

**PENERAPAN METODE FUZZY K-NN
PADA KLASIFIKASI JENIS MANGGA
BERDASARKAN TEKSTUR DAUN**

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer



Disusun Oleh :

ACHMAD ZAKI MUBAROK

NIM. 0810960025

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013**

LEMBAR PERSETUJUAN

**PENERAPAN METODE FUZZY K-NN
PADA KLASIFIKASI JENIS MANGGA
BERDASARKAN TEKSTUR DAUN**

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer



Disusun oleh :

ACHMAD ZAKI MUBAROK

NIM. 0810960025

Telah diperiksa dan disetujui oleh:

Pembimbing I,

Pembimbing II,

Drs. Achmad Ridok, M.Kom
NIP. 196808251994031002

Yusi Tyroni Mursityo, S.Kom., MS
NIP. 198002282006041001

LEMBAR PENGESAHAN

**PENERAPAN METODE FUZZY K-NN
PADA KLASIFIKASI JENIS MANGGA
BERDASARKAN TEKSTUR DAUN**

**SKRIPSI
LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI**

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Disusun Oleh:

ACHMAD ZAKI MUBAROK

NIM.0810960025

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 8 April 2013

Penguji I

Penguji II

Dian Eka Ratnawati, S.Si., M.Kom
NIP. 197306192002122001

Budi Dharma Setiawan, S.Kom., M. Cs.
NIK. 84101506110090

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom
NIK. 82062316110425

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.
NIP. 196708011992031001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan 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, 8 April 2013

Mahasiswa,

Achmad Zaki Mubarak
NIM. 0810960025

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena atas limpahan rahmat, dan hidayah-Nya, sehingga penulis bisa menyelesaikan skripsi ini dengan sebaik-baiknya dengan judul: **“Penerapan metode *Fuzzy* K-NN pada klasifikasi jenis mangga berdasarkan tekstur daun”**.

Skripsi ini diajukan sebagai syarat ujian seminar skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya Malang. Penulis menyadari sepenuhnya bahwa dalam menyelesaikan skripsi ini, penulis mendapat banyak bantuan dari semua pihak, untuk itu dalam kesempatan ini penulis mengucapkan terima kasih kepada :

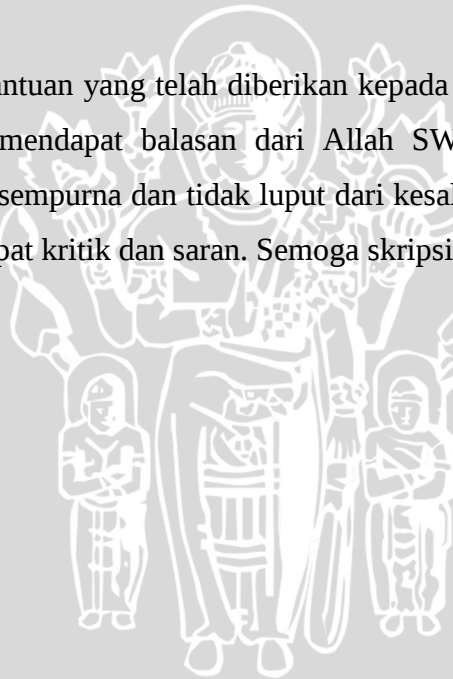
1. Bapak Drs. Achmad Ridok, M.Kom selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, dan motivasi hingga skripsi ini dapat terselesaikan.
2. Bapak Yusi Tyroni Mursityo, S.Kom., MS selaku Dosen Pembimbing II yang telah memberikan pengetahuan, bimbingan dan arahan untuk kesempurnaan penulisan skripsi ini.
3. Drs. Marji, MT. selaku Ketua Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Djoko Pramono, ST selaku Dosen Penasehat Akademik.
5. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Seluruh dosen pengajar dan karyawan Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.
7. Orang Tua penulis, serta keluarga yang selalu memberikan dukungan, nasihat dan doa.
8. Teman-teman Ilmu Komputer angkatan 2008 tercinta yang telah banyak memberikan bantuan dan pengalaman selama menjadi mahasiswa di Universitas Brawijaya.

9. Teman-teman KKN Desa Semen 2012 tercinta yang telah banyak memberikan momen-momen tak terlupakan.
10. Keluarga besar kos-kosan 292c terima kasih banyak untuk kenangannya selama 5 tahun.
11. Pak Eko, Bu Sofi, Mas Hery, Likha, Ardhy, Eko, Wp, dimas, dan semua pihak yang membantu terselesaikannya skripsi ini tetapi tidak disebutkan, penulis mohon maaf yang sebesar-besarnya.
12. Kepada siapa saja yang kenal penulis atau sebaliknya, penulis mohon maaf yang sebesar-besarnya jika pernah melakukan kesalahan yang disengaja maupun yang tidak disengaja dan semoga Kita bisa melewati segala ujian di dunia ini dengan nilai “cum laude” kelak ☺ Aamiin.

Semoga segala bantuan yang telah diberikan kepada penulis selama penulis menyusun skripsi ini mendapat balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini tidak sempurna dan tidak luput dari kesalahan, sehingga penulis menerima apabila terdapat kritik dan saran. Semoga skripsi ini bermanfaat.

Malang, 8 April 2013

Penulis



ABSTRAK

Achmad Zaki Mubarak. 2013. Penerapan Metode Fuzzy K-NN Pada Klasifikasi Jenis Mangga Berdasarkan Tekstur Daun. Skripsi Program Studi Informatika/Illmu komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Pembimbing : Drs. Achmad Ridok, M.Kom dan Yusi Tyroni Mursityo, S.Kom., MS

Bagi petani atau masyarakat awam yang berminat untuk menanam pohon mangga selalu berharap bahwa buah mangga yang dihasilkan dari pohon yang ditanamnya merupakan jenis buah mangga sesuai dengan yang diinginkan. Dalam memilih jenis pohon mangga secara manual kurang akurat sehingga perlu alat bantu berbasis komputer. Pada penelitian ini dikembangkan klasifikasi jenis mangga berdasarkan tekstur daun menggunakan *Fuzzy K-Nearest Neighbor* (FK-NN). Fitur tekstur yang digunakan dalam penelitian adalah : rata-rata intensitas, *smoothness*, *entropy*, *5 moment invariant*, *energy*, dan kontras. Hasil uji coba sistem dengan data 30 sampel daun mangga gadung dan 30 sampel daun mangga curut menunjukkan nilai rata-rata hasil akurasi keseluruhan 81.1574 %.

Kata kunci : *Fuzzy*, *K-Nearest Neighbor*, Klasifikasi, Daun mangga, Gadung, Curut.

ABSTRACT

Achmad Zaki Mubarak. 2013. *Implementation of Fuzzy K-NN Classification Methods on Mango Based Texture Leaf. Thesis of Informatic/Computer Science Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor : Drs. Achmad Ridok, M.Kom and Yusi Tyrone Mursityo, S.Kom., MS*

For farmers who are interested to plant mango trees have always hoped that the planted mangoes trees produced mango fruit types as desired. Manual selection was inaccurate, so we propose computer based system. In this research using Fuzzy K-Nearest Neighbor (FK-NN) on mango leaf texture feature dataset types of gadung and curut. Texture features in the study were: average of intensity, smoothness, entropy, 5 moment invariant, energy, and contrast. Results of testing the system using 30 samples of gadung mango leaves and 30 samples of curut mango leaves showed the average value of 81.1574% overall accuracy results.

Key words : Fuzzy, K-Nearest Neighbor, Classification, Mango Leaf, Gadung, Curut.



DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	II
LEMBAR PENGESAHAN	III
PERNYATAAN ORISINALITAS SKRIPSI.....	IV
KATA PENGANTAR.....	V
ABSTRAK	VII
ABSTRACT	VIII
DAFTAR ISI.....	IX
DAFTAR TABEL	XII
DAFTAR GAMBAR.....	XV
DAFTAR SOURCECODE.....	XVII
DAFTAR LAMPIRAN	XVIII
BAB I PENDAHULUAN.....	1
1.1. LATAR BELAKANG	1
1.2. PERUMUSAN MASALAH.....	3
1.3. TUJUAN PENELITIAN	3
1.4. BATASAN PERMASALAHAN	4
1.5. MANFAAT PENELITIAN.....	4
1.6. SISTEMATIKA PENULISAN.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. POHON MANGGA	6
2.1.1. Pengertian Pohon Mangga	6
2.1.2. Karakter Daun Mangga.....	6
2.1.3. Analisis Tekstur Citra	7
2.2. SISTEM <i>FUZZY</i>	10
2.2.1. Pengertian Logika Fuzzy.....	10
2.2.2. Himpunan <i>Crisp</i>	11
2.2.3. Himpunan <i>Fuzzy</i>	11
2.2.4. Fungsi Keanggotaan.....	12

2.3.	DATA MINING	14
2.3.1.	Pengertian Data Mining	14
2.3.2.	Klasifikasi	15
2.3.3.	K-Nearest Neighbor (KNN).....	15
2.3.3.1.	Pengertian K-Nearest Neighbor (KNN)	15
2.3.3.2.	Proses K-Nearest Neighbor (KNN).....	16
2.3.3.3.	Fungsi Kombinasi.....	17
2.3.3.4.	Algoritma K-Nearest Neighbor	18
2.3.4.	Fuzzy K-Nearest Neighbor Classifier	19
2.3.4.1.	Pengertian <i>Fuzzy K-Nearest Neighbor</i> (FK-NN)	19
2.3.4.2.	Algoritma <i>Fuzzy K-Nearest Neighbor</i> (FK-NN).....	19
2.4.	AKURASI PENGUJIAN K-FOLD CROSS VALIDATION	21
BAB III	METODOLOGI DAN PERANCANGAN	23
3.1.	STUDI LITERATUR	24
3.2.	PENGUMPULAN DATA	24
3.3.	ANALISIS DAN PERANCANGAN SISTEM	25
3.3.1.	Deskripsi Sistem	25
3.3.2.	Perancangan Proses.....	26
3.3.2.1.	Normalisasi nilai setiap atribut.....	28
3.3.2.1.1.	Hitung Max.....	29
3.3.2.1.2.	Hitung Min	31
3.3.2.2.	Klasifikasi K-Nearest Neighbor (K-NN)	32
3.3.2.2.1.	Proses Hasil Kuadrat	33
3.3.2.2.2.	Menghitung Euclidean Distance	33
3.3.2.2.3.	Proses Pembobotan.....	33
3.3.2.2.4.	Proses Pengurutan	33
3.3.2.2.5.	Proses Pembobotan sejumlah K	36
3.3.2.3.	Transformasi ke data fuzzy	38
3.3.2.3.1.	Proses Total	39
3.3.2.3.2.	Proses Hitung Derajat Keanggotaan.....	40
3.3.2.3.3.	Proses Hitung Hasil.....	40

3.4.	PERHITUNGAN MANUAL.....	43
3.5.	PERANCANGAN ANTARMUKA.....	50
3.6.	PERANCANGAN UJI COBA.....	53
BAB IV IMPLEMENTASI DAN UJI COBA SISTEM		56
4.1.	LINGKUNGAN IMPLEMENTASI.....	56
4.1.1.	Lingkungan implementasi perangkat keras.....	56
4.1.2.	Lingkungan implementasi perangkat lunak	56
4.2.	IMPLEMENTASI PROGRAM	57
4.2.1.	Implementasi konversi file *csv ke dalam database MySQL	57
4.2.2.	Implementasi normalisasi	58
4.2.3.	Implementasi klasifikasi.....	59
4.2.4.	Implementasi transformasi ke data fuzzy.....	61
4.3.	IMPLEMENTASI ANTARMUKA	62
5.1.	IMPLEMENTASI PENGUJIAN.....	67
5.1.1.	Hasil Uji	67
5.1.1.1.	4-Fold	67
5.1.1.2.	5-Fold	71
5.1.1.3.	6-Fold	76
5.1.1.4.	10-Fold	82
5.1.2.	Analisa Hasil	91
5.1.2.1.	4-Fold	91
5.1.2.2.	5-Fold	92
5.1.2.3.	6-Fold	94
5.1.2.4.	10-Fold	95
BAB VI PENUTUP		98
6.1.	KESIMPULAN	98
6.2.	SARAN.....	99
DAFTAR PUSTAKA		100
LAMPIRAN.....		103

DAFTAR TABEL

No.	Judul	Halaman
Tabel 3.1	Data input pelatihan sistem	43
Tabel 3.2	Data input uji sistem	44
Tabel 3.3	Data yang akan dilakukan klasifikasi.....	44
Tabel 3.4	Range Data Training	45
Tabel 3.5	Data Pelatihan yang telah di normalisasi	45
Tabel 3.6	Data Uji yang telah di normalisasi	46
Tabel 3.7	Nilai jarak terdekat “Euclidean distance” data uji dengan data pelatihan	47
Tabel 3.8	Nilai weighted	48
Tabel 3.9	Data setelah sorting dari yang terbesar	48
Tabel 3.10	Seleksi 4 data dengan nilai terbesar	49
Tabel 3.11	Perhitungan nilai membership neighbor	49
Tabel 3.12	Hasil	50
Tabel 3.13	Tabel Uji Pengaruh K-Fold Cross Validation Terhadap Akurasi	54
Tabel 5.1	Tabel Hasil Pengujian untuk 4-Fold cross validation bagian pertama data uji terhadap akurasi	68
Tabel 5.2	Tabel Hasil Pengujian untuk 4-Fold cross validation bagian kedua data uji terhadap akurasi	69
Tabel 5.3	Tabel Hasil Pengujian untuk 4-Fold cross validation bagian ketiga data uji terhadap akurasi	70
Tabel 5.4	Tabel Hasil Pengujian untuk 4-Fold cross validation bagian terakhir data uji terhadap akurasi	70
Tabel 5.5	Tabel Hasil Rata-rata untuk 4-Fold cross validation semua bagian data uji terhadap akurasi	71
Tabel 5.6	Tabel Hasil Pengujian untuk 5-Fold cross validation bagian pertama data uji terhadap akurasi	72
Tabel 5.7	Tabel Hasil Pengujian untuk 5-Fold cross validation bagian kedua data uji terhadap akurasi	73

Tabel 5.8 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian ketiga data uji terhadap akurasi	73
Tabel 5.9 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian keempat data uji terhadap akurasi	74
Tabel 5.10 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian terakhir data uji terhadap akurasi	75
Tabel 5.11 Tabel Hasil Rata-rata untuk 5-Fold cross validation semua bagian data uji terhadap akurasi	76
Tabel 5.12 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian pertama data uji terhadap akurasi	77
Tabel 5.13 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian kedua data uji terhadap akurasi	78
Tabel 5.14 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian ketiga data uji terhadap akurasi	79
Tabel 5.15 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian keempat data uji terhadap akurasi	79
Tabel 5.16 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian kelima data uji terhadap akurasi	80
Tabel 5.17 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian terakhir data uji terhadap akurasi	81
Tabel 5.18 Tabel Hasil Rata-rata untuk 6-Fold cross validation semua bagian data uji terhadap akurasi	82
Tabel 5.19 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian pertama data uji terhadap akurasi	83
Tabel 5.20 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kedua data uji terhadap akurasi	83
Tabel 5.21 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian ketiga data uji terhadap akurasi	84
Tabel 5.22 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian keempat data uji terhadap akurasi	85
Tabel 5.23 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kelima data uji terhadap akurasi	86

Tabel 5.24 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian keenam data uji terhadap akurasi	86
Tabel 5.25 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian ketujuh data uji terhadap akurasi	87
Tabel 5.26 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kedelapan data uji terhadap akurasi	88
Tabel 5.27 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kesembilan data uji terhadap akurasi	89
Tabel 5.28 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian terakhir data uji terhadap akurasi	89
Tabel 5.29 Tabel Hasil Rata-rata untuk 10-Fold cross validation semua bagian data uji terhadap akurasi	90



DAFTAR GAMBAR

No.	Judul	Halaman
Gambar 3.1	Diagram Alir Penelitian	23
Gambar 3.2	Daun pohon mangga gadung.....	24
Gambar 3.3	Daun pohon mangga curut	25
Gambar 3.4	Alur Proses Sistem secara umum	27
Gambar 3.5	Alur Proses Normalisasi setiap atribut	29
Gambar 3.6	Alur Proses Hitung Max.....	30
Gambar 3.7	Alur Proses Hitung Min	31
Gambar 3.8	Alur Proses Klasifikasi K-Nearest Neighbor	32
Gambar 3.9	Alur Proses Hasil Kuadrat.....	34
Gambar 3.10	Alur Proses Euclidean Distance	35
Gambar 3.11	Alur Proses Pembobotan	36
Gambar 3.12	Alur Proses Pengurutan Nilai Bobot	37
Gambar 3.13	Alur Proses Pembobotan sejumlah K.....	38
Gambar 3.14	Alur Proses Transformasi Ke Data Fuzzy.....	39
Gambar 3.15	Alur Proses Penjumlahan Bobot sejumlah K.....	40
Gambar 3.16	Alur Proses Menghitung Derajat Keanggotaan.....	41
Gambar 3.17	Alur Proses Menghitung Hasil FK-NN.....	42
Gambar 3.18	Antarmuka sistem untuk data	50
Gambar 3.19	Antarmuka sistem untuk perhitungan FK-NN	51
Gambar 3.20	Antarmuka sistem untuk grafik akurasi	52
Gambar 4.1	Antarmuka Tampilan untuk Data	63
Gambar 4.2	Antarmuka Tampilan untuk Perhitungan FKNN	64
Gambar 4.3	Antarmuka Tampilan untuk Grafik Akurasi	65
Gambar 5.1	Tingkat Akurasi untuk 4-Fold Cross Validation	91
Gambar 5.2	Detil Akurasi untuk 4-Fold Cross Validation	92
Gambar 5.3	Tingkat Akurasi untuk 5-Fold Cross Validation	93
Gambar 5.4	Detil Akurasi untuk 5-Fold Cross Validation	94
Gambar 5.5	Tingkat Akurasi untuk 6-Fold Cross Validation	94

Gambar 5.6 Detil Akurasi untuk 6-Fold Cross Validation	95
Gambar 5.7 Tingkat Akurasi untuk 10-Fold Cross Validation	96
Gambar 5.8 Detil Akurasi untuk 10-Fold Cross Validation	97



DAFTAR SOURCECODE

No.	Judul	Halaman
Sourcecode 4.1	Fungsi untuk konversi file *csv ke dalam database MySQL	57
Sourcecode 4.2	Fungsi untuk menentukan nilai minimum global	58
Sourcecode 4.3	Fungsi untuk menentukan nilai maksimum global	58
Sourcecode 4.4	Fungsi untuk menormalisasi nilai atribut	59
Sourcecode 4.5	Fungsi untuk mencari jarak tiap atribut	59
Sourcecode 4.6	Fungsi untuk menghitung jarak keseluruhan atribut	60
Sourcecode 4.7	Fungsi untuk menghitung bobot jarak euclidean	60
Sourcecode 4.8	Fungsi untuk mengurutkan bobot jarak euclidean	60
Sourcecode 4.9	Fungsi untuk menentukan nilai k yang diinginkan	61
Sourcecode 4.10	Fungsi untuk menjumlahkan nilai bobot sebanyak k	61
Sourcecode 4.11	Fungsi untuk mendapatkan nilai membership fuzzy	62
Sourcecode 4.12	Fungsi untuk mendapatkan hasil akhir	62

DAFTAR LAMPIRAN

No.	Judul	Halaman
	Lampiran 1 Dataset	103



BAB I PENDAHULUAN

1.1. Latar Belakang

Mangga adalah salah satu buah tropika yang sudah sangat lama dibudidayakan dan diperdagangkan di Indonesia. Hal tersebut tercermin dari daerah sebaran tanaman dan ragam varietas (kultivar) yang dapat dijumpai di hampir semua provinsi di Indonesia. Karena itu, mangga merupakan komoditas buah yang berpotensi untuk dikembangkan secara komersial sehingga mampu menggerakkan perekonomian, khususnya di daerah sentra produksi. Selain itu, mangga juga diharapkan mampu memberikan sumbangan peningkatan pendapatan kepada pelaku agribisnis pedesaan [AGR-08].

Pohon mangga sering ditanam di pekarangan rumah, selain berguna untuk penghijauan lingkungan, buah yang didapat juga mempunyai nilai gizi yang tinggi bagi manusia. Bagi petani atau masyarakat awam yang berminat untuk menanam pohon mangga selalu berharap bahwa buah mangga yang dihasilkan dari pohon yang ditanamnya merupakan jenis buah mangga sesuai dengan yang diinginkan.

Ahmad menuturkan pentingnya sistem komputer untuk membantu melakukan klasifikasi produk hortikultura, dengan bantuan software komputer tersebut diharapkan dapat meningkatkan standart mutu yang digunakan, selain itu dengan bantuan software ini maka pekerjaan yang awalnya dilakukan secara manual maka dapat dilakukan secara otomatis oleh sistem komputer [AHM,-10].

Penelitian yang sudah dilakukan pada klasifikasi produk hortikultura menggunakan data citra daun mangga dengan dua target klasifikasi jenis mangga berdasarkan tekstur daun menggunakan metode *K-Nearest Neighbor* (K-NN) dan *Backpropagation*. Klasifikasi menggunakan *K-Nearest Neighbor* (K-NN) ini menghasilkan akurasi sebesar 54.24% sedangkan dengan *Backpropagation* menghasilkan akurasi sebesar 65.19% [AGU-11].

Klasifikasi merupakan proses menemukan sebuah model atau fungsi yang mendeskripsikan dan membedakan data ke dalam kelas-kelas atau konsep-konsep.

Klasifikasi melibatkan proses pemeriksaan karakteristik dari objek dan memasukkan objek ke dalam salah satu kelas yang sudah didefinisikan sebelumnya [SAN-11]. Metode klasifikasi data mining yang digunakan antara lain adalah metode K-Nearest Neighbor, ID3, C45, Bayesian dan beberapa metode lainnya. Metode K-Nearest Neighbor (K-NN) adalah salah satu metode data mining yang dapat digunakan dalam pengklasifikasian jenis mangga berdasarkan tekstur daun [AGU-11]. K-Nearest Neighbor (K-NN) adalah suatu metode pengklasifikasian yang disupervisi, dimana hasil dari query akan diklasifikasikan berdasarkan mayoritas dari kategori. Algoritma ini bertujuan untuk mengklasifikasi objek baru berdasarkan atribut dan training sample. Diberikan suatu titik query, selanjutnya akan ditemukan sejumlah K objek atau (titik training) yang paling dekat dengan titik query. Nilai klasifikasi dari query instance akan ditentukan berdasarkan klasifikasi ketetanggaan. Algoritma ini memiliki kelebihan seperti tangguh terhadap training data yang noisy dan efektif apabila training data yang besar [RID-10]. Tetapi metode K-Nearest Neighbor (K-NN) masih perlu ditingkatkan lagi tingkat akurasi dengan menggabungkan teknik fuzzy dan K-Nearest Neighbor classifier yang ditemukan oleh James M. Keller dengan nama metode Fuzzy K-Nearest Neighbor (FK-NN) yang memiliki dua keunggulan utama daripada algoritma K-Nearest Neighbor (K-NN). Pertama, algoritma ini mampu mempertimbangkan sifat ambigu dari tetangga jika ada. Algoritma ini telah dirancang sedemikian rupa agar tetangga yang ambigu tidak memainkan peranan penting dalam klasifikasi saat ini. Keunggulan kedua yaitu sebuah instance akan memiliki derajat nilai keanggotaan pada setiap kelas sehingga akan lebih memberikan kekuatan atau kepercayaan suatu instance berada pada suatu kelas [YUK-11].

Selain dua keunggulan yang sudah dijelaskan mengapa penelitian ini menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN) adapula penelitian yang menjelaskan bahwa tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) lebih baik dari metode *K-Nearest Neighbor* (K-NN) [PRA-12] dan ingin membandingkan tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) dengan metode *Backpropagation*, maka dari itu penelitian ini menggunakan data klasifikasi jenis mangga berdasarkan tekstur daun dengan dua target klasifikasi

sama dengan data yang dipakai dalam penelitian yang sudah dilakukan adalah penelitian pengklasifikasian jenis mangga berdasarkan tekstur daun menggunakan metode *K-Nearest Neighbor* (K-NN) dan *Backpropagation*. Klasifikasi menggunakan *K-Nearest Neighbor* (K-NN) ini menghasilkan akurasi sebesar 54.24% sedangkan dengan *Backpropagation* menghasilkan akurasi sebesar 65.19% [AGU-11].

Oleh karena itu pada skripsi kali ini, ingin dicoba mengimplentasikan algoritma *Fuzzy K-Nearest Neighbor* (FK-NN). Dengan penerapan metode ini diharapkan akan memiliki tingkat keakuratan yang lebih tinggi daripada metode lain yang sudah pernah digunakan dalam penelitian sebelumnya.

Berdasarkan latar belakang yang telah di jelaskan di atas maka judul yang diambil untuk skripsi ini adalah **“Penerapan metode Fuzzy K-NN pada klasifikasi jenis mangga berdasarkan tekstur daun”**.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang ada, maka dapat dirumuskan permasalahannya, yaitu:

1. Bagaimana menerapkan metode *Fuzzy K-Nearest Neighbor* (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun?
2. Berapa tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun?
3. Berapa perbandingan tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun dengan metode yang pernah diterapkan sebelumnya?
4. Bagaimana metode analisa K-Fold Cross Validation menentukan K terbaik dalam metode klasifikasi *Fuzzy K-Nearest Neighbor* (FK-NN)?

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Untuk menerapkan metode *Fuzzy K-Nearest Neighbor* (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun.
2. Untuk mengukur tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-

NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun.

3. Untuk membandingkan tingkat akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) dengan metode yang diterapkan sebelumnya pada jurnal yang berjudul Klasifikasi Jenis Pohon Mangga Gadung Dan Curut Berdasarkan Tesktur Daun.
4. Untuk menentukan nilai K terbaik metode *Fuzzy K-Nearest Neighbor* (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun.

