nilai *distance* dengan akurasi terbaik dan akurasi arah GLCM terbaik terhadap klasifikasi mutu buah jeruk.

3.4 Implementasi

Implementasi klasifikasi mutu buah jeruk mengacu pada perancangan yang telah dibuat. Sistem akan melakukan proses ekstraksi ciri buah jeruk menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM), sistem akan mengenali pola umum citra jeruk dengan mutu tinggi atau jeruk dengan mutu rendah menggunakan data latih yang ada. Kemudian hasil ekstraksi ciri tersebut diproses menggunakan metode *Support Vector Machine* (SVM) untuk mengklasifikasi mutu buah jeruk pada data uji.

Implementasi dalam penelitian ini mengacu pada batasan sistem yaitu:

1. Pengolahan data menggunakan bahasa pemrograman Python 3 dengan library OpenCV untuk pengolahan citra digital, NumPy untuk pengolahan matriks dan scikit-learn untuk klasifikasi.
2. Masukan sistem dalam bentuk data gambar dengan format *.jpeg*.
3. Data diproses hingga memperoleh hasil klasifikasi dan akurasinya.
4. Dilakukan pengujian arah GLCM terhadap hasil klasifikasi mutu.

3.5 Pengujian & Analisis

Parameter pengujian yang akan dilakukan disesuaikan dengan perancangan algoritme. Terdapat beberapa rancangan skenario pengujian sebagai berikut:

1. Pengujian jumlah data latih.
2. Pengujian nilai *distance*.
3. Pengujian arah GLCM.

Berdasarkan rancangan skenario pengujian tersebut, dilihat kemampuan algoritme untuk menghasilkan nilai akurasi yang baik dengan dasar dari hasil pengujian jumlah data latih terbaik, nilai *distance* terbaik dan arah GLCM yang terbaik. Nilai akurasi setiap parameter pengujian pada seluruh data uji dihitung menggunakan Persamaan 3.1.

$$\text{Nilai akurasi} = \frac{\sum \text{Data terklasifikasi benar}}{\sum \text{Data uji}} \times 100\% \quad (3.1)$$

3.6 Kesimpulan dan Saran

Kesimpulan didapatkan setelah seluruh tahapan perancangan dan implementasi sistem telah selesai dilakukan. Serta telah dilakukan pengujian dan analisis terhadap metode yang digunakan yaitu *Gray Level Co-occurrence Matrix* (GLCM). Kesimpulan diambil dari hasil pengujian sistem dan analisis metode yang digunakan. Kemudian untuk saran ditulis dengan tujuan agar dapat memperbaiki segala kekurangan dan pertimbangan untuk pengembangan/penelitian selanjutnya.

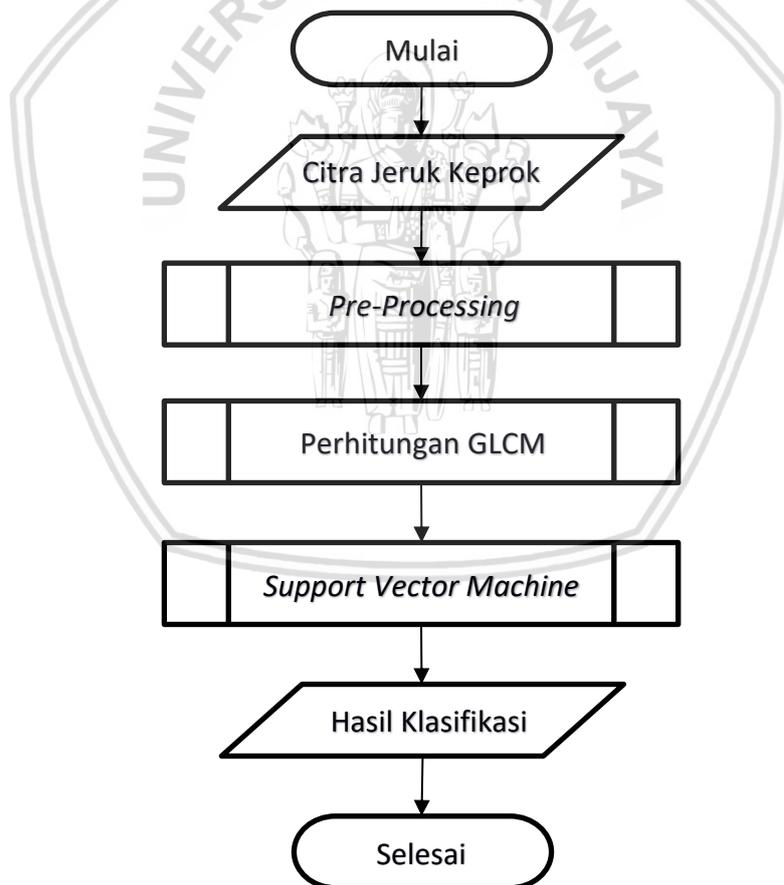


BAB 4 PERANCANGAN

Pada Bab 4 ini dijelaskan perancangan algoritme, perancangan skenario pengujian, dan perhitungan manual yang digunakan sebagai acuan dalam pengimplementasian sistem pemanfaatan Gray Level Co-occurrence Matrix (GLCM) citra jeruk keprok untuk klasifikasi mutu.

4.1 Perancangan Algoritme

Perancangan algoritme digambarkan dengan menggunakan *flowchart* (diagram alir). Dimana fungsi dari diagram alir ini adalah untuk menunjukkan alur dari program yang akan dibuat sehingga alur algoritme dapat lebih mudah dipahami. Perancangan algoritme secara umum pada penelitian ini adalah membaca masukan citra jeruk keprok, melakukan proses *pre-processing* pada citra masukan agar dapat diproses pada tahap selanjutnya, perhitungan GLCM, *support vector machine* (SVM) dan keluaran berupa hasil klasifikasi mutu buah jeruk. Perancangan secara keseluruhan dapat dilihat pada Gambar 4.1.



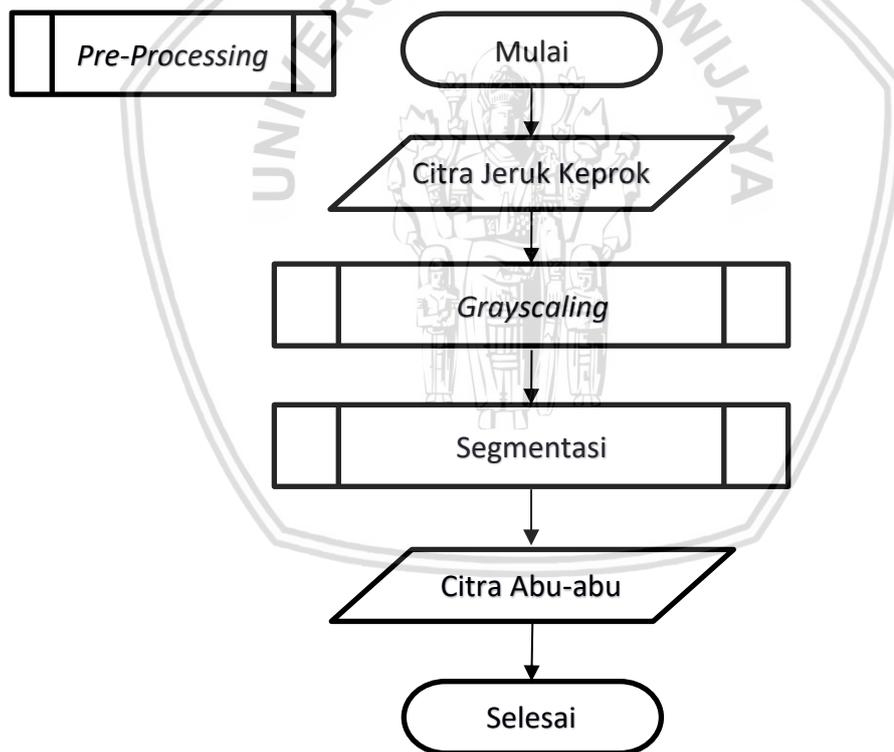
Gambar 4.1 Diagram Alir Sistem

Seperti yang telah ditunjukkan dengan diagram alir pada Gambar 4.1, perancangan algoritme pada penelitian ini dimulai dengan menerima masukan

data berupa citra jeruk keprok. Kemudian dilakukan *pre-processing* dengan melakukan *grayscale* pada citra masukan tersebut sehingga didapatkan citra *grayscale*/citra abu-abu dan juga segmentasi untuk mendapatkan citra tersegmentasi bada bagian tengah dari jeruk. Setelah itu, dilakukan proses perhitungan GLCM yang terdiri dari pembentukan matriks GLCM dan proses ekstraksi ciri *gray level co-occurrence matrix* (GLCM) untuk mendapatkan nilai ciri dari tiap-tiap citra. Terdapat 4 ciri yang didapatkan yaitu *contrast*, *homogeneity*, *energy* dan *entropy* dari 4 arah GLCM yaitu, arah 0°, 45°, 90° dan 135°. Selanjutnya yaitu proses identifikasi menggunakan metode *support vector machine* (SVM) untuk membedakan antara citra baik dan buruk sehingga didapatkan persentase cacat buah.

4.1.1 Pre-Processing

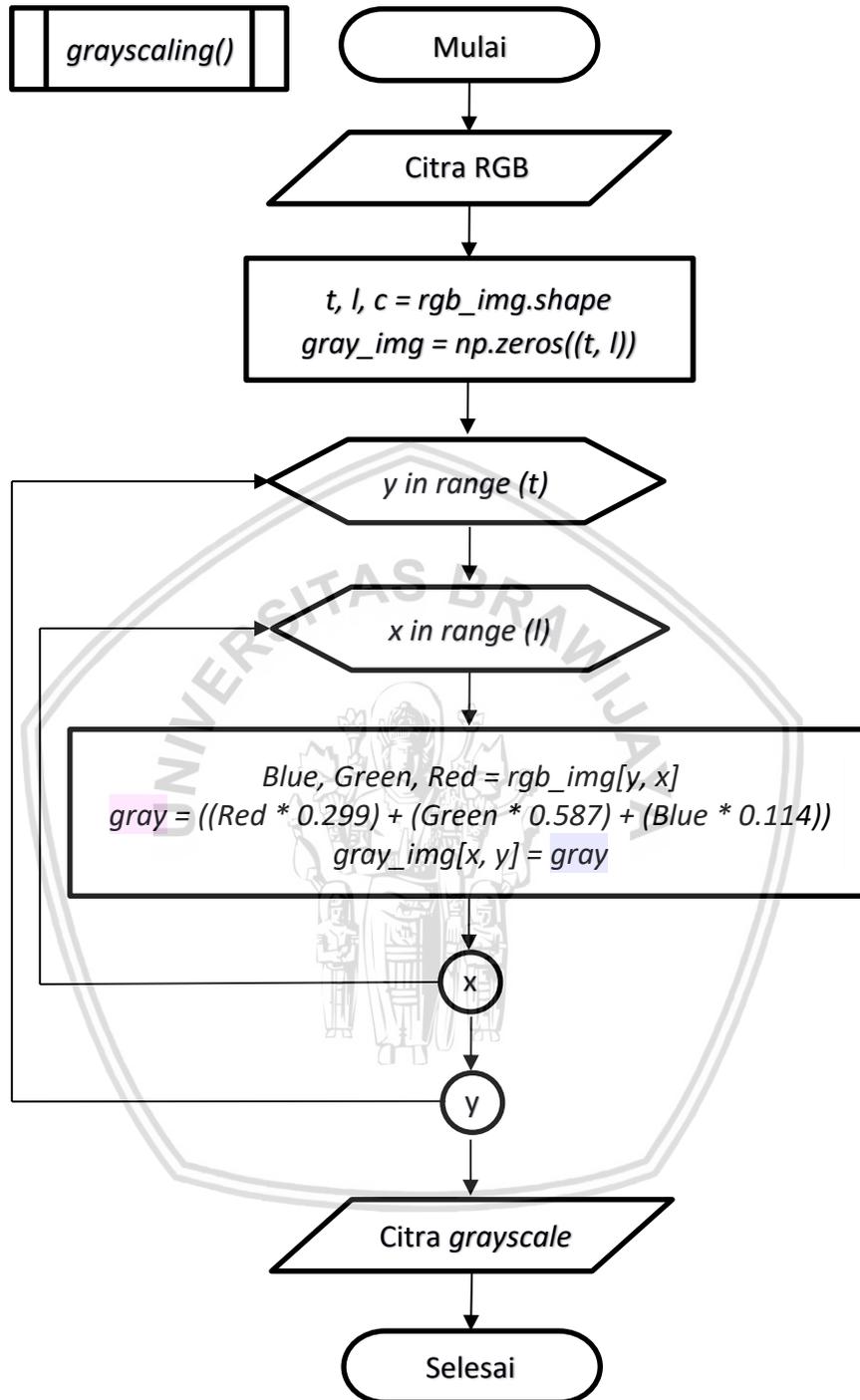
Pada tahap *pre-processing* dilakukan proses *grayscale* pada citra masukan yaitu citra jeruk keprok. *Grayscale* dilakukan untuk merubah citra awal berwarna RGB menjadi citra dengan skala keabuan. Diagram alir *pre-processing* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Alir *Pre-Processing*



4.1.1.1 Grayscale



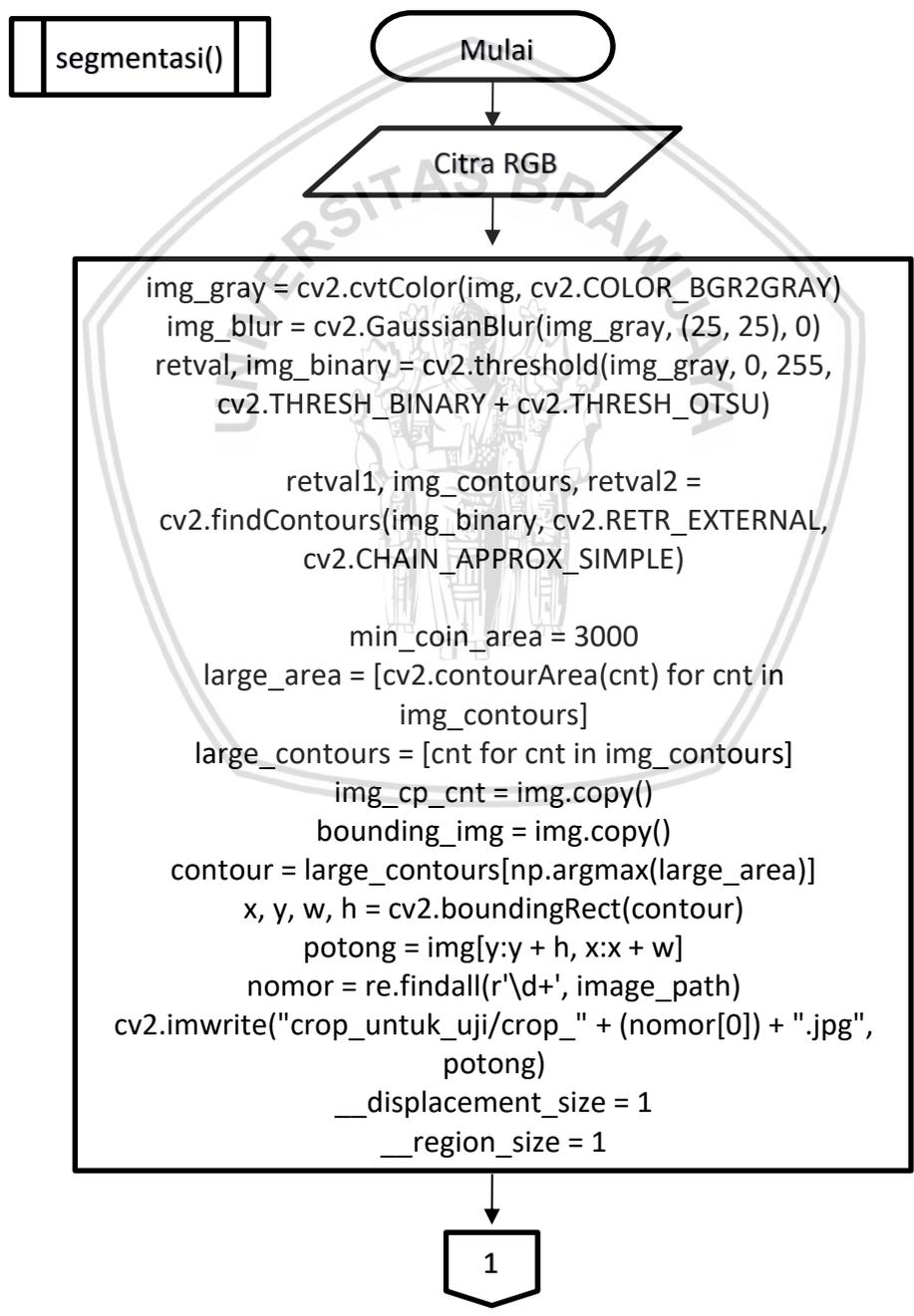
Gambar 4.3 Diagram Alir Grayscale

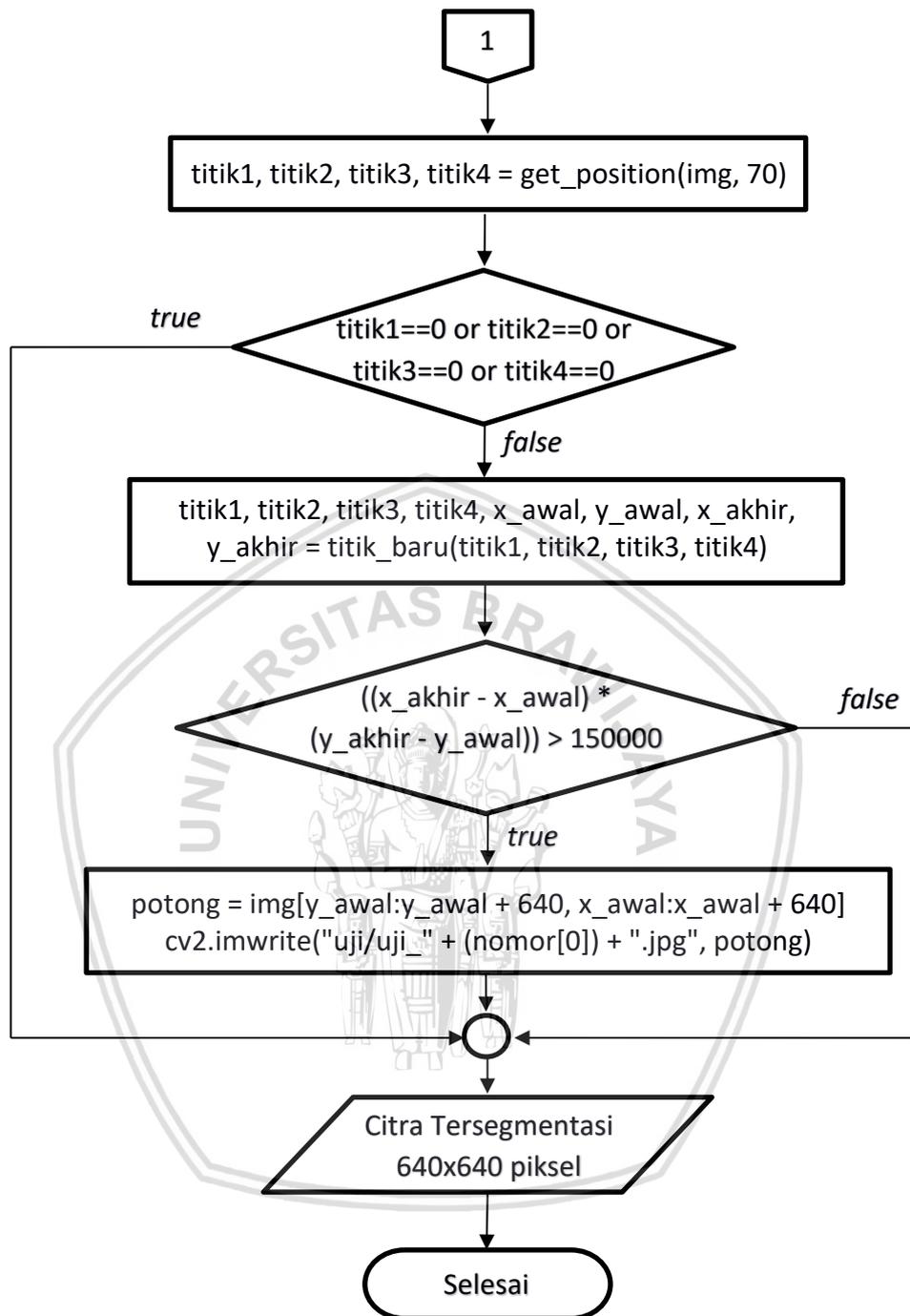
Berikut ini adalah penjelasan langkah-langkah/alur yang dilakukan dari diagram alir *grayscale* pada Gambar 4.3:

1. Menerima masukan berupa citra jeruk keprok dengan warna RGB.

2. Melakukan proses pembacaan ukuran citra RGB tersebut untuk mengetahui nilai tinggi (t), lebar (l) dan channel (c) dan juga membuat array citra baru yang berukuran sama untuk menyimpan hasil dari proses *grayscale*.
3. Melakukan perulangan untuk mengecek nilai pada tiap piksel citra.
4. Melakukan pengambilan nilai *red*, *green* dan *blue* dari tiap piksel dan melakukan proses perhitungan nilai keabuan/*grayscale* pada tiap piksel tersebut dengan cara menjumlahkan dari perhitungan nilai *red* dikalikan dengan 0.299, nilai *green* dikalikan dengan 0.587 dan nilai *blue* dikalikan dengan 0.114.
5. Memberikan keluaran berupa citra *grayscale*.