1.4. Batasan Permasalahan

Pada skripsi ini, permasalahan yang ada dibatasi pada data yang dijadikan pelatihan dan pengujian pada skripsi ini adalah dataset numerik jenis mangga gadung dan curut berdasarkan tekstur daun, didapatkan dari jurnal yang berjudul Klasifikasi Jenis Pohon Mangga Gadung Dan Curut Berdasarkan Tesktur Daun dengan menggunakan data inputan ekstraksi fitur tekstur daun sebanyak sepuluh fitur, yaitu rata-rata intensitas, *smoothness*, entropy, 5 komponen momen invariant meliputi: momen 1, momen 2, momen 4, momen 6, momen 7, energi, kontras.

1.5. Manfaat Penelitian

Manfaat yang dapat diambil dari skripsi ini adalah dapat mengukur keakuratan metode *Fuzzy K-Nearest Neighbor* (FK-NN) dari hasil klasifikasi jenis mangga berdasarkan tektur daun sehingga dapat diketahui kelayakan penerapan metode ini pada dataset citra tekstur daun mangga.

1.6. Sistematika Penulisan

1. BAB I : PENDAHULUAN

Bab ini berisi latar belakang penulisan, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan skripsi.

2. BAB II : TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori yang berkaitandengan topik penulisan skripsi menggunakan klasifikasi *Fuzzy K-Nearest Neighbor* (FK-NN), serta klasifikasi jenis mangga berdasarkan tektur daun.

3. BAB III : METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang metode-metode yang digunakan untuk klasifikasi jenis mangga berdasarkan tekstur daun menggunakan metode klasifikasi *Fuzzy K-Nearest Neighbor* (FK-NN), serta membahas tentang data penelitian, spesifikasi perangkat lunak, perancangan sistem, dan perancangan antarmuka.

4. BAB IV : IMPLEMENTASI DAN UJI COBA SISTEM

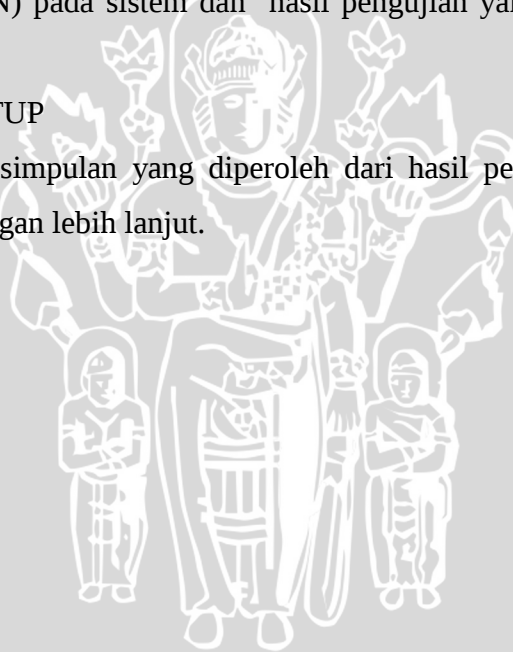
Bab ini berisi hasil implementasi perangkat lunak dan uji coba sistem.

5. BAB V : PEMBAHASAN DAN ANALISA

Bab ini berisi pembahasan dari implementasi klasifikasi jenis mangga berdasarkan tekstur daun menggunakan klasifikasi *Fuzzy K-Nearest Neighbor* (FK-NN) pada sistem dan hasil pengujian yang dilakukan serta analisa hasil.

6. BAB VI : PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian dan saran untuk pengembangan lebih lanjut.



BAB II

TINJAUAN PUSTAKA

2.1. Pohon Mangga

2.1.1. Pengertian Pohon Mangga

Tanaman mangga (*Mangifera indica*) yang banyak tumbuh di Indonesia berasal dari Tamil, India. Mangga termasuk tanaman buah berkeping dua (dikotil). Batangnya lurus, besar, kuat, dan berkulit tebal. Tinggi batangnya mencapai belasan hingga puluhan meter, dengan diameter pohon tuanya dapat berukuran lebih dari 100 cm. Daun mangga sangat rimbun. Bentuknya ada yang seperti kubah, bulat, panjang (oval), ada juga yang memanjang. Sementara itu, akarnya dapat menembus tanah hingga kedalaman puluhan meter. Umur tanaman mangga dapat mencapai lebih dari 50 tahun, jika berasal dari bibit biji [BRO-03].

Ada 62 spesies mangga yang dikenal, dan 16 spesies diantaranya dapat dimakan dan enak rasanya. *Mangifera indica* merupakan spesies yang paling enak dimakan diantara spesies lain. Diperkirakan spesies ini memiliki 650 kultivar. Namun, dari banyak kultivar tersebut hanya beberapa jenis yang cocok untuk ditanam di dalam pot, karena memiliki tajuk yang kompak, produktivitasnya tinggi, dan dapat berbuah di luar musimnya [AGR-08].

2.1.2. Karakter Daun Mangga

Daun pohon mangga umumnya tunggal, dengan letak tersebar, tanpa daun penumpu. Panjang tangkai daun bervariasi dari 1,25-12,5 cm, bagian pangkalnya membesar dan pada sisi sebelah atas ada alurnya. Aturan letak daun pada batang biasanya 3/8, tetapi makin mendekati ujung, letaknya makin berdekatan sehingga nampaknya seperti dalam lingkaran (roset). Helai daun bervariasi namun kebanyakan berbentuk jorong sampai lanset, 2-10 × 8-40 cm, agak liat seperti kulit, hijau tua berkilap, berpangkal melancip dengan tepi daun bergelombang dan ujung meluncip, dengan 12-30 tulang daun sekunder [RUK-97].

Beberapa variasi bentuk daun mangga:

- 1) Lonjong dan ujungnya seperti mata tombak.
- 2) Berbentuk bulat telur, ujungnya runcing seperti mata tombak.
- 3) Berbentuk segi empat, tetapi ujungnya runcing.
- 4) Berbentuk segi empat, ujungnya membulat.

Daun yang masih muda biasanya bewarna kemerahan, keunguan atau kekuningan; yang di kemudian hari akan berubah pada bagian permukaan sebelah atas menjadi hijau mengkilat, sedangkan bagian permukaan bawah berwarna hijau muda. Umur daun bisa mencapai 1 tahun atau lebih [AHM,-10].

2.1.3. Analisis Tekstur Citra

Analisis tekstur citra terbagi menjadi tiga bagian, yang pertama adalah pendekatan statistik, yang kedua adalah momen invariant yang terakhir adalah matriks co-occurrence.

a) Pendekatan Statistik

Pendekatan yang sering digunakan untuk analisis tekstur didasarkan pada properti statistik histogram intensitas [GON-08]. Satu kelas pengukuran didasarkan pada moment statistik. Untuk menghitung moment n th dari z terhadap mean diberikan oleh:

$$\mu^n = \sum_{i=0}^{L-1} (z_i - \bar{z})^n p(z_i) \quad (2-1)$$

di mana z_i adalah variabel random yang mengindikasikan intensitas, $p(z)$ adalah histogram level intensitas dalam region, L adalah jumlah level intensitas yang tersedia, mean (rata-rata) intensitas dihitung dengan formula:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p(z_i) \quad (2-2)$$

Ukuran *smoothness* relatif dari intensitas dalam region. R bernilai 0 untuk region dalam intensitas konstan dan mendekati 1 untuk region dengan ekskursi yang besar dalam nilai level intensitas. Dalam prakteknya, varian digunakan dalam ukuran ini yang dinormalisasikan dalam range [0,1] oleh pembagian dengan $(L-1)^2$. *Smoothness* dihitung dengan formula :

$$R = 1 - 1/(1 + \frac{\sigma^2}{(L-1)^2}) \quad (2-3)$$

Dimana σ adalah ukuran standart deviasi yang diukur dengan formula:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (z_i - \bar{z})^2} \quad (2-4)$$

Entropy digunakan untuk mengukur keacakan nilai intensitas citra, dihitung dengan formula:

$$e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (2-5)$$

b) Moment Invariants

Moment 2-D dari order (p + q) pada citra digital f(x,y) didefinisikan sebagai [GON-08]:

$$\mu_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad (2-6)$$

Untuk p,q = 0, 1, 2, ..., di mana penjumlahan lebih dari nilai koordinat spasial x dan y yang merentangkan citra. *Central moment* yang berhubungan didefinisikan sebagai:

$$\mu_{pq}^c = \sum_x \sum_y \left(x - \bar{x} \right)^p \left(y - \bar{y} \right)^q f(x,y) \quad (2-7)$$

dimana

$$\bar{x} = \frac{n \cdot 10}{n \cdot 100} \quad (2-8)$$

dan

$$\bar{y} = \frac{n \cdot 01}{n \cdot 100} \quad (2-9)$$

Normalized central moment dari order (p + q) didefinisikan sebagai:

$$\mu_{pq}^{(a)} = \frac{\sum \mu_{pq}^a}{\mu_{00}^a} \quad (2-10)$$

Untuk p, q = 0, 1, 2, ..., di mana:

$$n_r = \frac{a}{2} \cdot \frac{a}{2} + 1 \quad (2-11)$$

Untuk p+q = 2, 3, ...

Sejumlah tujuh moment invariant yang tidak sensitif terhadap translasi, perubahan skala, pencerminan, dan rotasi dapat diturunkan dari persamaan berikut:

$$\phi_1 = \eta_{20}^2 + \eta_{02}^2 \quad (2-12)$$

$$\phi_2 = \left(\frac{\eta_{20}}{\eta_{02}} - \frac{\eta_{02}}{\eta_{20}} \right)^2 + 4 \frac{\eta_{11}^2}{\eta_{11}^2} \quad (2-13)$$

$$\phi_3 = \left(\frac{\eta_{30}}{\eta_{12}} - 3 \frac{\eta_{12}}{\eta_{30}} \right)^2 + \left(3 \frac{\eta_{21}}{\eta_{03}} - \frac{\eta_{03}}{\eta_{21}} \right)^2 \quad (2-14)$$

$$\phi_4 = \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right)^2 + \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right)^2 \quad (2-15)$$

$$\phi_5 = \left(\frac{\eta_{30}}{\eta_{12}} - 3 \frac{\eta_{12}}{\eta_{30}} \right) \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right) + \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right) \left(\frac{\eta_{21}}{\eta_{03}} - \frac{\eta_{03}}{\eta_{21}} \right) + 3 \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right) \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right) + (3\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2-16)$$

$$\phi_6 = \left(\frac{\eta_{20}}{\eta_{02}} + \frac{\eta_{02}}{\eta_{20}} \right) \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right)^2 - \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right)^2 + 4 \frac{\eta_{11}^2}{\eta_{11}^2} \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right) (\eta_{21} + \eta_{03})$$

$$\phi_7 = \left(3 \frac{\eta_{30}}{\eta_{12}} - \frac{\eta_{12}}{\eta_{30}} \right) \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right) + \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right) \left(\frac{\eta_{21}}{\eta_{03}} - \frac{\eta_{03}}{\eta_{21}} \right) + 3 \left(\frac{\eta_{21}}{\eta_{03}} + \frac{\eta_{03}}{\eta_{21}} \right) \left(\frac{\eta_{30}}{\eta_{12}} + \frac{\eta_{12}}{\eta_{30}} \right) + (3\eta_{12} + \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2-17)$$

c) Matrik Co-occurrence

Matriks intensitas co-occurrence adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [AHM-05]. Matriks intensitas co-occurrence $p(i_1, i_2)$ didefinisikan dengan dua langkah sederhana sebagai berikut. Langkah pertama adalah menentukan lebih dulu jarak antara dua titik dalam arah vertikal dan horizontal (vektor $d=(dx,dy)$), di mana besaran dx dan dy dinyatakan dalam piksel sebagai unit terkecil dalam citra digital. Langkah kedua adalah menghitung pasangan piksel-piksel yang mempunyai nilai intensitas i_1 dan i_2 dan berjarak di piksel dalam citra. Kemudian hasil setiap pasangan nilai intensitas diletakkan pada matriks sesuai dengan koordinat-nya, di mana absis untuk nilai intensitas i_1 dan ordinat untuk nilai intensitas i_2 .

Energi dalam matrik co-occurrence yaitu fitur untuk mengukur konsentrasi pasangan intensitas pada matriks co-occurrence, dan didefinisikan dengan:

$$\text{energi} = \sum_{i_1=1}^n \sum_{i_2=1}^n p_{i_1, i_2}^2 \quad (2-18)$$

Kontras yang digunakan untuk mengukur kekuatan perbedaan intensitas dalam citra dan dinyatakan dengan:

$$\text{kontras} = \sum_{i_1=1}^n \sum_{i_2=1}^n (i_1 - i_2)^2 p_{i_1, i_2} \quad (2-19)$$

2.2. Sistem Fuzzy

2.2.1. Pengertian Logika Fuzzy

Sistem fuzzy pertama kali diperkenalkan oleh Prof. L. A. Zadeh dari Berkelay pada tahun 1965. Sistem fuzzy merupakan penduga numerik yang terstruktur dan dinamis. Sistem ini mempunyai kemampuan untuk mengembangkan sistem intelijen lingkungan yang tidak pasti. Sistem ini menduga suatu fungsi dengan logika fuzzy. Dalam logika fuzzy terdapat beberapa proses

yaitu penentuan himpunan *fuzzy*, penerapan aturan IF-THEN dan proses inferensi *fuzzy* [MAR-05].

2.2.2. Himpunan *Crisp*

Himpunan *Crisp* (Crisp Set) A didefinisikan oleh item-item yang ada pada himpunan itu. Jika $A \ni a$, maka nilai yang berhubungan dengan a adalah 1. Namun, jika $A \not\ni a$, maka nilai yang berhubungan dengan a adalah 0. $A = \{x \mid P(x)\}$ menunjukkan bahwa A berisi item x dengan $P(x)$ benar. Jika X_A merupakan fungsi karakteristik A dan properti P, dapat dikatakan bahwa $P(x)$ benar, jika dan hanya jika $X_A(x) = 1$ [KUS-04].

2.2.3. Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:

- 1) Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- 2) Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Jika pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaannya terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A(x) = 1$ berarti x menjadi anggota penuh pada himpunan A. Himpunan *fuzzy* memiliki 2 atribut, yaitu:

- 1) *Linguistik*, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA
- 2) *Numerik*, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* yaitu:

- a) Variabel *fuzzy* merupakan variabel yang akan dibahas dalam sistem *fuzzy* misalnya umur, temperatur, permintaan, dsb.
- b) Himpunan *fuzzy* merupakan suatu group yang mewakili suatu kondisi tertentu dalam variabel *fuzzy*, misalnya variabel umur dibagi atas 3

himpunan *fuzzy* yaitu muda, parobaya dan tua.

- c) Semesta Pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy* misalnya semesta pembicaraan variabel umur adalah 0 sampai 100.
- d) *Domain* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam himpunan *fuzzy* misalnya domain umur muda 20–45, *domain* parobaya 25–65 dan *domain* tua 45–70 [KUS-04].

2.2.4. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melakukan pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan antara lain:

- a) Representasi Linier

Pada representasi linier, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan fuzzy yang linier. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Persamaan (2-20) merupakan persamaan dari fungsi keanggotaan representasi linier naik.

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2-20)$$

Sedangkan pada linier turun, garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah dengan fungsi keanggotaan. Persamaan (2-21) merupakan persamaan dari fungsi keanggotaan representasi linier turun.

$$\mu[x] = \begin{cases} \left(\frac{b-x}{b-a}\right); & a \leq x \leq b \\ 0; & x \leq a \text{ or } x \geq b \end{cases} \quad (2-21)$$

b) Representasi Kurva Segitiga

Fungsi keanggotaan segitiga ditandai oleh adanya 3 (tiga) parameter {a,b,c} yang akan menentukan koordinat x dari tiga sudut. Kurva ini pada dasarnya merupakan gabungan antara dua garis (linier). Persamaan (2-22) merupakan persamaan dari fungsi keanggotaan segitiga.

$$\mu[x] = \begin{cases} \left(\frac{x-a}{b-a}\right); & a \leq x \leq b \\ \left(\frac{c-x}{c-b}\right); & b \leq x \leq c \\ 0; & x \leq a \text{ or } x \geq c \end{cases} \quad (2-22)$$

c) Representasi Kurva Bell

Fungsi keanggotaan generalized bell, disifati oleh parameter {a,b,c}. Persamaan (2-23) merupakan persamaan dari fungsi keanggotaan bell.

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-a}{b} \right|^{2c}} \quad (2-23)$$

parameter b selalu positif, supaya kurva menghadap kebawah.

d) Representasi Kurva Bahu

Representasi fungsi keanggotaan *fuzzy* dengan menggunakan kurva bahu pada dasarnya adalah gabungan dari kurva segitiga dan kurva trapesium. Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang pada salah sisi dari variabel *fuzzy* yang ditinjau ini terdapat nilai yang konstan, yaitu pada himpunan ekstrim kiri dan ekstrim kanan [KUS-04].

2.3. Data Mining

2.3.1. Pengertian Data Mining

Definisi data mining adalah suatu algoritma di dalam menggali informasi berharga yang terpendam atau tersembunyi pada suatu koleksi data (database) yang sangat besar sehingga ditemukan suatu pola yang menarik yang sebelumnya tidak diketahui [HAI-04].

Menurut Kantardzic (2003), tujuan data mining yang berkaitan dengan klasifikasi adalah menghasilkan model yang dapat digunakan untuk klasifikasi atau estimasi. Sedangkan tujuan data mining yang berkaitan dengan deskripsi adalah memperoleh pemahaman sistem yang dianalisa dengan mengungkap pola dan hubungan dalam kumpulan data yang besar.

Terdapat beberapa fungsionalitas *data mining*, sebagai berikut [HAN-00]:

1. Analisis asosiasi

Menemukan aturan (*rule*) asosiasi yang menunjukkan kondisi nilai atribut yang sering ada bersamaan dalam satu kumpulan data.

2. Klasifikasi

Fungsi pembelajaran yang memetakan (mengklasifikasi) sebuah item data ke dalam salah satu dari beberapa kelas yang sudah didefinisikan.

3. *Clustering*

Melakukan pengelompokan data tanpa berdasarkan kelas data tertentu.

4. *Pendeteksian* perubahan dan deviasi

Berfokus pada penemuan perubahan yang paling signifikan di dalam

data dari nilai-nilai yang telah diukur sebelumnya.

2.3.2. Klasifikasi

Menurut Han dan Kamber (2000) klasifikasi adalah proses untuk menemukan suatu kumpulan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep, agar dapat menggunakan model tersebut untuk mengklasifikasi kelas objek yang kelas labelnya tidak diketahui.

Klasifikasi merupakan suatu teknik yang dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang telah diklasifikasi dan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunakan supervised induction, yang memanfaatkan kumpulan pengujian dari record yang terklasifikasi untuk menentukan kelas data baru [KUS-07].

Menurut Pramudiono (2003), proses klasifikasi biasanya dibagi menjadi dua fase, yaitu:

1. *Learning*

Pada fase ini sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan.

2. *Test*

Pada fase ini model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi, model ini dapat dipakai untuk klasifikasi kelas data yang belum diketahui.

Terdapat beberapa metode klasifikasi, antara lain decision tree, Bayesian, fuzzy, neural network, support vector machine (SVM) dan k-nearest neighbor.

2.3.3. K-Nearest Neighbor (KNN)

2.3.3.1. Pengertian K-Nearest Neighbor (KNN)

K-Nearest Neighbor merupakan algoritma yang sering digunakan untuk klasifikasi [MOR-09]. Menurut Sarkar dan Leong (2000), kelebihan dari algoritma KNN adalah sederhana dan mudah diimplementasikan. Algoritma ini mencari k training record (tetangga) yang memiliki jarak terdekat dari record

baru, untuk mengklasifikasi kelas dari record baru tersebut.

Salah satu masalah yang dihadapi K-NN adalah pemilihan nilai K yang sulit, cara voting mayoritas dari K-tetangga untuk nilai K yang besar bisa mengakibatkan distorsi data yang besar, jika K terlalu kecil bisa menyebabkan algoritma terlalu sensitif terhadap noise. Meskipun begitu, sebenarnya penentuan nilai K yang terbaik juga dapat dilakukan dengan menggunakan teknik cross validation [TAN-06].

2.3.3.2. Proses K-Nearest Neighbor (KNN)

Pada algoritma K-NN, data berdimensi N, dapat dihitung jarak dari data tersebut ke data yang lain, nilai jarak ini yang digunakan sebagai nilai kedekatan antara data uji dengan data latih. Nilai K pada K-NN berarti K-data terdekat dari data uji.

Pada algoritma K-NN, sebuah data uji $z = (x', y')$, dimana x' ada vektor/atribut data uji, sedangkan y' adalah label kelas data uji yang belum diketahui, kemudian menghitung jarak (atau kemiripan) data uji ke setiap data latih $d(x', x)$, kemudian mengambil K tetangga terdekat pertama dalam D_z . Setelah itu dihitung jumlah data yang mengikuti kelas yang ada dari K tetangga tersebut. Kelas dengan data terbanyak yang mengikutinya menjadi kelas pemenang yang diberikan sebagai label kelas pada data uji y' .

Langkah awal yang dilakukan pada metode ini adalah menghitung jarak antara record baru dengan tiap record pada data latih. Namun sebelum dihitung jaraknya, setiap record dilakukan normalisasi terlebih dahulu. Normalisasi yang digunakan pada penelitian ini adalah *Min-max normalization*. Untuk menghitung nilai normalisasi ditunjukkan oleh persamaan (2-24).

$$V[i][j]' = \frac{V[i][j] - \min[V[i][j]]}{\max[V[i][j]] - \min[V[i][j]]} \quad (2-24)$$

Dimana:

$V[i][j]'$ = hasil normalisasi yang nilainya berkisar antara 0 dan 1

$V[i][j]$ = nilai atribut yang akan dinormalisasi

$\min [j]$ = nilai minimum dari suatu atribut j
 $\max [j]$ = nilai maksimum dari suatu atribut j

Selanjutnya, atribut yang telah di normalisasi tersebut digunakan pada proses klasifikasi. Awalnya dihitung jarak terdekatnya terlebih dahulu. Untuk menghitung jarak digunakan fungsi jarak Euclidean yang ditunjukkan oleh persamaan (2-25).

$$\begin{aligned} x_1 &= (x_{11}, x_{12}, \dots, x_{1n}) \\ x_2 &= (x_{21}, x_{22}, \dots, x_{2n}) \\ d(x_1, x_2) &= \sqrt{\sum_{r=1}^n (ar(x_1) - ar(x_2))^2} \end{aligned} \quad (2-25)$$

di mana x_1 dan x_2 merupakan dua record dengan n atribut. Persamaan (2-25) menghitung jarak antara x_1 dan x_2 , dengan tujuan untuk menentukan perbedaan antara nilai-nilai atribut pada record x_1 dan x_2 . Setelah diketahui jarak antar record, kemudian diambil sebanyak k tetangga terdekat untuk mengklasifikasi label kelas dari record baru menggunakan label kelas tetangga. Klasifikasi tersebut diketahui dengan fungsi kombinasi yang dibahas pada subbab 2.3.3.3.

2.3.3.3. Fungsi Kombinasi

Untuk memberikan keputusan klasifikasi bagi *record* baru, dilakukan kombinasi terhadap *record* yang serupa, dengan fungsi kombinasi. Terdapat dua jenis fungsi kombinasi *unweighted voting* dan *weighted voting*. Pada *unweighted voting*, label kelas untuk record yang baru, dipilih berdasarkan label kelas terbanyak (mayoritas) yang dimiliki tetangga. Sedangkan *weighted voting* dilakukan dengan memberikan bobot terhadap beberapa tetangga yang dekat dengan record yang baru. Pemberian bobot ini dapat memberikan pengaruh lebih dalam menentukan label kelas. *Weighted* ditunjukkan oleh persamaan (2-26).

$$weight(y) = \frac{1}{DISTANCE(x, a)} \quad (2-26)$$

Dimana:

$y = \text{neighbor}$

$a = \text{new Record}$

DISTANCE (*neighbor*, *newRecord*) merupakan jarak antara record yang baru dengan tetangganya. Persamaan (2-26) dapat menghitung bobot dari semua tetangganya, dengan menjumlahkan bobot dari tetangga yang memiliki label kelas yang sama. Label kelas dari *record* yang baru adalah label kelas dari *record* yang jumlah bobot tetangganya paling besar [MOR-09].

2.3.3.4. Algoritma K-Nearest Neighbor

Berikut ini adalah algoritma *K-Nearest Neighbor* menurut James M Keller (1985):

$W = \{x_1, x_2, \dots, x_n\}$

BEGIN

Input y , of unknown classification

Set K , $1 \leq K \leq n$

Initialize $i = 1$

DO UNTIL (K-nearest neighbors found)

 Compute distance from y to x_i

 IF ($i \leq K$) THEN

 Include x_i in the set of K-nearest neighbor

 ELSE IF (x_i closer to y than any previous nearest neighbor) THEN

 Delete the farthest of the K-nearest neighbors

 Include x_i in the set of K-nearest neighbors

 END IF

 Increment i

END DO UNTIL

Determine the majority class represented in the set of K-Nearest neighbor

IF (a tie exists) THEN

 Compute sum of distance of neighbor in each class which tied

 IF (no tie occurs) THEN

 Classify y in the class of minimum sum

 ELSE

 Classify y in the class of last minimum found

 END IF

ELSE

 Classify y in the majority class

```
END IF
END
```

2.3.4. Fuzzy K-Nearest Neighbor Classifier

2.3.4.1. Pengertian *Fuzzy K-Nearest Neighbor* (FK-NN)

Fuzzy K-NN merupakan metode klasifikasi yang menggabungkan teknik fuzzy dengan k-Nearest Neighbor classifier. Algoritma fuzzy K-nearest neighbor memberikan nilai keanggotaan kelas pada vector sampel dan bukan menempatkan vector pada kelas tertentu. Keuntungannya adalah nilai-nilai keanggotaan vector seharusnya memberikan tingkat jaminan pada hasil klasifikasi. Sebagai contoh, jika vector diberikan nilai keanggotaan 0.9 pada kelas pertama dan 0.05 pada dua kelas lainnya kita dapat cukup yakin bahwa kelas dengan nilai keanggotaan 0.9 adalah kelas milik vektor tersebut. Di sisi lain, jika sebuah vektor diberi nilai keanggotaan 0.55 pada kelas pertama, 0.44 pada kelas kedua, dan 0.01 pada kelas ketiga, maka kita harus ragu-ragu untuk menetapkan vektor berdasarkan hasil ini. Namun, dapat diyakinkan bahwa vektor tersebut bukan milik kelas ketiga. Dalam kasus seperti ini, perlu diperiksa lebih lanjut untuk menentukan klasifikasinya, karena memiliki derajat keanggotaan yang tinggi pada dua kelas yaitu satu dan dua. Pemberian nilai keanggotaan oleh algoritma ini jelas berguna dalam proses klasifikasi [YUK-11].