4.1.1.2 Segmentasi





Gambar 4.4 Diagram Alir Segmentasi

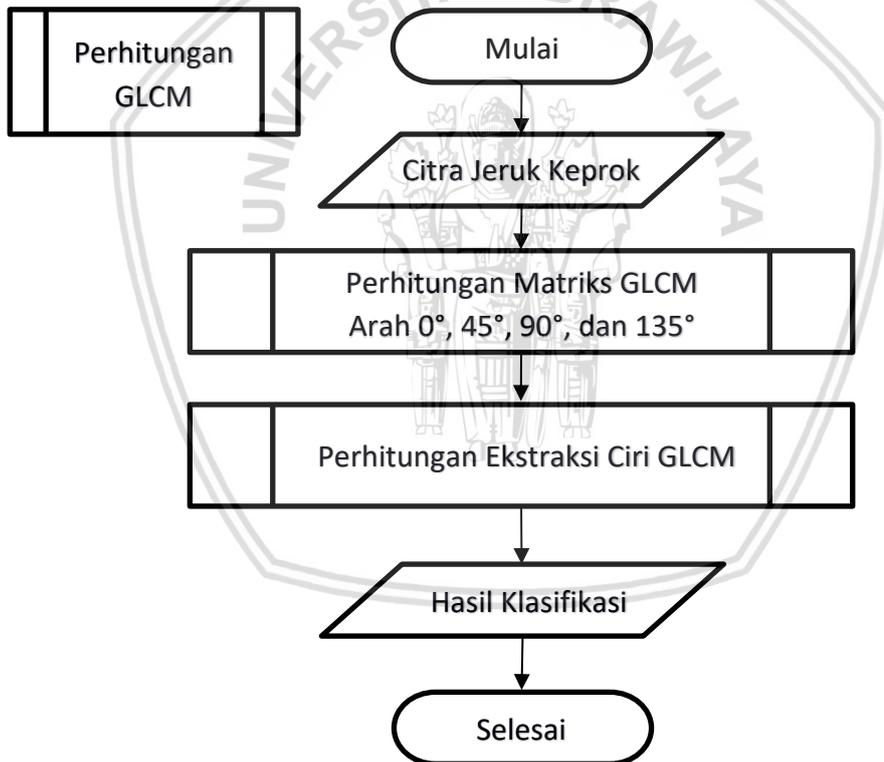
Berikut ini adalah penjelasan langkah-langkah/alur yang dilakukan dari diagram alir *grayscale* pada Gambar 4.4:

1. Menerima masukan berupa citra jeruk keprok dengan warna RGB.
2. Melakukan proses segmentasi *cropping* citra yang pertama untuk mendapatkan objek buah jeruk dengan menggunakan *thresholding Binary* dan *Otsu* untuk memisahkan objek dengan latar belakangnya.

3. Melakukan proses segmentasi *cropping* citra yang kedua untuk mendapatkan bagian tengah sekitar tangkai dari objek buah jeruk.
4. Memberikan keluaran berupa citra hasil segmentasi berukuran 640x640 piksel.

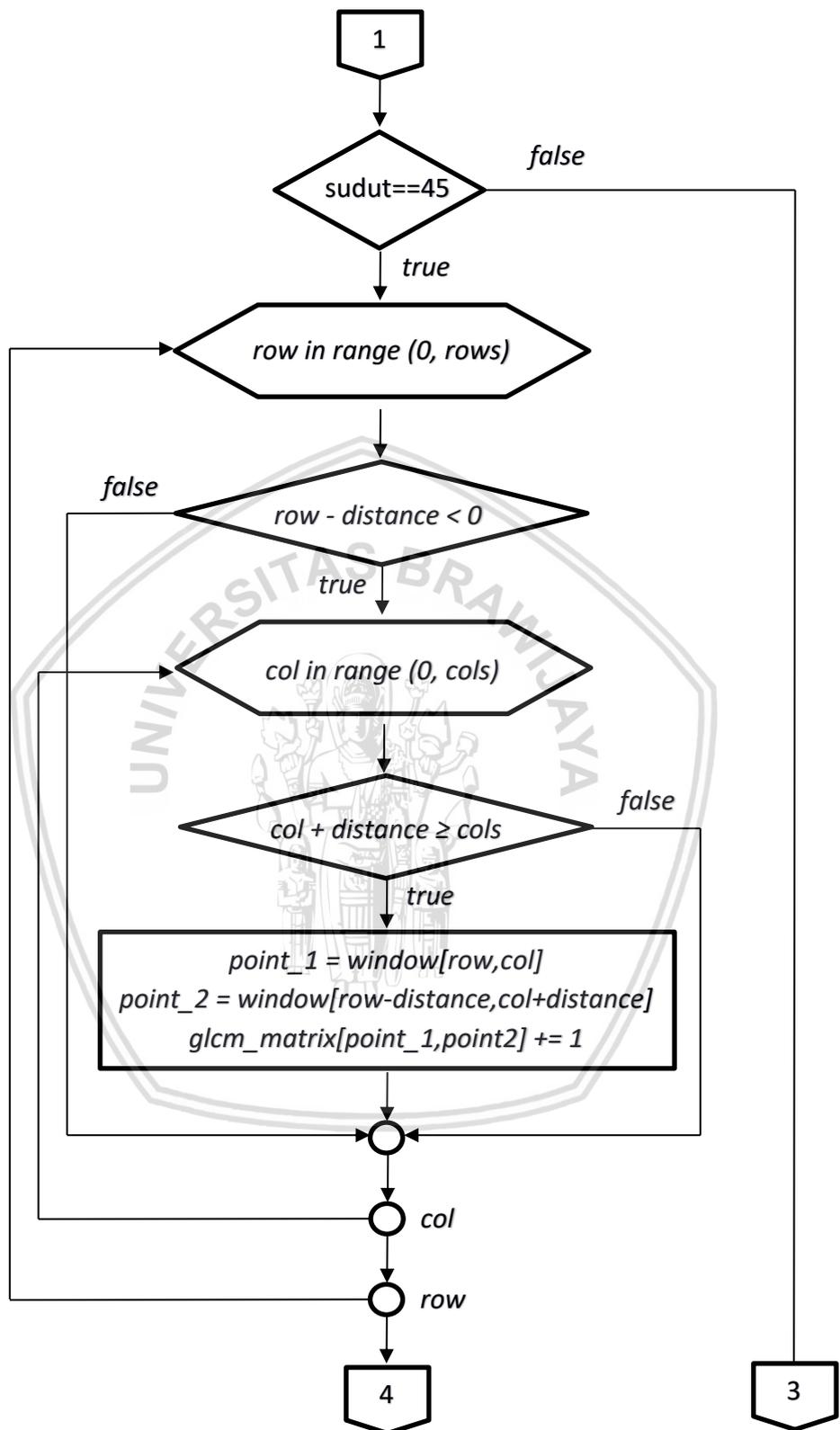
4.1.2 Perhitungan GLCM

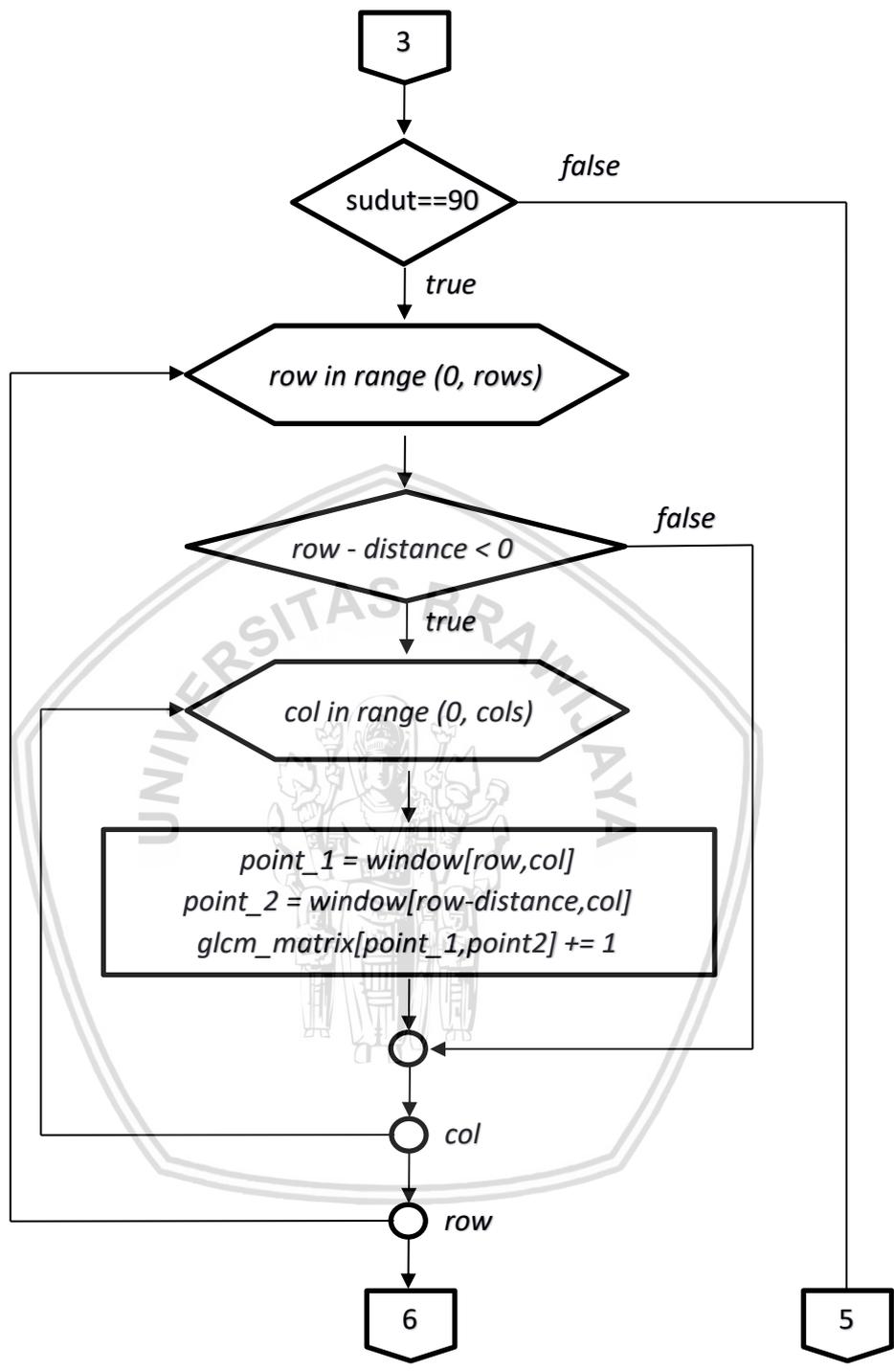
Pada tahap ini dilakukan perhitungan ciri GLCM dari citra hasil *pre-processing* berupa citra *grayscale*. Pertama-tama dilakukan proses perhitungan matriks awal pada arah 0° , 45° , 90° dan 135° . Setelah itu, dilakukan pembentukan matriks GLCM yang simetris dengan cara menjumlahkan antara matriks awal dengan matriks transposnya. Setelah didapatkan matriks yang simetris tersebut selanjutnya perlu dilakukan normalisasi agar jumlah seluruh elemen matriks bernilai 1, dengan cara membagi setiap elemen atau tiap nilai pada matriks dengan total keseluruhan pasangan piksel. Kemudian dilakukan perhitungan ciri GLCM. Terdapat 4 ciri yang akan digunakan, yaitu *contrast*, *homogeneity*, *energy* dan *entropy*. Diagram alir perhitungan GLCM dapat dilihat pada Gambar 4.5.

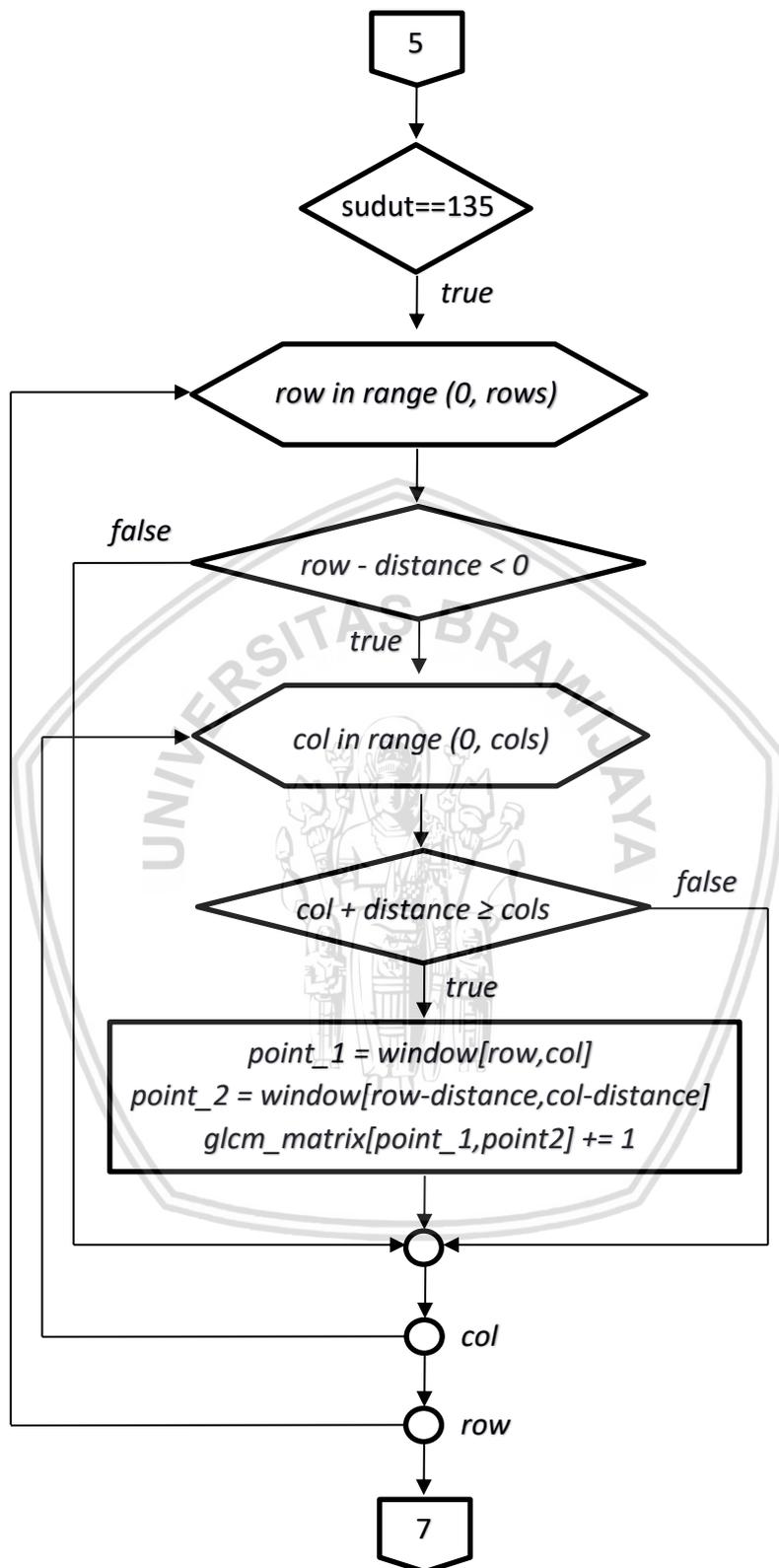


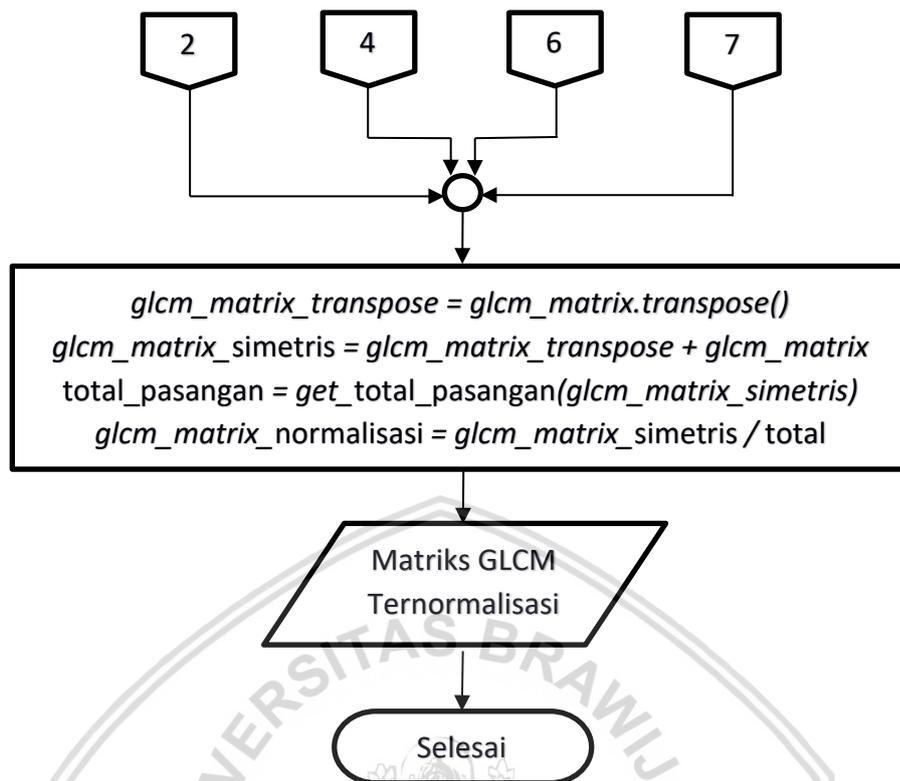
Gambar 4.5 Diagram Alir Perhitungan GLCM











Gambar 4.6 Diagram Alir Perhitungan Matriks GLCM

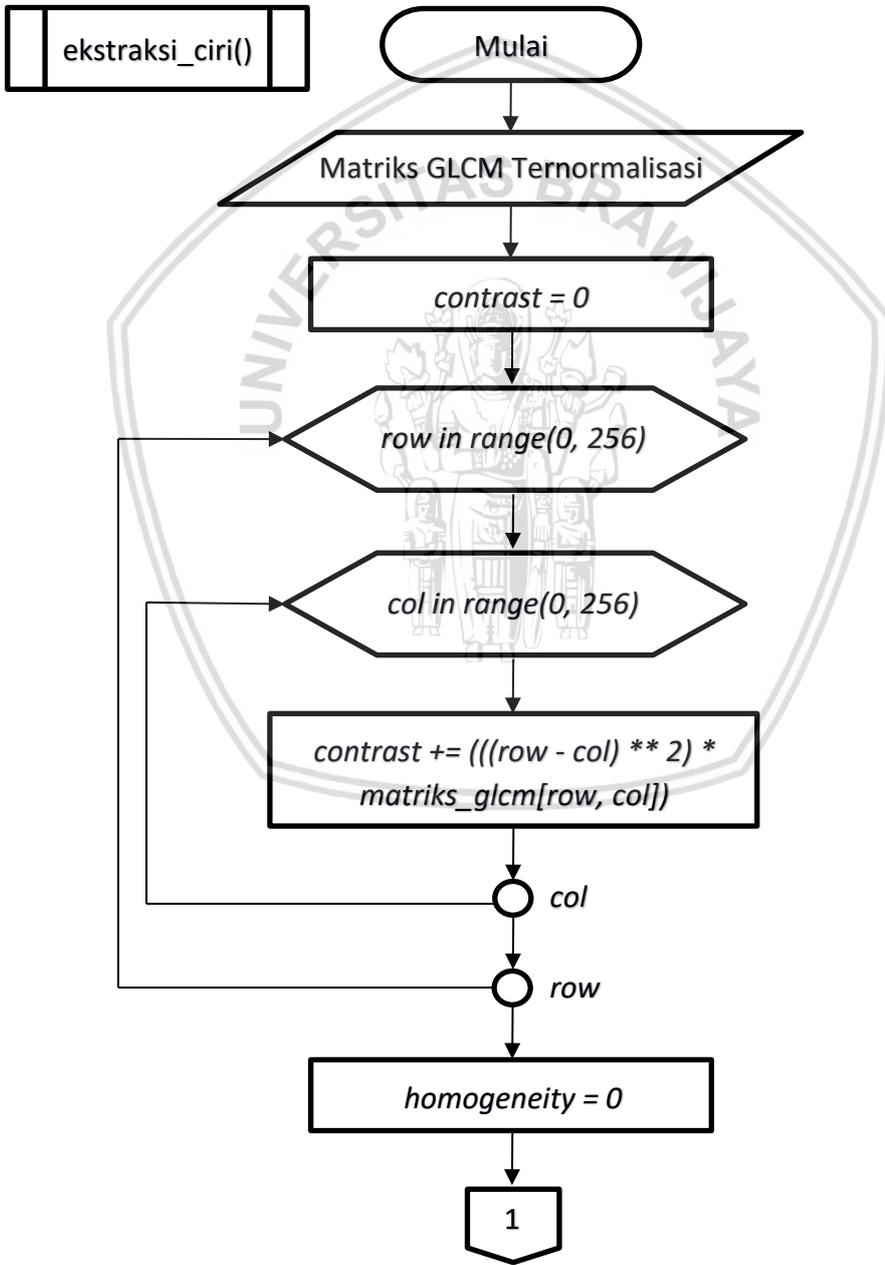
Berikut ini adalah penjelasan langkah-langkah/alur yang dilakukan pada diagram alir perhitungan matriks GLCM pada Gambar 4.6:

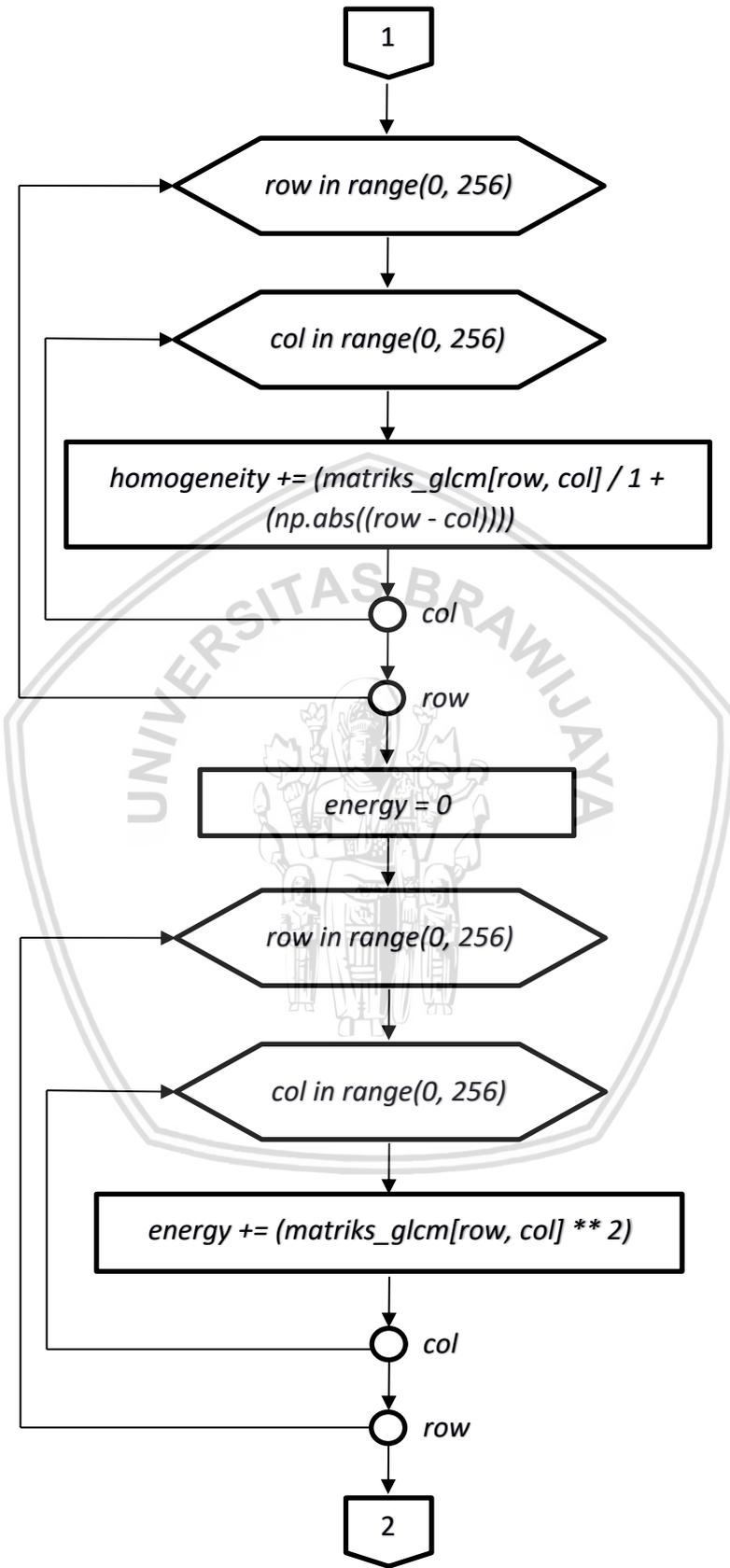
1. Menerima masukan berupa citra *grayscale* jeruk keprok, nilai sudut dan nilai *distance*.
2. Melakukan proses untuk mengetahui ukuran citra *grayscale* jeruk keprok dan membuat array berukuran 256x256 untuk menyimpan hasil perhitungan matriks GLCM.
3. Melakukan proses perhitungan matriks awal GLCM sesuai dengan sudut/arah GLCM dan jarak ketetanggaan (*distance*) yang diinginkan. Terdapat 4 arah sudut GLCM yaitu, 0°, 45°, 90° dan 135°. Untuk arah sudut 0° dilakukan pengambilan nilai yang didapatkan dari banyaknya kemunculan nilai pasangan suatu piksel dengan piksel tetangganya pada sisi kanan, yang berjarak sesuai dengan nilai *distance*-nya. Untuk arah sudut 45° dilakukan pengambilan nilai yang didapatkan dari banyaknya kemunculan nilai pasangan suatu piksel dengan piksel tetangganya pada sisi diagonal kanan atas, yang berjarak sesuai dengan nilai *distance*-nya. Untuk arah sudut 90° dilakukan pengambilan nilai yang didapatkan dari banyaknya kemunculan nilai pasangan suatu piksel dengan piksel tetangganya pada sisi atas, yang berjarak sesuai dengan nilai *distance*-nya. Untuk arah sudut 135° dilakukan pengambilan nilai yang didapatkan dari banyaknya kemunculan nilai pasangan suatu piksel dengan piksel tetangganya pada sisi diagonal kiri atas, yang berjarak sesuai dengan nilai *distance*-nya.

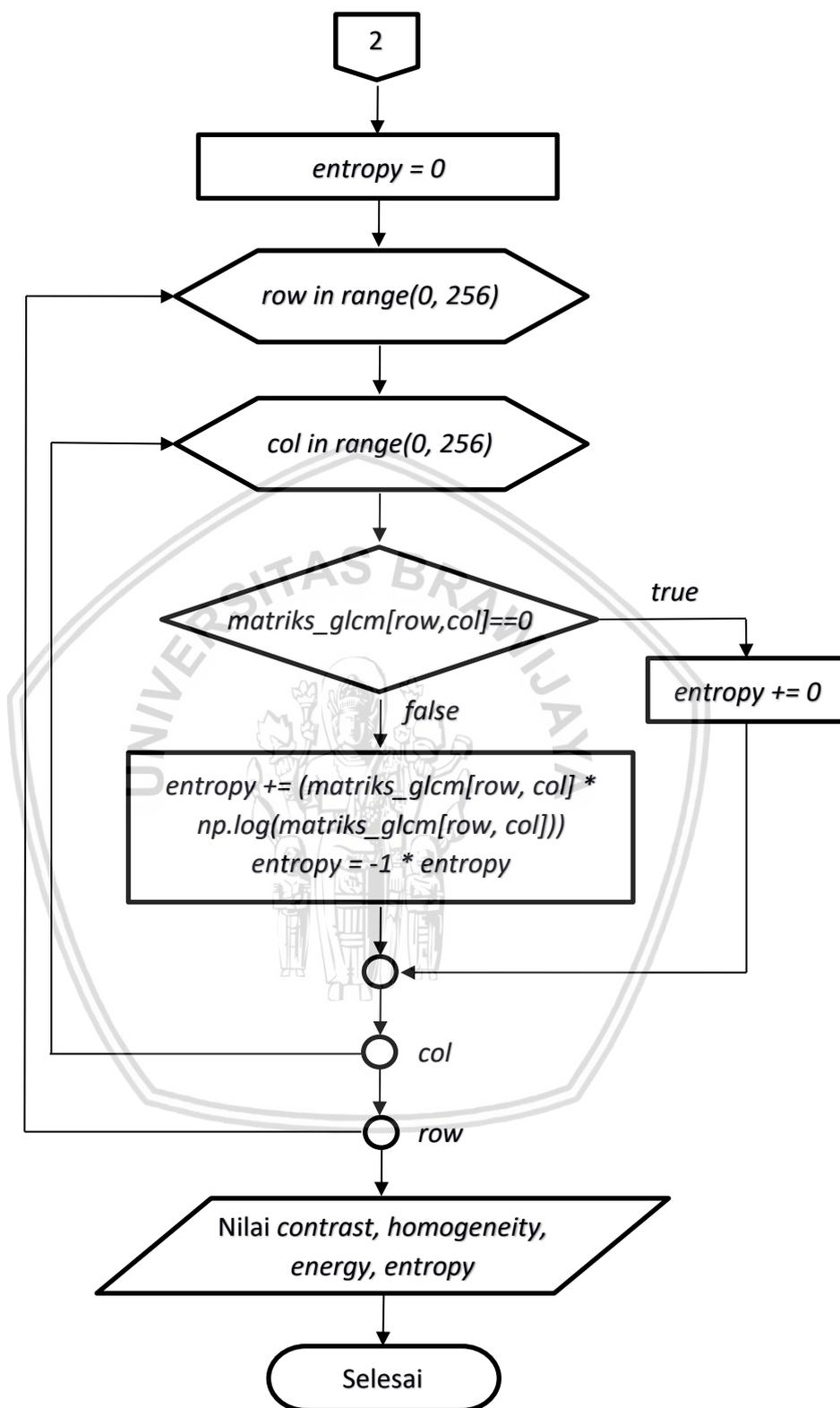
4. Melakukan pembentukan matriks GLCM yang simetris dengan cara menjumlahkan antara matriks awal dengan matriks transposnya.
5. Melakukan normalisasi matriks agar jumlah seluruh elemen matriks bernilai 1, dengan cara membagi setiap elemen atau tiap nilai pada matriks dengan total keseluruhan pasangan piksel.
6. Memberikan keluaran berupa matriks GLCM yang telah ternormalisasi.

4.1.2.2 Perhitungan Ekstraksi Ciri GLCM

Diagram alir untuk ekstraksi ciri *Gray Level Co-occurrence Matrix* (GLCM) dapat dilihat pada gambar 4.7.







Gambar 4.7 Diagram Alir Perhitungan Ekstraksi Ciri GLCM

Berikut ini adalah penjelasan langkah-langkah/alur yang dilakukan pada diagram alir perhitungan ekstraksi ciri GLCM pada Gambar 4.7:

1. Menerima masukan berupa matriks GLCM yang sudah ternormalisasi.
2. Melakukan proses perhitungan ekstraksi ciri *contrast* dengan rumus yang telah ditentukan untuk mengetahui nilai kekuatan perbedaan intensitas dalam citra.
3. Melakukan proses perhitungan ekstraksi ciri *homogeneity* dengan rumus yang telah ditentukan untuk mengetahui nilai kehomogenan variasi intensitas dalam citra.
4. Melakukan proses perhitungan ekstraksi ciri *energy* dengan rumus yang telah ditentukan untuk mengetahui nilai konsentrasi pasangan intensitas pada matriks GLCM.
5. Melakukan proses perhitungan ekstraksi ciri *entropy* dengan rumus yang telah ditentukan untuk mengetahui nilai keteracakan dari distribusi intensitas.
6. Memberikan keluaran berupa nilai dari *contrast*, *homogeneity*, *energy* dan *entropy*.

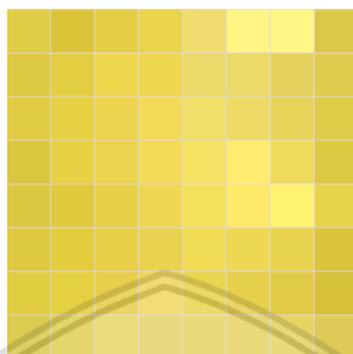
4.1.3 Identifikasi Cacat Buah

Penelitian ini menggunakan metode *support vector machine* (SVM) untuk mengidentifikasi cacat buah pada suatu area *windowing* citra, dilakukan klasifikasi apakah area *windowing* tersebut masuk kedalam kelas baik atau kelas buruk/cacat. Ukuran *window* yang digunakan adalah 64x64 piksel. Ukuran tersebut dipilih karena dapat merepresentasikan tekstur dari suatu buah jeruk dengan baik. Berdasarkan percobaan ukuran *window* yang telah penulis lakukan, apabila menggunakan ukuran *window* 32x32 piksel tekstur dari buah jeruk kurang terlihat karena ukuran terlalu kecil, sedangkan apabila ukuran *window* lebih besar seperti 128x128 piksel maka terdapat banyak data *noise*, sehingga dipilih ukuran *windowing* 64x64 piksel. Kemudian dilakukan *sliding window* pada data uji untuk mendapatkan jumlah citra baik dan buruk. Metode SVM pada penelitian ini menggunakan *library scikit-learn*. Kernel SVM yang digunakan adalah kernel *linear*. Kernel *linear* dipilih karena dapat mengklasifikasikan 2 buah kelas data yaitu citra baik dan citra buruk/cacat secara lebih baik, apabila dibandingkan dengan kernel RBF.

Klasifikasi mutu ditentukan berdasarkan persentase cacat buah. Persentase tersebut didapatkan dari luas area cacat terhadap luas permukaan buah jeruk. Luas area cacat dihitung dari jumlah citra 64x64 piksel yang teridentifikasi buruk dikalikan dengan luas citra 64x64 piksel. Dilakukan konversi dari piksel kedalam *centimeter*. Total luas citra 64x64 piksel yang telah didapatkan lalu kemudian dibandingkan dengan luas permukaan lingkaran atau luas permukaan jeruk untuk mendapatkan persentasenya. Persentase tersebut yang digunakan untuk menentukan *grade* buah jeruk, dimana 0% untuk *Grade Super*, maksimal 10% untuk *Grade A* dan maksimal 15% untuk *Grade B*.

4.2 Perhitungan Manual

Perhitungan manual atau manualisasi diterapkan pada citra jeruk keprok dengan dilakukan pemotongan suatu titik acak dengan ukuran 8x8 piksel. Potongan gambar yang digunakan untuk perhitungan manual ditunjukkan pada Gambar 4.8.



Gambar 4.8 Potongan Citra Jeruk Keprok dengan Ukuran 8x8 Piksel

4.2.1 Pre-Processing

Pada proses *pre-processing* dilakukan *grayscale* untuk mengubah citra masukan RGB menjadi citra keabuan/*grayscale*. Pertama, dilakukan pengambilan nilai *Red*, *Green* dan *Blue* tiap piksel pada citra. Masing-masing nilai tersebut ditunjukkan pada Tabel 4.1, Tabel 4.2 dan Tabel 4.3.

Tabel 4.1 Nilai *Red* dari Gambar 4.8

y,x	0	1	2	3	4	5	6	7
0	218	211	223	230	238	253	252	212
1	214	220	234	234	231	230	223	215
2	216	224	231	241	236	234	225	215
3	211	225	230	243	240	252	234	213
4	211	217	227	235	240	250	251	224
5	214	220	227	230	238	230	224	211
6	217	222	225	238	227	225	218	206
7	211	220	228	226	227	226	226	214

Tabel 4.2 Nilai *Green* dari Gambar 4.8

y,x	0	1	2	3	4	5	6	7
0	213	203	214	217	228	255	255	208
1	211	215	222	220	229	228	219	214



2	213	218	218	223	235	228	220	214
3	209	219	217	224	236	244	228	211
4	209	209	215	221	237	242	255	222
5	211	212	215	215	230	225	220	205
6	215	217	214	218	222	215	211	201
7	211	216	220	215	219	217	218	210

Tabel 4.3 Nilai Blue dari Gambar 4.8

y,x	0	1	2	3	4	5	6	7
0	45	30	44	51	86	109	110	46
1	42	41	53	54	81	80	67	49
2	42	44	54	66	79	77	65	46
3	37	45	53	67	74	86	67	40
4	38	36	49	59	68	80	87	49
5	38	38	48	56	63	58	52	33
6	41	41	50	67	53	52	44	29
7	58	62	80	83	77	77	78	64

Setelah berhasil didapatkan nilai Red, Green dan Blue dari tiap piksel pada citra, langkah selanjutnya adalah melakukan perhitungan nilai keabuan/grayscale.

$$grayscale = (R * 0.229) + (G * 0.587) + (B * 0.114)$$

$$grayscale = (226 * 0.229) + (205 * 0.587) + (70 * 0.114) = 195.889 \approx 195$$

Perhitungan diatas adalah perhitungan nilai grayscale pada piksel (0,0) dengan nilai Red=226, Green=205 dan Blue=70 yang menghasilkan nilai grayscale sebesar 195. Untuk hasil perhitungan keseluruhan nilai grayscale pada Gambar 4.8 ditunjukkan pada Tabel 4.4.

Tabel 4.4 Hasil Perhitungan Grayscale

y,x	0	1	2	3	4	5	6	7
0	195	192	194	189	190	192	195	193
1	185	196	199	200	191	194	198	199
2	197	206	203	202	199	199	198	206
3	201	205	210	211	206	201	206	203
4	214	212	217	218	218	213	204	205
5	237	211	212	228	225	207	199	203



6	237	202	203	211	234	202	194	204
7	190	195	195	192	202	187	182	194

4.2.2 Perhitungan GLCM

Dari hasil pre-processing, selanjutnya dilakukan proses perhitungan GLCM. Perhitungan GLCM dilakukan dengan 2 tahapan, pertama adalah perhitungan matriks GLCM dan kedua adalah perhitungan ekstraksi ciri GLCM.

4.2.2.1 Perhitungan Matriks GLCM

- Perhitungan matriks GLCM dilakukan untuk 4 arah yaitu 0° , 45° , 90° dan 135° . Digunakan nilai ketetangaan piksel (*distance*) sebesar 1. Matriks GLCM berukuran 256×256 . Inisialisasi nilai awal matriks GLCM didapatkan dari jumlah pasangan piksel. Berikut ini merupakan manualisasi pembentukan matriks GLCM pada tiap-tiap sudut.

a) Sudut 0°

195	192	194	189	190	192	195	193
185	196	199	200	191	194	198	199
197	206	203	202	199	199	198	206
201	205	210	211	206	201	206	203
214	212	217	218	218	213	204	205
237	211	212	228	225	207	199	203
237	202	203	211	234	202	194	204
190	195	195	192	202	187	182	194

Gambar 4.9 Perhitungan Pasangan Piksel Pada Sudut 0°

Berdasarkan Gambar 4.9 dapat dilihat bahwa jumlah pasangan piksel (195,192) pada sudut 0° dengan nilai ketetangaan piksel (*distance*) sebesar 1 adalah sebanyak 2 pasang. Nilai jumlah pasangan tersebut dan pasangan-pasangan lainnya dimasukkan kedalam matriks GLCM seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Matriks GLCM Sudut 0°

y,x	0	...	192	195	199	...	255
0	0	...	0	0	0	...	0
...
192	0	...	0	1	0	...	0
195	0	...	2	1	0	...	0
199	0	...	0	0	1	...	0



...
255	0	...	0	0	0	...	0

b) Sudut 45°

195	192	194	189	190	192	195	193
185	196	199	200	191	194	198	199
197	206	203	202	199	199	198	206
201	205	210	211	206	201	206	203
214	212	217	218	218	213	204	205
237	211	212	228	225	207	199	203
237	202	203	211	234	202	194	204
190	195	195	192	202	187	182	194

Gambar 4.10 Perhitungan Pasangan Pixel Pada Sudut 45°

Berdasarkan Gambar 4.10 dapat dilihat bahwa jumlah pasangan pixel (199,198) pada sudut 45° dengan nilai ketetanggaan pixel (*distance*) sebesar 1 adalah sebanyak 1 pasang. Nilai jumlah pasangan tersebut dan pasangan-pasangan lainnya dimasukkan kedalam matriks GLCM seperti yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Matriks GLCM Sudut 45°

y,x	0	...	198	199	201	...	255
0	0	...	0	0	0	...	0
...
198	0	...	0	1	0	...	0
199	0	...	1	0	0	...	0
201	0	...	1	0	0	...	0
...
255	0	...	0	0	0	...	0

c) Sudut 90°

195	192	194	189	190	192	195	193
185	196	199	200	191	194	198	199
197	206	203	202	199	199	198	206
201	205	210	211	206	201	206	203



214	212	217	218	218	213	204	205
237	211	212	228	225	207	199	203
237	202	203	211	234	202	194	204
190	195	195	192	202	187	182	194

Gambar 4.11 Perhitungan Pasangan Pikel Pada Sudut 90°

Berdasarkan Gambar 4.11 dapat dilihat bahwa jumlah pasangan piksel (203,206) pada sudut 90° dengan nilai ketetangaan piksel (*distance*) sebesar 1 adalah sebanyak 1 pasang. Nilai jumlah pasangan tersebut dan pasangan-pasangan lainnya dimasukkan kedalam matriks GLCM seperti yang ditunjukkan pada Tabel 4.7.