2.3.4.2. Algoritma *Fuzzy K-Nearest Neighbor* (FK-NN)

Dasar dari algoritma ini adalah untuk menetapkan nilai keanggotaan sebagai fungsi jarak vektor dari *K-Nearest Neighbor* dan keanggotaan tetangga mereka di kelas-kelas yang memungkinkan. Berikut ini adalah algoritma *fuzzy K-NN* [KEL-85]:

```
BEGIN
  Input x, of unknown classification
  Set K,  $1 \leq K \leq n$ 
  Initialize i = 1
  DO UNTIL (K-nearest neighbors to x found)
    Compute distance from x to  $x_i$ 
    IF ( $i \leq K$ ) THEN
      Include  $x_i$  in the set of K-nearest neighbor
```

```

ELSE IF ( $x_i$  closer to  $x$  than any previous nearest neighbor) THEN
    Delete the farthest of the K-nearest neighbors
    Include  $x$ , in the set of K-nearest neighbors
END IF
END DO UNTIL
Set  $i = 1$ 
DO UNTIL ( $x$  assigned membership in all classes)
    Compute  $u_i(x)$  using (2.10)
    Increment  $i$ 
END DO UNTIL
END

```

Berikut ini merupakan rumus (2-27), yang merupakan fungsi rumus logika fuzzy untuk menentukan nilai keanggotaan kelas. Seperti halnya pada teori fuzzy, sebuah data mempunyai nilai keanggotaan pada setiap kelas, yang artinya sebuah data bisa dimiliki oleh kelas yang berbeda dengan nilai derajat keanggotaan dalam interval $[0,1]$. Teori himpunan fuzzy men-generalisasi teori K-NN klasik dengan mendefinisikan nilai keanggotaan sebuah data pada masing-masing kelas. Dimana U_{ij} merupakan keanggotaan kelas ke- i pada pattern ke- j . Adapun nilai dari U_{ij} ditentukan oleh :

$$U_{ij}(x) = \begin{cases} 0.5 + \frac{n_j}{k} * 0.49 & j = i \\ \frac{n_j}{k} * 0.49 & j \neq i \end{cases} \quad (2-27)$$

dimana n_j dinotasikan sebagai jumlah neighbor pada kelas ke- j .

Namun algoritma F-kNN ini masih mempunyai beberapa kelemahan, salah satunya yaitu keakuratannya tergantung pada pemilihan nilai k . Ketika pemilihan nilai k terlalu kecil akan menghasilkan akurasi yang rendah karena hasil dari klasifikasi hanya tergantung pada sejumlah k dan tidak memperhatikan besar data training yang tersedia. Sedangkan pemilihan nilai k yang terlalu tinggi juga akan menyebabkan akurasi rendah karena semakin banyak data yang tidak relevan (noise), sehingga hasil kategorisasi klasifikasi baru lebih terpengaruh dengan kelas yang lebih besar. Meskipun begitu, sebenarnya penentuan nilai K yang terbaik juga dapat dilakukan dengan menggunakan teknik cross validation [LEU-07].

Nilai keanggotaan suatu data pada kelas sangat dipengaruhi oleh jarak data itu ke tetangga terdekatnya, semakin dekat ke tetangganya maka semakin besar nilai keanggotaan data tersebut pada kelas tetangganya, begitu pula sebaliknya. Jarak tersebut diukur dengan N dimensi (fitur) data.

Pengukuran jarak (kemiripan) dua data yang digunakan dalam FK-NN digeneralisasi dengan persamaan [2-28]:

$$d(x_i, x_j) = \left(\sum_{l=1}^N |x_{il} - x_{jl}|^p \right)^{\frac{1}{p}} \quad (2-28)$$

Dimana N adalah dimensi (jumlah fitur) data. Untuk p adalah penentu jarak yang digunakan, jika $p=1$ maka jarak yang digunakan adalah Manhattan, jika $p=2$ maka jarak yang digunakan adalah Euclidean, jika $p=\infty$ maka jarak yang digunakan adalah Chebyshev.

Meskipun FK-NN menggunakan nilai keanggotaan untuk menyatakan keanggotaan data pada setiap kelas, tetapi untuk memberikan keluaran akhir, FK-NN tetap harus memberikan kelas akhir hasil klasifikasi, untuk keperluan ini, FKNN memilih kelas dengan nilai keanggotaan terbesar pada data tersebut [PRA-12].

2.4. Akurasi Pengujian K-Fold Cross Validation

Ada beberapa teknik untuk mengestimasi tingkat kesalahan yang telah dikembangkan pada bidang pengenalan pola, salah satunya adalah k-fold cross validation. Cara kerjanya adalah melakukan pengelompokan antara data latih dan data uji yang saling asing atau terpisah dan tidak ada irisan, kemudian dilakukan proses pengujian yang diulang sebanyak k kali. Hasil pengujian itu kemudian dirata-ratakan untuk menghasilkan sebuah nilai [FAU-10].

Langkah-langkah teknik k-fold cross validation [PRA-10] adalah sebagai berikut:

1. Membagi data yang ada menjadi k kelompok
2. Untuk setiap k, buat sejumlah T himpunan data yang memuat semua

- data latih kecuali yang berada di kelompok ke-k.
3. Kerjakan algoritma yang dimiliki dengan sejumlah T data latih.
 4. Uji algoritma ini dengan menggunakan data pada kelompok 'k' sebagai data uji.
 5. Lakukan pencatatan hasil algoritma.

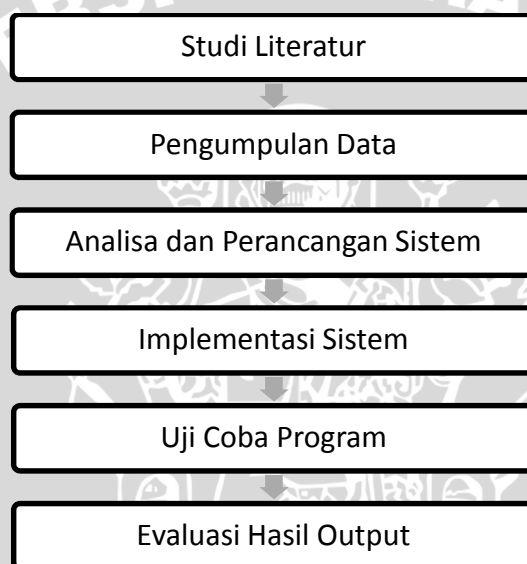
K-fold cross validation sangat tepat dan berguna ketika menentukan nilai yang tepat untuk k. Teknik ini tidak membutuhkan waktu banyak untuk membuat data uji yang ada. Keuntungan teknik k-fold cross validation adalah bahwa semua elemen pada basis data digunakan untuk pelatihan sekaligus pengujian [PRA-10].



BAB III

METODOLOGI DAN PERANCANGAN

Metodologi dan perancangan sistem dilakukan untuk memperoleh gambaran yang tepat dari apa yang akan dilakukan oleh sistem serta bertujuan untuk mengetahui bagaimana seluk beluk sistem yang akan diteliti dan dibangun. Secara umum, tahapan dari penelitian ini adalah ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Rancangan gambar 3.1 yang digunakan dan langkah-langkah yang dilakukan dalam penelitian tentang sistem klasifikasi jenis mangga berdasarkan tekstur daun menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN), yaitu :

1. Mempelajari literatur tentang metode *Fuzzy K-Nearest Neighbor* (FK-NN) sebagai suatu algoritma yang menyelesaikan masalah klasifikasi.
2. Mengumpulkan data-data yang dibutuhkan untuk sistem klasifikasi jenis mangga berdasarkan tekstur daun.
3. Menganalisa dan melakukan perancangan sistem yang akan dibuat dengan metode *Fuzzy K-Nearest Neighbor* (FK-NN).

4. Mengimplementasikan hasil perancangan sistem yang telah dibuat dalam perangkat lunak.
5. Melakukan uji coba terhadap perangkat lunak.
6. Mengevaluasi hasil output pada saat uji coba perangkat lunak yang dihasilkan oleh sistem.

3.1. Studi Literatur

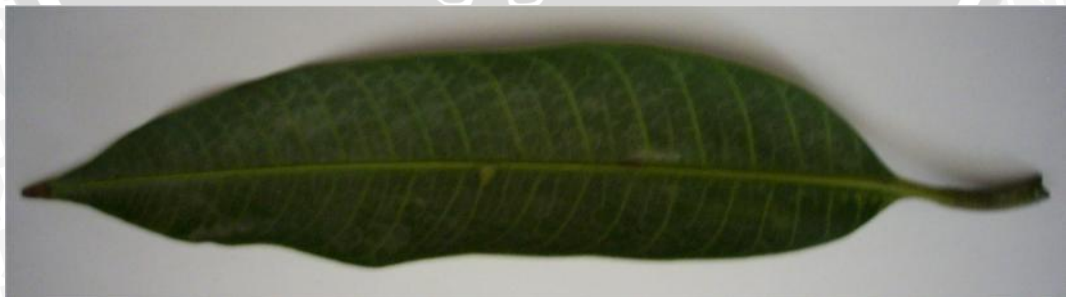
Tahap awal untuk melakukan penelitian ini yaitu dilakukannya studi pustaka secara teoritis. Pada tahap studi pustaka dilakukan kajian secara teoritis mengenai metode-metode yang dapat mendukung untuk penyelesaian permasalahan dalam penelitian ini. Studi pustaka yang digunakan dapat diperoleh dari buku, jurnal, internet atau penelitian sebelumnya. Tinjauan Pustaka memberikan pengetahuan mendasar bagi peneliti untuk meneliti obyek penelitian, sehubungan dengan itu maka dilakukan studi pustaka mengenai:

1. Dataset numerik jenis mangga gadung dan curut berdasarkan tekstur daun.
2. Metode *Fuzzy K-Nearest Neighbor* (FK-NN).

3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data dari jurnal penelitian yang berjudul Klasifikasi Jenis Mangga Gadung dan Curut Berdasarkan Tekstur Daun. Berikut contoh dari kedua gambar daun pohon mangga gadung dan curut:

1. Daun pohon mangga gadung



Gambar 3.2 Daun pohon mangga gadung

2. Daun pohon mangga curut



Gambar 3.3 Daun pohon mangga curut

Awalnya atribut data numerik jenis mangga gadung dan curut yang didapat ada 17 atribut meliputi: rata-rata intensitas, rata-rata kontras, *smoothness*, *third moment*, *uniformity*, entropi dimana 6 atribut tadi mewakili pendekatan statistik dan deskriptor tekstur berdasarkan histogram intensitas region, 7 momen invarian dimisalkan M1, M2, M3, M4, M5, M6, M7, matriks intensitas co-occurrence untuk mendapatkan nilai entropi, energi, kontras, dan yang terakhir homogenitas.

Untuk pemilihan atribut yang ingin disertakan dalam penelitian sebelum dilakukan pengimplementasian metode algoritma *Fuzzy K-Nearest Neighbor* (FK-NN) akhirnya terpilih 10 atribut berdasarkan subjektifitas peneliti dan diambil sama untuk keperluan pengujian serta perbandingan metode. Atribut-atribut tersebut adalah rata-rata intensitas, *smoothness*, entropi, M1, M2, M4, M6, M7, energi, kontras dan akhirnya siap untuk dilakukan tahapan berikutnya yaitu analisa dan perancangan sistem.

3.3. Analisis Dan Perancangan Sistem

3.3.1. Deskripsi Sistem

Secara umum sistem yang dibangun adalah perangkat lunak yang mengimplementasikan algoritma *Fuzzy K-Nearest Neighbor* (FK-NN) untuk mengklasifikasikan jenis mangga gadung dan curut berdasarkan tekstur daun dengan menggunakan data inputan fitur tekstur daun sebanyak sepuluh fitur, yaitu rata-rata intensitas, *smoothness*, entropi, 5 komponen momen invariant meliputi:

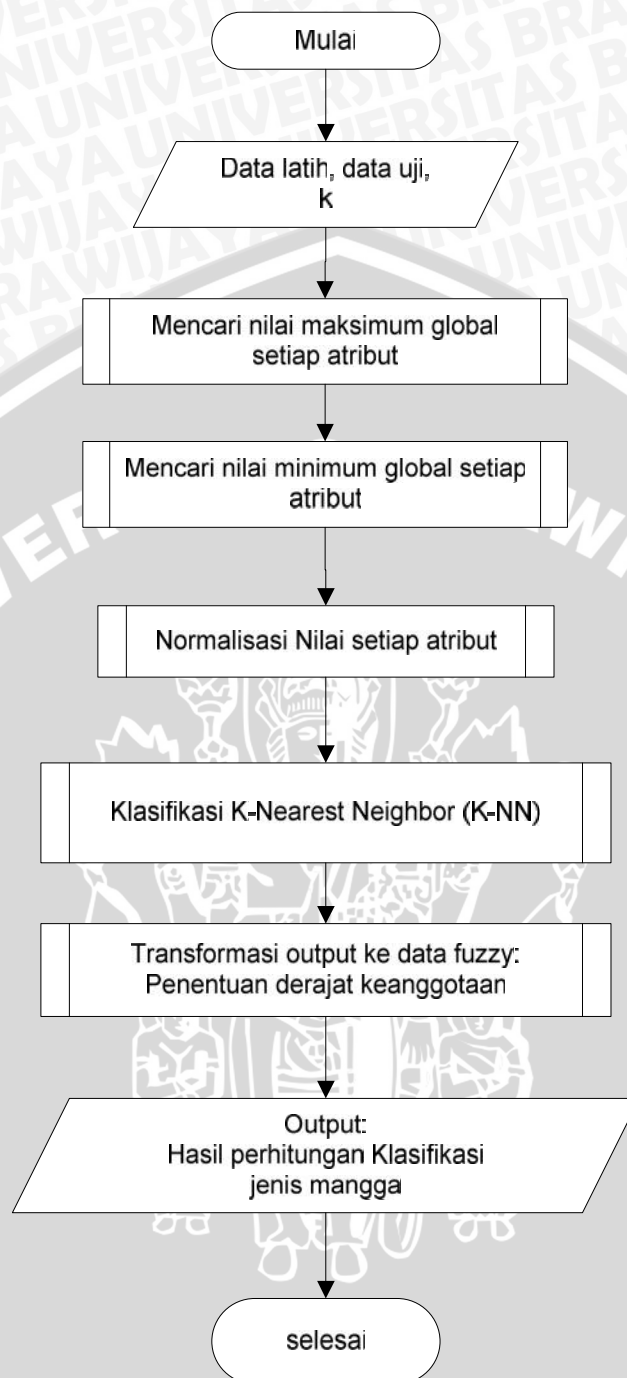
momen 1, momen 2, momen 4, momen 6, momen 7, energy, kontras. Pengujian dilakukan pada 2 jenis daun mangga: gadung, dan curut, masing-masing diambil sampel 30 citra daun dari 10 daun, sehingga total ada 60 citra daun yang digunakan sebagai data set dalam penelitian yang nantinya menghasilkan suatu informasi bahwa data tersebut masuk dalam klasifikasi jenis mangga gadung atau curut.

Sistem ini menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN). Metode ini merupakan gabungan antara penalaran logika fuzzy dan algoritma klasifikasi *K-Nearest Neighbors* (K-NN). Tingkat keberhasilan dari sistem ini diukur dengan menggunakan tingkat keakuratan yang dihasilkan oleh output sistem. Semakin besar akurasi yang dihasilkan maka akan semakin besar pula tingkat keberhasilan dari sistem.

Pada Proses penentuan klasifikasi ini, sistem harus memperoleh data terlebih dahulu sebagai basis pengetahuan yang nantinya akan disimpan ke dalam Ms. Excel yang kemudian di import ke MySQL. Data numerik yang tersimpan dalam MySQL akan dipanggil di program untuk dilakukan normalisasi terlebih dahulu. Data yang telah dinormalisasi ini kemudian akan di klasifikasi *K-Nearest Neighbor* (K-NN). Setelah diketahui nilai klasifikasi *K-Nearest Neighbor* (K-NN) maka akan digunakan untuk menghitung lebih lanjut menggunakan transformasi output ke data fuzzy penentuan derajat keanggotaan. Data-data ini akan digunakan untuk menghitung hasil nilai output.

3.3.2. Perancangan Proses

Subbab ini menjelaskan tentang proses-proses yang dilakukan oleh sistem. Alur proses dari sistem ditampilkan oleh Gambar 3.4.



Gambar 3.4 Alur Proses Sistem secara umum

Berdasarkan Gambar 3.4, proses pada sistem meliputi tiga proses utama, yaitu:

1. Normalisasi nilai setiap atribut

Merupakan proses transformasi berupa normalisasi terhadap nilai-nilai atribut

untuk membakukan skala pengaruh yang ada pada atribut, terhadap hasil.

Proses ini dijelaskan pada subbab 3.3.2.1.

2. Klasifikasi K-Nearest Neighbor

Merupakan proses klasifikasi pada dataset, dimana dataset ini di kelaskan ke dalam 2 kelas yaitu mangga gadung dan mangga curut. Merupakan proses klasifikasi data uji menggunakan KNN dengan k tetangga terdekat. Proses ini dijelaskan pada subbab 3.3.2.2.

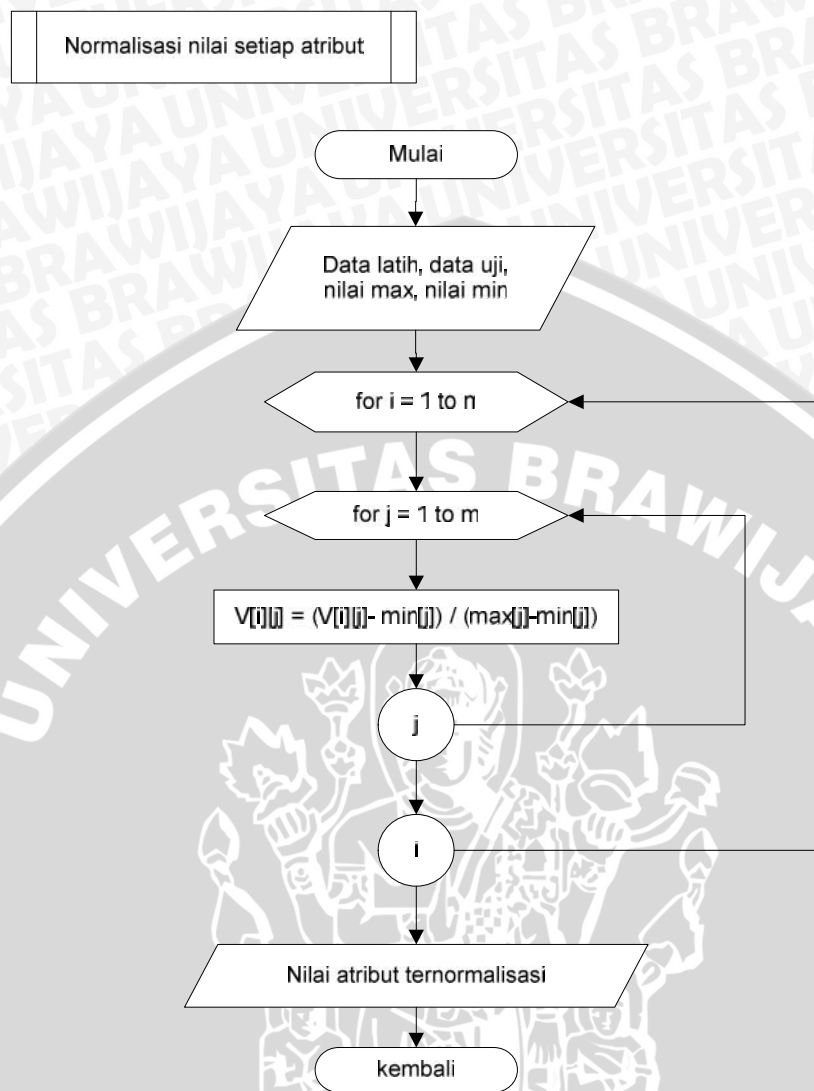
3. Transformasi ke data fuzzy

Proses ini mengubah nilai kuantitatif pada setiap atribut dalam dataset menjadi derajat keanggotaan fuzzy beserta linguistiknya berdasarkan fungsi keanggotaan atribut. Proses ini dijelaskan pada subbab 3.3.2.3.

3.3.2.1. Normalisasi nilai setiap atribut

Proses ini merupakan normalisasi terhadap nilai-nilai atribut untuk membakukan skala pengaruh yang ada pada atribut, terhadap hasil. Untuk mencari hasil yang telah dinormalisasikan digunakan rumus (2-24).

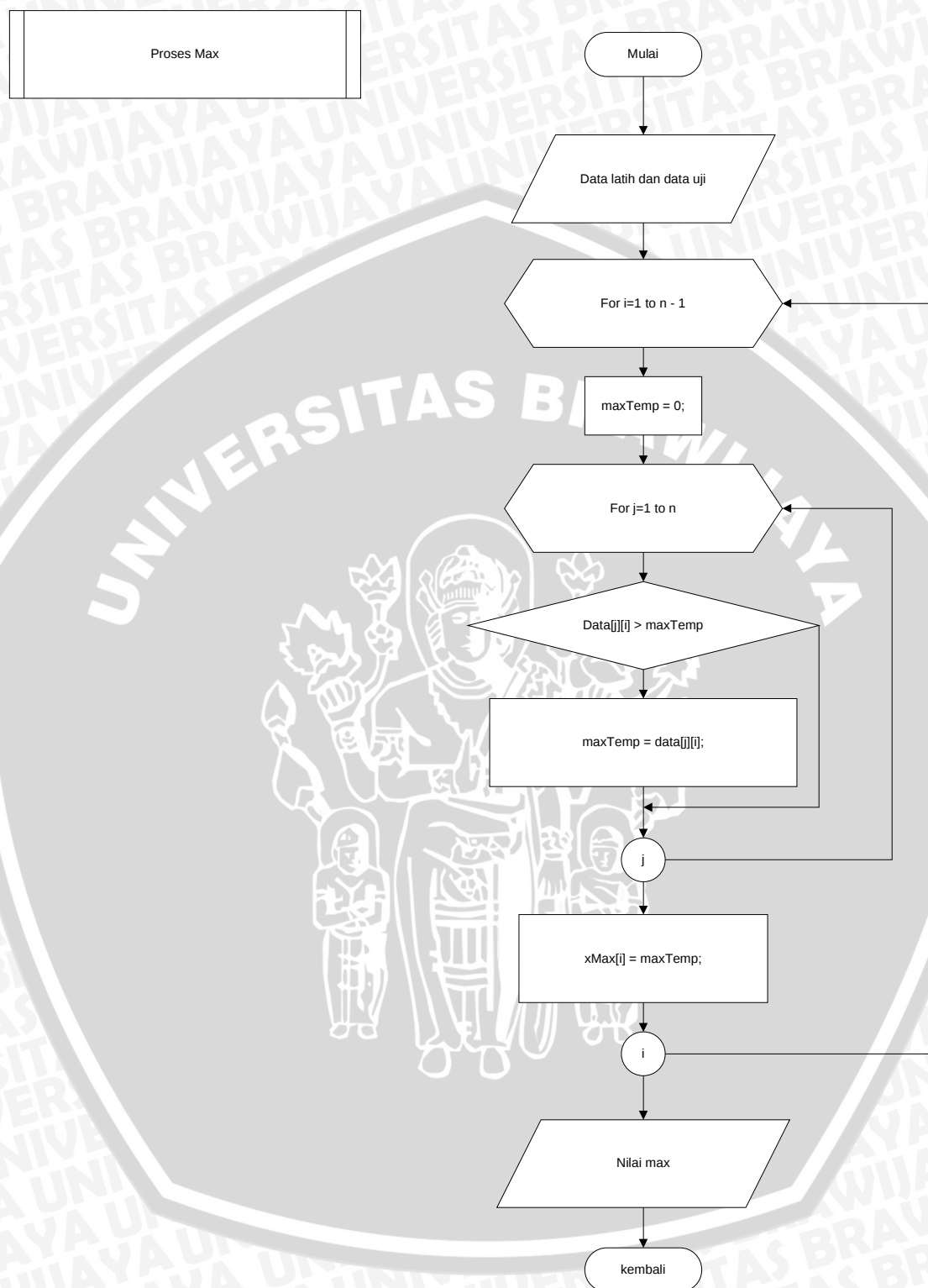
Selanjutnya, atribut yang telah di normalisasi tersebut digunakan pada proses klasifikasi. Alur proses dari sistem ditampilkan oleh Gambar 3.5.