Tabel 4.7 Matriks GLCM Sudut 90°

y,x	0	...	203	205	206	...	255
0	0	...	0	0	0	...	0
...
203	0	...	0	0	1	...	0
205	0	...	1	0	1	...	0
206	0	...	0	0	0	...	0
...
255	0	...	0	0	0	...	0

d) Sudut 135°

195	192	194	189	190	192	195	193
185	196	199	200	191	194	198	199
197	206	203	202	199	199	198	206
201	205	210	211	206	201	206	203
214	212	217	218	218	213	204	205
237	211	212	228	225	207	199	203
237	202	203	211	234	202	194	204
190	195	195	192	202	187	182	194

Gambar 4.12 Perhitungan Pasangan Pikel Pada Sudut 135°

Berdasarkan Gambar 4.12 dapat dilihat bahwa jumlah pasangan piksel (212,212) pada sudut 135° dengan nilai ketetangaan piksel (*distance*) sebesar 1 adalah sebanyak 1 pasang. Nilai jumlah pasangan tersebut dan pasangan-

pasangan lainnya dimasukkan kedalam matriks GLCM seperti yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Matriks GLCM Sudut 135°

y,x	0	...	211	212	214	...	255
0	0	...	0	0	0	...	0
...
211	0	...	0	1	1	...	0
212	0	...	0	1	0	...	0
214	0	...	0	0	0	...	0
...
255	0	...	0	0	0	...	0

2. Proses selanjutnya adalah membentuk matriks GLCM yang simetris dengan cara menjumlahkan matriks GLCM dengan nilai matriks transposnya. Berikut ini merupakan manualisasi pembentukan matriks GLCM yang simetris pada tiap-tiap sudut.

a) Sudut 0°

$$M = \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 2 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 2 & 0 & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 3 & 0 & \dots & 0 \\ 0 & \dots & 3 & 2 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

b) Sudut 45°

$$M = \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$



$$= \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 2 & 1 & \dots & 0 \\ 0 & \dots & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

c) Sudut 90°

$$M = \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 & 1 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 & 1 & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

d) Sudut 135°

$$M = \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 2 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

3. Kemudian dilakukan normalisasi agar jumlah seluruh elemen pada matriks bernilai 1. Normalisasi matriks GLCM dilakukan dengan cara membagi setiap elemen matriks dengan jumlah total pasangan piksel. Berikut ini merupakan manualisasi normalisasi matriks GLCM pada tiap-tiap sudut.

a) Sudut 0°



Jumlah pasangan piksel dari matriks GLCM sudut 0° adalah 112. Maka matriks GLCM ternormalisasi adalah sebagai berikut.

$$M = \frac{\begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 2 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}}{112}$$

$$= \begin{pmatrix} 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0089 & 0.0089 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0089 & 0.0178 & 0.0000 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0089 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

b) Sudut 45°

Jumlah pasangan piksel dari matriks GLCM sudut 45° adalah 98. Maka matriks GLCM ternormalisasi adalah sebagai berikut.

$$M = \frac{\begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 2 & 1 & \dots & 0 \\ 0 & \dots & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}}{98}$$

$$= \begin{pmatrix} 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0204 & 0.0102 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0204 & 0.0000 & 0.0000 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0102 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

c) Sudut 90°

Jumlah pasangan piksel dari matriks GLCM sudut 90° adalah 112. Maka matriks GLCM ternormalisasi adalah sebagai berikut.

$$M = \frac{\begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 & 1 & \dots & 0 \\ 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}}{112}$$

$$= \begin{pmatrix} 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0089 & 0.0089 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0089 & 0.0000 & 0.0089 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0089 & 0.0089 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

d) Sudut 135°

Jumlah pasangan piksel dari matriks GLCM sudut 135° adalah 98. Maka matriks GLCM ternormalisasi adalah sebagai berikut.

$$M = \frac{\begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 & \dots & 0 \\ 0 & \dots & 1 & 2 & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}}{98}$$

$$= \begin{pmatrix} 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0102 & 0.0102 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0102 & 0.0204 & 0.0000 & \dots & 0.0000 \\ 0.0000 & \dots & 0.0102 & 0.0000 & 0.0000 & \dots & 0.0000 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.0000 & \dots & 0.0000 & 0.0000 & 0.0000 & \dots & 0.0000 \end{pmatrix}$$

4.2.2.2 Perhitungan Ekstraksi Ciri GLCM

Perhitungan ekstraksi ciri GLCM dilakukan setelah didapatkan matriks GLCM yang ternormalisasi. Ciri yang digunakan yaitu, *contrast*, *homogeneity*, *energy* dan *entropy*. Perhitungan ciri dilakukan pada tiap-tiap sudut yaitu, 0°, 45°, 90° dan 135°. Masing-masing ciri tersebut dihitung menggunakan Persamaan 2.2 hingga Persamaan 2.5. Berikut ini merupakan manualisasi perhitungan ekstraksi ciri GLCM pada tiap-tiap sudut.

a) Sudut 0°

$$\begin{aligned}
 \text{Contrast} &= ((0 - 0)^2 * 0.0000) + ((192 - 195)^2 * 0.0089) \\
 &\quad + ((192 - 199)^2 * 0.0089) + ((195 - 192)^2 * 0.0089) \\
 &\quad + ((195 - 195)^2 * 0.0178) + ((199 - 192)^2 * 0.0089) + \dots \\
 &\quad + ((255 - 255)^2 * 0.0000) \\
 &= 0 + 0.0801 + 0.4361 + 0.0801 + 0 + 0.4361 + \dots + 0 \\
 &= 100.0535
 \end{aligned}$$

$$\begin{aligned}
 \text{Homogeneity} &= \left(\frac{0.0000}{1 + (0 - 0)^2} \right) + \left(\frac{0.0089}{1 + (192 - 195)^2} \right) \\
 &\quad + \left(\frac{0.0089}{1 + (192 - 199)^2} \right) + \left(\frac{0.0089}{1 + (195 - 192)^2} \right) \\
 &\quad + \left(\frac{0.0178}{1 + (195 - 195)^2} \right) + \left(\frac{0.0089}{1 + (199 - 192)^2} \right) + \dots \\
 &\quad + \left(\frac{0.0000}{1 + (255 - 255)^2} \right) \\
 &= 0 + 0.00089 + 0.000178 + 0.00089 + 0.0178 + \\
 &\quad 0.000178 + \dots + 0 \\
 &= 0.2692
 \end{aligned}$$

$$\begin{aligned}
 \text{Energy} &= 0.0000^2 + 0.0089^2 + 0.0089^2 + 0.0089^2 + 0.0178^2 + 0.0089^2 \\
 &\quad + \dots + 0.0000^2 \\
 &= 0 + 0.00007 + 0.00007 + 0.00007 + 0.00031 + 0.00007 + \\
 &\quad \dots + 0 \\
 &= 0.01163
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} &= 0 + (-0.0089 * \log(0.0089)) + (-0.0089 * \log(0.0089)) \\
 &\quad + (-0.0089 * \log(0.0089)) + (-0.0178 * \log(0.0178)) \\
 &\quad + (-0.0089 * \log(0.0089)) + \dots + 0 \\
 &= 0 + 0.0182 + 0.0182 + 0.0182 + 0.0311 + 0.0182 + \dots + 0 \\
 &= 4.5234
 \end{aligned}$$

b) Sudut 45°

$$\begin{aligned}
 \text{Contrast} &= ((0 - 0)^2 * 0.0000) + ((198 - 199)^2 * 0.0204) \\
 &\quad + ((198 - 201)^2 * 0.0102) + ((199 - 198)^2 * 0.0204) \\
 &\quad + ((201 - 198)^2 * 0.0102) + \dots \\
 &\quad + ((255 - 255)^2 * 0.0000) \\
 &= 0 + 0.0204 + 0.0918 + 0.0204 + 0.0918 + \dots + 0 \\
 &= 154.0612
 \end{aligned}$$

$$\begin{aligned}
 \text{Homogeneity} &= \left(\frac{0.0000}{1 + (0 - 0)^2} \right) + \left(\frac{0.0204}{1 + (198 - 199)^2} \right) \\
 &\quad + \left(\frac{0.0102}{1 + (198 - 201)^2} \right) + \left(\frac{0.0204}{1 + (199 - 198)^2} \right)
 \end{aligned}$$



$$\begin{aligned}
 & + \left(\frac{0.0102}{1 + (201 - 198)^2} \right) + \dots + \left(\frac{0.0000}{1 + (255 - 255)^2} \right) \\
 & = 0 + 0.0102 + 0.00102 + 0.0102 + 0.00102 + \dots + 0 \\
 & = 0.2096
 \end{aligned}$$

$$\begin{aligned}
 \text{Energy} & = 0.0000^2 + 0.0204^2 + 0.0102^2 + 0.0204^2 + 0.0102^2 \\
 & \quad + 0.0000^2 \\
 & = 0 + 0.000416 + 0.000104 + 0.000416 + 0.000104 + \\
 & \quad \dots + 0 \\
 & = 0.01187
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} & = 0 + (-0.0204 * \log(0.0204)) + (-0.0102 * \log(0.0102)) \\
 & \quad + (-0.0204 * \log(0.0204)) + (-0.0102 * \log(0.0102)) \\
 & \quad + \dots + 0 \\
 & = 0 + 0.0344 + 0.0203 + 0.0344 + 0.0203 + \dots + 0 \\
 & = 4.4718
 \end{aligned}$$

c) Sudut 90°

$$\begin{aligned}
 \text{Contrast} & = ((0 - 0)^2 * 0.0000) + ((203 - 205)^2 * 0.0089) \\
 & \quad + ((203 - 206)^2 * 0.0089) + ((205 - 203)^2 * 0.0089) \\
 & \quad + ((205 - 206)^2 * 0.0089) + ((206 - 203)^2 * 0.0089) \\
 & \quad + ((206 - 205)^2 * 0.0089) + \dots \\
 & \quad + ((255 - 255)^2 * 0.0000) \\
 & = 0 + 0.0356 + 0.0801 + 0.0356 + 0.0089 + 0.0801 + \\
 & \quad 0.0089 + \dots + 0 \\
 & = 127.2678
 \end{aligned}$$

$$\begin{aligned}
 \text{Homogeneity} & = \left(\frac{0.0000}{1 + (0 - 0)^2} \right) + \left(\frac{0.0089}{1 + (203 - 205)^2} \right) \\
 & \quad + \left(\frac{0.0089}{1 + (203 - 206)^2} \right) + \left(\frac{0.0089}{1 + (205 - 203)^2} \right) \\
 & \quad + \left(\frac{0.0089}{1 + (205 - 206)^2} \right) + \left(\frac{0.0089}{1 + (206 - 203)^2} \right) \\
 & \quad + \left(\frac{0.0089}{1 + (206 - 205)^2} \right) + \dots + \left(\frac{0.0000}{1 + (255 - 255)^2} \right) \\
 & = 0 + 0.00178 + 0.00089 + 0.00178 + 0.00445 + \\
 & \quad 0.00089 + 0.00445 + \dots + 0 \\
 & = 0.2001
 \end{aligned}$$

$$\begin{aligned}
 \text{Energy} & = 0.0000^2 + 0.0089^2 + 0.0089^2 + 0.0089^2 + 0.0089^2 \\
 & \quad + 0.0089^2 + 0.0089^2 + \dots + 0.0000^2 \\
 & = 0 + 0.00007 + 0.00007 + 0.00007 + 0.00007 + 0.00007 + \\
 & \quad 0.00007 + \dots + 0
 \end{aligned}$$



$$= 0.01116$$

$$\begin{aligned} \text{Entropy} &= 0 + (-0.0089 * \log(0.0089)) + (-0.0089 * \log(0.0089)) \\ &\quad + (-0.0089 * \log(0.0089)) + (-0.0089 * \log(0.0089)) \\ &\quad + (-0.0089 * \log(0.0089)) + (-0.0089 * \log(0.0089)) \\ &\quad + \dots + 0 \end{aligned}$$

$$= 0 + 0.0182 + 0.0182 + 0.0182 + 0.0182 + 0.0182 + 0.0182 + \dots + 0$$

$$= 4.5606$$

d) Sudut 135°

$$\begin{aligned} \text{Contrast} &= ((0 - 0)^2 * 0.0000) + ((211 - 212)^2 * 0.0102) \\ &\quad + ((211 - 214)^2 * 0.0102) + ((212 - 211)^2 * 0.0102) \\ &\quad + ((212 - 212)^2 * 0.0204) + ((214 - 211)^2 * 0.0102) \\ &\quad + \dots + ((255 - 255)^2 * 0.0000) \end{aligned}$$

$$= 0 + 0.0102 + 0.0918 + 0.0102 + 0 + 0.0918 + \dots + 0$$

$$= 176.4693$$

$$\text{Homogeneity} = \left(\frac{0.0000}{1 + (0 - 0)^2} \right) + \left(\frac{0.0102}{1 + (211 - 212)^2} \right)$$

$$+ \left(\frac{0.0102}{1 + (211 - 214)^2} \right) + \left(\frac{0.0102}{1 + (212 - 211)^2} \right)$$

$$+ \left(\frac{0.0204}{1 + (212 - 212)^2} \right) + \left(\frac{0.0102}{1 + (214 - 211)^2} \right) + \dots$$

$$+ \left(\frac{0.0000}{1 + (255 - 255)^2} \right)$$

$$= 0 + 0.0051 + 0.00102 + 0.0051 + 0 + 0.00102 + \dots + 0$$

$$= 0.2056$$

$$\begin{aligned} \text{Energy} &= 0.0000^2 + 0.0102^2 + 0.0102^2 + 0.0102^2 + 0.0204^2 \\ &\quad + 0.0102^2 + 0.0000^2 \end{aligned}$$

$$= 0 + 0.000104 + 0.000104 + 0.000104 + 0.000416 + 0.000104 + \dots + 0$$

$$= 0.01103$$

$$\begin{aligned} \text{Entropy} &= 0 + (-0.0102 * \log(0.0102)) + (-0.0102 * \log(0.0102)) \\ &\quad + (-0.0102 * \log(0.0102)) + (-0.0204 * \log(0.0204)) \\ &\quad + (-0.0102 * \log(0.0102)) + \dots + 0 \end{aligned}$$

$$= 0 + 0.0203 + 0.0203 + 0.0203 + 0.0344 + 0.0203 + \dots + 0$$

$$= 4.5283$$



4.2.3 Klasifikasi Mutu Jeruk

Dari hasil perhitungan GLCM, selanjutnya dilakukan proses pembelajaran SVM. Dimana nantinya didapatkan model SVM yang akan digunakan untuk proses identifikasi citra baik dan buruk. SVM digunakan pada proses sliding window 64x64 untuk mengidentifikasi berapa banyak citra baik dan citra buruk. Setelah didapatkan jumlah citra buruk maka dilakukan perhitungan persentase untuk menentukan *grade* atau mutu dari buah jeruk tersebut. Dimisalkan pada jeruk A terdapat 10 citra 64x64 yang teridentifikasi buruk.

Pertama dilakukan konversi nilai 64 piksel kedalam *centimeter*, yang nilainya adalah 0.22 cm. Kemudian dilakukan perhitungan luas citra 64x64.

$$\text{luas citra } 64 \times 64 = 0.22 * 0.22 = 0.0484 \text{ cm}$$

Selanjutnya total area cacat pada jeruk A adalah sebagai berikut:

$$\text{area cacat jeruk A} = (0.0484 * 10) = 0.484 \text{ cm}$$

Setelah itu, dilakukan perhitungan luas permukaan jeruk.

$$\text{luas permukaan jeruk A} = \pi * r^2$$

r adalah jari-jari yang didapatkan dari perhitungan diameter. Untuk mendapatkan diameter dilakukan konversi dengan perbandingan 288 piksel adalah 1 cm. Nilai ini didapatkan dari perhitungan manual antara ukuran asli jeruk yang dibandingkan dengan hasil foto dalam piksel.

$$\text{diameter jeruk A} = \frac{2150}{288} = 7.465 \text{ cm}$$

$$r = \frac{7.465}{2} = 3.7325$$

$$\text{luas permukaan jeruk A} = 3.14 * (3.7325)^2 = 43.745$$

Setelah didapatkan luas permukaan jeruk, dilakukan perhitungan persentase area cacat pada jeruk A.

$$\text{persentase area cacat jeruk A} = \frac{0.484}{43.745} * 100\% = 1.106\%$$

Didapatkan nilai persentase area cacat jeruk A adalah sebesar 1.106% dimana nilai tersebut adalah dibawah 10%. Sehingga mutu jeruk A diklasifikasikan sebagai *Grade A*.

4.3 Perancangan Skenario Pengujian

Pada penelitian ini dilakukan pengujian terkait algoritme yang digunakan. Pengujian tersebut antara lain:

1. Pengujian untuk mengetahui jumlah data latih yang terbaik.
2. Pengujian untuk mengetahui nilai *distance* yang terbaik.
3. Pengujian untuk mengetahui arah GLCM yang terbaik.

4.3.1 Pengujian Jumlah Data Latih

Pengujian ini dilakukan untuk mengetahui pengaruh jumlah data latih terhadap hasil klasifikasi. Tujuan dari pengujian ini adalah untuk mendapatkan jumlah data latih yang tepat terhadap nilai *distance* dan hasil klasifikasi pada arah GLCM. Variasi data latih yang digunakan adalah 10, 20, 30, 40, 50 dan 60 data. Pengujian akan dilakukan pada 40 data uji, dengan variasi nilai *d* yang digunakan yaitu, $d=1$, $d=2$ dan $d=3$ pada 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° .

4.3.2 Pengujian Nilai *Distance* (*d*)

Pengujian ini dilakukan untuk mengetahui pengaruh nilai ketetanggaan piksel (*distance*) terhadap hasil klasifikasi. Tujuan dari pengujian ini adalah untuk mendapatkan nilai *distance* yang tepat terhadap hasil klasifikasi. Pengujian akan dilakukan pada 40 data uji, dengan variasi nilai *d* yang digunakan yaitu, $d=1$, $d=2$ dan $d=3$ pada 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° .

4.3.3 Pengujian Arah GLCM

Pengujian ini dilakukan untuk mengetahui pengaruh arah GLCM terhadap hasil klasifikasi. Tujuan dari pengujian ini adalah untuk mendapatkan arah GLCM yang paling tepat terhadap hasil klasifikasi. Pengujian akan dilakukan pada 40 data uji, dengan nilai *d* yang digunakan adalah $d=1$, jumlah data latih sebanyak 20 dan tentunya diaplikasikan pada 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° . Dari 4 arah GLCM tersebut dihitung total data terklasifikasi benar dan total data terklasifikasi salah untuk mendapatkan persentase akurasinya.

Ketiga pengujian tersebut yaitu jumlah data latih, nilai *distance* (*d*) dan arah GLCM dilakukan bersamaan pada tiap-tiap parameternya. Diantaranya adalah pengujian jumlah data latih 20 dengan $d=1$ pada 4 arah GLCM, pengujian jumlah data latih 20 dengan $d=2$ pada 4 arah GLCM, pengujian jumlah data latih 20 dengan $d=3$ pada 4 arah GLCM, pengujian jumlah data latih 40 dengan $d=1$ pada 4 arah GLCM, pengujian jumlah data latih 40 dengan $d=2$ pada 4 arah GLCM, pengujian jumlah data latih 40 dengan $d=3$ pada 4 arah GLCM, pengujian jumlah data latih 60 dengan $d=1$ pada 4 arah GLCM, pengujian jumlah data latih 60 dengan $d=2$ pada 4 arah GLCM, dan pengujian jumlah data latih 60 dengan $d=3$ pada 4 arah GLCM. Yang hasilnya dituliskan kedalam tabel sesuai dengan rancangan pada Tabel 4.9. Rancangan rangkuman dari seluruh pengujian dapat dilihat pada Tabel 4.10.

Tabel 4.9 Rancangan Pengujian

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1								
2								
3								

4								
5								
...								
40								
Total Benar (%)								
Total Salah (%)								

Tabel 4.10 Rancangan Pengujian Data Latih, *Distance* dan Arah GLCM

Banyak Data Latih	<i>d</i>	Hasil Klasifikasi (%)							
		Arah GLCM							
		0°		45°		90°		135°	
		B	S	B	S	B	S	B	S
10	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								
20	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								
30	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								
40	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								
50	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								
60	<i>d</i> =1								
	<i>d</i> =2								
	<i>d</i> =3								

BAB 5 IMPLEMENTASI

Pada Bab 5 ini dijelaskan mengenai implementasi berdasarkan perancangan yang telah disusun sebelumnya pada Bab Perancangan. Bab ini terdiri dari penjelasan mengenai lingkungan implementasi, batasan implementasi dan implementasi aplikasi. Pada sub bab lingkungan implementasi berisi penjelasan tentang spesifikasi kebutuhan yang digunakan pada penelitian ini. Kebutuhan tersebut terdiri dari kebutuhan *hardware* dan kebutuhan *software*. Sub bab batasan implementasi berisi penjelasan tentang batasan-batasan apa saja yang diterapkan dalam penelitian. Dan pada sub bab implementasi aplikasi berisi implementasi kode program yang terdiri dari *pre-processing* dan ekstraksi ciri GLCM.

5.1 Lingkungan Implementasi

Pada sub bab ini berisi penjelasan mengenai lingkungan implementasi sistem pemanfaatan ciri GLCM citra buah jeruk keprok untuk klasifikasi mutu yang terdiri dari lingkungan *hardware* dan lingkungan *software*.

5.1.1 Lingkungan *Hardware*

Lingkungan *hardware* yang digunakan dalam pembuatan sistem pemanfaatan ciri GLCM citra buah jeruk keprok untuk klasifikasi mutu adalah sebagai berikut:

1. *Processor*: Intel Core i5-4210U CPU @ 1.70GHz 2.40GHz
2. *Memory*: 8.00 GB
3. *VGA*: NVIDIA GEFORCE 820M
4. *ROM* 500 GB HDD
5. *Monitor* 14 inch
6. *Keyboard*
7. *Mouse*

5.1.2 Lingkungan *Software*

Lingkungan *software* yang digunakan dalam pembuatan sistem pemanfaatan ciri GLCM citra buah jeruk keprok untuk klasifikasi mutu adalah sebagai berikut:

1. *Operating system* Microsoft Windows 10 Pro.
2. Microsoft Office Word 2016 yang digunakan untuk membuat laporan penelitian.
3. Microsoft Office PowerPoint 2016 yang digunakan untuk membuat presentasi penelitian.
4. Microsoft Office Excel 2016 yang digunakan untuk menyimpan data pelatihan.
5. JetBrains PyCharm Community Edition yang digunakan untuk menulis program.
6. CorelDRAW X8 digunakan untuk *cropping* citra.

5.2 Batasan Implementasi

Batasan-batasan dalam implementasi sistem pemanfaatan ciri GLCM citra buah jeruk keprok untuk klasifikasi mutu adalah sebagai berikut:

1. Jumlah data yang digunakan pada penelitian ini sebanyak 100 data yang terbagi menjadi 60 data latih dan 40 data uji.
2. Penggunaan data latih pada setiap parameter pengujian data latih ditentukan berdasarkan nama *file* secara urut, 10 data = *file* 1-10, 20 data = *file* 1-20, 30 data = *file* 1-30, 40 data = *file* 1-40, 50 data = *file* 1-50 dan 60 data = *file* 1-60.
3. Metode *pre-processing* yang digunakan adalah *grayscale* dan segmentasi.
4. Ekstraksi ciri *gray level co-occurrence matrix* (GLCM) menghasilkan 4 ciri yaitu, *contrast*, *homogeneity*, *energy* dan *entropy*.
5. Identifikasi cacat buah (klasifikasi citra baik/buruk) menggunakan metode *support vector machine* (SVM) dengan *library scikit-learn*.
6. Kernel SVM yang digunakan adalah kernel *linear*.
7. Klasifikasi mutu berdasarkan persentase dari total hasil identifikasi cacat buah yang didapatkan dari jumlah citra buruk (hasil *sliding window* 64x64 pada citra masukan) yang nilainya dikonversi kedalam *centimeter*, terhadap luas permukaan jeruk.
8. Ukuran asli data citra adalah 4032x3024 piksel dan dilakukan *resize* menjadi 2000x1500 piksel. Citra hasil *resize* tersebut yang digunakan mulai proses *development* hingga *application*.
9. Citra yang digunakan untuk pengujian adalah citra hasil segmentasi yang berukuran 640x640 piksel.

5.3 Implementasi Aplikasi

Pada sub bab ini berisi penjelasan mengenai implementasi kode program berdasarkan perancangan yang telah dibuat pada bab sebelumnya. Implementasi aplikasi terdiri dari *pre-processing* dan ekstraksi ciri GLCM. Pada subsub bab *pre-processing* terdiri atas implementasi *grayscale* dan implementasi segmentasi. Dan pada subsub bab perhitungan GLCM terdiri dari implementasi pembentukan matriks GLCM dan implementasi ekstraksi ciri.

5.3.1 Pre-Processing

5.3.1.1 Implementasi Grayscale

Hal pertama yang dilakukan pada pengimplementasian sistem pemanfaatan ciri GLCM citra buah jeruk keprok untuk klasifikasi mutu adalah mengubah citra masukan yang berwarna (RGB) menjadi citra keabuan/*grayscale*. Implementasi kode program *grayscale* terdapat pada fungsi *grayscale()* pada *Sourcecode* 5.1.

No	
1	<code>def grayscaleing(rgb_img):</code>
2	<code> t, l, c = rgb_img.shape</code>
3	<code> gray_img = np.zeros((t, l), dtype=np.uint8)</code>
4	<code> for y in range(t):</code>
5	<code> for x in range(l):</code>
6	<code> blue, green, red = rgb_img[y, x]</code>
7	<code> gray = ((red * 0.299) + (green * 0.587) + (blue</code>
8	<code> * 0.114))</code>
9	<code> gray_img[x, y] = gray</code>
	<code> return gray_img</code>

Sourcecode 5.1 Implementasi Fungsi *grayscaleing()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.1:

1. Baris 1 adalah deklarasi fungsi *grayscaleing()* dengan parameter berupa citra masukan RGB dengan format JPEG.
2. Baris 2 adalah inisialisasi variabel *t*, *l*, dan *c* yang masing-masing digunakan untuk mendapatkan nilai tinggi, lebar dan *channel* yaitu BGR dari citra.
3. Baris 3 adalah inisialisasi variabel *gray_img* yang berupa array berisi nilai 0 dengan ukuran tinggi dan lebar sesuai dengan nilai yang didapatkan dari citra masukan.
4. Baris 4 dan 5 adalah deklarasi perulangan untuk tiap point *x,y*.
5. Baris 6 adalah inisialisasi variabel *blue*, *green* dan *red* untuk menyimpan nilai RGB dari tiap piksel.
6. Baris 7 adalah perhitungan nilai *gray* (keabuan) dari tiap-tiap piksel dengan cara menjumlahkan perkalian variabel *red* dengan 0.299, *green* dengan 0.587 dan *blue* dengan 0.114.
7. Baris 8 adalah proses penyimpanan nilai *gray* kedalam variabel *gray_img*.
8. Baris 9 digunakan untuk mengembalikan nilai variabel *gray_img*.

5.3.1.2 Implementasi Segmentasi

Dari citra masukan perlu dilakukan segmentasi untuk mengambil bagian objek dari jeruk. Terutama bagian tengah sekitar tangkai dimana bagian tersebut adalah letak cacat dari buah. Implementasi kode program segmentasi terdapat pada fungsi *segmentasi()* pada *Sourcecode* 5.2.

No	
1	<code>for image_path in glob.glob("data_uji/*.jpg"):</code>
2	<code> img = cv2.imread(image_path)</code>
3	<code> print(image_path)</code>
4	<code> img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)</code>
5	<code> img_blur = cv2.GaussianBlur(img_gray, (25, 25), 0)</code>
6	<code> retval, img_binary = cv2.threshold(img_gray, 0, 255,</code>
7	<code> cv2.THRESH_BINARY + cv2.THRESH_OTSU)</code>
8	<code> retval1, img_contours, retval2 =</code>
	<code> cv2.findContours(img_binary, cv2.RETR_EXTERNAL,</code>
	<code> cv2.CHAIN_APPROX_SIMPLE)</code>
	<code> min_coin_area = 3000</code>

```

9      large_area = [cv2.contourArea(cnt) for cnt in
10      img_contours]
11     large_contours = [cnt for cnt in img_contours]
12     print("large_contours", np.shape(large_contours))
13     img_cp_cnt = img.copy()
14     bounding_img = img.copy()
15     contour = large_contours[np.argmax(large_area)]
16     x, y, w, h = cv2.boundingRect(contour)
17
18     potong = img[y:y + h, x:x + w]
19     nomor = re.findall(r'\d+', image_path)
20     cv2.imwrite("crop_untuk_uji/crop_" + (nomor[0]) +
21     ".jpg", potong)
22
23     for image_path in glob.glob("crop_untuk_uji/*.jpg"):
24     nomor = re.findall(r'\d+', image_path)
25     img = cv2.imread(image_path)
26     __displacement_size = 1
27     __region_size = 1
28     titik1, titik2, titik3, titik4 = get_position(img, 70)
29     if titik1 == 0 or titik2 == 0 or titik3 == 0 or titik4
30     == 0:
31         continue
32     else:
33         titik1, titik2, titik3, titik4, x_awal, y_awal,
34         x_akhir, y_akhir = titik_baru(titik1, titik2,
35         titik3, titik4)
36         if ((x_akhir - x_awal) * (y_akhir - y_awal)) >
37         150000:
38             print("\t", (x_akhir - x_awal) * (y_akhir -
39             y_awal))
40             potong = img[y_awal:y_awal + 640, x_awal:x_awal
41             + 640]
42             cv2.imwrite("uji/uji_" + (nomor[0]) + ".jpg",
43             potong)

```

Sourcecode 5.2 Implementasi Fungsi segmentasi()

Berikut ini merupakan penjelasan dari *Sourcecode* 5.2:

1. Baris 1-17 merupakan proses segmentasi untuk cropping citra yang pertama yaitu membedakan objek dengan latar belakang untuk mendapatkan citra objek buah jeruk saja.
2. Baris 18-32 merupakan proses segmentasi untuk cropping citra yang kedua yaitu mengambil bagian tengah sekitar tangkai dari objek buah jeruk dengan ukuran 640x640 piksel.

Implementasi kode program untuk proses mendapatkan posisi pengecekan titik objek jeruk dengan fungsi *get_position()* ditunjukkan pada *Sourcecode* 5.3.

No	
1	def get_position(img, threshold):
2	rows, column, _ = img.shape
	# titik 0, 0
3	for i in range(rows):
4	potong = img[i:i + 1, i:i + 1]
5	if potong[:, :, 0][0] < threshold:
6	titik1 = (i, i)
7	break



```

8         else:
9             continue
10        # titik 0, x
11        for i in range(rows):
12            potong = img[i:i + 1, (column - 2) - i:(column - 1)
13                    - i]
14            if potong[:, :, 0][0] < threshold:
15                titik2 = (i, (column - 1) - (2 * i))
16                break
17            else:
18                continue
19        # titik y,x
20        for i in range(rows):
21            potong = img[(rows - 2) - i:(rows - 1) - i, (column
22                    - 2) - i:(column - 1) - i]
23            if len(potong) == 0:
24                continue
25            elif potong[:, :, 0][0] < threshold:
26                titik3 = ((rows - 2) - i, (column - 2) - i)
27                break
28        # titik y,0
29        for i in range(rows):
30            potong = img[(rows - 2) - i:(rows - 1) - i, i:i + 1]
31            if potong[:, :, 0][0] < threshold:
32                titik4 = ((rows - 2) - i, i)
33                break
34            else:
35                continue
36        return titik1, titik2, titik3, titik4

```

Sourcecode 5.3 Implementasi Fungsi *get_position()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.3:

1. Baris 1 merupakan deklarasi fungsi *get_position()* dengan parameter *img* yang merupakan citra masukan dan *threshold* yang merupakan nilai ambang batas yang digunakan dalam proses pencarian posisi objek.
2. Baris 2 adalah inisialisasi variabel *rows* dan *column* untuk mendapatkan nilai baris dan kolom dari citra.
3. Baris 3-9 merupakan proses pencarian titik 0,0.
4. Baris 10-16 merupakan proses pencarian titik 0,x.
5. Baris 17-23 merupakan proses pencarian titik y,x.
6. Baris 24-30 merupakan proses pencarian titik y,0.
7. Baris 31 mengembalikan nilai variabel *titik1*, *titik2*, *titik3* dan *titik4*.

Implementasi kode program untuk proses mendapatkan titik terpendek pada bagian tengah objek jeruk dengan fungsi *titik_baru()* ditunjukkan pada *Sourcecode* 5.4.

No	
1	def titik_baru(titik1, titik2, titik3, titik4):
2	titik1s, titik2s, titik3s, titik4s = [0, 0], [0, 0], [0, 0], [0, 0]



```

3      # x
4      if titik1[1] < titik4[1]:
5          titik1s[1] = titik1[1]
6          titik4s[1] = titik1[1]
7          x_awal = titik1[1]
8          y_awal = titik1[0]
9      else:
10         titik1s[1] = titik4[1]
11         titik4s[1] = titik4[1]
12         x_awal = titik4[1]
13         y_awal = titik1[0]
14     if titik2[1] < titik3[1]:
15         titik3s[1] = titik2[1]
16         titik2s[1] = titik2[1]
17         x_akhir = titik2[1]
18     else:
19         titik3s[1] = titik3[1]
20         titik2s[1] = titik3[1]
21         x_akhir = titik3[1]
22     # y
23     if titik1[0] > titik2[0]:
24         titik1s[0] = titik1[0]
25         titik2s[0] = titik1[0]
26     else:
27         titik1s[0] = titik2[0]
28         titik2s[0] = titik2[0]
29     if titik3[0] > titik4[0]:
30         titik3s[0] = titik3[0]
31         titik4s[0] = titik3[0]
32         y_akhir = titik3[0]
33     else:
34         titik3s[0] = titik4[0]
35         titik4s[0] = titik4[0]
36         y_akhir = titik4[0]
37     return tuple(titik1s), tuple(titik2s), tuple(titik3s),
38         tuple(titik4s), x_awal, y_awal, x_akhir, y_akhir

```

Sourcecode 5.4 Implementasi Fungsi titik_baru()

Berikut ini merupakan penjelasan dari Sourcecode 5.4:

1. Baris 1 merupakan deklarasi fungsi titik_baru() dengan parameter *titik1*, *titik2*, *titik3* dan *titik4* yang merupakan posisi titik dari bagian tengah jeruk.
2. Baris 2 adalah inialisasi variabel *titik1s*, *titik2s*, *titik3s* dan *titik4s* dengan nilai [0,0].
3. Baris 3-20 merupakan proses pencarian titik terpendek pada sumbu x.
4. Baris 21-34 merupakan proses pencarian titik terpendek pada sumbu y.
5. Baris 35 digunakan untuk mengembalikan nilai variabel *titik1s*, *titik2s*, *titik3s*, *titik4s*, *x_awal*, *y_awal*, *x_akhir* dan *y_akhir*.

5.3.2 Perhitungan GLCM

Pada tahap ini dilakukan perhitungan GLCM dari citra hasil *pre-processing*. Pertama-tama dilakukan proses perhitungan matriks awal pada arah 0°, 45°, 90° dan 135°. Setelah itu, dilakukan pembentukan matriks GLCM yang simetris dengan

cara menjumlahkan antara matriks awal dengan matriks transposnya. Setelah didapatkan matriks yang simetris tersebut selanjutnya perlu dilakukan normalisasi agar jumlah seluruh elemen matriks bernilai 1, dengan cara membagi setiap elemen atau tiap nilai pada matriks dengan total keseluruhan pasangan piksel. Kemudian dilakukan perhitungan ciri GLCM. Terdapat 4 ciri yang akan digunakan, yaitu *contrast*, *homogeneity*, *energy* dan *entropy*.

5.3.2.1 Implementasi Pembentukan Matriks GLCM

Fungsi yang kedua adalah `matriks_glcm()`. Dimana pada fungsi ini berisi proses pembentukan matriks GLCM dengan hasil akhir berupa matriks GLCM yang telah ternormalisasi. Nilai dari matriks GLCM berdasarkan pembentukan dari arah 0° , 45° , 90° dan 135° . Implementasi kode program pembentukan matriks GLCM terdapat pada fungsi `matriks_glcm()` pada *Sourcecode* 5.5.

No	
1	<code>def matriks_glcm(image, sudut, distance):</code>
2	<code> rows, cols = window.shape</code>
3	<code> glcm_matrix = np.zeros((256, 256))</code>
4	<code> if sudut == 0:</code>
5	<code> for row in range(0, rows):</code>
6	<code> for col in range(0, cols):</code>
7	<code> if col + distance >= cols:</code>
8	<code> continue</code>
9	<code> point_1 = image [row, col]</code>
10	<code> point_2 = image[row, col + distance]</code>
11	<code> glcm_matrix[point_1, point_2] += 1</code>
12	<code> elif sudut == 45:</code>
13	<code> for row in range(0, rows):</code>
14	<code> if row - distance < 0:</code>
15	<code> continue</code>
16	<code> for col in range(0, cols):</code>
17	<code> if col + distance >= cols:</code>
18	<code> continue</code>
19	<code> point_1 = image[row, col]</code>
20	<code> point_2 = image[row - distance, col +</code>
	<code> distance]</code>
21	<code> glcm_matrix[point_1, point_2] += 1</code>
22	<code> elif sudut == 90:</code>
23	<code> for row in range(0, rows):</code>
24	<code> if row - distance < 0:</code>
25	<code> continue</code>
26	<code> for col in range(0, cols):</code>
27	<code> point_1 = image[row, col]</code>
28	<code> point_2 = image[row - distance, col]</code>
29	<code> glcm_matrix[point_1, point_2] += 1</code>
30	<code> elif sudut == 135:</code>
31	<code> for row in range(0, rows):</code>
32	<code> if row - distance < 0:</code>
33	<code> continue</code>
34	<code> for col in range(0, cols):</code>
35	<code> if col - distance < 0:</code>
36	<code> continue</code>
37	<code> point_1 = image[row, col]</code>
38	<code> point_2 = image[row - distance, col -</code>
	<code> distance]</code>

```

39         glcm_matrix[point_1, point_2] += 1
40     glcm_matrix_transpose = glcm_matrix.transpose()
41     glcm_matrix_simetris = glcm_matrix +
42     glcm_matrix_transpose
43     total_pasangan =
44     get_total_pasangan(glcm_matrix_simetris)
45     glcm_matrix_normalisasi = glcm_matrix_simetris /
46     total_pasangan
47     return glcm_matrix_normalisasi

```

Sourcecode 5.5 Implementasi Fungsi matriks_glcm()

Berikut ini merupakan penjelasan dari Sourcecode 5.5:

1. Baris 1 adalah deklarasi fungsi matriks_glcm() dengan parameter *image* yang merupakan citra masukan, sudut yang merupakan arah GLCM, dan *distance* yang merupakan jarak ketetanggaan.
2. Baris 2 adalah inialisasi variabel *rows* dan *cols* untuk mendapatkan nilai baris dan kolom dari citra.
3. Baris 3 adalah inialisasi variabel array *glcm_matrix* dengan panjang 256x256 dengan nilai 0.
4. Baris 4-11 merupakan proses perhitungan matriks awal GLCM apabila masukan parameter sudut bernilai 0.
5. Baris 12-21 merupakan proses perhitungan matriks awal GLCM apabila masukan parameter sudut bernilai 45.
6. Baris 22-29 merupakan proses perhitungan matriks awal GLCM apabila masukan parameter sudut bernilai 90.
7. Baris 30-39 merupakan proses perhitungan matriks awal GLCM apabila masukan parameter sudut bernilai 135.
8. Baris 40 adalah inialisasi variabel *glcm_matrix_transpose* dengan nilai dari variabel *glcm_matrix* yang di transposkan.
9. Baris 41 adalah inialisasi variabel *glcm_matrix_simetris* dengan nilai penjumlahan variabel *glcm_matrix_transpose* dengan *glcm_matrix* yang bertujuan untuk mendapatkan matriks GLCM yang simetris.
10. Baris 42 adalah inialisasi variabel *total_pasangan* untuk mendapatkan nilai total pasangan pada matriks GLCM.
11. Baris 43 adalah inialisasi variabel *glcm_matrix_normalisasi* dengan nilai hasil pembagian antara variabel *glcm_matrix_simetris* dengan variabel *total_pasangan* untuk mendapatkan matriks GLCM yang ternormalisasi.
12. Baris 44 digunakan untuk mengembalikan nilai variabel *glcm_matrix_normalisasi*.

Implementasi kode program untuk perhitungan total pasangan piksel matriks GLCM dengan fungsi `get_total_pasangan()` ditunjukkan pada *Sourcecode* 5.6.

No	
1	<code>def get_total_pasangan(glcm_matriks):</code>
2	<code> y, x = glcm_matriks.shape</code>
3	<code> total = 0</code>
4	<code> for row in range(0, y):</code>
5	<code> for col in range(0, x):</code>
6	<code> total += glcm_matriks[row, col]</code>
7	<code> return total</code>

Sourcecode 5.6 Implementasi Fungsi `get_total_pasangan()`

Berikut ini merupakan penjelasan dari *Sourcecode* 5.6:

1. Baris 1 adalah deklarasi fungsi `get_total_pasangan()` dengan parameter `glcm_matriks` yang merupakan matriks GLCM.
2. Baris 2 adalah inisialisasi variabel `y` dan `x` untuk mendapatkan nilai ukuran matriks.
3. Baris 3 adalah inisialisasi variabel `total` dengan nilai 0.
4. Baris 4-6 merupakan proses perhitungan jumlah total pasangan piksel matriks GLCM.
5. Baris 7 digunakan untuk mengembalikan nilai variabel `total`.

5.3.2.2 Implementasi Ekstraksi Ciri

Fungsi berikutnya adalah `ekstraksi_ciri()`. Dimana pada fungsi ini berisi proses ekstraksi ciri *gray level co-occurrence matrix* (GLCM). Ekstraksi ciri GLCM ini menghasilkan 4 ciri yaitu, *contrast*, *homogeneity*, *energy* dan *entropy*. Implementasi kode program ekstraksi ciri terdapat dalam fungsi `ekstraksi_ciri()` pada *Sourcecode* 5.7.

No	
1	<code>def ekstraksi_ciri(glcm_matriks):</code>
2	<code> contrast = get_contrast(glcm_matriks)</code>
3	<code> homogeneity = get_homogeneity(glcm_matriks)</code>
4	<code> energy = get_energy(glcm_matriks)</code>
5	<code> entropy = get_entropy(glcm_matriks)</code>
6	<code> print("contrast: ", contrast, "\thomogeneity: ",</code> <code> homogeneity, "\tenergy: ", energy, "\tentropy",</code> <code> entropy)</code>
7	<code> return contrast, homogeneity, energy, entropy</code>

Sourcecode 5.7 Implementasi Fungsi `ekstraksi_ciri()`

Berikut ini merupakan penjelasan dari *Sourcecode* 5.7:

1. Baris 1 adalah deklarasi fungsi `ekstraksi_ciri()` dengan parameter `glcm_matriks` yang merupakan matriks GLCM.
2. Baris 2 adalah inisialisasi variabel `contrast` bernilai ciri GLCM *contrast* dengan hasil memanggil fungsi `get_contrast()`.



3. Baris 3 adalah inialisasi variabel *homogeneity* bernilai ciri GLCM *homogeneity* dengan hasil memanggil fungsi *get_homogeneity()*.
4. Baris 4 adalah inialisasi variabel *energy* bernilai ciri GLCM *energy* dengan hasil memanggil fungsi *get_energy()*.
5. Baris 5 adalah inialisasi variabel *entropy* bernilai ciri GLCM *entropy* dengan hasil memanggil fungsi *get_entropy()*.
6. Baris 6 digunakan untuk mencetak nilai variabel *contrast*, *homogeneity*, *energy*, dan *entropy*.
7. Baris 7 digunakan untuk mengembalikan nilai variabel *contrast*, *homogeneity*, *energy*, dan *entropy*.