Gambar 3.5 Alur Proses Normalisasi setiap atribut

3.3.2.1.1. Hitung Max

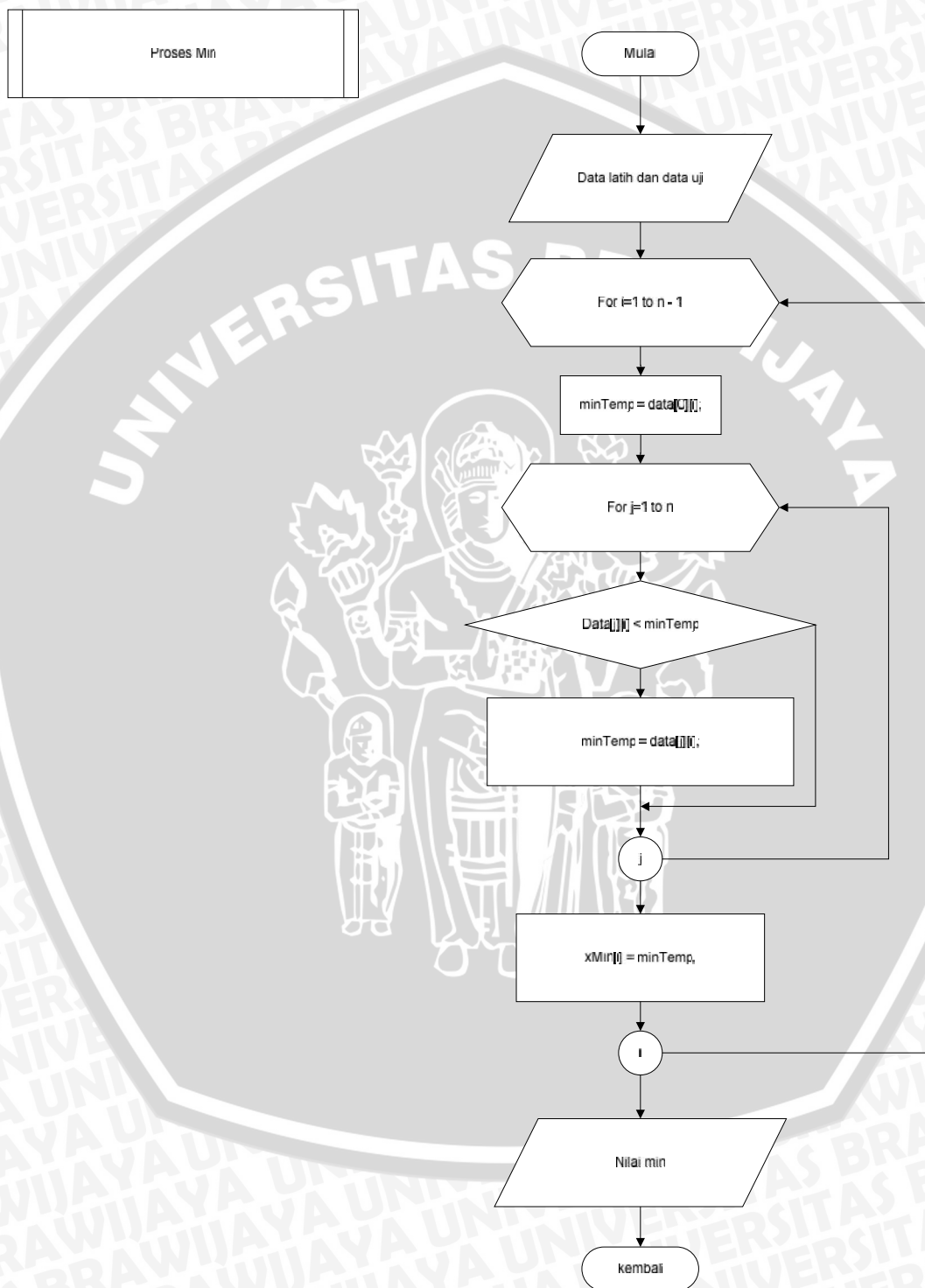
Proses hitung max merupakan proses mencari nilai maksimum data setiap atribut. Alur proses dari sistem ditampilkan oleh Gambar 3.6.



Gambar 3.6 Alur Proses Hitung Max

3.3.2.1.2. Hitung Min

Proses hitung min merupakan proses mencari nilai minimum data setiap atribut. Alur proses dari sistem ditampilkan oleh Gambar 3.7.

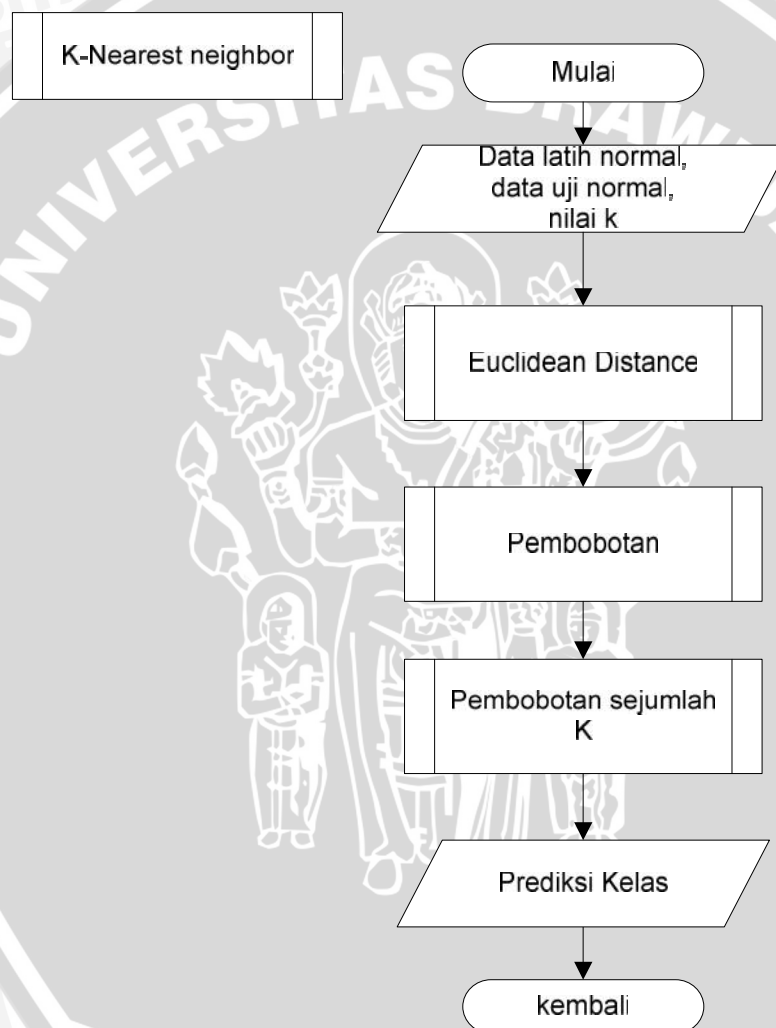


Gambar 3.7 Alur Proses Hitung Min

3.3.2.2. Klasifikasi K-Nearest Neighbor (K-NN)

Proses ini mengklasifikasikan dataset ke dalam 2 kelas. Dimana setiap kelas memiliki anggota yang terdapat dalam range yang sama sesuai perhitungan data. Pada Klasifikasi ini juga dihitung distance atau jarak similarity dengan menggunakan Euclidean Distance dan weighted vote.

Flowchart untuk proses Klasifikasi K-Nearest Neighbor (K-NN) ditampilkan oleh Gambar 3.8.



Gambar 3.8 Alur Proses Klasifikasi K-Nearest Neighbor

Langkah-langkah dalam proses klasifikasi dengan K-Nearest Neighbor adalah sebagai berikut:

1. Masukan (input) berupa data latih dan data uji yang telah

dinormalisasi serta nilai k yang dimasukkan oleh user.

2. Dilakukan iterasi untuk menghitung distance (jarak) antara record baru dengan tiap record pada data latih menggunakan persamaan (2-25).
3. Menghitung nilai weighted vote sesuai dengan persamaan (2-26).
4. Mengambil k record dengan distance terkecil yang akan digunakan sebagai tetangga.
5. Menentukan klasifikasi kelas.
6. Keluaran (output) berupa klasifikasi kelas dari data uji.

3.3.2.2.1. Proses Hasil Kuadrat

Proses ini untuk menghitung hasil kuadrat dari setiap atribut. Flowchart untuk proses Hasil Kuadrat ditampilkan oleh Gambar 3.9.

3.3.2.2.2. Menghitung Euclidean Distance

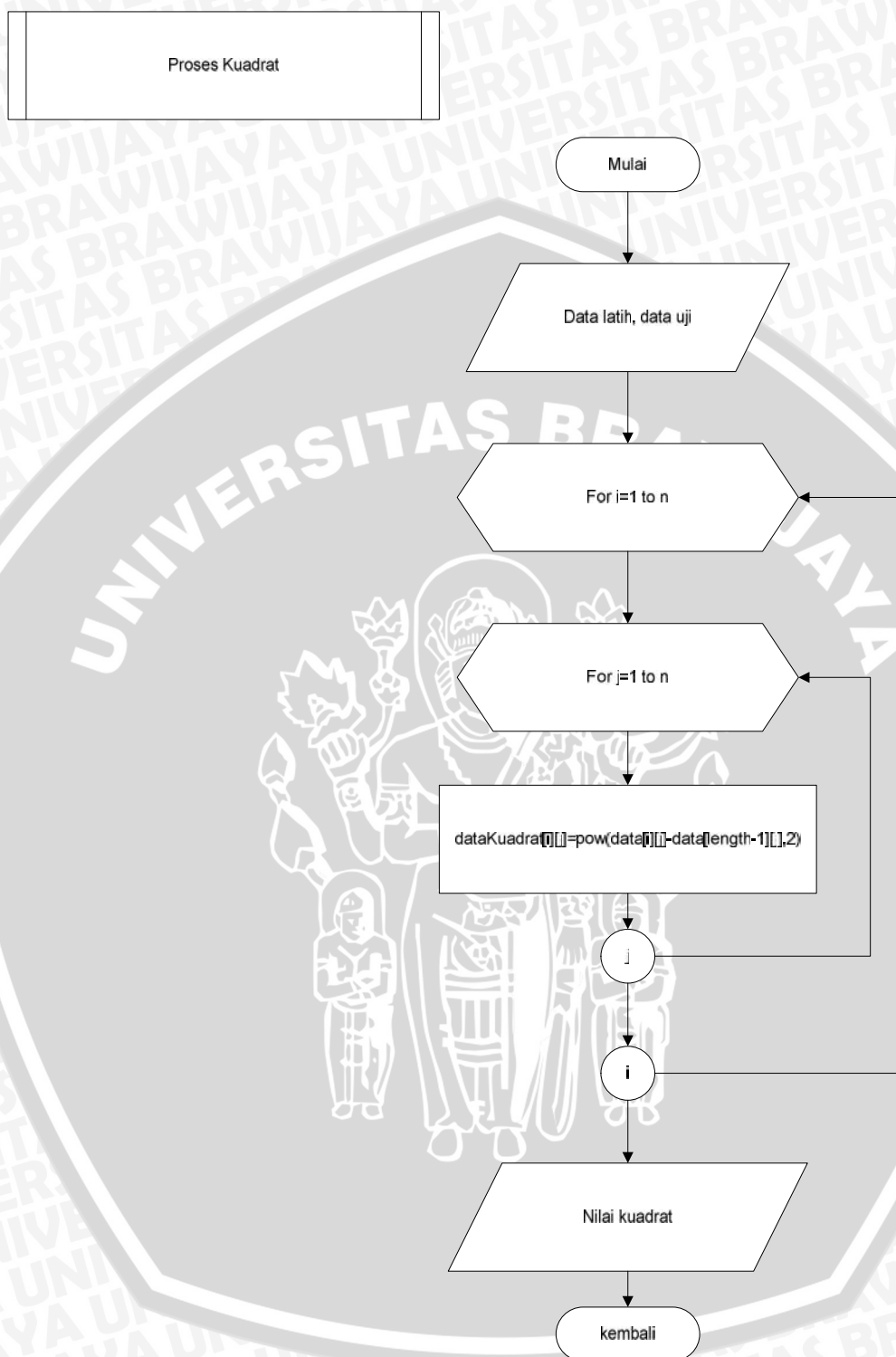
Proses ini untuk menghitung distance (jarak) antara record data uji dengan tiap record pada data latih. Dimana untuk mendapatkan nilai jarak terdekat, antara data latih dan data uji, dihitung dari setiap atribut. Flowchart untuk proses Euclidean Distance ditampilkan oleh Gambar 3.10.

3.3.2.2.3. Proses Pembobotan

Proses ini untuk menghitung pembobotan dari setiap nilai euclidean. Flowchart untuk proses Hasil Pembobotan ditampilkan oleh Gambar 3.11.

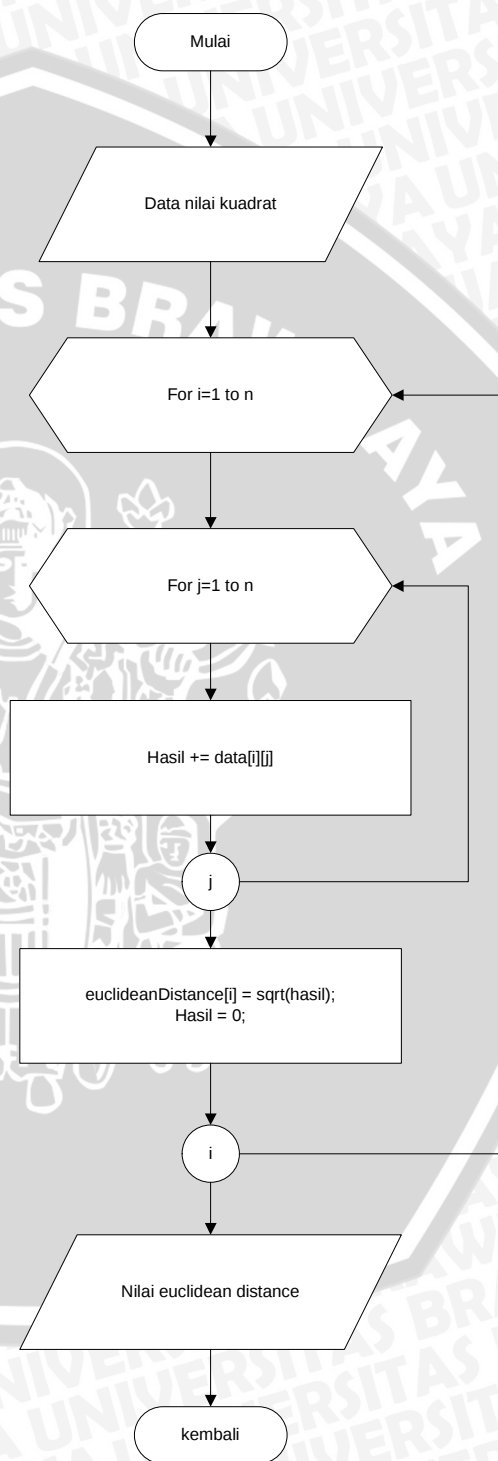
3.3.2.2.4. Proses Pengurutan

Proses ini untuk mengurutkan nilai pembobotan dari setiap nilai euclidean. Flowchart untuk proses Pengurutan Pembobotan ditampilkan oleh Gambar 3.12.

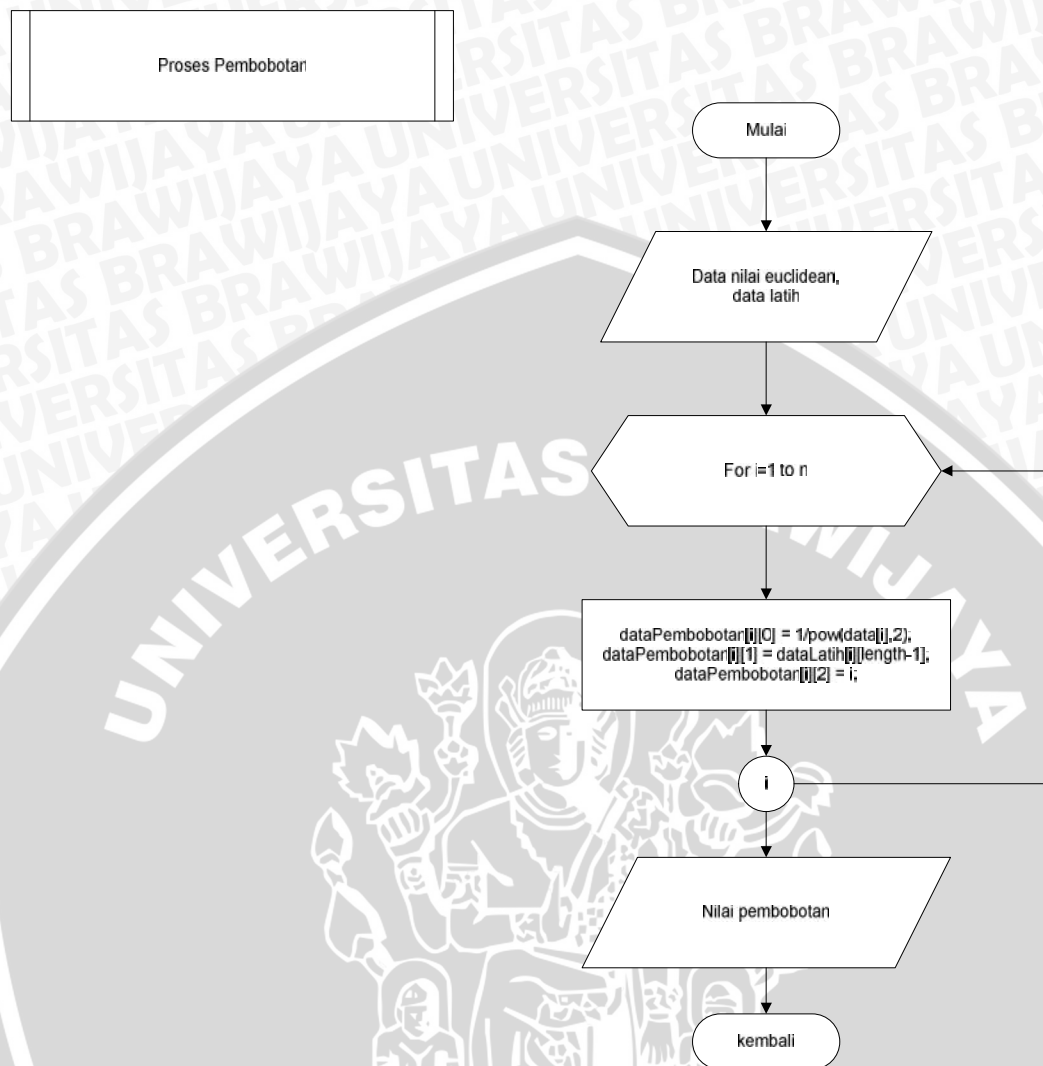


Gambar 3.9 Alur Proses Hasil Kuadrat

Proses Euclidean



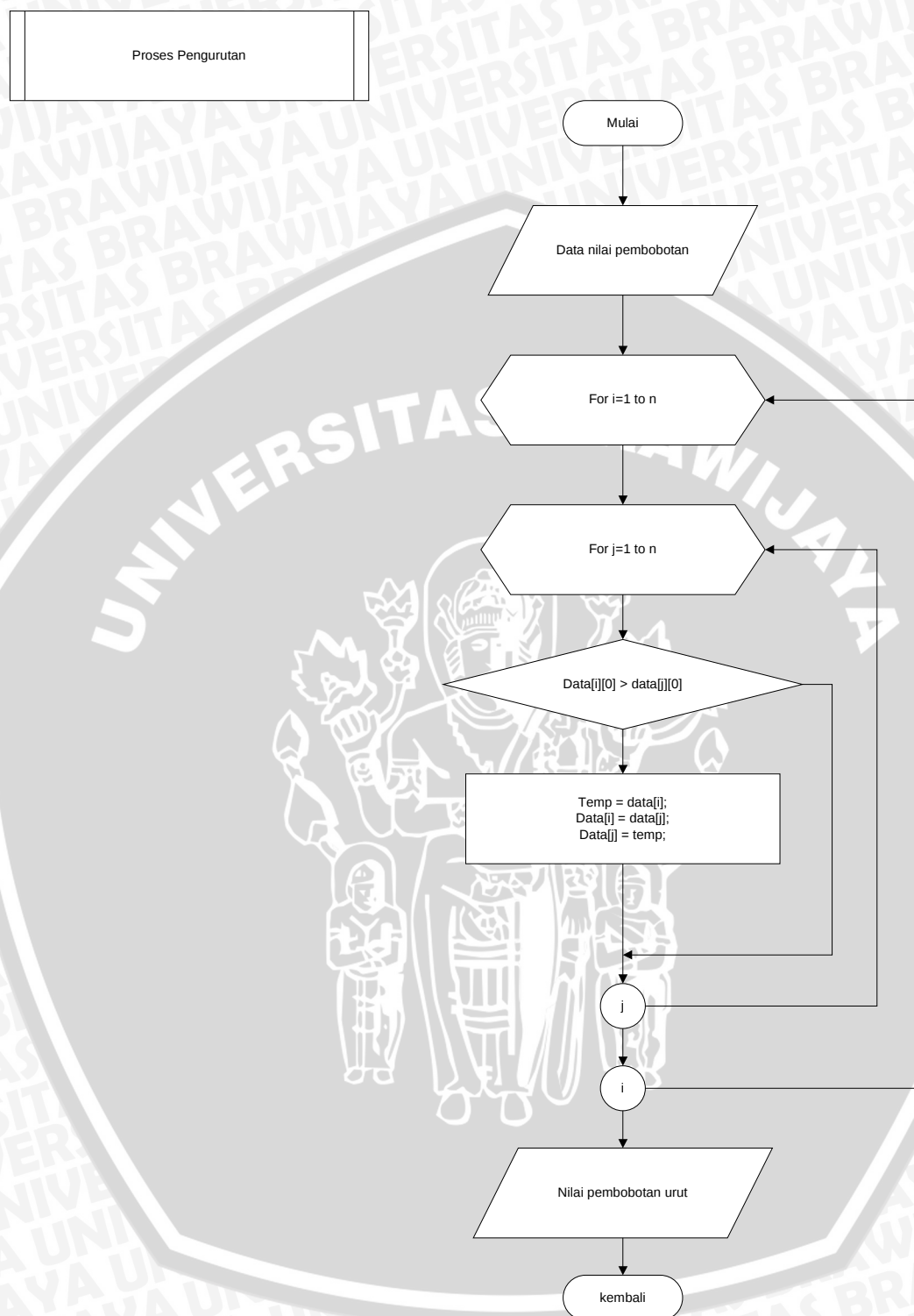
Gambar 3.10 Alur Proses Euclidean Distance



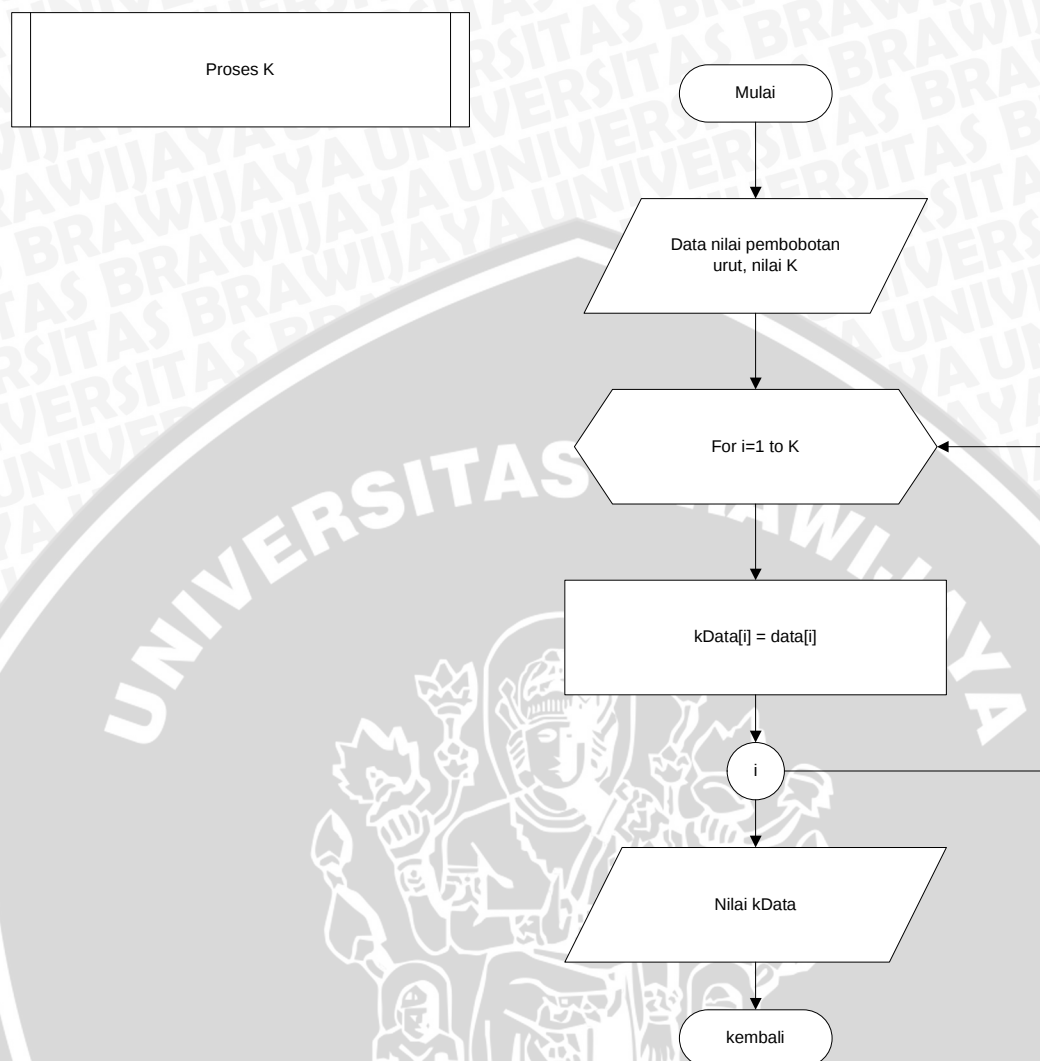
Gambar 3.11 Alur Proses Pembobotan

3.3.2.2.5. Proses Pembobotan sejumlah K

Proses ini untuk mengambil nilai pembobotan dari setiap nilai euclidean sejumlah K. Flowchart untuk proses Pembobotan sejumlah K ditampilkan oleh Gambar 3.13.