Implementasi kode program untuk perhitungan nilai ciri GLCM *contrast* dengan fungsi *get_contrast()* ditunjukkan pada *Sourcecode* 5.8.

No	
1	<code>def get_contrast(glcm_matriks):</code>
2	<code> contrast = 0</code>
3	<code> for row in range(0, 256):</code>
4	<code> for col in range(0, 256):</code>
5	<code> contrast += ((row - col) ** 2) * glcm_matriks[row, col])</code>
6	<code> return contrast</code>

Sourcecode 5.8 Implementasi Fungsi *get_contrast()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.8:

1. Baris 1 adalah deklarasi fungsi *get_contrast()* dengan parameter *glcm_matriks* yang merupakan matriks GLCM.
2. Baris 2 adalah inialisasi variabel *contrast* dengan nilai 0.
3. Baris 3-5 merupakan proses perhitungan ciri GLCM *contrast*.
4. Baris 6 digunakan untuk mengembalikan nilai variabel *contrast*.

Implementasi kode program untuk perhitungan nilai ciri GLCM *homogeneity* dengan fungsi *get_homogeneity()* ditunjukkan pada *Sourcecode* 5.9.

No	
1	<code>def get_homogeneity(glcm_matriks):</code>
2	<code> homogeneity = 0</code>
3	<code> for row in range(0, 256):</code>
4	<code> for col in range(0, 256):</code>
5	<code> homogeneity += (glcm_matriks[row, col] / (1 + (np.abs((row - col))))))</code>
6	<code> return homogeneity</code>

Sourcecode 5.9 Implementasi Fungsi *get_homogeneity()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.9:

1. Baris 1 adalah deklarasi fungsi *get_homogeneity()* dengan parameter *glcm_matriks* yang merupakan matriks GLCM.
2. Baris 2 adalah inialisasi variabel *homogeneity* dengan nilai 0.



3. Baris 3-5 merupakan proses perhitungan ciri GLCM *homogeneity*.
4. Baris 6 digunakan untuk mengembalikan nilai variabel *homogeneity*.

Implementasi kode program untuk perhitungan nilai ciri GLCM *energy* dengan fungsi *get_energy()* ditunjukkan pada *Sourcecode* 5.10.

No	
1	<code>def get_energy(glcm_matriks):</code>
2	<code> energy = 0</code>
3	<code> for row in range(0, 256):</code>
4	<code> for col in range(0, 256):</code>
5	<code> energy += (glcm_matriks[row, col] ** 2)</code>
6	<code> return energy</code>

Sourcecode 5.10 Implementasi Fungsi *get_energy()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.10:

1. Baris 1 adalah deklarasi fungsi *get_energy()* dengan parameter *glcm_matriks* yang merupakan matriks GLCM.
2. Baris 2 adalah inisialisasi variabel *energy* dengan nilai 0.
3. Baris 3-5 merupakan proses perhitungan ciri GLCM *energy*.
4. Baris 6 digunakan untuk mengembalikan nilai variabel *energy*.

Implementasi kode program untuk perhitungan nilai ciri GLCM *entropy* dengan fungsi *get_entropy()* ditunjukkan pada *Sourcecode* 5.11.

No	
1	<code>def get_entropy(glcm_matriks):</code>
2	<code> entropy = 0.0</code>
3	<code> for row in range(0, 255):</code>
4	<code> for col in range(0, 255):</code>
5	<code> if glcm_matriks[row, col] == 0</code>
6	<code> entropy += 0</code>
7	<code> else:</code>
8	<code> entropy += (glcm_matriks[row, col] * np.log(glcm_matriks[row, col]))</code>
9	<code> entropy = -1 * entropy</code>
10	<code> return entropy</code>

Sourcecode 5.11 Implementasi Fungsi *get_entropy()*

Berikut ini merupakan penjelasan dari *Sourcecode* 5.11:

1. Baris 1 adalah deklarasi fungsi *get_entropy()* dengan parameter *glcm_matriks* yang merupakan matriks GLCM.
2. Baris 2 adalah inisialisasi variabel *entropy* dengan nilai 0.0.
3. Baris 3-8 merupakan proses perhitungan ciri GLCM *entropy*.
4. Baris 9 merupakan proses mengalikan nilai variabel *entropy* dengan -1.
5. Baris 10 digunakan untuk mengembalikan nilai variabel *entropy*.

5.3.3 Identifikasi Cacat Buah

Penelitian ini menggunakan metode *support vector machine* (SVM) untuk mengidentifikasi cacat buah pada suatu area *windowing* citra, apakah area tersebut baik atau buruk. Pengklasifikasian mutu jeruk berdasarkan persentase cacat buah yang dihitung dari luas *windowing* 64x64 dikalikan jumlah citra teridentifikasi buruk, kemudian dibandingkan dengan luas permukaan jeruk. Lalu didapatkan persentase sebagai acuan dalam pengklasifikasian mutu *grade Super*, *grade A* atau *grade B*. Metode SVM pada penelitian ini menggunakan *library scikit-learn* dengan kernel *linear*. Implementasi kode program identifikasi cacat buah terdapat dalam fungsi identifikasi() pada *Sourcecode* 5.12.

No	
1	def identifikasi(image, theta, d, nama_file, region=64, displacement=64):
2	rows, cols = image.shape
3	baik = 0
4	buruk = 0
5	luas64 = 0.0484
6	diameter = D.diameter(image)
7	jari_jari = diameter / 2
8	luas_lingkaran = 3.14 * (jari_jari ** 2)
9	model = load_model(nama_file)
10	# SLIDING WINDOW
11	for row in range(0, rows, displacement):
12	if row + region >= rows:
13	continue
14	for column in range(0, cols, displacement):
15	if column + region >= cols:
16	continue
17	# Melakukan windowing pada citra uji
18	window = image[row:row + region, column:column + region]
19	# Matriks GLCM
20	matrix_glcm = matriks_glcm(image=window, sudut=theta, distance=d)
21	# Ekstraksi ciri
22	ciri = np.array(ekstraksi_ciri(matrix_glcm)).reshape(-1, 4)
23	pred = model.predict(ciri)
24	print("predict: ", pred)
25	if pred == 'baik':
26	baik += 1
27	elif pred == 'buruk':
28	buruk += 1
29	luas_buruk = buruk * luas64
30	persentase = (luas_buruk / luas_lingkaran) * 100
31	if persentase == 0:
32	grade = 'Grade Super'
33	elif persentase <= 10:
34	grade = 'Grade A'
35	elif persentase > 10 and persentase <= 15:
36	grade = 'Grade B'
37	print("Buruk:", buruk, "Persentase Cacat Buah: ", persentase, "%\tKlasifikasi: ", grade)

Sourcecode 5.12 Implementasi Fungsi identifikasi()

Berikut ini merupakan penjelasan dari *Sourcecode* 5.12:

1. Baris 1 adalah deklarasi fungsi identifikasi() dengan parameter *image* yang merupakan citra masukan, *theta* yang merupakan sudut atau arah GLCM yang diinginkan, *d* yang merupakan jarak ketetanggaan, *region* yaitu ukuran *windowing* dan *displacement* yaitu jarak perpindahan tiap *window*.
2. Baris 2 adalah inisialisasi variabel *rows* dan *cols* untuk mendapatkan nilai ukuran citra.
3. Baris 3 adalah inisialisasi variabel *baik* untuk menyimpan nilai identifikasi citra baik dengan nilai 0.
4. Baris 4 adalah inisialisasi variabel *buruk* untuk menyimpan nilai identifikasi citra buruk dengan nilai 0.
5. Baris 5 adalah inisialisasi variabel *luas64* untuk menyimpan nilai luas citra 64x64 dalam ukuran *centimeter*.
6. Baris 6 adalah inisialisasi variabel *diameter* untuk menyimpan nilai diameter dari jeruk.
7. Baris 7 adalah inisialisasi variabel *jari_jari* untuk menyimpan nilai jari-jari lingkaran/jeruk.
8. Baris 8 adalah inisialisasi variabel *luas_lingkaran* untuk menyimpan nilai luas permukaan dari jeruk.
9. Baris 9 adalah inisialisasi variabel *model* untuk melakukan *load* model dengan memanggil fungsi *load_model()*.
10. Baris 10-28 adalah proses identifikasi citra baik dan buruk dengan cara melakukan *sliding window* dengan *region* tertentu pada citra masukan.
11. Baris 29 adalah inisialisasi variabel *luas_buruk* untuk menyimpan total nilai dari citra terklasifikasi buruk.
12. Baris 30 adalah inisialisasi variabel *persentase_cacat* untuk menghitung nilai persentase cacat buah.
13. Baris 31-36 adalah inisialisasi kondisi untuk *grade* buah berdasarkan persentase cacatnya.
14. Baris 37 adalah deklarasi *print* untuk mencetak jumlah teridentifikasi buruk, persentase cacat buah dan juga klasifikasi *grade*-nya.

Implementasi kode program untuk proses perhitungan diameter jeruk dengan fungsi *diameter()* ditunjukkan pada *Sourcecode* 5.13.

No	
1	<code>def diameter(nama_file):</code>
2	<code> nomor = re.findall(r'\d+', nama_file)</code>
3	<code> img = cv2.imread('data_jeruk_2_asli/jeruk_' + nomor[0]</code> <code> + '_atas.jpg')</code>
4	<code> y, x, c = img.shape</code>
5	<code> x_tengah = int(x / 2)</code>
6	<code> count = 0</code>

```

7     hijau = False
8     for row in range(0, y, 10):
9         img_temp = img.copy()
10        cv2.rectangle(img_temp, (x_tengah, row), (x_tengah
11        + 10, row + 10), (0, 0, 255), 3)
12        if not hijau:
13            cek = ck.cek_warna(img[row:row + 10,
14            x_tengah:x_tengah + 10])
15            if cek[0] <= 100:
16                hijau = True
17        elif hijau:
18            cv2.rectangle(img_temp, (x_tengah, row),
19            (x_tengah + 10, row + 10), (0, 255, 0), 3)
20            cek = ck.cek_warna(img[row:row + 10,
21            x_tengah:x_tengah + 10])
22            if cek[1] <= 100:s
23                hijau = False
24                count += 10
25        elif count > 0 and cek[0] >= 120 and not hijau:
26            break
27    d = count / 288
28    return d

```

Sourcecode 5.13 Implementasi Fungsi diameter()

Berikut ini merupakan penjelasan dari *Sourcecode* 5.13:

15. Baris 1 adalah deklarasi fungsi diameter() dengan parameter *nama_file* yang merupakan nama dari *file* data jeruk yang akan di cek diameternya.
16. Baris 2 adalah inisialisasi variabel *nomor* yang digunakan untuk mengambil nilai angka pada variabel *nama_file* dengan memanggil fungsi *findall()* dari *library re*.
17. Baris 3 adalah inisialisasi variabel *y, x, c* yang digunakan untuk mengambil nilai *y, x* dan *channel* dari citra.
18. Baris 4 digunakan untuk menyimpan model SVM dengan melakukan pemanggilan fungsi *joblib.dumb()* berparameter nilai variabel *classifier* dan *nama_file*.
19. Baris 5 adalah inisialisasi variabel *x_tengah* yang digunakan untuk menyimpan nilai tengah dari citra.
20. Baris 6 adalah inisialisasi variabel *count* yang digunakan untuk menyimpan total piksel dari objek jeruk.
21. Baris 7 adalah inisialisasi variabel hijau yang digunakan untuk menyimpan kondisi false.
22. Baris 8-22 adalah proses perhitungan total piksel dari objek buah jeruk.
23. Baris 23 adalah inisialisasi variabel *d* yang digunakan untuk menyimpan nilai diameter dari hasil pembagian total piksel (variabel *count*) dengan 288.
24. Baris 24 digunakan untuk mengembalikan nilai variabel *d*.

Implementasi kode program untuk proses pembelajaran *support vector machine* dengan fungsi SVM() ditunjukkan pada *Sourcecode* 5.14.

No	
1	def SVM(data, label, nama_file):
2	classifier = svm.SVC(kernel='linear')
3	classifier.fit(data, label)
4	joblib.dump(classifier, nama_file)

Sourcecode 5.14 Implementasi Fungsi SVM()

Berikut ini merupakan penjelasan dari *Sourcecode* 5.14:

1. Baris 1 adalah deklarasi fungsi svm() dengan parameter data yang merupakan data nilai hasil ekstraksi ciri, *label* yang merupakan kelas data, dan *nama_file* yang merupakan nama dari *file* yang diinisialisasi untuk menyimpan model SVM.
2. Baris 2 adalah inialisasi variabel *classifier* yang digunakan untuk inialisasi model SVM dengan memanggil fungsi SVC() dari *library scikit-learn*, kernel yang digunakan adalah kernel *linear*.
3. Baris 3 digunakan untuk melakukan pembelajaran SVM dari data *training* yang berhasil tersimpan.
4. Baris 4 digunakan untuk menyimpan model SVM dengan melakukan pemanggilan fungsi *joblib.dump()* berparameter nilai variabel *classifier* dan *nama_file*.

Implementasi kode program untuk proses *load* model SVM dengan fungsi *load_model()* ditunjukkan pada *Sourcecode* 5.15.

No	
1	def load_model(nama_file):
2	return joblib.load(nama_file)

Sourcecode 5.15 Implementasi Fungsi load_model()

Berikut ini merupakan penjelasan dari *Sourcecode* 5.15:

1. Baris 1 adalah deklarasi fungsi *load_model()* dengan parameter *nama_file* yang merupakan *file* model SVM yang telah disimpan untuk dipanggil.
2. Baris 2 digunakan untuk mengembalikan nilai dari model SVM dengan pemanggilan fungsi *joblib.load()*.

BAB 6 PENGUJIAN DAN ANALISIS

Pada Bab 6 ini dijelaskan tentang pengujian dan analisis yang dilakukan pada sistem pemanfaatan Gray Level Co-occurrence Matrix (GLCM) citra jeruk keprok untuk klasifikasi mutu. Pengujian dilakukan berdasarkan skenario pengujian yang telah dirancang pada Bab 4.

6.1 Pengujian

Pada sub-bab ini membahas mengenai pengujian yang telah dilakukan sesuai dengan skenario pengujian. Skenario pengujian yang dilakukan adalah sebagai berikut:

6.1.1 Pengujian Jumlah Data Latih

Pada pengujian pertama yaitu pengujian jumlah data latih, yang dilakukan adalah menguji banyaknya jumlah data latih terhadap akurasi hasil klasifikasi. Pengujian ini bertujuan untuk mengetahui berapa jumlah data latih yang terbaik yang dibutuhkan untuk permasalahan pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu. Pengujian ini diterapkan pada 40 data uji citra buah jeruk keprok batu 55, yang terdiri dari 12 data jeruk *Grade Super* dan 28 data jeruk *Grade A*.

Data latih citra jeruk utuh adalah sebanyak 60 data. Kemudian dilakukan *cropping* citra secara manual dengan ukuran 64x64 piksel. Dari tiap-tiap data latih tersebut diambil masing-masing 1 *cropping* citra baik dan 1 *cropping* citra buruk/cacat. Sehingga didapatkan sejumlah 60 data *cropping* citra baik dan 60 data *cropping* citra buruk/cacat. Data tersebut diberi nama urut mulai data 1 sampai dengan data 60. Parameter pengujian penelitian ini adalah 10, 20, 30, 40, 50 dan 60 data latih. Untuk parameter 10 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-10 dan data *cropping* citra buruk/cacat urutan 1-10 sebagai data pelatihan sistem. Untuk parameter 20 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-20 dan data *cropping* citra buruk/cacat urutan 1-20 sebagai data pelatihan sistem. Untuk parameter 30 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-30 dan data *cropping* citra buruk/cacat urutan 1-10 sebagai data pelatihan sistem. Untuk parameter 40 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-40 dan data *cropping* citra buruk/cacat urutan 1-40 sebagai data pelatihan sistem. Untuk parameter 50 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-50 dan data *cropping* citra buruk/cacat urutan 1-50 sebagai data pelatihan sistem. Untuk parameter 60 data dilakukan pengujian menggunakan data *cropping* citra baik urutan 1-60 dan data *cropping* citra buruk/cacat urutan 1-60 sebagai data pelatihan sistem. Jumlah tersebut diujikan pada 3 jarak ketetanggaan piksel (*distance*) yaitu, $d=1$, $d=2$ dan $d=3$, serta pada 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° .

Pada setiap data uji dilakukan *sliding window* berukuran 64x64 piksel untuk mendapatkan hasil identifikasi citra baik atau citra buruk/cacat, dengan menggunakan metode *support vector machine* (SVM) dengan kernel *linear*. Dari jumlah *window* teridentifikasi buruk dihitung luasnya dengan mengkonversikan piksel kedalam *centimeter*, untuk dibandingkan dengan luas keseluruhan permukaan jeruk sehingga didapatkan persentase cacat buah. Kemudian dihitung berapa persentase akurasi hasil klasifikasi pada tiap parameter pengujian. Dari pengujian tersebut diperoleh jumlah data latih yang terbaik pada sistem pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu ini. Terdapat tabel hasil uji coba terhadap jumlah data latih secara lengkap yang dapat dilihat pada Lampiran A hingga Lampiran R dan tabel ringkasnya dapat dilihat pada pada Tabel 6.1 sampai dengan Tabel 6.18 serta tabel kesimpulan dapat dilihat pada Tabel 6.19.

6.1.2 Pengujian Nilai *Distance* (d)

Pada pengujian kedua yaitu pengujian nilai *distance* (d), yang dilakukan adalah menguji nilai jarak ketetangaan piksel (*distance*) terhadap akurasi hasil klasifikasi. Jarak ketetangaan piksel merupakan jarak yang digunakan dalam penentuan pasangan piksel pada proses pembentukan matriks awal GLCM. Pengujian ini bertujuan untuk mengetahui berapa nilai *distance* yang terbaik yang dibutuhkan untuk permasalahan pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu. Nilai *distance* yang digunakan pada pengujian ini yaitu, $d=1$, $d=2$ dan $d=3$. Pengujian ini diterapkan pada 40 data uji citra buah jeruk keprok batu 55, yang terdiri dari 12 data jeruk *Grade Super* dan 28 data jeruk *Grade A*. Nilai *distance* tersebut diujikan pada parameter data latih 10, 20, 30, 40, 50 dan 60 data serta pada 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° .

Dari pengujian tersebut diperoleh nilai *distance* yang terbaik untuk sistem pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu. Terdapat tabel hasil uji coba terhadap jumlah data latih secara lengkap yang dapat dilihat pada Lampiran A hingga Lampiran R dan tabel ringkasnya dapat dilihat pada pada Tabel 6.1 sampai dengan Tabel 6.18 serta tabel kesimpulan dapat dilihat pada Tabel 6.19.

6.1.3 Pengujian Arah GLCM

Pada pengujian ketiga yaitu pengujian arah GLCM, yang dilakukan adalah menguji arah *gray level co-occurrence matrix* (GLCM) terhadap akurasi hasil klasifikasi. Pengujian ini bertujuan untuk mengetahui arah GLCM yang terbaik pada permasalahan pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu. Pengujian ini diterapkan pada 40 data uji citra buah jeruk keprok batu 55, yang terdiri dari 12 data jeruk *Grade Super* dan 28 data jeruk *Grade A*. Terdapat 4 arah GLCM yang digunakan pada pengujian ini yaitu, 0° , 45° , 90° dan 135° . Dari 4 arah tersebut diujikan dengan nilai *distance* sebesar $d=1$, $d=2$ dan $d=3$ dan jumlah data latih sebanyak 20, 40 dan 60 data citra baik dan data citra buruk.

Dari pengujian tersebut diperoleh arah GLCM yang terbaik untuk sistem pemanfaatan *gray level co-occurrence matrix* (GLCM) citra buah jeruk untuk klasifikasi mutu. Terdapat tabel hasil uji coba terhadap jumlah data latih secara lengkap yang dapat dilihat pada Lampiran A hingga Lampiran R dan tabel ringkasnya dapat dilihat pada pada Tabel 6.1 sampai dengan Tabel 6.18 serta tabel kesimpulan dapat dilihat pada Tabel 6.19.

6.2 Hasil Pengujian

Pada sub-bab ini berisi hasil pengujian berdasarkan skenario yang telah dirancang pada Bab 4. Hasil pengujian tersebut dituliskan dalam bentuk tabel yang dapat dilihat pada Tabel 6.1 sampai dengan Tabel 6.19.

Tabel 6.1 Hasil Pengujian 10 Data Latih dengan d=1 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.896	Salah	0.896	Salah	0.149	Salah	0.298	Salah
3	1.772	Salah	2.151	Salah	1.645	Salah	1.392	Salah
4	0.473	Benar	0.473	Benar	0.315	Benar	0.157	Benar
5	1.278	Benar	1.278	Benar	0.767	Benar	0.639	Benar
...
40	1.365	Benar	1.240	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	70		80		75		75	
Total Salah (%)	30		20		25		25	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.1 terdapat hasil pengujian dengan parameter 10 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem



mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.1 hasil akurasi terbaik pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 45° dan 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Dan akurasi terkecil pada arah 0° yaitu 70% dengan nilai *error* 30%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran A.