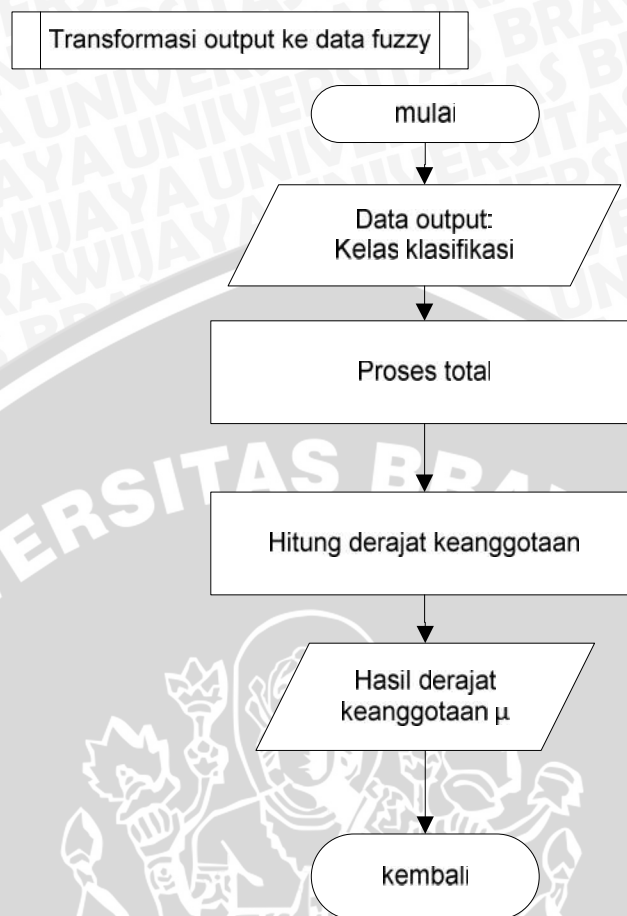
Gambar 3.12 Alur Proses Pengurutan Nilai Bobot



Gambar 3.13 Alur Proses Pembobotan sejumlah K

3.3.2.3. Transformasi ke data fuzzy

Alur proses transformasi ke data fuzzy dapat dilihat pada gambar 3.14.

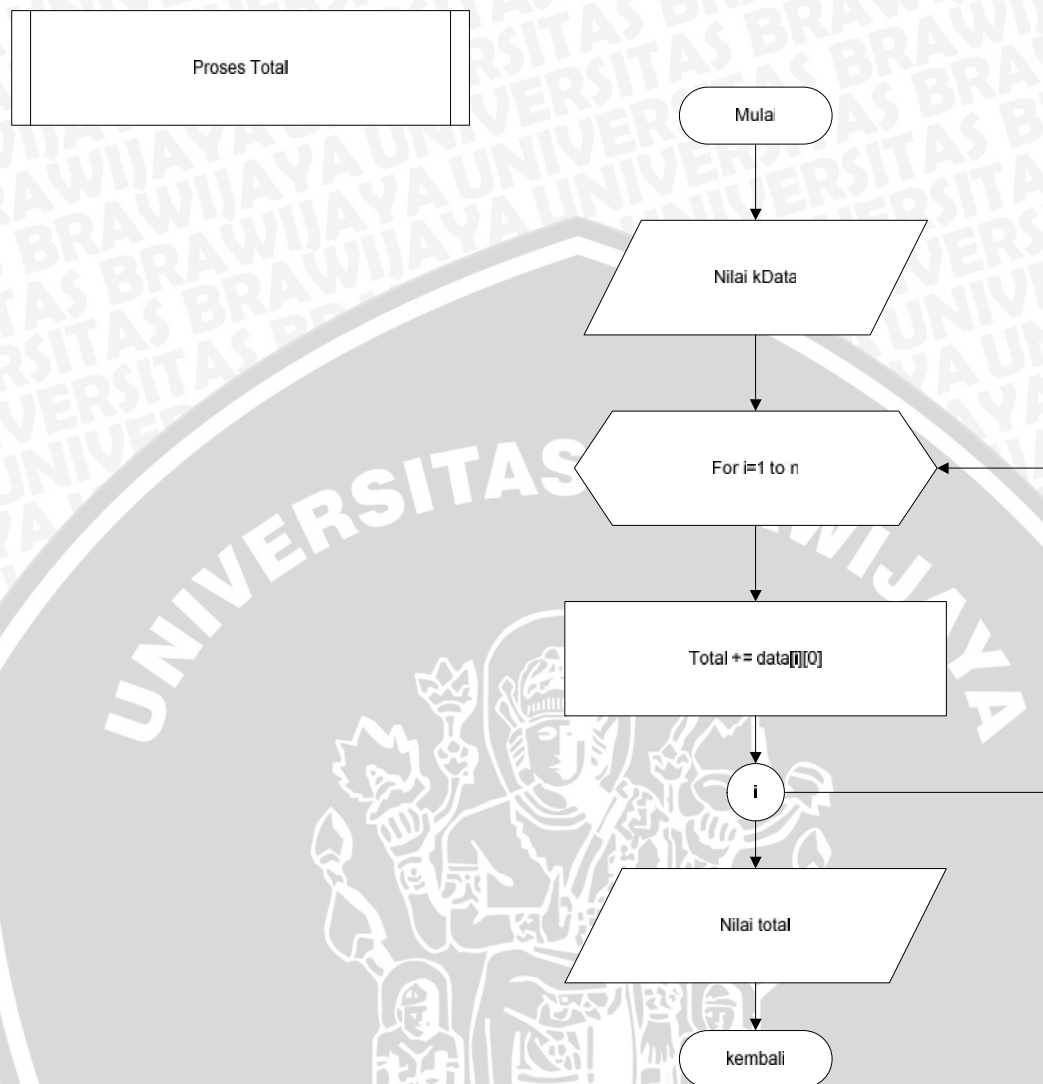


Gambar 3.14 Alur Proses Transformasi Ke Data Fuzzy

Proses ini mengubah nilai kuantitatif pada setiap atribut dalam dataset menjadi derajat keanggotaan fuzzy beserta linguistiknya berdasarkan fungsi keanggotaan atribut.

3.3.2.3.1. Proses Total

Proses ini untuk mengambil nilai pembobotan dari setiap nilai euclidean sejumlah K lalu dijumlahkan. Flowchart untuk proses menjumlahkan Pembobotan sejumlah K ditampilkan oleh Gambar 3.15.



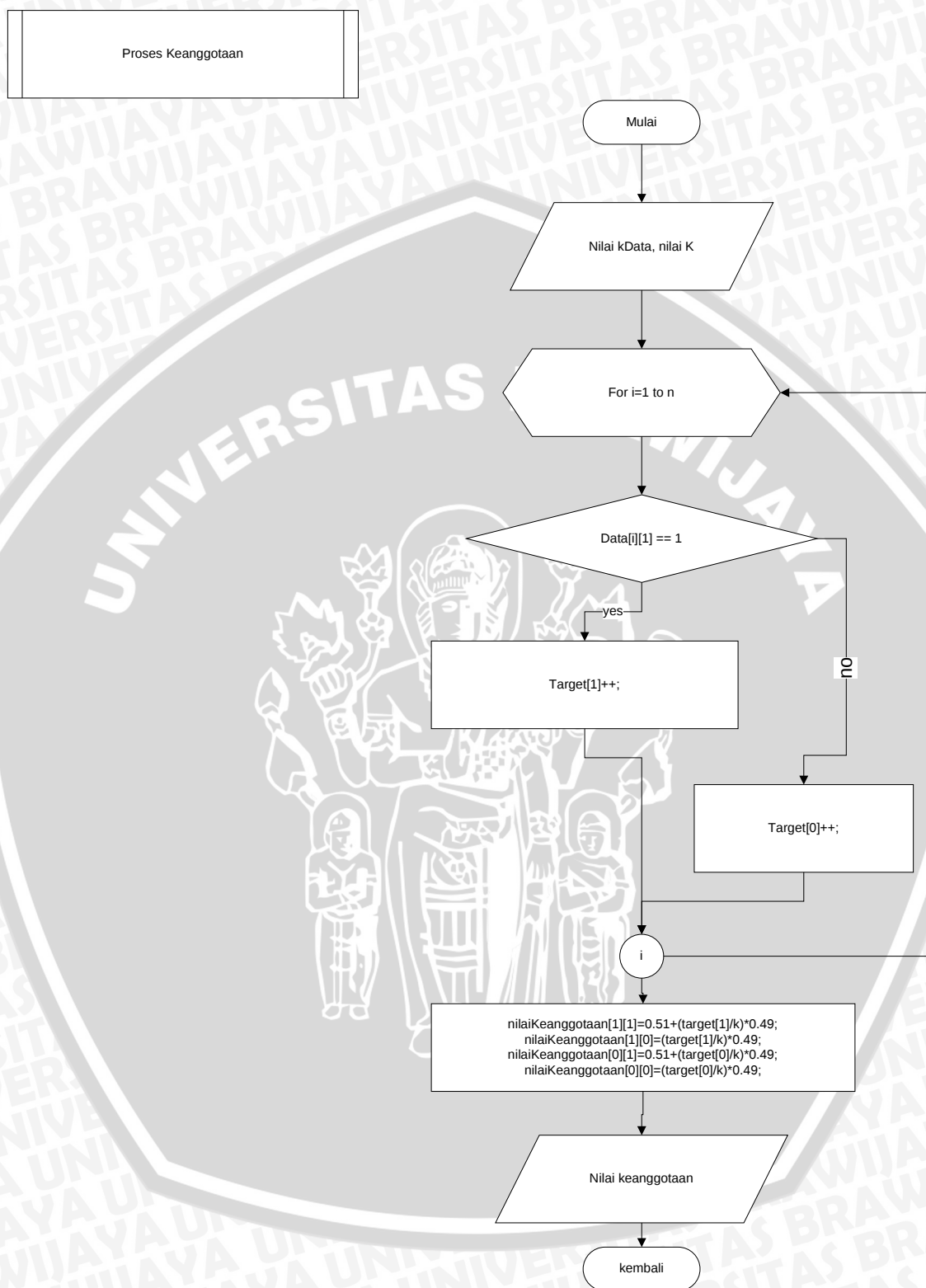
Gambar 3.15 Alur Proses Penjumlahan Bobot sejumlah K

3.3.2.3.2. Proses Hitung Derajat Keanggotaan

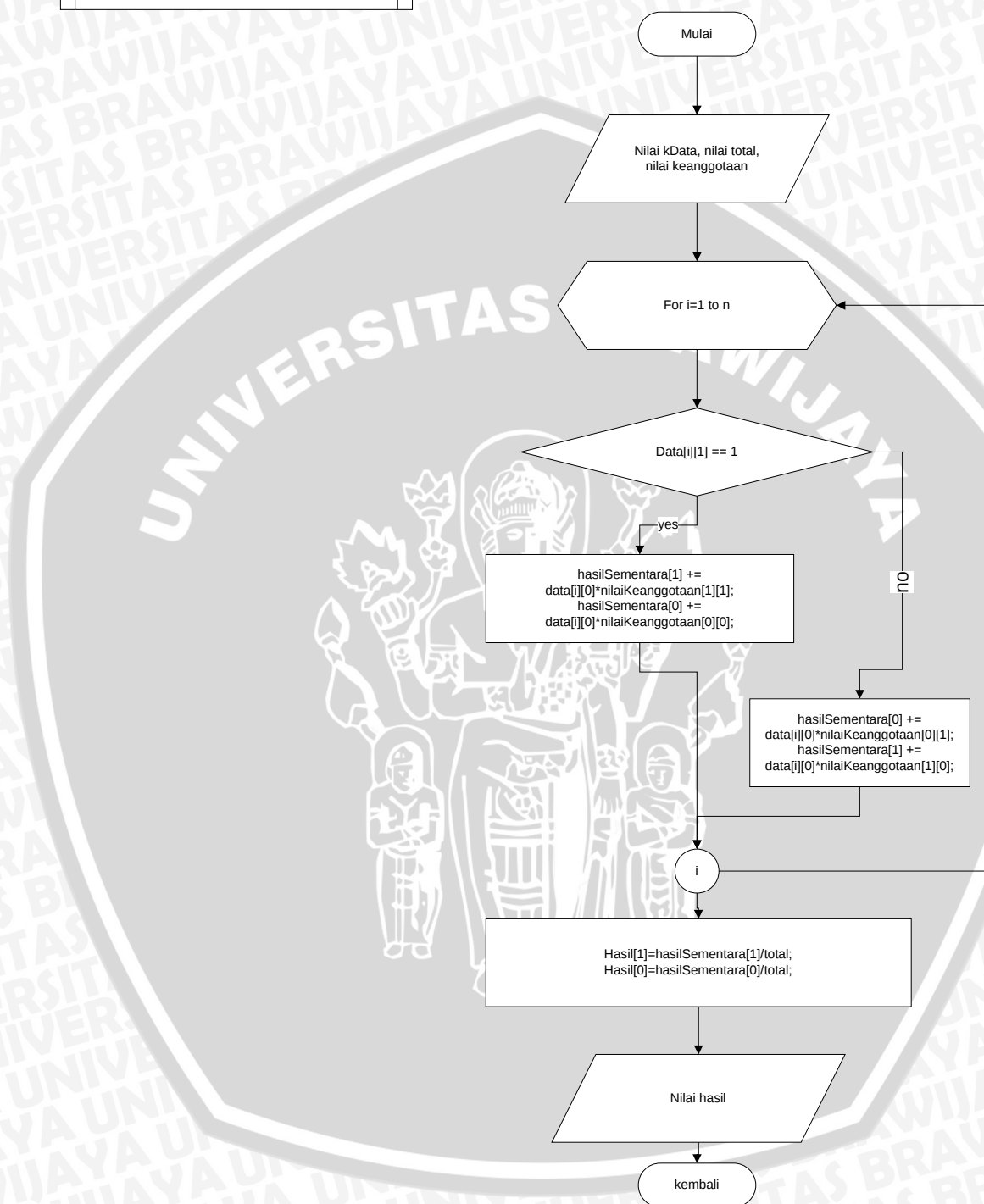
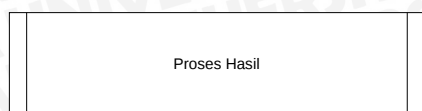
Proses ini untuk menghitung derajat keanggotaan. Flowchart untuk proses menghitung derajat keanggotaan ditampilkan oleh Gambar 3.16.

3.3.2.3.3. Proses Hitung Hasil

Proses ini untuk menghitung hasil FK-NN. Flowchart untuk proses menghitung hasil FK-NN ditampilkan oleh Gambar 3.17.



Gambar 3.16 Alur Proses Menghitung Derajat Keanggotaan



Gambar 3.17 Alur Proses Menghitung Hasil FK-NN

3.4. Perhitungan Manual

Pada subbab 3.4 akan menampilkan contoh perhitungan manual untuk proses klasifikasi jenis mangga gadung dan curut berdasarkan tekstur daun dengan metode *Fuzzy K-Nearest Neighbor* (FK-NN) menggunakan 10 data pelatihan yang tercantum dalam tabel 3.1 dan data uji tercantum dalam tabel 3.2.

Tabel 3.1 Data input pelatihan sistem

No.	Fitur										Target
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	
1	34.4043	0.0144	4.2666	5.4251	11.3855	23.2463	29.6159	46.1905	0.1704	86.0139	0
2	27.2064	0.0103	3.7283	5.3438	11.2981	22.0147	29.7404	43.1424	0.205	68.9388	0
3	26.4059	0.0102	3.664	5.3492	11.3047	21.3572	27.5919	42.5218	0.2204	57.0943	0
4	27.9246	0.01	3.9025	5.3477	11.3233	21.2033	27.3976	42.338	0.185	63.55	0
5	30.2173	0.0111	4.0013	5.4143	11.4539	21.3308	27.4956	42.5375	0.1755	59.8338	0
6	22.9308	0.0104	3.5174	5.0157	10.2957	20.7524	26.0901	42.9884	0.2885	46.912	1
7	34.0326	0.0171	3.7517	5.4777	11.3749	22.7801	30.1354	44.5894	0.229	56.1978	1
8	30.1586	0.0165	3.2758	5.4947	11.4474	21.7318	27.5095	45.766	0.2855	48.5624	1
9	33.8422	0.021	3.3648	5.7678	12.1696	22.5781	28.8225	46.1383	0.2895	57.2624	1
10	30.6568	0.0159	3.5851	5.6038	11.87	23.9468	30.0445	47.65	0.2584	68.567	1

Keterangan kolom :

RTA : Rata-rata intensitas

ENT : Entropy

KON : Kontras

M2 : Moment 2

M6 : Moment 6

SMO : Smoothness

ENE : Energy

M1 : Moment 1

M3 : Moment 4

M7 : Moment 7

Dari 10 data tersebut, memiliki 2 buah kelas sesuai dengan nilai target output yang mana merupakan jenis mangga gadung dan mangga curut. Dengan representasi nilai 0 sebagai mangga curut dan 1 sebagai mangga gadung.

Tabel 3.2 Data input uji sistem

No.	Fitur										Target
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	
1	21.8585	0.0101	3.4356	5.0127	10.2944	20.7491	26.1038	41.6225	0.3058	45.4907	?

Langkah-langkah perhitungan untuk algoritma *K-Nearest Neighbor Classification* adalah sebagai berikut:

- Langkah 1, buat tabel yang berisi data-data yang akan diklasifikasikan, seperti pada tabel 3.3.

Tabel 3.3 Data yang akan dilakukan klasifikasi

No.	Fitur										Target
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	
1	34.4043	0.0144	4.2666	5.4251	11.3855	23.2463	29.6159	46.1905	0.1704	86.0139	0
2	27.2064	0.0103	3.7283	5.3438	11.2981	22.0147	29.7404	43.1424	0.205	68.9388	0
3	26.4059	0.0102	3.664	5.3492	11.3047	21.3572	27.5919	42.5218	0.2204	57.0943	0
4	27.9246	0.01	3.9025	5.3477	11.3233	21.2033	27.3976	42.338	0.185	63.55	0
5	30.2173	0.0111	4.0013	5.4143	11.4539	21.3308	27.4956	42.5375	0.1755	59.8338	0
6	22.9308	0.0104	3.5174	5.0157	10.2957	20.7524	26.0901	42.9884	0.2885	46.912	1
7	34.0326	0.0171	3.7517	5.4777	11.3749	22.7801	30.1354	44.5894	0.229	56.1978	1
8	30.1586	0.0165	3.2758	5.4947	11.4474	21.7318	27.5095	45.766	0.2855	48.5624	1
9	33.8422	0.021	3.3648	5.7678	12.1696	22.5781	28.8225	46.1383	0.2895	57.2624	1
10	30.6568	0.0159	3.5851	5.6038	11.87	23.9468	30.0445	47.65	0.2584	68.567	1

- Kemudian dilakukan normalisasi untuk setiap atribut, dimana untuk menghitung hasil normalisasi digunakan persamaan 2.24. Nilai range didapat dari nilai max data training dikurangi nilai min data training pada tabel 3.4. Dimana data training terdiri dari 60 record. Berikut ini adalah contoh, perhitungan normalisasi nilai record pertama untuk atribut RTA.

$V = 34,4043$ (nilai record pertama untuk atribut RTA)

$\min [j] = 21,8585$

$\max [j] = 37,027$

Setelah diketahui nilai maksimum dan minimum, kemudian dilakukan perhitungan menggunakan persamaan 2.24.

$$\begin{aligned} V' &= \frac{(V - \min [j])}{\text{range } [j]} \\ &= \frac{(34,4043 - 21,8585)}{(37,027 - 21,8585)} \\ &= 0.8270956258 \end{aligned}$$

Dari perhitungan ini diperoleh nilai normalisasi untuk record pertama pada atribut RTA, yaitu 0.8270956258. Proses yang sama juga dilakukan untuk nilai record pada seluruh atribut. Hasil perhitungan normalisasi ditampilkan pada tabel 3.5 dan tabel 3.6.

Tabel 3.4 Range Data Training

	Fitur									
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON
Max	37,027	0,023	4,2666	5,8222	12,3151	24,5504	33,9108	48,8011	0,3465	86,0139
Min	21,8585	0,01	3,1779	5,0127	10,2944	19,8087	25,4705	40,7678	0,1573	37,6108
Range	15,1685	0,013	1,0887	0,8095	2,0207	4,7417	8,4403	8,0333	0,1892	48,4031

Tabel 3.5 Data Pelatihan yang telah di normalisasi

No.	Fitur									
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON
1	0.8270956258	0.3384615385	1	0.5094502779	0.5399613995	0.7249720564	0.4911436797	0.6750276972	0.0692389006	1
2	0.3525661733	0.0230769231	0.5055570864	0.4090179123	0.4967090612	0.4652339878	0.5058943402	0.2955945875	0.2521141649	0.6472312724
3	0.2997923328	0.0153846154	0.4464958207	0.4156886967	0.4999752561	0.3265706392	0.251341777	0.2183411549	0.3335095137	0.4025258713

4	0.3999142961	0	0.6655644346	0.4138357011	0.5091799871	0.2941139254	0.2283212682	0.195461392	0.1464059197	0.5358995602
5	0.5510630583	0.0846153846	0.7563148709	0.4961087091	0.5738110556	0.3210030158	0.2399322299	0.2202955199	0.0961945032	0.4591234859
6	0.0706925536	0.0307692308	0.3118398089	0.0037059914	0.0006433414	0.199021448	0.0734097129	0.2764243835	0.6934460888	0.1921612459
7	0.8025908956	0.5461538462	0.5270506108	0.5744286597	0.5347156926	0.6266528882	0.5526936246	0.4757198163	0.3789640592	0.3840043303
8	0.5471931964	0.5	0.0899237623	0.5954292773	0.5705943485	0.4055718413	0.2415790908	0.6221851543	0.677589852	0.2262582355
9	0.7900385668	0.8461538462	0.1716726371	0.9327980235	0.9279952492	0.5840521332	0.3971422817	0.6685297449	0.6987315011	0.4059987893
10	0.5800375779	0.4538461538	0.3740240654	0.7302038295	0.7797297966	0.8727038826	0.5419238653	0.856708949	0.5343551797	0.6395499462

Tabel 3.6 Data Uji yang telah di normalisasi

No.	Fitur									
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON
1	0	0.0076923077	0.2367043263	0	0	0.1983254951	0.075032878	0.1063946323	0.7848837209	0.1627974241

- c. Kemudian, di cari terlebih dahulu jarak terdekat “*Euclidean distance*” dari data uji dengan data pelatihan. Dimana rumus untuk *Euclidean distance*, dapat dilihat pada persamaan (2-25). Berikut ini adalah contoh, perhitungan nilai jarak terdekat dari *record* pertama data latih dengan record uji.

$$d(x_1, x_2) = \sqrt{\sum_{r=1}^n (a_r(x_1) - a_r(x_2))^2}$$

$$\begin{aligned}
 d &= \sqrt{(0.8270956258 - 0)^2 + (0.3384615385 - 0.0076923077)^2 + (1 - 0.2367043263)^2 +} \\
 &\quad (0.5094502779 - 0)^2 + (0.5399613995 - 0)^2 + (0.7249720564 - 0.1983254951)^2 + \\
 &\quad (0.4911436797 - 0.075032878)^2 + (0.6750276972 - 0.1063946323)^2 + \\
 &\quad (0.0692389006 - 0.7848837209)^2 + (1 - 0.1627974241)^2} \\
 &= \sqrt{0.68408717421749362564 + 0.10940828404402366864 + 0.58262028548913687169 +} \\
 &\quad 0.25953958565238722841 + 0.29155831294999860025 + 0.27735660052911465769 + \\
 &\quad 0.17314819929141672289 + 0.32334356249756761201 + 0.51214750882221929209 +} \\
 &\quad 0.70090815309359526081} \\
 &= \sqrt{3.91411766658695354012} \\
 d &= 1.9784129161291
 \end{aligned}$$

Selanjutnya dihitung lagi jarak antara *record* uji dengan *record* data latih yang lain. Untuk hasil perhitungan nilai dari jarak terdekat dapat dilihat pada tabel 3.7.

Tabel 3.7 Nilai jarak terdekat “Euclidean distance” data uji dengan data pelatihan

No.	Nilai	Target
1	1.9784129161291	0
2	1.1924910326881	0
3	0.93701201000849	0
4	1.1670049714416	0
5	1.3312710869497	0
6	0.22209408013514	1
7	1.5479055524183	1
8	1.2632155506405	1
9	1.9229663661318	1
10	1.7948070202315	1

- d. Selanjutnya dari data yang telah ditemukan nilai *Euclidean distance* dilakukan pencarian nilai *weight* (*w*) dengan menggunakan persamaan (2-26). Berikut ini adalah contoh, perhitungan nilai *weight* dari hasil *record* pertama perhitungan *Euclidean distance*.

$$\text{weight} = \frac{1}{1.9784129161291^2} = 0.25548542101993$$

Untuk hasil perhitungan nilai *weight* dari hasil *Euclidean Distance* untuk seluruh *record* dapat dilihat pada tabel 3.8.