Tabel 6.2 Hasil Pengujian 10 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0.118	Salah	0	Benar	0	Benar
2	0.747	Salah	0.896	Salah	0.298	Salah	0.298	Salah
3	1.518	Salah	2.151	Salah	1.518	Salah	1.392	Salah
4	0.473	Benar	0.473	Benar	0.315	Benar	0.157	Benar
5	1.150	Benar	1.278	Benar	0.767	Benar	0.639	Benar
...
40	1.116	Benar	1.116	Benar	1.116	Benar	0.868	Benar
Total Benar (%)	75		80		77.5		72.5	
Total Salah (%)	25		20		22.5		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.2 terdapat hasil pengujian dengan parameter 10 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.2 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0°, 90° dan 135° masing-masing didapatkan akurasi sebesar 75%, 77.5% dan 72.5% dengan nilai



error sebesar 25%, 22.5% dan 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran B.

Tabel 6.3 Hasil Pengujian 10 Data Latih dengan $d=3$ pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.597	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.518	Salah	0.632	Salah	1.518	Salah	1.392	Salah
4	0.315	Benar	0.157	Benar	0.157	Benar	0.157	Benar
5	0.894	Benar	0.255	Benar	0.639	Benar	0.639	Benar
...
40	0.868	Benar	0.744	Benar	0.992	Benar	0.744	Benar
Total Benar (%)	72.5		75		75		72.5	
Total Salah (%)	22.5		25		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.3 terdapat hasil pengujian dengan parameter 10 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.3 hasil akurasi terbaik didapatkan pada arah 45° dan 90° sebesar 75% dengan nilai *error* sebesar 25%. Sedangkan pada arah 0° dan 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran C.

Tabel 6.4 Hasil Pengujian 20 Data Latih dengan d=1 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.149	Salah	0.149	Salah	0.149	Salah
3	0.759	Salah	1.012	Salah	1.645	Salah	0.253	Salah
4	0.315	Benar	0.157	Benar	0.315	Benar	0.157	Benar
5	0.767	Benar	0.084	Benar	0.767	Benar	0.127	Benar
...
40	1.116	Benar	0.992	Benar	1.365	Benar	0.744	Benar
Total Benar (%)	72.5		75		75		72.5	
Total Salah (%)	27.5		25		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.4 terdapat hasil pengujian dengan parameter 20 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.4 hasil akurasi terbaik pada arah 0° dan 135° sebesar 72.5% dengan nilai *error* sebesar 27.5%. Sedangkan pada arah 45° dan 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran D.

Tabel 6.5 Hasil Pengujian 20 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK



1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.747	Salah	0.298	Salah	0.298	Salah
3	1.012	Salah	2.151	Salah	1.518	Salah	1.392	Salah
4	0.157	Benar	0.473	Benar	0.315	Benar	0.157	Benar
5	0.767	Benar	1.278	Benar	0.767	Benar	0.639	Benar
...
40	0.868	Benar	1.116	Benar	1.116	Benar	0.868	Benar
Total Benar (%)	75		82.5		77.5		72.5	
Total Salah (%)	25		17.5		22.5		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.5 terdapat hasil pengujian dengan parameter 20 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.5 hasil akurasi terbaik didapatkan pada arah 45° sebesar 82.5% dengan nilai *error* sebesar 17.5%. Sedangkan pada arah 0°, 90° dan 135° masing-masing didapatkan akurasi sebesar 75%, 77.5% dan 72.5% dengan nilai *error* sebesar 25%, 22.5% dan 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran E.

Tabel 6.6 Hasil Pengujian 20 Data Latih dengan d=3 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.597	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.518	Salah	1.772	Salah	1.518	Salah	1.392	Salah
4	0.315	Benar	0.315	Benar	0.157	Benar	0.157	Benar



5	0.894	Benar	0.894	Benar	0.639	Benar	0.639	Benar
...
40	0.868	Benar	0.868	Benar	0.992	Benar	0.744	Benar
Total Benar (%)	72.5		80		75		72.5	
Total Salah (%)	22.5		20		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.6 terdapat hasil pengujian dengan parameter 20 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan "Benar", sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan "Salah". Dari Tabel 6.6 didapatkan hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° dan 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Dan pada arah 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran F.

Tabel 6.7 Hasil Pengujian 30 Data Latih dengan d=1 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.149	Salah	0.298	Salah	0.298	Salah
3	0.886	Salah	1.139	Salah	1.898	Salah	1.392	Salah
4	0.315	Benar	0.157	Benar	0.315	Benar	0.315	Benar
5	1.022	Benar	0.894	Benar	1.150	Benar	0.639	Benar
...
40	1.116	Benar	0.992	Benar	1.365	Benar	0.992	Benar



Total Benar (%)	72.5	72.5	72.5	75
Total Salah (%)	27.5	27.5	27.5	25

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.7 terdapat hasil pengujian dengan parameter 30 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.7 hasil akurasi terbaik didapatkan pada arah 135° sebesar 75% dengan nilai *error* sebesar 25%. Sedangkan pada arah 0°, 45° dan 90° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran G.

Tabel 6.8 Hasil Pengujian 30 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.298	Salah	0.747	Salah	0.298	Salah	0.298	Salah
3	1.265	Salah	2.151	Salah	1.898	Salah	1.518	Salah
4	0.315	Benar	0.473	Benar	0.473	Benar	0.157	Benar
5	1.022	Benar	1.278	Benar	1.278	Benar	0.639	Benar
...
40	1.116	Benar	1.116	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	72.5		82.5		80		72.5	
Total Salah (%)	27.5		17.5		20		27.5	



Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.8 terdapat hasil pengujian dengan parameter 30 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.8 hasil akurasi terbaik didapatkan pada arah 45° sebesar 82.5% dengan nilai *error* sebesar 17.5%. Dan pada arah 90° didapatkan akurasi sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° dan 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran H.

Tabel 6.9 Hasil Pengujian 30 Data Latih dengan d=3 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0.118	Salah	0	Benar	0	Benar	0	Benar
2	0.896	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.645	Salah	2.025	Salah	2.025	Salah	1.772	Salah
4	0.631	Benar	0.473	Benar	0.473	Benar	0.631	Benar
5	1.022	Benar	1.022	Benar	1.150	Benar	0.894	Benar
...
40	0.992	Benar	0.992	Benar	1.365	Benar	0.868	Benar
Total Benar (%)	67.5		80		77.5		77.5	
Total Salah (%)	32.5		20		22.5		22.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.9 terdapat hasil pengujian dengan parameter 30 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada



masing-masing arah GLCM, yaitu 0° , 45° , 90° dan 135° . Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.9 hasil akurasi terbaik didapatkan pada arah 45° sebesar 82.5% dengan nilai *error* sebesar 17.5%. Dan pada arah 90° didapatkan akurasi sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° didapatkan akurasi sebesar 67.5% dengan nilai *error* sebesar 32.5% dan pada arah 135° didapatkan akurasi sebesar 77.5% dengan nilai *error* sebesar 22.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran I.

Tabel 6.10 Hasil Pengujian 40 Data Latih dengan $d=1$ pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.149	Salah	0.149	Salah	0.298	Salah
3	0.886	Salah	1.139	Salah	1.645	Salah	1.392	Salah
4	0.315	Benar	0.157	Benar	0.315	Benar	0.315	Benar
5	1.022	Benar	0.894	Benar	0.767	Benar	0.639	Benar
...
40	1.116	Benar	0.992	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	72.5		72.5		72.5		75	
Total Salah (%)	27.5		27.5		27.5		25	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.10 terdapat hasil pengujian dengan parameter 40 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0° , 45° , 90° dan 135° . Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka

dituliskan “Salah”. Dari Tabel 6.10 hasil akurasi terbaik didapatkan pada arah 135° sebesar 75% dengan nilai *error* sebesar 25%. Sedangkan pada arah 0°, 45° dan 90° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran J.

Tabel 6.11 Hasil Pengujian 40 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.298	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.265	Salah	1.898	Salah	1.898	Salah	1.518	Salah
4	0.315	Benar	0.315	Benar	0.315	Benar	0.157	Benar
5	0.894	Benar	1.022	Benar	1.022	Benar	0.639	Benar
...
40	1.116	Benar	0.992	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	72.5		80		75		72.5	
Total Salah (%)	27.5		20		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.11 terdapat hasil pengujian dengan parameter 40 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.11 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° dan 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Dan pada arah 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran K.



Tabel 6.12 Hasil Pengujian 40 Data Latih dengan d=3 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.747	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.645	Salah	1.772	Salah	1.898	Salah	1.772	Salah
4	0.631	Benar	0.315	Benar	0.473	Benar	0.631	Benar
5	0.894	Benar	0.894	Benar	1.022	Benar	0.894	Benar
...
40	0.992	Benar	0.992	Benar	1.116	Benar	0.868	Benar
Total Benar (%)	72.5		80		77.5		77.5	
Total Salah (%)	27.5		20		22.5		22.5	

Keterangan:

- PB : Persentase Buruk
- HK : Hasil Klasifikasi

Pada Tabel 6.12 terdapat hasil pengujian dengan parameter 40 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.12 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 90° dan 135° didapatkan akurasi sebesar 77.5% dengan nilai *error* sebesar 22.5%. Dan pada arah 0° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran L.

Tabel 6.13 Hasil Pengujian 50 Data Latih dengan d=1 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK

1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.149	Salah	0.149	Salah	0.298	Salah
3	1.139	Salah	1.139	Salah	1.645	Salah	1.392	Salah
4	0.315	Benar	0.157	Benar	0.315	Benar	0.315	Benar
5	1.022	Benar	0.894	Benar	0.767	Benar	0.639	Benar
...
40	1.24	Benar	0.992	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	72.5		75		72.5		75	
Total Salah (%)	27.5		25		27.5		25	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.13 terdapat hasil pengujian dengan parameter 50 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.13 hasil akurasi terbaik didapatkan pada arah 45° dan 135° sebesar 75% dengan nilai *error* sebesar 25%. Sedangkan pada arah 0°, 45° dan 90° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran M.

Tabel 6.14 Hasil Pengujian 50 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.448	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.392	Salah	1.898	Salah	1.898	Salah	1.518	Salah
4	0.473	Benar	0.315	Benar	0.315	Benar	0.157	Benar

5	1.022	Benar	1.022	Benar	1.022	Benar	0.639	Benar
...
40	1.116	Benar	0.992	Benar	1.36	Benar	0.992	Benar
Total Benar (%)	75		80		75		72.5	
Total Salah (%)	25		20		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.14 terdapat hasil pengujian dengan parameter 50 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan "Benar", sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan "Salah". Dari Tabel 6.14 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° dan 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Dan pada arah 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran N.

Tabel 6.15 Hasil Pengujian 50 Data Latih dengan d=3 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.747	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.645	Salah	1.772	Salah	1.898	Salah	1.772	Salah
4	0.631	Benar	0.315	Benar	0.473	Benar	0.631	Benar
5	0.894	Benar	0.894	Benar	1.022	Benar	0.894	Benar
...
40	0.992	Benar	0.992	Benar	1.116	Benar	0.868	Benar



Total Benar (%)	72.5	80	77.5	77.5
Total Salah (%)	27.5	20	22.5	22.5

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.15 terdapat hasil pengujian dengan parameter 50 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.15 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 90° dan 135° didapatkan akurasi sebesar 77.5% dengan nilai *error* sebesar 22.5%. Dan pada arah 0° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran O.

Tabel 6.16 Hasil Pengujian 60 Data Latih dengan d=1 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.149	Salah	0.149	Salah	0.149	Salah	0.298	Salah
3	1.265	Salah	1.012	Salah	1.645	Salah	1.392	Salah
4	0.315	Benar	0.157	Benar	0.315	Benar	0.315	Benar
5	1.022	Benar	0.894	Benar	0.767	Benar	0.639	Benar
...
40	1.24	Benar	0.992	Benar	1.365	Benar	0.992	Benar
Total Benar (%)	72.5		72.5		72.5		75	



Total Salah (%)	27.5	27.5	27.5	25
-----------------	------	------	------	----

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.16 terdapat hasil pengujian dengan parameter 60 data latih dan nilai *distance* 1 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.16 hasil akurasi terbaik didapatkan pada arah 135° sebesar 75% dengan nilai *error* sebesar 25%. Sedangkan pada arah 0°, 45° dan 90° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran P.

Tabel 6.17 Hasil Pengujian 60 Data Latih dengan d=2 pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.448	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.392	Salah	1.898	Salah	1.898	Salah	1.518	Salah
4	0.473	Benar	0.315	Benar	0.315	Benar	0.157	Benar
5	1.022	Benar	1.022	Benar	1.022	Benar	0.639	Benar
...
40	1.116	Benar	0.992	Benar	1.36	Benar	0.992	Benar
Total Benar (%)	75		80		75		72.5	
Total Salah (%)	25		20		25		27.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi



Pada Tabel 6.17 terdapat hasil pengujian dengan parameter 60 data latih dan nilai *distance* 2 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.17 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 0° dan 90° didapatkan akurasi sebesar 75% dengan nilai *error* sebesar 25%. Dan pada arah 135° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran Q.

Tabel 6.18 Hasil Pengujian 60 Data Latih dengan $d=3$ pada 4 Arah GLCM

Data Uji	Arah GLCM							
	0°		45°		90°		135°	
	PB	HK	PB	HK	PB	HK	PB	HK
1	0	Benar	0	Benar	0	Benar	0	Benar
2	0.747	Salah	0.448	Salah	0.298	Salah	0.298	Salah
3	1.645	Salah	1.772	Salah	1.898	Salah	1.772	Salah
4	0.631	Benar	0.315	Benar	0.473	Benar	0.631	Benar
5	0.894	Benar	0.894	Benar	1.022	Benar	0.894	Benar
...
40	0.992	Benar	0.992	Benar	1.116	Benar	0.868	Benar
Total Benar (%)	72.5		80		77.5		77.5	
Total Salah (%)	27.5		20		22.5		22.5	

Keterangan:

- PB : Persentase Buruk
- HK: Hasil Klasifikasi

Pada Tabel 6.18 terdapat hasil pengujian dengan parameter 60 data latih dan nilai *distance* 3 pada 4 arah GLCM. Pada tabel tersebut terdapat nilai persentase buruk atau persentase cacat buah pada kolom PB dari tiap-tiap data uji pada masing-masing arah GLCM, yaitu 0°, 45°, 90° dan 135°. Dari nilai persentase buruk yang didapat tersebut kemudian dicocokkan dengan hasil pengamatan pakar pada data uji, apabila sistem dapat mengklasifikasikan mutu jeruk sesuai dengan hasil

pengamatan pakar maka dituliskan “Benar”, sebaliknya apabila sistem mengklasifikasikan mutu jeruk tidak sesuai dengan hasil pengamatan pakar maka dituliskan “Salah”. Dari Tabel 6.18 hasil akurasi terbaik didapatkan pada arah 45° sebesar 80% dengan nilai *error* sebesar 20%. Sedangkan pada arah 90° dan 135° didapatkan akurasi sebesar 77.5% dengan nilai *error* sebesar 22.5%. Dan pada arah 0° didapatkan akurasi sebesar 72.5% dengan nilai *error* sebesar 27.5%. Untuk hasil pengujian lengkap terhadap 40 data uji dapat dilihat pada Lampiran R.

Tabel 6.19 Rangkuman Pengujian Data Latih, *Distance* dan Arah GLCM

Banyak Data Latih	<i>d</i>	Hasil Klasifikasi (%)							
		Arah GLCM							
		0°		45°		90°		135°	
		B	S	B	S	B	S	B	S
10	<i>d</i> =1	70	30	80	20	75	25	75	25
	<i>d</i> =2	75	25	80	20	77.5	22.5	72.5	27.5
	<i>d</i> =3	72.5	27.5	75	25	75	25	72.5	27.5
20	<i>d</i> =1	72.5	27.5	75	25	75	25	72.5	27.5
	<i>d</i> =2	75	25	82.5	17.5	77.5	22.5	72.5	27.5
	<i>d</i> =3	72.5	27.5	80	20	75	25	72.5	27.5
30	<i>d</i> =1	72.5	27.5	72.5	27.5	72.5	27.5	75	25
	<i>d</i> =2	72.5	27.5	82.5	17.5	80	20	72.5	27.5
	<i>d</i> =3	67.5	32.5	80	20	77.5	22.5	77.5	22.5
40	<i>d</i> =1	72.5	27.5	72.5	27.5	72.5	27.5	75	25
	<i>d</i> =2	72.5	27.5	80	20	75	25	72.5	27.5
	<i>d</i> =3	72.5	27.5	80	20	77.5	22.5	77.5	22.5
50	<i>d</i> =1	72.5	27.5	75	25	72.5	27.5	75	25
	<i>d</i> =2	75	25	80	20	75	25	72.5	27.5
	<i>d</i> =3	72.5	27.5	80	20	77.5	22.5	77.5	22.5
60	<i>d</i> =1	72.5	27.5	72.5	27.5	72.5	27.5	75	25
	<i>d</i> =2	75	25	80	20	75	25	72.5	27.5
	<i>d</i> =3	72.5	27.5	80	20	77.5	22.5	77.5	22.5

Keterangan:

- B : Benar
- S : Salah

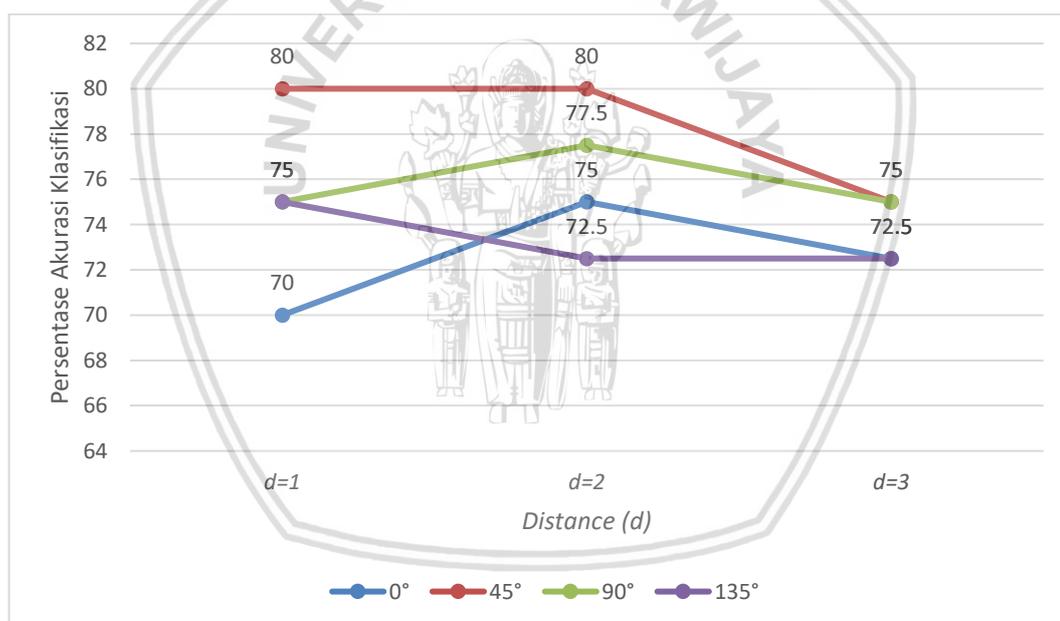


Pada Tabel 6.19 terdapat rangkuman hasil pengujian data latih, distance, dan arah GLCM. Pada tabel tersebut terdapat nilai akurasi klasifikasi dan nilai *error* dari tiap-tiap parameter pengujian. Nilai akurasi klasifikasi terdapat pada kolom B dan nilai *error* terdapat pada kolom S.

6.3 Analisis Pengujian

Pada sub-bab ini dijelaskan analisis mengenai hasil pengujian yang telah dilakukan. Analisis berdasarkan pengujian yang dilakukan dengan jumlah data latih sebanyak 20, 40 dan 60 data pada 3 *distance* yaitu, $d=1$, $d=2$ dan $d=3$ dan 4 arah GLCM yaitu, 0° , 45° , 90° dan 135° . Didapatkan hasil akurasi klasifikasi. Kemudian hasil akurasi klasifikasi dari tiap parameter tersebut dibandingkan.

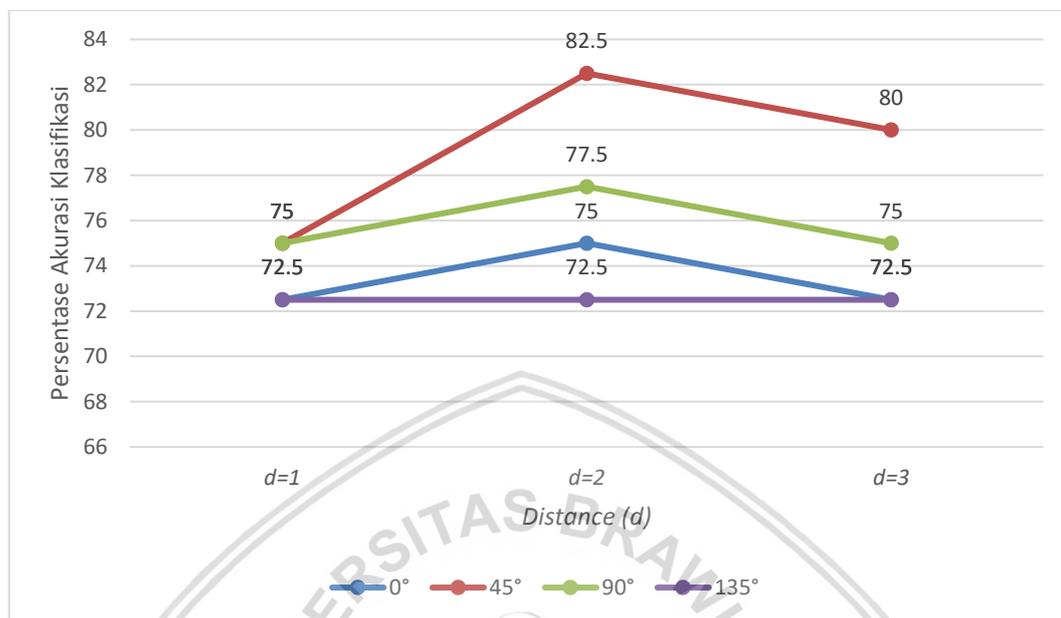
Dari hasil pengujian yang dilakukan maka dibuat grafik untuk melihat bagaimana kecenderungan nilai hasil klasifikasi berdasarkan jumlah data latih tertentu, *distance* tertentu dan arah GLCM tertentu. Karena ketiga parameter pengujian saling berkaitan maka berdasarkan Tabel 6.19 dibuat grafik dengan memisahkan data latih 10, 20, 30, 40, 50 dan 60 dengan 3 nilai *distance* dan 4 arah GLCM yang dapat dilihat pada Gambar 6.1 hingga Gambar 6.6.