Tabel 3.8 Nilai weighted

No.	Nilai	Target
1	0.25548542101993	0
2	0.70321763972873	0
3	1.138963181467	0
4	0.73426797633096	0
5	0.56424406562201	0
6	20.273374111077	1
7	0.41736024997991	1
8	0.6266789012198	1
9	0.27043109306314	1
10	0.31043056894982	1

- e. Pengurutan nilai pembobotan ke dalam semua kelas dalam metode *Fuzzy K-Nearest Neighbor* (FK-NN)

Tabel 3.9 Data setelah sorting dari yang terbesar

No.	Nilai	Target
1	20.273374111077	1
2	1.138963181467	0
3	0.73426797633096	0
4	0.70321763972873	0
5	0.6266789012198	1
6	0.56424406562201	0
7	0.41736024997991	1
8	0.31043056894982	1
9	0.27043109306314	1
10	0.25548542101993	0

- f. Misal, pada saat awal di inputkan nilai $k=4$, maka pada data setelah mengalami proses pemecahan nilai pembobotan data tersebut di urutkan sesuai dengan nilai yang paling terbesar. Kemudian $k=4$ dimaksudkan, di ambil 4 nilai data yang terbesar. Data hasil sorting dilihat pada tabel 3.9 kemudian data setelah penentuan $k=4$ dapat dilihat pada tabel 3.10.

Tabel 3.10 Seleksi 4 data dengan nilai terbesar

No.	Nilai	Target
1	20.273374111077	1
2	1.138963181467	0
3	0.73426797633096	0
4	0.70321763972873	0

- g. Selanjutnya dari data yang telah ditemukan nilai K dilakukan pencarian nilai membership kelas pada tiap neighbor dengan menggunakan persamaan (2-27). Berikut ini adalah contoh, perhitungan nilai membership neighbor.

$$\begin{aligned}
 u_{ij}(x) &= 0.51 + \left(\frac{1}{4}\right) \cdot 0.49 \\
 &= 0.6325
 \end{aligned}$$

Untuk hasil perhitungan nilai membership neighbor untuk seluruh record dapat dilihat pada tabel 3.11.

Tabel 3.11 Perhitungan nilai membership neighbor

Target	Sama	Tidak Sama
1	0.6325	0.1225
0	0.8775	0.3675

- h. Masukan tersebut dicari dengan nilai $k=4$ dan didapat hasil 0.57499457022077 jika dipersentase menjadi 57.5 % dengan target mangga gadung representasi dari nilai 1.

Tabel 3.12 Hasil

Hasil	Target
0.57499457022077	1
0.42500542977923	0

3.5. Perancangan Antarmuka

Antarmuka (*interface*) untuk sistem dibuat sebanyak 3 halaman, yaitu halaman untuk data, perhitungan metode *Fuzzy K-Nearest Neighbor* (FK-NN), dan grafik akurasi ditunjukkan pada gambar 3.18, 3.19, 3.20. Dalam perancangan antarmuka ini terdapat 3 fungsi tab dengan skenario mengunggah data uji dan data latih selanjutnya membuka tab perhitungan untuk melihat hasilnya dan seterusnya sehingga akhirnya didapatkan hasil grafik di tab grafik.

The interface consists of three tabs: **Data** (1), **Perhitungan FKNN** (2), and **Grafik** (3). The **Data** tab is selected and contains two main sections:

- Data Uji** (4): A table with two columns labeled **Tabel** (5) and **Cell Content 1** / **Cell content 2**. Below the table is a **Pilih file** button (6) and a **Submit** button (7).
- Data Latih** (8): A table with two columns labeled **Tabel** (9) and **Cell Content 1** / **Cell content 2**. Below the table is a **Pilih file** button (10) and a **Submit** button (11).

Gambar 3.18 Antarmuka sistem untuk data

Pada gambar 3.18, antarmuka sistem untuk data terdiri dari :

1. Tombol Tab data untuk memasukkan data uji, dan data latih.
2. Tombol Tab perhitungan FK-NN untuk menampilkan perhitungan dari proses *Fuzzy K-Nearest Neighbor* (FK-NN).
3. Tombol Tab grafik untuk menampilkan grafik akurasi metode yang digunakan dalam kasus ini menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN).
4. *Label* data uji untuk judul tabel.
5. Tabel data uji untuk menampilkan detail fitur dan target data uji.
6. Tombol pilih file untuk mengambil data uji yang digunakan pada proses.
7. Tombol *submit* untuk memproses semua data uji yang telah dimasukkan
8. *Label* data latih untuk judul tabel.
9. Tabel data uji untuk menampilkan detail fitur dan target data latih.
10. Tombol pilih file untuk mengambil data latih yang digunakan pada proses.
11. Tombol submit untuk memproses semua data latih yang telah dimasukkan.

Tabel	Tabel
Cell Content 1	Cell Content 1
Cell content 2	Cell content 2

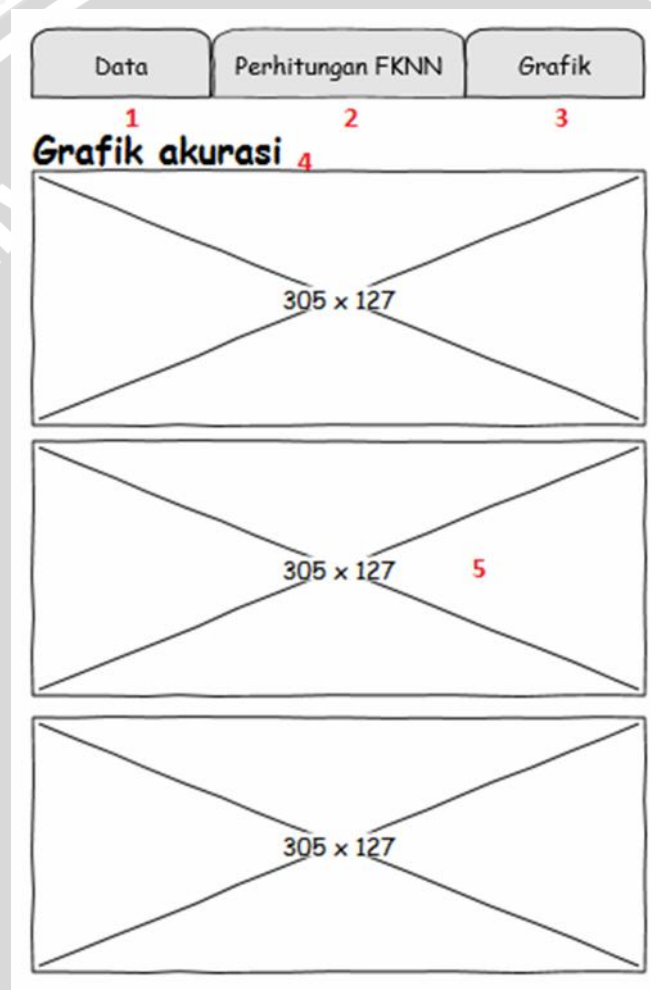
Rata-rata akurasi :

Gambar 3.19 Antarmuka sistem untuk perhitungan FK-NN

Pada gambar 3.19, antarmuka sistem untuk perhitungan FKNN terdiri dari :

1. Tombol Tab data untuk memasukkan data uji, dan data latih.
2. Tombol Tab perhitungan FK-NN untuk menampilkan perhitungan dari proses Fuzzy K-Nearest Neighbor (FK-NN).

3. Tombol Tab grafik untuk menampilkan grafik akurasi metode yang digunakan dalam kasus ini menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN).
4. Label hasil perhitungan untuk judul tabel dengan isi kolom nilai K, benar, salah, dan akurasi.
5. Tabel hasil perhitungan berisi kolom nilai K, benar, salah, dan akurasi.
6. Rata-rata akurasi dengan nilai K sampai 10.



Gambar 3.20 Antarmuka sistem untuk grafik akurasi

Pada gambar 3.20, antarmuka sistem untuk grafik akurasi terdiri dari :

1. Tombol Tab data untuk memasukkan data uji, dan data latih.
2. Tombol Tab perhitungan FK-NN untuk menampilkan perhitungan dari proses *Fuzzy K-Nearest Neighbor* (FK-NN).

3. Tombol Tab grafik untuk menampilkan grafik akurasi metode yang digunakan dalam kasus ini menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN).
4. Label grafik akurasi setiap K-fold
5. Gambar yang menunjukkan grafik perbandingan akurasi tiap K-fold.

3.6. Perancangan Uji Coba

Setelah sistem selesai dibuat, langkah selanjutnya adalah melakukan pengujian terhadap sistem tersebut. Pengujian dilakukan untuk mengetahui tingkat akurasi dari hasil klasifikasi jenis mangga berdasarkan tekstur daun menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN). Pengujian tersebut dilakukan terhadap dataset klasifikasi jenis mangga berdasarkan tekstur daun yang terdiri dari 2 kelas klasifikasi, yaitu mangga gadung dan mangga curut.

Pengujian untuk mengetahui pengaruh k-fold cross validation terhadap tingkat akurasi klasifikasi. Dimana dalam pengujian ini akan dilakukan 4 kali percobaan . Untuk jumlah 4-fold cross validation, yaitu 4, 5, 6, dan 10, akan dilakukan 10 kali percobaan dengan nilai k mulai 1 sampai 10. Dari nilai tersebut akan diambil masing-masing nilai akurasi, dan kemudian nilai setiap akurasi tersebut akan dihitung nilai rata-ratanya.

Uji pengaruh k-fold cross validation terhadap tingkat akurasi klasifikasi dilakukan pada dataset klasifikasi jenis mangga berdasarkan tekstur daun. Pada pengujian ini di 4-fold cross validation, yaitu 4, 5, 6, dan 10, dilakukan dengan 10 kali percobaan untuk mendapatkan nilai akurasi k yang paling baik mulai 1 sampai 10. Kemudian hasil akurasi dari 10 percobaan tersebut dilakukan perhitungan rata-rata untuk mendapatkan akurasi rata-ratanya. Rancangan tabel yang akan digunakan untuk mencatat hasil dari pengujian ini ditampilkan oleh Tabel 3.13.

Tabel 3.13 Tabel Uji Pengaruh K-Fold Cross Validation Terhadap Akurasi

K-Fold Cross Validation	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
		Benar	Salah		
4	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
5	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
6	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
10	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				

Keterangan:

1. K-fold cross validation : 4-fold cross validation, yaitu 4, 5, 6, dan 10, akan dilakukan 10 kali percobaan dengan nilai k mulai 1 sampai 10.
2. Kolom nilai akurasi menunjukkan hasil akurasi dari 10 percobaan yang telah dilakukan mulai k sama dengan 1 sampai 10. Perhitungan tingkat keakurasian ini didapatkan.
3. Akurasi Rata-rata: tingkat akurasi rata-rata dalam persen dari beberapa kali pengujian dengan nilai k-fold cross validation yang berbeda.

K-Fold Cross Validation dilakukan untuk membagi data pelatihan dan data pengujian. K-Fold Cross Validation membagi data sample secara acak ke dalam K subset yang saling bebas. Satu subset sebagai data pengujian dan K-1 subset sebagai data pelatihan. Proses cross-validation akan diulang sampai K kali. Dalam penelitian ini terdapat total 60 record dengan dua kelas masing-masing 30 record, jika diterapkan metode K-fold cross validation ini maka total record akan dibagi sejumlah K. Contoh penelitian ini total jumlah data 60 record akan ditentukan $K = 4$ maka pengujian dilakukan sebanyak 4 kali dengan data uji dan data latih sebanyak 4 maka tidak ada data yang tumpuk sehingga semua akan jadi data uji dan data latih.

BAB IV

IMPLEMENTASI DAN UJI COBA SISTEM

Bab ini berisi implementasi sistem. Implementasi sistem merupakan penerapan dari rancangan sistem yang sudah dibahas pada bab sebelumnya.

4.1. Lingkungan Implementasi

Lingkungan implementasi yang dijelaskan dalam sub bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1. Lingkungan implementasi perangkat keras

Perangkat keras yang digunakan dalam penerapan metode Fuzzy K-Nearest Neighbor (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun ini adalah sebagai berikut :

1. Prosesor Intel Core™ I5 2.50 GHz
2. Memori 2.00 GB
3. Harddisk dengan kapasitas 512 GB
4. Monitor 14"
5. Keyboard
6. Mouse

4.1.2. Lingkungan implementasi perangkat lunak

Perangkat lunak yang digunakan dalam penerapan metode Fuzzy K-Nearest Neighbor (FK-NN) dalam klasifikasi jenis mangga berdasarkan tekstur daun ini adalah sebagai berikut :

1. Sistem operasi Microsoft Windows 7
2. XAMPP
3. Apache Web Server
4. PHP
5. HTML5

6. CSS3
7. JavaScript
8. Microsoft Office Excel 2007
9. Netbeans 7.1.1 IDE
10. Code Igniter Framework 2.1.2
11. MySQL
12. Internet Browser

4.2. Implementasi Program

Sesuai dengan pembahasan pada BAB III, proses pada sistem meliputi empat proses utama, yaitu konversi data file *.csv ke dalam database MySQL, normalisasi nilai atribut, klasifikasi, dan transformasi ke data fuzzy. Pada bab ini dibahas implementasi dari proses-proses tersebut.

4.2.1. Implementasi konversi file *.csv ke dalam database MySQL

Agar dapat digunakan pada proses perhitungan, maka data pada file *.csv diubah ke dalam database MySQL. Data pada tiap record file *.csv dipisahkan oleh tanda semicolon (titik koma). Proses tersebut diimplementasikan pada fungsi uji_input() yang ditunjukkan pada Sourcecode 4.1.

```

Skripsi.php
function uji_input() {
    if (!empty($_FILES['userfile']['name'])) {
        $config['upload_path'] = './asset/upload/file/';
        $config['allowed_types'] = 'csv';
        $config['overwrite'] = TRUE;
        $this->load->library('upload', $config);

        if ($this->upload->do_upload()) {
            $file_name = $this->upload->file_name;
            $this->upload->data();
        }
        $filename = null;
        $filename = $_SERVER['DOCUMENT_ROOT'] . "/asset/upload/file/" .
$file_name;

        $handle = fopen("$filename", "r");

        $this->Skripsi_model->empty_input();
        while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
            $this->Skripsi_model->uji_input($data);
        }
        fclose($handle);
        redirect('');
    }
    redirect('');
}

```

Sourcecode 4.1 Fungsi untuk konversi file *.csv ke dalam database MySQL

4.2.2. Implementasi normalisasi

Sebelum dilakukan klasifikasi terlebih dahulu dilakukan tranformasi terhadap data berupa normalisasi menggunakan metode min-max. Langkah awal dalam proses normalisasi ini adalah dicari nilai minimum dan nilai maksimum pada tiap atribut data latih dan data uji menggunakan fungsi proses_normalisasi(). Fungsi tersebut dapat dilihat pada Sourcecode 4.4.

Untuk menentukan nilai normalisasi harus melalui beberapa tahap algoritma min dan max. Dibawah ini merupakan fungsi proses_min() untuk mencari nilai minimal global dari semua data fitur yang diinputkan. Fungsi tersebut dapat dilihat pada Sourcecode 4.2.

```
Fknn_lib.php
function proses_min($data) {
    $xMin = null;
    for ($i = 0; $i < count($data[0]) - 1; $i++) {
        $minTemp = $data[0][$i];
        for ($j = 1; $j < count($data); $j++) {
            if ($data[$j][$i] < $minTemp) {
                $minTemp = $data[$j][$i];
            }
        }
        $xMin[$i] = $minTemp;
    }
    return $xMin;
}
```

Sourcecode 4.2 Fungsi untuk menentukan nilai minimum global

Setelah diketahui nilai minimal global maka ditentukan juga nilai maksimal global melalui fungsi algoritma dibawah ini dengan nama proses_max() untuk mencari nilai maksimal global dari semua data fitur yang diinputkan. Fungsi tersebut dapat dilihat pada Sourcecode 4.3.

```
Fknn_lib.php
function proses_max($data) {
    $xMax = null;
    for ($i = 0; $i < count($data[0]) - 1; $i++) {
        $maxTemp = 0;
        for ($j = 0; $j < count($data); $j++) {
            if ($data[$j][$i] > $maxTemp) {
                $maxTemp = $data[$j][$i];
            }
        }
        $xMax[$i] = $maxTemp;
    }
    return $xMax;
}
```

Sourcecode 4.3 Fungsi untuk menentukan nilai maksimum global

Setelah nilai maksimal global dan minimal global didapatkan selanjutnya akan dilakukan normalisasi tiap fitur data uji dan data latih. Proses tersebut diimplementasikan dengan fungsi proses_normalisasi() yang ditunjukkan pada Sourcecode 4.4.

Fknn_lib.php

```
function proses_normalisasi($data) {
    $dataNormalisasi = null;

    for ($i = 0; $i < count($data); $i++) {
        for ($j = 0; $j < count($data[0]) - 1; $j++) {
            $dataNormalisasi[$i][$j] = ($data[$i][$j] - $this->minGlobal[$j]) / ($this->maxGlobal[$j] - $this->minGlobal[$j]);
        }
    }
    return $dataNormalisasi;
}
```

Sourcecode 4.4 Fungsi untuk menormalisasi nilai atribut

4.2.3. Implementasi klasifikasi

Proses klasifikasi diawali dengan dihitung jarak antara data latih dan data uji menggunakan weighted Euclidean distance yang diimplementasikan pada fungsi proses_kuadrat(). Fungsi tersebut ditunjukkan pada Sourcecode 4.5. Jarak yang diperoleh kemudian diurutkan untuk menentukan prediksi kelasnya.

Fknn_lib.php

```
function proses_kuadrat($data) {
    $dataKuadrat = null;

    for ($i = 0; $i < count($data) - 1; $i++) {
        for ($j = 0; $j < count($data[0]); $j++) {
            $dataKuadrat[$i][$j] = pow($data[$i][$j] - $data[count($data) - 1][$j], 2);
        }
    }
    return $dataKuadrat;
}
```

Sourcecode 4.5 Fungsi untuk mencari jarak tiap atribut

Jarak euclidean yang telah diperoleh dari antara jarak data uji dan data latih tiap atribut kemudian dihitung secara keseluruhan dengan fungsi proses_euclidean(). Fungsi tersebut ditunjukkan pada Sourcecode 4.6.

Fknn_lib.php

```
function proses_euclidean($data) {
    $euclideanDistance = null;
    $hasil = null;

    for ($i = 0; $i < count($data); $i++) {
        for ($j = 0; $j < count($data[0]); $j++) {
            $hasil += $data[$i][$j];
        }
        $euclideanDistance[$i] = sqrt($hasil);
        $hasil = 0;
    }
    return $euclideanDistance;
}
```

Sourcecode 4.6 Fungsi untuk menghitung jarak keseluruhan atribut

Proses pembobotan ini dilakukan setelah jarak sebenarnya sudah didapatkan sebanyak data yang dilatihkan menggunakan fungsi proses_pembobotan(). Fungsi tersebut ditunjukkan pada Sourcecode 4.7.

Fknn_lib.php

```
function proses_pembobotan($data) {
    $dataPembobotan = null;
    $target = $this->dataLokal;

    for ($i = 0; $i < count($data); $i++) {
        $dataPembobotan[$i][0] = 1 / pow($data[$i], 2);
        $dataPembobotan[$i][1] = $target[$i][count($target[0]) - 1];
    }
    return $dataPembobotan;
}
```

Sourcecode 4.7 Fungsi untuk menghitung bobot jarak euclidean

Langkah selanjutnya dilakukan proses pengurutan nilai jarak yang sudah dilakukan pembobotan. Proses pengurutan dilakukan pada fungsi proses_pengurutan(). Fungsi tersebut ditunjukkan pada Sourcecode 4.8.

Fknn_lib.php

```
function proses_pengurutan($data) {
    $dataPembobotanUrut = null;

    for ($i = 0; $i < count($data); $i++) {
        for ($j = 0; $j < count($data); $j++) {
            if ($data[$i][0] > $data[$j][0]) {
                $temp = $data[$i];
                $data[$i] = $data[$j];
                $data[$j] = $temp;
            }
        }
    }
    $dataPembobotanUrut = $data;

    return $dataPembobotanUrut;
}
```

Sourcecode 4.8 Fungsi untuk mengurutkan bobot jarak euclidean

Proses algoritma selanjutnya adalah menentukan nilai K yang diinginkan tetapi dalam sistem yang dibuat ini akan otomatis memberikan nilai K mulai dari 1 sampai 10 yang akhirnya akan dilihat mana nilai K yang paling maksimal. Proses penentuan nilai K dilakukan pada fungsi proses_k(). Fungsi tersebut ditunjukkan pada Sourcecode 4.9.

```
Fknn_lib.php
function proses_k($data, $k) {
    $kData = null;

    for ($i = 0; $i < $k; $i++) {
        $kData[$i] = $data[$i];
    }
    return $kData;
}
```

Sourcecode 4.9 Fungsi untuk menentukan nilai k yang diinginkan

4.2.4. Implementasi transformasi ke data fuzzy

Proses total dibawah ini menjumlahkan bobot jarak euclidean yang sudah ditentukan nilai k. Fungsi tersebut ditunjukkan pada Sourcecode 4.10.

```
Fknn_lib.php
function proses_total($data) {
    $total = null;

    for ($i = 0; $i < count($data); $i++) {
        $total+=$data[$i][0];
    }
    return $total;
}
```

Sourcecode 4.10 Fungsi untuk menjumlahkan nilai bobot sebanyak k

Cara penentuan kelas keanggotaan dengan menggunakan rumus 2.8. Berikut adalah implementasinya dalam algoritma fungsi dengan nama proses_keanggotaan(). Fungsi tersebut ditunjukkan pada Sourcecode 4.11.

```
Fknn_lib.php
function proses_keanggotaan($data, $k) {
    $target[0] = null;
    $target[1] = null;

    for ($i = 0; $i < count($data); $i++) {
        if ($data[$i][1] == 1) {
            $target[1]++;
        } else {
            $target[0]++;
        }
    }
    $nilaiKeanggotaan = null;

    $nilaiKeanggotaan[1][1] = 0.51 + ($target[1] / $k) * 0.49;
    $nilaiKeanggotaan[1][0] = ($target[1] / $k) * 0.49;
    $nilaiKeanggotaan[0][1] = 0.51 + ($target[0] / $k) * 0.49;
}
```

```

$nilaiKeanggotaan[0][0] = ($target[0] / $k) * 0.49;

return $nilaiKeanggotaan;
}

```

Sourcecode 4.11 Fungsi untuk mendapatkan nilai membership fuzzy

Langkah terakhir untuk menentukan nilai hasil dari proses klasifikasi menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN) yaitu menjumlahkan semua nilai target dan akhirnya didapatkan hasil persentase tiap kelas klasifikasi. Berikut adalah implementasinya dalam algoritma fungsi dengan nama proses_hasil(). Fungsi tersebut ditunjukkan pada Sourcecode 4.12.

```

Fknn_lib.php

function proses_hasil($data, $total, $nilaiKeanggotaan) {
    $hasilSementara[0] = null;
    $hasilSementara[1] = null;

    for ($i = 0; $i < count($data); $i++) {
        if ($data[$i][1] == 1) {
            $hasilSementara[1] += $data[$i][0] * $nilaiKeanggotaan[1][1];
            $hasilSementara[0] += $data[$i][0] * $nilaiKeanggotaan[0][0];
        } else {
            $hasilSementara[0] += $data[$i][0] * $nilaiKeanggotaan[0][1];
            $hasilSementara[1] += $data[$i][0] * $nilaiKeanggotaan[1][0];
        }
    }
    $hasil[1] = $hasilSementara[1] / $total;
    $hasil[0] = $hasilSementara[0] / $total;

    return $hasil;
}

```

Sourcecode 4.12 Fungsi untuk mendapatkan hasil akhir

4.3. Implementasi Antarmuka

Sistem yang dibuat memiliki antarmuka dengan sebuah halaman yang terbagi menjadi tiga buah tab, yaitu data, perhitungan FKNN, dan grafik. Tab data terdiri dari dua groupbox, yaitu data uji dan data latih.

Pada groupbox data uji, terdapat tombol choose file untuk memilih data uji yang akan digunakan. Data uji dapat ditampilkan pada tabel dengan menekan tombol submit. Data uji yang akan diuji dalam sistem aplikasi ini sebanyak 25 data uji, dengan rincian 4 fold, 5 fold, 6 fold, 10 fold jadi total sebanyak 25 data uji. Tampilan gambaran antarmuka bisa dilihat pada gambar 4.1.

Data	FKNN	Grafik
------	------	--------

Data Uji

#	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	HASIL
1	22.9308	0.0104	3.5174	5.0157	10.2957	20.7524	26.0901	42.9884	0.2885	46.912	1
2	26.7442	0.0126	3.512	5.3106	11.0406	19.8087	25.4705	40.7678	0.2681	45.6608	0
3	21.8585	0.0101	3.4356	5.0127	10.2944	20.7491	26.1038	41.6225	0.3058	45.4907	1
4	31.7645	0.0191	3.2132	5.4151	11.1654	20.9969	27.427	41.6468	0.2994	55.6361	0
5	25.9265	0.0131	3.5267	5.2557	10.8722	22.2616	28.5474	44.76	0.2885	65.813	1
6	29.8966	0.013	3.6046	5.3714	11.2581	21.0658	26.8844	43.0076	0.2242	46.8415	0
7	24.2602	0.0129	3.2604	5.3013	10.9709	23.4557	30.0072	46.1636	0.3267	42.6226	1
8	29.186	0.0123	3.6001	5.3612	11.2564	21.1341	26.9547	43.0248	0.2223	45.285	0
9	22.1514	0.011	3.3673	5.1437	10.6013	21.5305	27.1572	42.7366	0.3252	37.6108	1
10	30.0384	0.0125	3.5232	5.3783	11.3044	20.3286	26.0532	41.9308	0.2153	43.3852	0
11	26.4007	0.0147	3.5961	5.3119	10.9793	22.6928	28.7302	44.9686	0.3038	66.0016	1
12	29.9325	0.0124	3.5354	5.397	11.3635	20.4353	26.1743	42.2122	0.2161	40.9603	0
13	30.7381	0.0185	3.5061	5.4946	11.3941	20.9706	26.7292	47.4832	0.2978	75.9769	1
14	31.1821	0.0133	3.8381	5.321	11.0843	21.513	27.3052	43.0113	0.2073	64.5236	0
15	27.9176	0.018	3.2303	5.4865	11.374	21.1735	26.9448	45.7487	0.3465	73.9455	1

No file chosen

Data Latih

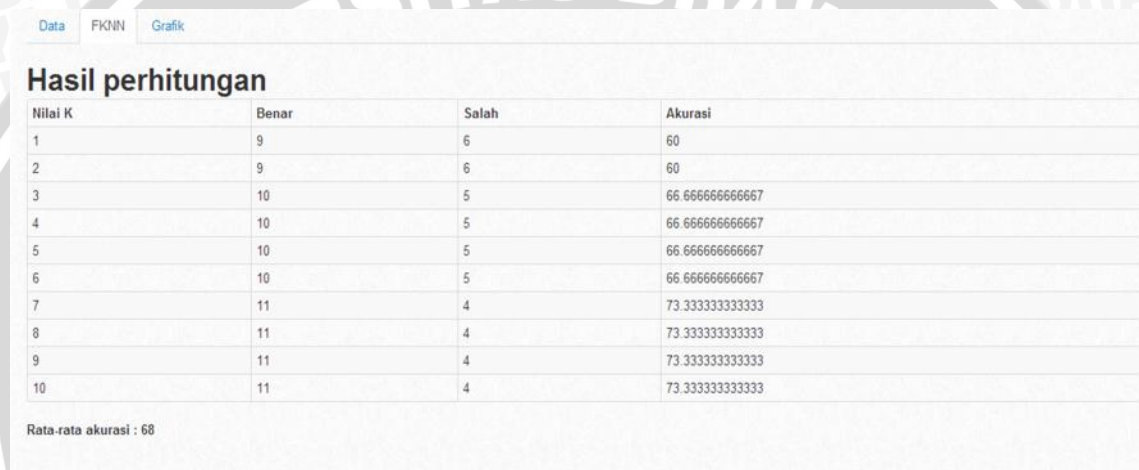
#	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	HASIL
1	31.8636	0.014	3.8424	5.364	11.1819	21.7538	27.5027	43.6303	0.2093	63.1219	0
2	32.6357	0.0161	3.6921	5.4356	11.2893	22.2441	33.9108	43.6082	0.2339	53.6529	1
3	26.7316	0.0114	3.603	5.3124	11.1195	23.7571	29.5838	48.2862	0.2418	58.346	0
4	34.0326	0.0171	3.7517	5.4777	11.3749	22.7801	30.1354	44.5894	0.229	56.1978	1
5	28.7469	0.0134	3.6849	5.3928	11.2746	23.44	31.7583	48.8011	0.2464	61.4687	0
6	31.9312	0.0154	3.6765	5.4126	11.2388	22.9342	29.1025	45.0863	0.2329	55.4962	1
7	29.3415	0.0132	3.6274	5.3961	11.3008	23.5487	32.2958	45.7782	0.2352	61.8965	0
8	31.3799	0.0156	3.6244	5.4179	11.2478	23.0439	28.8683	47.9436	0.2452	52.4196	1
9	28.3529	0.0132	3.5706	5.4094	11.3367	24.3451	30.6306	48.1419	0.2511	62.9698	0
10	27.9487	0.0149	3.1779	5.4533	11.3738	21.1046	26.7936	44.814	0.2987	47.1112	1
11	32.8221	0.0156	3.7852	5.5173	11.5596	21.6605	27.5898	44.0239	0.2217	68.3875	0
12	29.704	0.0156	3.4396	5.4471	11.373	22.0218	27.7131	46.1071	0.2722	73.2776	1
13	33.4647	0.0153	3.8466	5.5244	11.5918	21.7392	27.6525	44.2706	0.2074	65.4779	0
14	30.1586	0.0165	3.2758	5.4947	11.4474	21.7318	27.5095	45.766	0.2855	48.5624	1
15	33.8383	0.0156	3.8677	5.5295	11.5959	21.681	27.6063	44.2342	0.2069	69.1944	0
16	30.3328	0.0159	3.4121	5.51	11.4994	21.7591	27.5463	46.4551	0.2706	53.0058	1
17	33.4208	0.0156	3.8452	5.5306	11.5986	21.753	27.6848	44.2779	0.2128	68.653	0
18	36.3493	0.0218	3.5343	5.8069	12.2852	22.8607	29.0412	47.7408	0.261	62.4773	1
19	32.5496	0.0158	3.5935	5.4015	11.1916	21.2392	27.249	42.8177	0.2321	72.1401	0
20	37.027	0.0222	3.5208	5.8172	12.3067	23.1994	29.4193	47.5247	0.2574	63.4462	1
21	31.3359	0.0138	3.6672	5.3463	11.0873	21.1085	27.0275	42.6598	0.2151	70.1101	0
22	35.5074	0.0223	3.3934	5.8222	12.3151	23.1075	29.3006	47.6737	0.2815	61.0906	1
23	33.0079	0.0158	3.6762	5.4128	11.2234	21.5474	27.697	43.1419	0.2238	75.2294	0
24	33.8422	0.021	3.3648	5.7678	12.1696	22.5781	28.8225	46.1383	0.2895	57.2624	1
25	33.0212	0.0153	3.6986	5.4074	11.2113	21.6648	27.785	43.4036	0.216	71.0647	0
26	30.3104	0.0166	3.4706	5.456	11.2992	21.5149	27.3216	44.2937	0.2805	52.0211	1
27	28.3778	0.0109	3.5533	5.2621	10.9682	20.3656	26.0351	41.417	0.2059	60.7486	0
28	29.7622	0.0167	3.3821	5.4524	11.2859	21.827	27.7229	44.4432	0.2923	52.3289	1
29	30.7768	0.0115	3.7104	5.3203	11.1108	21.3824	27.3508	42.5751	0.1816	70.368	0
30	31.405	0.0186	3.3846	5.5088	11.3972	22.4403	28.4759	45.2711	0.2937	56.0111	1
31	30.9377	0.0112	3.7929	5.3175	11.1123	21.6944	27.8371	42.8845	0.1737	70.0519	0
32	35.7292	0.0196	3.7924	5.5175	11.4263	22.6851	29.1374	45.213	0.2364	79.7554	1
33	31.5011	0.0108	3.8728	5.31	11.1034	21.4415	27.4029	42.6441	0.1573	68.8046	0
34	29.1729	0.0155	3.4826	5.5427	11.6998	22.9311	29.0055	44.8459	0.2786	67.7212	1
35	35.2274	0.0153	4.265	5.4728	11.5021	23.216	29.9284	45.8194	0.1738	83.7451	0
36	33.9193	0.023	3.3328	5.7097	11.9975	21.7963	28.5323	44.7507	0.3116	84.3237	1
37	34.4043	0.0144	4.2666	5.4251	11.3855	23.2463	29.6159	46.1905	0.1704	86.0139	0
38	29.0943	0.0156	3.5126	5.5885	11.8344	24.5504	30.7731	48.3301	0.2784	69.3236	1
39	27.2064	0.0103	3.7283	5.3438	11.2981	22.0147	29.7404	43.1424	0.205	68.9388	0
40	29.0359	0.0148	3.4111	5.5716	11.8278	24.376	30.9883	47.6693	0.2721	78.6404	1
41	26.4059	0.0102	3.664	5.3492	11.3047	21.3572	27.5919	42.5218	0.2204	57.0943	0
42	30.0713	0.0158	3.5678	5.5993	11.8586	24.2964	30.4569	48.0587	0.2659	68.6455	1
43	27.9246	0.01	3.9025	5.3477	11.3233	21.2033	27.3976	42.338	0.185	63.55	0
44	30.6568	0.0159	3.5851	5.6038	11.87	23.9468	30.0445	47.65	0.2584	68.567	1
45	30.2173	0.0111	4.0013	5.4143	11.4539	21.3308	27.4956	42.5375	0.1755	59.8338	0

No file chosen

Gambar 4.1 Antarmuka Tampilan untuk Data

Seperti halnya pada groupbox data uji, pada groupbox data latih juga terdapat tombol choose file untuk memilih data latih yang akan digunakan. Selain itu terdapat pula tombol submit. Data latih yang akan dilatihkan dalam sistem aplikasi ini sebanyak 25 data latih, dengan rincian 4 fold, 5 fold, 6 fold, 10 fold jadi total sebanyak 25 data latih dan bersifat berpasangan untuk menghindari data yang sama dengan data uji.

Dalam memilih file data uji dan data latih format file yang dipilih adalah csv (comma-separated values) untuk selanjutnya diproses oleh sistem dan memasukkannya ke dalam database kemudian ditampilkan dalam tabel.



Nilai K	Benar	Salah	Akurasi
1	9	6	60
2	9	6	60
3	10	5	66.6666666667
4	10	5	66.6666666667
5	10	5	66.6666666667
6	10	5	66.6666666667
7	11	4	73.3333333333
8	11	4	73.3333333333
9	11	4	73.3333333333
10	11	4	73.3333333333

Rata-rata akurasi : 68

Gambar 4.2 Antarmuka Tampilan untuk Perhitungan FKNN

Tingkat akurasi dari klasifikasi ditampilkan pada tab FKNN, yang ditunjukkan pada gambar 4.2. Pada tab tersebut menampilkan hasil perhitungan Fuzzy K-Nearest Neighbor (FK-NN) meliputi kolom K, benar, salah, akurasi, dan yang terakhir rata-rata akurasi. Hasil yang ditampilkan dalam tabel merupakan perhitungan per fold jadi akan dilakukan sebanyak 25 perhitungan untuk mendapatkan hasil seluruh fold. Dalam penelitian ini menggunakan 4 fold cross validation, yaitu 4 fold, 5 fold, 6 fold, dan 10 fold. Untuk semua perhitungan sebanyak 25 kali tersebut akan dihitung sejumlah K sebanyak 10.



Gambar 4.3 Antarmuka Tampilan untuk Grafik Akurasi

Tab terakhir dari sistem yang dibuat untuk menampilkan grafik akurasi dari perhitungan ditunjukkan pada gambar 4.3 sudah dilakukan sebanyak 25 kali berasal dari 4 perhitungan untuk 4 fold, 5 perhitungan untuk 5 fold, 6 perhitungan untuk 6 fold, dan yang terakhir 10 perhitungan untuk 10 fold. Pada pojok kanan atas terdapat dua tombol yang berguna untuk menyimpan gambar grafik berupa gambar dengan format PNG, JPG, SVG, maupun PDF serta bisa langsung dicetak dikertas ditunjukkan pada ikon gambar sebelah kiri sedangkan sebelah kanan untuk yang sudah dijelaskan sebelumnya yaitu untuk menyimpan grafik berupa gambar berformat PNG, JPG, SVG, maupun PDF. Garis vertikal atau y untuk melihat akurasi dari setiap garis fold sedangkan garis horizontal atau x untuk melihat nilai K mulai 1 – 10. Jumlah garis dalam grafik akurasi berjumlah sejumlah K-Fold.



BAB V

PEMBAHASAN DAN ANALISA

Bab ini berisi pembahasan dan analisa mengenai hasil uji coba terhadap sistem.

5.1. Implementasi Pengujian

Pada pengujian klasifikasi jenis mangga berdasarkan tekstur daun dengan menggunakan metode Fuzzy K-Nearest Neighbor (FK-NN) digunakan dataset mangga gadung dan mangga curut terdiri dari 60 record dengan rincian 30 record untuk mangga gadung dan 30 record untuk mangga curut.

5.1.1. Hasil Uji

Seperti yang telah dibahas pada bab 3, pengujian dilakukan menggunakan K-Fold cross validation. Digunakan sebanyak 4 cross validation, yaitu 4-Fold, 5-Fold, 6-Fold, dan 10-Fold.

5.1.1.1. 4-Fold

Digunakan 4-fold cross validation berarti data dibagi menjadi 4 bagian sebagai data uji dan dilakukan secara bergantian. Berikut hasil dari fold yang pertama menunjukkan hasil akurasi rata-rata sebesar 80 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 5$ dengan hasil benar sebanyak 13 dan 2 salah menghasilkan persentase 86.6667 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 2$ dengan hasil benar sebanyak 11 dan 4 salah menghasilkan persentase 73.3333 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.1.

Tabel 5.1 Tabel Hasil Pengujian untuk 4-Fold cross validation bagian pertama data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
4	1	2	11	4	73.3333	80
		3	12	3	80	
		4	12	3	80	
		5	13	2	86.6667	
		6	12	3	80	
		7	12	3	80	
		8	12	3	80	
		9	12	3	80	
		10	12	3	80	

Hasil akurasi dari fold yang kedua menunjukkan hasil akurasi rata-rata sebesar 68.8889 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 7, 8, 9, 10$ dengan hasil benar sebanyak 11 dan 4 salah menghasilkan persentase 73.3333 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 2$ dengan hasil benar sebanyak 9 dan 6 salah menghasilkan persentase 60 %. Rincian hasil lebih detail bisa dilihat pada tabel 5.2.

Tabel 5.2 Tabel Hasil Pengujian untuk 4-Fold cross validation bagian kedua data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
4	2	2	9	6	60	68,8889
		3	10	5	66.6667	
		4	10	5	66.6667	
		5	10	5	66.6667	
		6	10	5	66.6667	
		7	11	4	73.3333	
		8	11	4	73.3333	
		9	11	4	73.3333	
		10	11	4	73.3333	

Hasil akurasi dari fold yang ketiga menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 15 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.3.

Tabel 5.3 Tabel Hasil Pengujian untuk 4-Fold cross validation bagian ketiga data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
4	3	2	15	0	100	100
		3	15	0	100	
		4	15	0	100	
		5	15	0	100	
		6	15	0	100	
		7	15	0	100	
		8	15	0	100	
		9	15	0	100	
		10	15	0	100	

Hasil akurasi dari fold yang terakhir menunjukkan hasil akurasi rata-rata sebesar 71.8519 %. Nilai akurasi tertinggi diperoleh saat nilai k = 10 dengan hasil benar sebanyak 14 dan 1 salah menghasilkan persentase 93.3333 %. Sedangkan nilai akurasi terendah diperoleh saat nilai k = 2, 3, 4, 5, 6 dengan hasil benar sebanyak 10 dan 5 salah menghasilkan persentase 66.6667 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.4.

Tabel 5.4 Tabel Hasil Pengujian untuk 4-Fold cross validation bagian terakhir data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
4	4	2	10	5	66.6667	71.8519
		3	10	5	66.6667	

	4	10	5	66.6667
	5	10	5	66.6667
	6	10	5	66.6667
	7	11	4	73.3333
	8	11	4	73.3333
	9	11	4	73.3333
	10	14	1	93.3333

Langkah terakhir dari semua perhitungan tiap fold dari 1 sampai 4 dilakukan perhitungan rata-rata untuk 4-fold cross validation ini menghasilkan akurasi rata-rata 80.1852 %. Rincian hasil lebih detail bisa dilihat pada tabel 5.5.

Tabel 5.5 Tabel Hasil Rata-rata untuk 4-Fold cross validation semua bagian data uji terhadap akurasi

Fold	Akurasi (%)
1	80
2	68.8889
3	100
4	71.8519
Rata-rata akurasi (%)	80.1852

5.1.1.2. 5-Fold

Digunakan 5-fold cross validation berarti data dibagi menjadi 5 bagian sebagai data uji dan dilakukan secara bergantian. Berikut hasil dari fold yang pertama menunjukkan hasil akurasi rata-rata sebesar 75 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 5$ dengan hasil benar sebanyak 10 dan 2 salah

menghasilkan persentase 83.3333 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 2$ dengan hasil benar sebanyak 8 dan 4 salah menghasilkan persentase 66.6667 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.6.

Tabel 5.6 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian pertama data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
5	1	2	8	4	66.6667	75
		3	9	3	75	
		4	9	3	75	
		5	10	2	83.3333	
		6	9	3	75	
		7	9	3	75	
		8	9	3	75	
		9	9	3	75	
		10	9	3	75	

Hasil akurasi dari fold yang kedua menunjukkan hasil akurasi rata-rata sebesar 45.3704 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 2, 3, 4, 9$ dengan hasil benar sebanyak 6 dan 6 salah menghasilkan persentase 50 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 5, 6, 7, 8, 10$ dengan hasil benar sebanyak 5 dan 7 salah menghasilkan persentase 41.6667 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.7.

Tabel 5.7 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian kedua data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
5	2	2	6	6	50	45.3704
		3	6	6	50	
		4	6	6	50	
		5	5	7	41.6667	
		6	5	7	41.6667	
		7	5	7	41.6667	
		8	5	7	41.6667	
		9	6	6	50	
		10	5	7	41.6667	

Hasil akurasi dari fold yang ketiga menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 12 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.8.

Tabel 5.8 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian ketiga data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
5	3	2	12	0	100	100
		3	12	0	100	
		4	12	0	100	

	5	12	0	100
	6	12	0	100
	7	12	0	100
	8	12	0	100
	9	12	0	100
	10	12	0	100

Hasil akurasi dari fold yang keempat menunjukkan hasil akurasi rata-rata sebesar 91.6667 %. Untuk hasil rata-rata 91.6667 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 91.6667 % dengan hasil benar sebanyak 11 dan 1 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.9.

Tabel 5.9 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian keempat data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
5	4	2	11	1	91.6667	91.6667
		3	11	1	91.6667	
		4	11	1	91.6667	
		5	11	1	91.6667	
		6	11	1	91.6667	
		7	11	1	91.6667	
		8	11	1	91.6667	
		9	11	1	91.6667	
		10	11	1	91.6667	

Hasil akurasi dari fold yang terakhir menunjukkan hasil akurasi rata-rata sebesar 71.2963 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 10$ dengan hasil benar sebanyak 12 dan 0 salah menghasilkan persentase 100 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 2$ dengan hasil benar sebanyak 6 dan 6 salah menghasilkan persentase 50 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.10.

Tabel 5.10 Tabel Hasil Pengujian untuk 5-Fold cross validation bagian terakhir data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
5	5	2	6	6	50	71.2963
		3	8	4	66.6667	
		4	8	4	66.6667	
		5	8	4	66.6667	
		6	8	4	66.6667	
		7	9	3	75	
		8	9	3	75	
		9	9	3	75	
		10	12	0	100	

Langkah terakhir dari semua perhitungan tiap fold dari 1 sampai 5 dilakukan perhitungan rata-rata untuk 5-fold cross validation ini menghasilkan akurasi rata-rata 76.166 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.11.

Tabel 5.11 Tabel Hasil Rata-rata untuk 5-Fold cross validation semua bagian data uji terhadap akurasi

Fold	Akurasi (%)
1	75
2	45.3704
3	100
4	91.6667
5	71.2963
Rata-rata akurasi (%)	76.6667

5.1.1.3. 6-Fold

Digunakan 6-fold cross validation berarti data dibagi menjadi 6 bagian sebagai data uji dan dilakukan secara bergantian. Berikut hasil dari fold yang pertama menunjukkan hasil akurasi rata-rata sebesar 71.1111 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 7$ dengan hasil benar sebanyak 8 dan 2 salah menghasilkan persentase 80 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k =$ selain nilai k dengan nilai akurasi tertinggi dengan hasil benar sebanyak 7 dan 3 salah menghasilkan persentase 70 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.12.

Tabel 5.12 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian pertama data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	1	2	7	3	70	71.1111
		3	7	3	70	
		4	7	3	70	
		5	7	3	70	
		6	7	3	70	
		7	8	2	80	
		8	7	3	70	
		9	7	3	70	
		10	7	3	70	

Hasil akurasi dari fold yang kedua menunjukkan hasil akurasi rata-rata sebesar 74.4444 %. Nilai akurasi tertinggi diperoleh saat nilai k = 2, 3, 5 dengan hasil benar sebanyak 9 dan 1 salah menghasilkan persentase 90 %. Sedangkan nilai akurasi terendah diperoleh saat nilai k = 8, 9, 10 dengan hasil benar sebanyak 6 dan 4 salah menghasilkan persentase 60 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.13.

Tabel 5.13 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian kedua data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	2	2	9	1	90	74.4444
		3	9	1	90	
		4	8	2	80	
		5	9	1	90	
		6	7	3	70	
		7	7	3	70	
		8	6	4	60	
		9	6	4	60	
		10	6	4	60	

Hasil akurasi dari fold yang ketiga menunjukkan hasil akurasi rata-rata sebesar 85.5556 %. Nilai akurasi tertinggi diperoleh saat nilai k = 2, 4 dengan hasil benar sebanyak 10 dan 0 salah menghasilkan persentase 100 %. Sedangkan nilai akurasi terendah diperoleh saat nilai k = 3, 6, 7, 8, 9, 10 dengan hasil benar sebanyak 8 dan 2 salah menghasilkan persentase 80 %. Rincian hasil lebih detail bisa dilihat pada tabel 5.14.

Tabel 5.14 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian ketiga data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	3	2	10	0	100	85.5556
		3	8	2	80	
		4	10	0	100	
		5	9	1	90	
		6	8	2	80	
		7	8	2	80	
		8	8	2	80	
		9	8	2	80	
		10	8	2	80	

Hasil akurasi dari fold yang keempat menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 10 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.15.

Tabel 5.15 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian keempat data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	4	2	10	0	100	100
		3	10	0	100	
		4	10	0	100	

	5	10	0	100
	6	10	0	100
	7	10	0	100
	8	10	0	100
	9	10	0	100
	10	10	0	100

Hasil akurasi dari fold yang kelima menunjukkan hasil akurasi rata-rata sebesar 90 %. Untuk hasil rata-rata 90 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 90 % dengan hasil benar sebanyak 9 dan 1 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.16.

Tabel 5.16 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian kelima data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	5	2	90	1	90	90
		3	90	1	90	
		4	90	1	90	
		5	90	1	90	
		6	90	1	90	
		7	90	1	90	
		8	90	1	90	
		9	90	1	90	
		10	90	1	90	

Hasil akurasi dari fold yang terakhir menunjukkan hasil akurasi rata-rata sebesar 70 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 10$ dengan hasil benar sebanyak 10 dan 0 salah menghasilkan persentase 100 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 2, 3, 4, 5, 6$ dengan hasil benar sebanyak 6 dan 4 salah menghasilkan persentase 60 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.17.

Tabel 5.17 Tabel Hasil Pengujian untuk 6-Fold cross validation bagian terakhir data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
6	6	2	6	4	60	70
		3	6	4	60	
		4	6	4	60	
		5	6	4	60	
		6	6	4	60	
		7	7	3	70	
		8	7	3	70	
		9	9	1	90	
		10	10	0	100	

Langkah terakhir dari semua perhitungan tiap fold dari 1 sampai 6 dilakukan perhitungan rata-rata untuk 6-fold cross validation ini menghasilkan akurasi rata-rata 82.166 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.18.

Tabel 5.18 Tabel Hasil Rata-rata untuk 6-Fold cross validation semua bagian data uji terhadap akurasi

Fold	Akurasi (%)
1	71.1111
2	74.4444
3	85.5556
4	100
5	90
6	70
Rata-rata akurasi (%)	81.8519

5.1.1.4. 10-Fold

Digunakan 10-fold cross validation berarti data dibagi menjadi 10 bagian sebagai data uji dan dilakukan secara bergantian. Berikut hasil dari fold yang pertama menunjukkan hasil akurasi rata-rata sebesar 61.1111 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 2$ dan 9 dengan hasil benar sebanyak 5 dan 1 salah menghasilkan persentase 83.3333 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 4, 5, 6, 7, 8$ dengan hasil benar sebanyak 3 dan 3 salah menghasilkan persentase 50 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.19.

Tabel 5.19 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian pertama data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	1	2	5	1	83.3333	61.1111
		3	4	2	66.6667	
		4	3	3	50	
		5	3	3	50	
		6	3	3	50	
		7	3	3	50	
		8	3	3	50	
		9	5	1	83.3333	
		10	4	2	66.6667	

Hasil akurasi dari fold yang kedua menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 6 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.20.

Tabel 5.20 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kedua data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	2	2	6	0	100	100
		3	6	0	100	
		4	6	0	100	

	5	6	0	100
	6	6	0	100
	7	6	0	100
	8	6	0	100
	9	6	0	100
	10	6	0	100

Hasil akurasi dari fold yang ketiga menunjukkan hasil akurasi rata-rata sebesar 96.2963 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 2$ sampai 8 dengan hasil benar sebanyak 6 dan 0 salah menghasilkan persentase 100 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 9$ dan 10 dengan hasil benar sebanyak 5 dan 1 salah menghasilkan persentase 83.3333 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.21.

Tabel 5.21 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian ketiga data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	3	2	6	0	100	96.2963
		3	6	0	100	
		4	6	0	100	
		5	6	0	100	
		6	6	0	100	
		7	6	0	100	
		8	6	0	100	
		9	5	1	83.3333	
		10	5	1	83.3333	

Hasil akurasi dari fold yang keempat menunjukkan hasil akurasi rata-rata sebesar 22.2223 %. Nilai akurasi tertinggi diperoleh saat nilai $k = 2$ dengan hasil benar sebanyak 4 dan 2 salah menghasilkan persentase 66.6667 %. Sedangkan nilai akurasi terendah diperoleh saat nilai $k = 3$ sampai 10 dengan hasil benar sebanyak 1 dan 5 salah menghasilkan persentase 16.6667 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.22.

Tabel 5.22 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian keempat data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	4	2	4	2	66.6667	22.2223
		3	1	5	16.6667	
		4	1	5	16.6667	
		5	1	5	16.6667	
		6	1	5	16.6667	
		7	1	5	16.6667	
		8	1	5	16.6667	
		9	1	5	16.6667	
		10	1	5	16.6667	

Hasil akurasi dari fold yang kelima menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 6 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.23.

Tabel 5.23 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kelima data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	5	2	6	0	100	100
		3	6	0	100	
		4	6	0	100	
		5	6	0	100	
		6	6	0	100	
		7	6	0	100	
		8	6	0	100	
		9	6	0	100	
		10	6	0	100	

Hasil akurasi dari fold yang keenam menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 6 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.24.