Gambar 6.1 Grafik Hasil Pengujian Data Latih 10 Pada 3 *Distance* dan 4 Arah GLCM

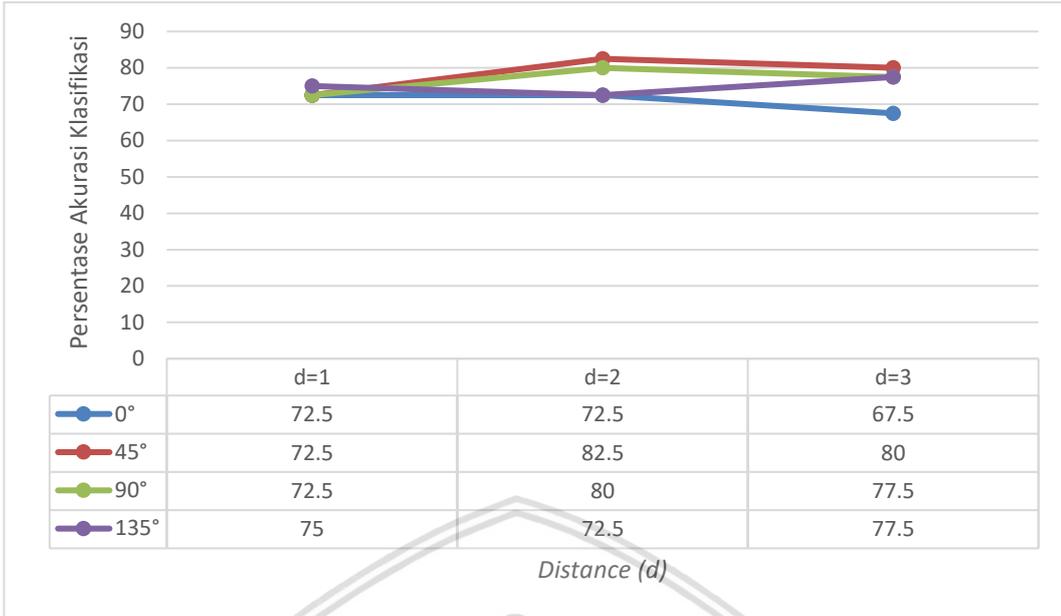
Berdasarkan Gambar 6.1 dapat dilihat bahwa nilai akurasi cenderung lebih tinggi pada $d=2$ di semua arah, sedangkan nilai pada $d=3$ nilai akurasi cenderung turun atau rendah. Hal tersebut dikarenakan ketika $d=2$ matriks awal dapat merepresentasikan nilai yang tepat sehingga dapat menghasilkan ekstraksi ciri yang baik, sebaliknya ketika $d=1$ dan $d=3$ matriks awal tidak dapat merepresentasikan nilai yang tepat sehingga hasil ekstraksi ciri kurang baik. Kemudian pada arah 0° dan 90° nilai akurasi cenderung sama, dikarenakan pada arah tersebut pasangan piksel yang dihasilkan kurang menggambarkan perbedaan

nilai dari tekstur citra sehingga matriks GLCM dan ekstraksi ciri yang dihasilkan kurang baik. Hasil akurasi tertinggi sebesar 80% didapatkan pada arah 45° dengan $d=1$ dan $d=2$ dan jumlah data latih 10.



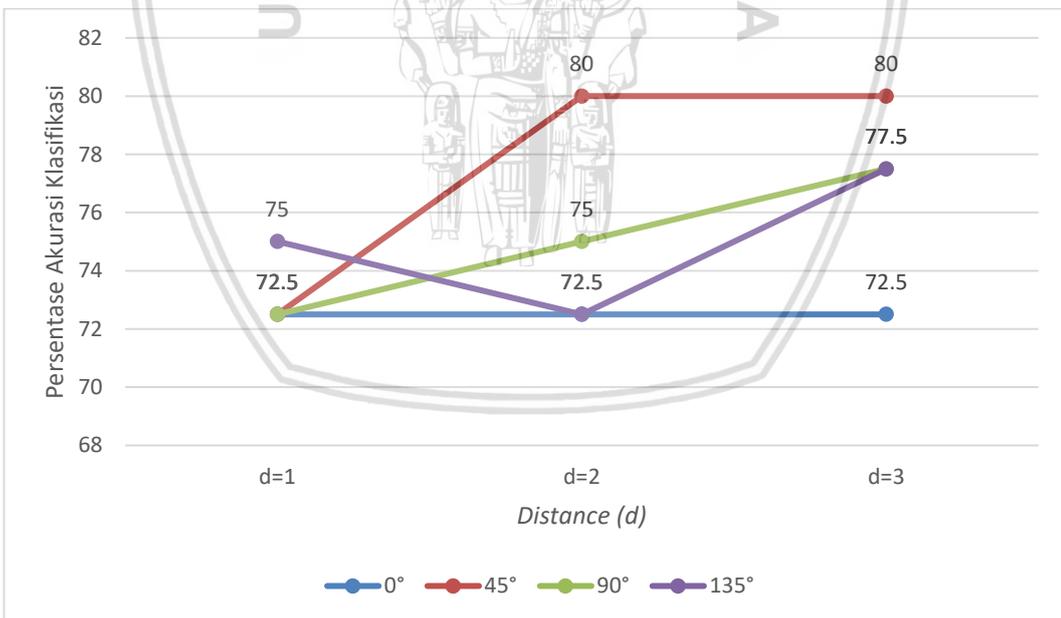
Gambar 6.2 Grafik Hasil Pengujian Data Latih 20 Pada 3 Distance dan 4 Arah GLCM

Berdasarkan Gambar 6.2 dapat dilihat bahwa nilai akurasi cenderung lebih tinggi pada nilai $d=2$ di semua arah, sedangkan pada $d=1$ dan $d=3$ nilai akurasinya cenderung rendah. Hal tersebut dikarenakan ketika $d=2$ matriks awal dapat merepresentasikan nilai yang tepat sehingga dapat menghasilkan ekstraksi ciri yang baik, sebaliknya ketika $d=1$ dan $d=3$ matriks awal tidak dapat merepresentasikan nilai yang tepat sehingga hasil ekstraksi ciri kurang baik. Kemudian pada arah 0° dan 90° nilai akurasi sama dan cenderung rendah, dikarenakan pada arah tersebut kurang menggambarkan perbedaan nilai pasangan piksel dari tekstur citra sehingga matriks GLCM dan ekstraksi ciri yang dihasilkan kurang baik. Selanjutnya dapat dilihat bahwa pada arah 135° baik dengan nilai $d=1$, $d=2$, maupun $d=3$ hasil akurasi sama, hal ini dikarenakan nilai matriks awal ketiganya cenderung sama. Hasil akurasi tertinggi sebesar 82.5% didapatkan pada arah 45° dengan $d=2$ dan jumlah data latih 20.



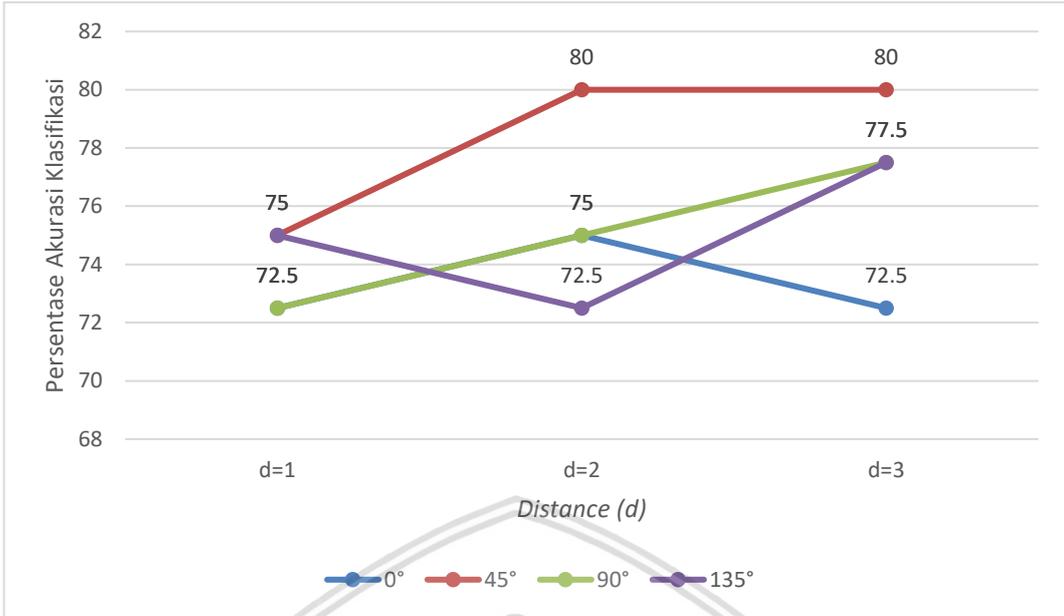
Gambar 6.3 Grafik Hasil Pengujian Data Latih 30 Pada 3 Distance dan 4 Arah GLCM

Berdasarkan Gambar 6.3 dapat dilihat bahwa selisih nilai akurasi antar arah pada setiap nilai *distance* cenderung dekat, dan juga didapatkan akurasi tertinggi sebesar 82.5% pada arah GLCM 45° dengan nilai *d*=2 dan jumlah data latih 30.

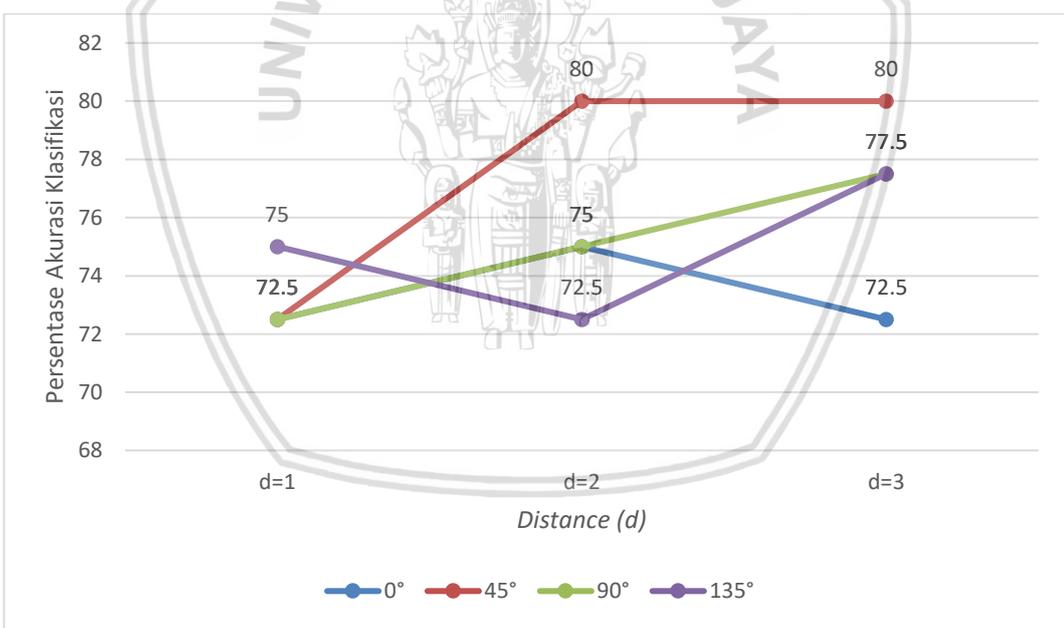


Gambar 6.4 Grafik Hasil Pengujian Data Latih 40 Pada 3 Distance dan 4 Arah GLCM





Gambar 6.5 Grafik Hasil Pengujian Data Latih 50 Pada 3 Distance dan 4 Arah GLCM



Gambar 6.6 Grafik Hasil Pengujian Data Latih 60 Pada 3 Distance dan 4 Arah GLCM

Berdasarkan Gambar 6.4, Gambar 6.5 dan Gambar 6.6 dapat dilihat bahwa nilai akurasi pada 4 arah GLCM, 3 nilai distance dengan jumlah data latih 40, 50 dan 60 relatif sama. Hal ini dikarenakan bahwa semakin banyak jumlah data latih, maka semakin kecil perbedaan atau variasi nilai antar pikselnya dan juga terdapat kemiripan pada nilai tekstur. Pada arah 0° nilai akurasi cenderung kecil, dikarenakan kurang menggambarkan perbedaan nilai pasangan piksel dari tekstur citra sehingga matriks GLCM dan ekstraksi ciri yang dihasilkan kurang baik.



Kemudian pada arah 45° dan 90° hasil klasifikasi cenderung meningkat pada setiap nilai *distance*, dikarenakan peningkatan nilai *distance* mempengaruhi matriks awal yang dihasilkan. Lalu pada arah 135° hasil klasifikasi cenderung tinggi ketika $d=1$ dan $d=3$ dibandingkan ketika $d=2$, hal ini dikarenakan matriks awal dengan $d=1$ dan $d=3$ lebih baik nilainya daripada $d=2$ sehingga berpengaruh pada ekstraksi cirinya. Hasil akurasi tertinggi sebesar 80% didapatkan pada arah 45° pada $d=2$ dan $d=3$ dengan jumlah data latih 40, 50 dan 60.

Dari pengujian diatas didapatkan bahwa arah GLCM 45° menghasilkan nilai akurasi terbaik, dikarenakan arah 45° dapat merepresentasikan tekstur citra dengan baik dan menghasilkan matriks GLCM yang baik. Dapat disimpulkan pula bahwa jarak ketetanggaan piksel (*distance*) mempengaruhi pembentukan matriks GLCM sehingga juga mempengaruhi hasil akurasi pada pemanfaatan ekstraksi ciri GLCM untuk klasifikasi mutu jeruk ini. Kemudian semakin banyak jumlah data latih, maka semakin kecil perbedaan variasi antar data pada satu kelas dan terdapat kemiripan tekstur, sehingga hasil akurasi klasifikasi cenderung sama atau bahkan lebih rendah.

Karakteristik dari GLCM adalah mengetahui perbedaan nilai pada suatu piksel dengan piksel lainnya pada citra. Apabila nilai antar piksel tidak homogen (nilai homogenitas kecil) maka nilai kontrasnya besar, begitupun sebaliknya apabila homogen (nilai homogenitas besar) maka nilai kontrasnya kecil. Pada penelitian ini didapatkan bahwa pada arah diagonal ($45^\circ/135^\circ$) nilai homogenitas kecil sehingga variasi intensitas dalam citra (kontras) tinggi. Ciri yang dihasilkan GLCM merepresentasikan nilai tekstur pada citra buah jeruk. Sehingga nilai-nilai tersebut dapat dimanfaatkan untuk mengklasifikasikan mutu jeruk.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Ekstraksi ciri *gray level co-occurrence matrix* (GLCM) citra buah jeruk keprok dapat dimanfaatkan untuk klasifikasi mutu, karakteristik dari GLCM adalah mengetahui perbedaan nilai pada suatu piksel dengan piksel lainnya pada citra. Apabila nilai antar piksel tidak homogen (nilai homogenitas kecil) maka nilai kontrasnya besar, begitupun sebaliknya apabila homogen (nilai homogenitas besar) maka nilai kontrasnya kecil. Pada penelitian ini didapatkan bahwa pada arah diagonal ($45^\circ/135^\circ$) nilai homogenitas kecil sehingga variasi intensitas dalam citra (kontras) tinggi. Ciri yang dihasilkan GLCM merepresentasikan nilai tekstur pada citra buah jeruk. Sehingga nilai-nilai tersebut dapat dimanfaatkan untuk mengklasifikasikan mutu jeruk.
2. Pemanfaatan hasil ekstraksi ciri *gray level co-occurrence matrix* (GLCM) citra jeruk keprok untuk klasifikasi mutu dapat memberikan hasil terbaik sebesar 82.5% dengan jumlah data latih sebesar 20 dan 30 data, pada jarak ketetanggaan piksel (*distance*) sebesar 2 dan pada arah GLCM 45° . Hal ini menunjukkan bahwa sistem yang telah dibuat mampu mengklasifikasikan mutu dengan baik.

7.2 Saran

Penelitian ini masih belum sempurna dan memiliki beberapa kekurangan. Adapun saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Perlakuan pengambilan data harus lebih diperhatikan terutama faktor pencahayaan agar warna yang dihasilkan seperti aslinya.
2. Perlu dilakukan penambahan metode untuk menghasilkan fitur warna agar didapatkan fitur yang lebih representatif, dapat menggunakan metode *color moments*, RGB to HSV, maupun metode lainnya.
3. Algoritme diimplementasikan pada penelitian lanjutan untuk menuju sistem klasifikasi mutu secara otomatis, seperti diaplikasikan pada *mobile* untuk proses klasifikasi mutu dengan gadget ataupun dalam bentuk penciptaan alat khusus klasifikasi mutu buah jeruk otomatis.

DAFTAR PUSTAKA

- Agustian, W., Setyaningsih, S. & Qur'ania, A., 2017. Klasifikasi Buah Jeruk Menggunakan Metode Naive Bayes Berdasarkan Analisis Tekstur dan Normalisasi Warna. *Program Studi Ilmu Komputer FMIPA Universitas Pakuan*.
- Astuti, S., 2015. *Tabloid Sinar Tani*. [Online] Available at: <http://www.tabloidsinartani.com> [Accessed 1 Maret 2018].
- Dewi, R. K. & Ginardi, R. V. H., 2014. Identifikasi Penyakit Pada Daun Tebu Dengan Gray Level Co-occurrence Matrix dan Color Moments. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 1(2), pp. 70-77.
- Endarto, O. & Martini, E., 2016. *Pedoman Budi Daya Jeruk Sehat*. Bogor: World Agroforestry Centre (ICRAF) Southeast Asia Regional Program.
- Kadir, A. & Susanto, A., 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit Andi.
- Kementrian Pertanian RI, 2017. *Kementrian Pertanian RI*. [Online] Available at: <http://www.pertanian.go.id/Indikator/tabel-2-prod-lspn-prodvitas-horti.pdf> [Accessed 6 Maret 2018].
- Neneng, Adi, K. & Isnanto, R. R., 2016. Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM). *Jurnal Sistem Informasi Bisnis*, 6(1), pp. 1-10.
- Nugroho, A. S., Witarto, A. B. & Handoko, D., 2003. *Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika*. [Online] Available at: <http://ilmukomputer.com> [Accessed 3 Maret 2018].
- Prasetyo, E., 2011. *Pengolahan Citra Digital Dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Prasetyo, E., 2014. *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Putra, D., 2010. *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- Raheja, J. L., Kumar, S. & Chaudhary, A., 2013. Fabric defect detection based on GLCM and Gabor filter: A comparison. *Elsevier*, pp. 6469-6474.
- Resita, R., Juratminingsih & Gasim, 2016. Identifikasi Jenis Buah Jeruk Menggunakan Metode Jaringan Syaraf Tiruan Berdasarkan Tekstur Kulit. *STMIK GI MDP*.

- Saifudin & Fadlil, A., 2015. Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Cooccurrence Matrix (GLCM) Dengan Klasifikasi Jarak Euclidean. Volume 19.
- Sembiring, K., 2007. Tutorial SVM Bahasa Indonesia. *Teknik Informatika ITB*.
- Siqueira, F. R. d., Schwartz, W. R. & Pedrini, H., 2013. Multi-scale Gray Level Co-occurrence Matrices for Texture Description. *Neurocomputing*, Volume 120, pp. 336-345.
- Standar Nasional Indonesia, 2009. *SNI 3165:2009*, Jakarta: Badan Standardisasi Nasional (BSN).
- Surya, R. A., Fadhil, A. & Yudhana, A., 2017. Ekstraksi Ciri Metode Gray Level Co-Occurrence Matrix (GLCM) dan Filter Gabor Untuk Klasifikasi Citra Batik Pekalongan. *Jurnal Informatika: Jurnal Pengembangan IT*, Volume 02.
- Sutoyo, T. et al., 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- Taufik, Y., 2015. *Statistik Produksi Holtikultura Tahun 2014*. Jakarta: Direktorat Jenderal Hortikultura Kementerian Pertanian.
- Wiharja, Y. P. & Harjoko, A., 2014. Pemrosesan Citra Digital untuk Klasifikasi Mutu Buah Pisang Menggunakan Jaringan Syaraf Tiruan. *IJEIS*, Volume 4, pp. 57-68.