Tabel 5.24 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian keenam data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	6	2	6	0	100	100
		3	6	0	100	
		4	6	0	100	

	5	6	0	100
	6	6	0	100
	7	6	0	100
	8	6	0	100
	9	6	0	100
	10	6	0	100

Hasil akurasi dari fold yang ketujuh menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 6 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.25.

Tabel 5.25 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian ketujuh data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	7	2	6	0	100	100
		3	6	0	100	
		4	6	0	100	
		5	6	0	100	
		6	6	0	100	
		7	6	0	100	
		8	6	0	100	
		9	6	0	100	
		10	6	0	100	

Hasil akurasi dari fold yang kedelapan menunjukkan hasil akurasi rata-rata sebesar 83.3333 %. Untuk hasil rata-rata 83.3333 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 83.3333 % dengan hasil benar sebanyak 5 dan 1 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.26.

Tabel 5.26 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kedelapan data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	8	2	5	1	83.3333	83.3333
		3	5	1	83.3333	
		4	5	1	83.3333	
		5	5	1	83.3333	
		6	5	1	83.3333	
		7	5	1	83.3333	
		8	5	1	83.3333	
		9	5	1	83.3333	
		10	5	1	83.3333	

Hasil akurasi dari fold yang kesembilan menunjukkan hasil akurasi rata-rata sebesar 96.2963 %. Nilai akurasi tertinggi diperoleh saat nilai k = 3 sampai 10 dengan hasil benar sebanyak 6 dan 0 salah menghasilkan persentase 100 %. Sedangkan nilai akurasi terendah diperoleh saat nilai k = 2 dengan hasil benar sebanyak 4 dan 2 salah menghasilkan persentase 66.6667 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.27.

Tabel 5.27 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian kesembilan data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	9	2	4	2	66.6667	96.2963
		3	6	0	100	
		4	6	0	100	
		5	6	0	100	
		6	6	0	100	
		7	6	0	100	
		8	6	0	100	
		9	6	0	100	
		10	6	0	100	

Hasil akurasi dari fold yang terakhir menunjukkan hasil akurasi rata-rata sebesar 100 %. Untuk hasil rata-rata 100 % berarti menunjukkan semua nilai k mulai 2 sampai 10 menghasilkan akurasi masing-masing sebesar 100 % dengan hasil benar sebanyak 6 dan 0 salah. Rincian hasil lebih detil bisa dilihat pada tabel 5.28.

Tabel 5.28 Tabel Hasil Pengujian untuk 10-Fold cross validation bagian terakhir data uji terhadap akurasi

K-Fold Cross Validation	Fold	Nilai K	Hasil		Akurasi (%)	Rata-rata akurasi (%)
			Benar	Salah		
10	10	2	6	0	100	100
		3	6	0	100	
		4	6	0	100	

	5	6	0	100
	6	6	0	100
	7	6	0	100
	8	6	0	100
	9	6	0	100
	10	6	0	100

Langkah terakhir dari semua perhitungan tiap fold dari 1 sampai 10 dilakukan perhitungan rata-rata untuk 10-fold cross validation ini menghasilkan akurasi rata-rata 86.333 %. Rincian hasil lebih detil bisa dilihat pada tabel 5.29.

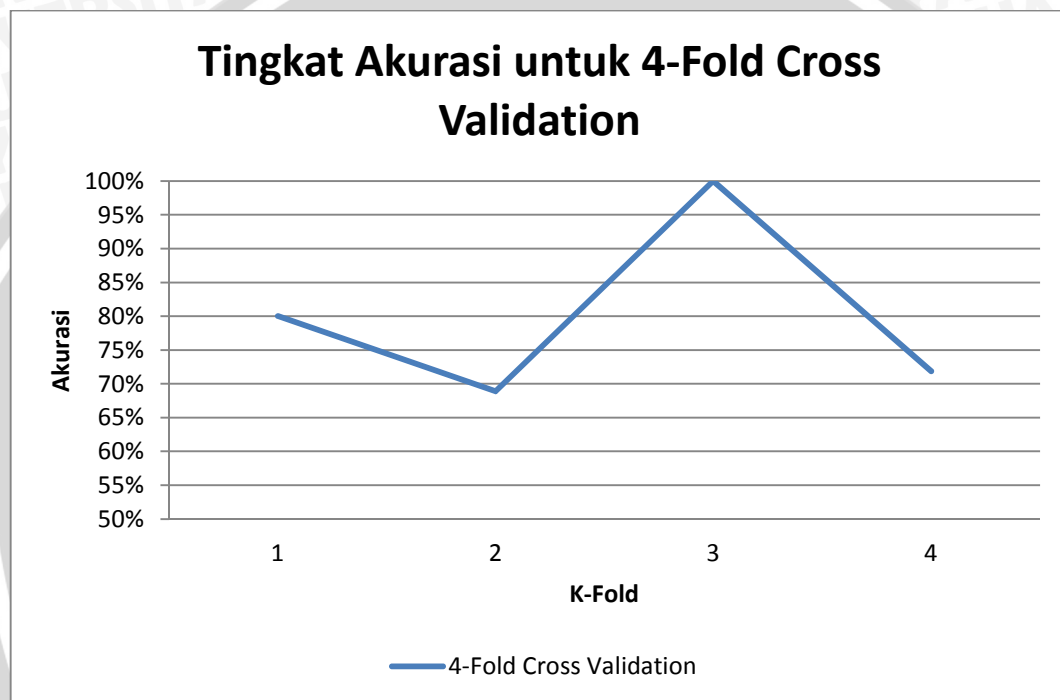
Tabel 5.29 Tabel Hasil Rata-rata untuk 10-Fold cross validation semua bagian data uji terhadap akurasi

Fold	Akurasi (%)
1	61.1111
2	100
3	96.2963
4	22.2223
5	100
6	100
7	100
8	83.3333
9	96.2963
10	100
Rata-rata akurasi (%)	85.9259

5.1.2. Analisa Hasil

5.1.2.1. 4-Fold

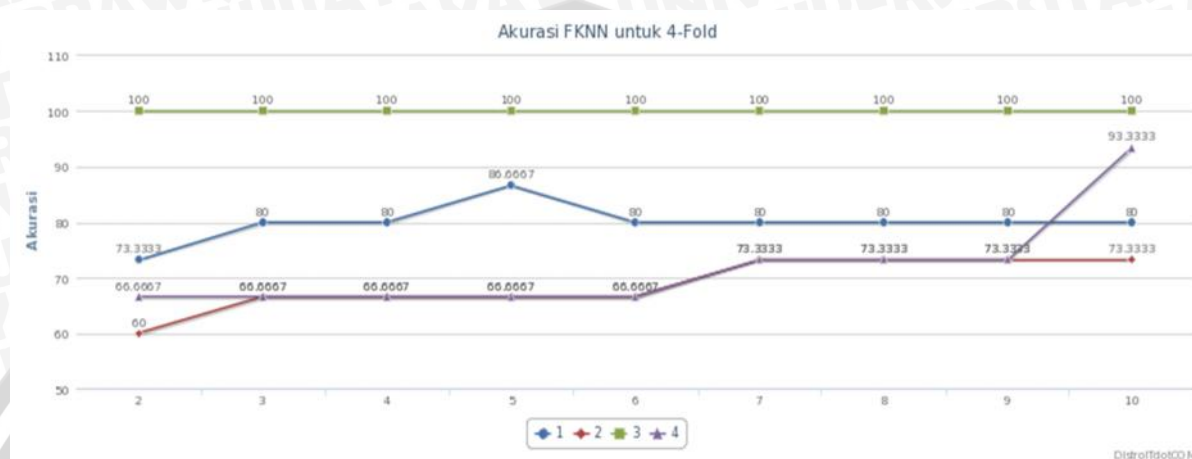
Gambar 5.1 menunjukkan bahwa hasil pengujian dataset citra jenis mangga gadung dan curut berdasarkan tekstur daun, diketahui bahwa akurasi paling baik berada pada K-Fold bagian ketiga dengan nilai 100 % dan yang paling jelek berada pada K-Fold bagian kedua dengan nilai 68.8889 %. Sehingga menghasilkan hasil rata-rata akhir sebesar 80.1852 %.



Gambar 5.1 Tingkat Akurasi untuk 4-Fold Cross Validation

Gambaran grafik yang ditunjukkan gambar 5.1 pada tingkat akurasi tiap fold untuk 4-Fold Cross Validation dipengaruhi oleh gambar 5.2 yang ditunjukkan dibawah ini, yaitu untuk fold pertama dengan tingkat akurasi 80 % dikarenakan pengujian tiap k berada pada kisaran 80 % kecuali k = 2 dan k = 5 yang tidak 80 % tetapi jika dijumlahkan menghasilkan nilai akurasi 80 % jadi untuk fold yang pertama memiliki akurasi 80 %. Untuk fold kedua dengan tingkat akurasi 68.8889 % dikarenakan pengujian tiap k berada pada kisaran antara 60 % - 73.3333 % dengan dominasi 66.6667 % dan 73.3333 % jadi untuk fold yang kedua memiliki akurasi 68.8889 %. Untuk fold ketiga dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang ketiga

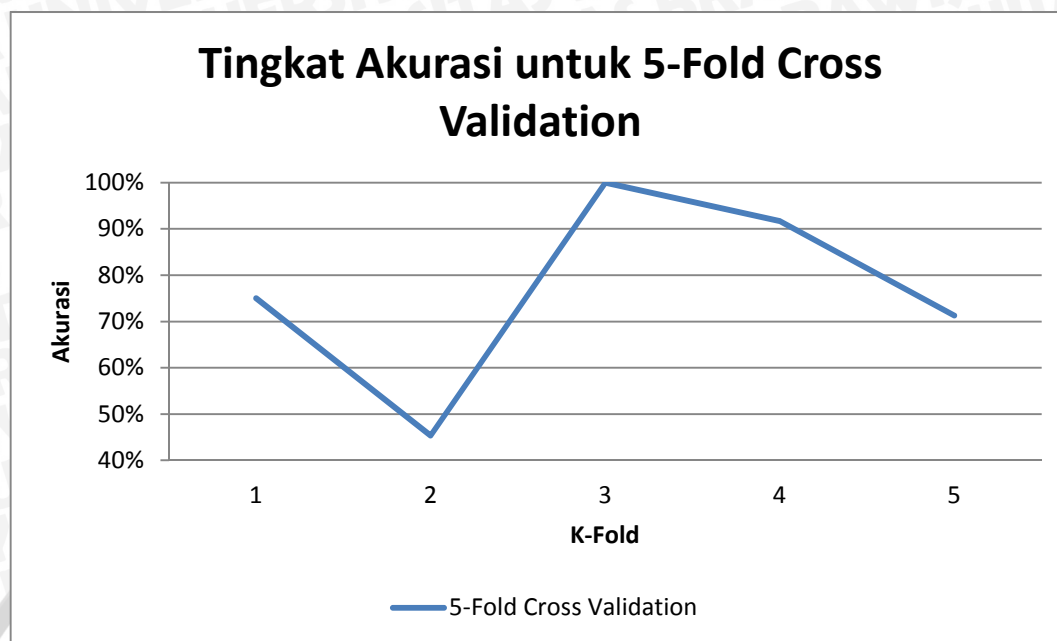
memiliki akurasi 100 %. Untuk fold yang terakhir dengan tingkat akurasi 71.8519 % dikarenakan pengujian tiap k berada pada kisaran antara 66.6667 % - 93.3333 % dengan dominasi 66.6667 % dan 73.3333 % jadi untuk fold yang terakhir memiliki akurasi 71.8519 %.



Gambar 5.2 Detil Akurasi untuk 4-Fold Cross Validation

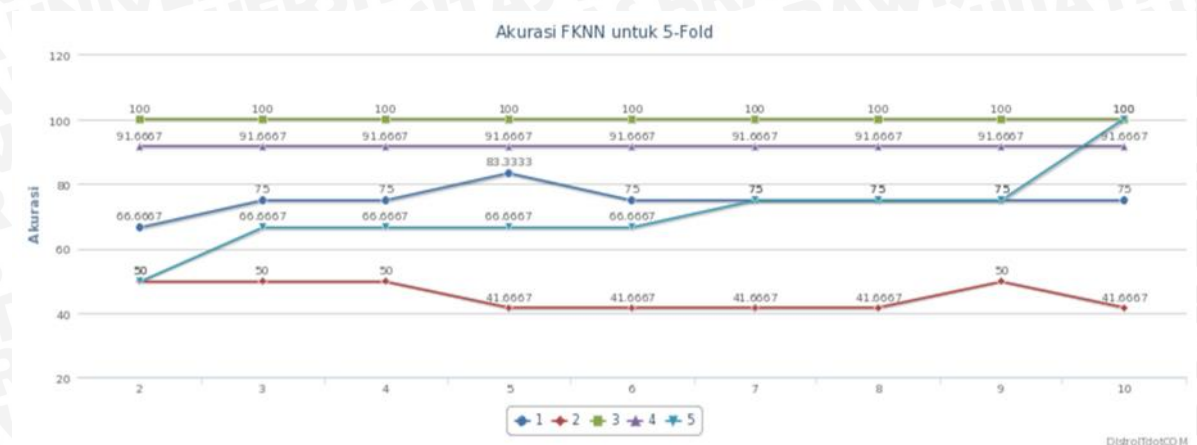
5.1.2.2. 5-Fold

Gambar 5.3 menunjukkan bahwa hasil pengujian dataset citra jenis mangga gadung dan curut berdasarkan tekstur daun, diketahui bahwa akurasi paling baik berada pada K-Fold bagian ketiga dengan nilai 100 % dan yang paling jelek berada pada K-Fold bagian kedua dengan nilai 45.3704 %. Sehingga menghasilkan hasil rata-rata akhir sebesar 76.6667 %.



Gambar 5.3 Tingkat Akurasi untuk 5-Fold Cross Validation

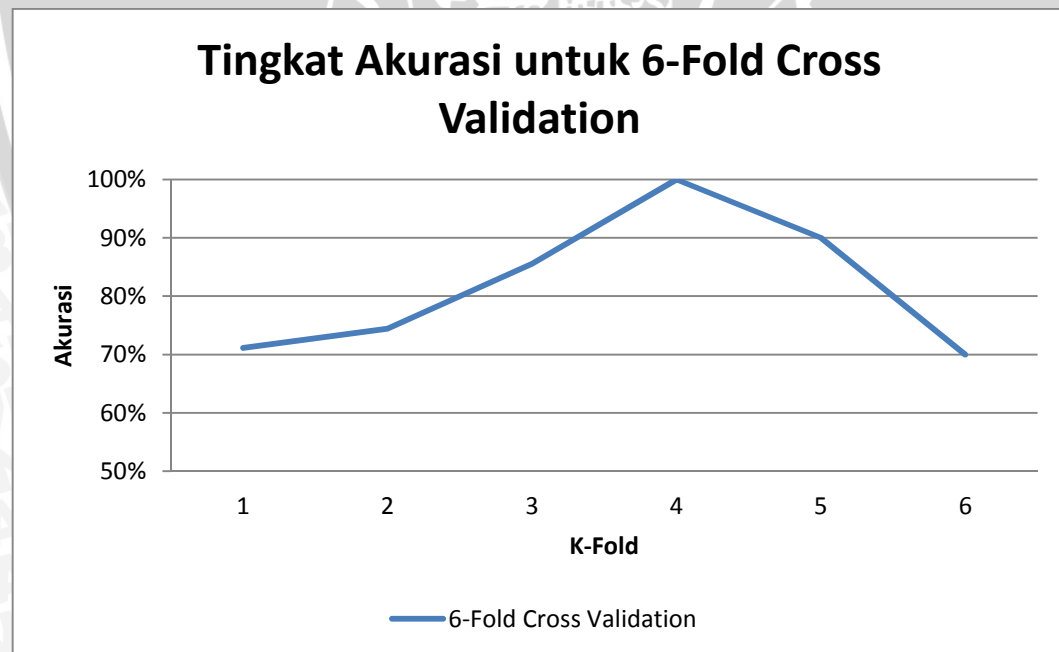
Gambaran grafik yang ditunjukkan gambar 5.3 pada tingkat akurasi tiap fold untuk 5-Fold Cross Validation dipengaruhi oleh gambar 5.4 yang ditunjukkan dibawah ini, yaitu untuk fold pertama dengan tingkat akurasi 75 % dikarenakan pengujian tiap k berada pada kisaran 75 % kecuali $k = 2$ dan $k = 5$ yang tidak 75 % tetapi jika dijumlahkan menghasilkan nilai akurasi 75 % jadi untuk fold yang pertama memiliki akurasi 75 %. Untuk fold kedua dengan tingkat akurasi 45.3704 % dikarenakan pengujian tiap k berada pada kisaran antara 41.6667 % - 50 % dengan dominasi 41.6667 % dan 50 % jadi untuk fold yang kedua memiliki akurasi 45.3704 %. Untuk fold ketiga dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang ketiga memiliki akurasi 100 %. Untuk fold keempat dengan tingkat akurasi 91.6667 % dikarenakan pengujian semua k menghasilkan nilai akurasi 91.6667 % jadi untuk fold yang keempat memiliki akurasi 91.6667 %. Untuk fold yang terakhir dengan tingkat akurasi 71.2963 % dikarenakan pengujian tiap k berada pada kisaran antara 50 % - 100 % dengan dominasi 66.6667 % dan 75 % jadi untuk fold yang terakhir memiliki akurasi 71.2963 %.



Gambar 5.4 Detil Akurasi untuk 5-Fold Cross Validation

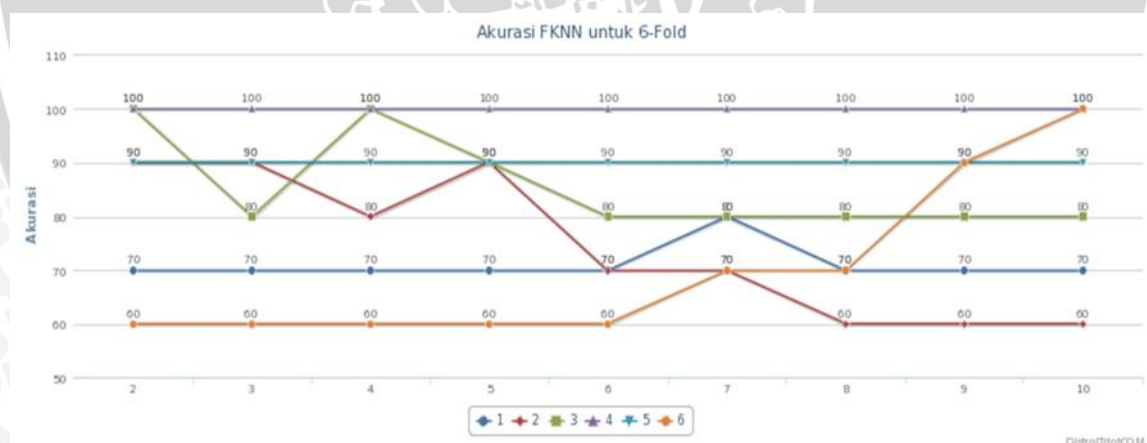
5.1.2.3. 6-Fold

Gambar 5.5 menunjukkan bahwa hasil pengujian dataset citra jenis mangga gadung dan curut berdasarkan tekstur daun, diketahui bahwa akurasi paling baik berada pada K-Fold bagian keempat dengan nilai 100 % dan yang paling jelek berada pada K-Fold bagian keenam dengan nilai 70 %. Sehingga menghasilkan hasil rata-rata akhir sebesar 81.8519 %.



Gambar 5.5 Tingkat Akurasi untuk 6-Fold Cross Validation

Gambaran grafik yang ditunjukkan gambar 5.5 pada tingkat akurasi tiap fold untuk 6-Fold Cross Validation dipengaruhi oleh gambar 5.6 yang ditunjukkan dibawah ini, yaitu untuk fold pertama dengan tingkat akurasi 71.1111 % dikarenakan pengujian tiap k berada pada kisaran 70 % kecuali k = 7 yang tidak 70 % tetapi nilai akurasi 80 % jadi untuk fold yang pertama memiliki akurasi 71.1111 %. Untuk fold kedua dengan tingkat akurasi 74.4444 % dikarenakan pengujian tiap k berada pada kisaran antara 60 % - 90 % dengan dominasi 60 % dan 90 % jadi untuk fold yang kedua memiliki akurasi 74.4444 %. Untuk fold ketiga dengan tingkat akurasi 85.5556 % dikarenakan pengujian tiap k berada pada kisaran antara 80 % - 100 % dengan dominasi 80 % jadi untuk fold yang ketiga memiliki akurasi 85.5556 %. Untuk fold keempat dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang keempat memiliki akurasi 100 %. Untuk fold kelima dengan tingkat akurasi 90 % dikarenakan pengujian semua k menghasilkan nilai akurasi 90 % jadi untuk fold yang kelima memiliki akurasi 90 %. Untuk fold yang terakhir dengan tingkat akurasi 70 % dikarenakan pengujian tiap k berada pada kisaran antara 60 % - 100 % dengan dominasi 60 % jadi untuk fold yang terakhir memiliki akurasi 70 %.

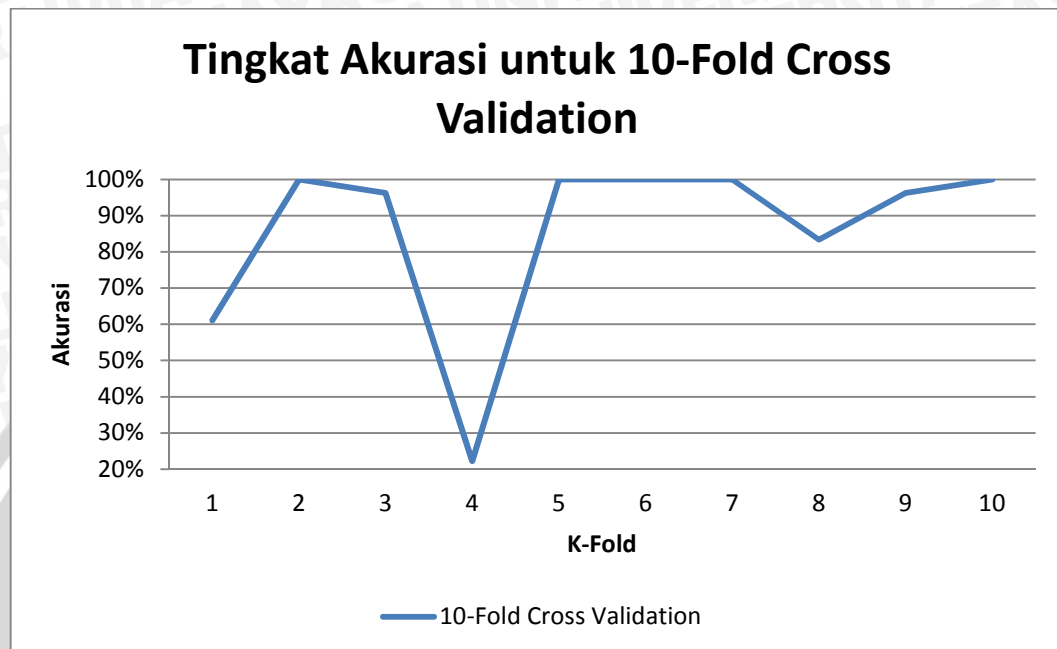


Gambar 5.6 Detil Akurasi untuk 6-Fold Cross Validation

5.1.2.4. 10-Fold

Gambar 5.7 menunjukkan bahwa hasil pengujian dataset citra jenis mangga gadung dan curut berdasarkan tekstur daun, diketahui bahwa akurasi paling baik berada pada K-Fold bagian kedua, kelima sampai ketujuh, dan yang terakhir

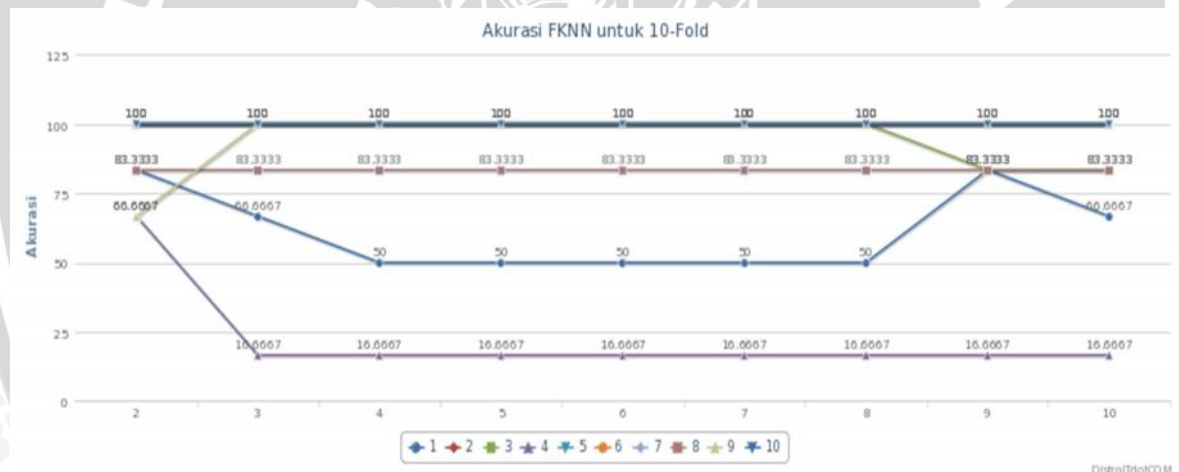
bagian kesepuluh dengan nilai 100 % dan yang paling jelek berada pada K-Fold bagian keempat dengan nilai 22.2223 %. Sehingga menghasilkan hasil rata-rata akhir sebesar 85.9259 %.



Gambar 5.7 Tingkat Akurasi untuk 10-Fold Cross Validation

Gambaran grafik yang ditunjukkan gambar 5.7 pada tingkat akurasi tiap fold untuk 10-Fold Cross Validation dipengaruhi oleh gambar 5.8 yang ditunjukkan dibawah ini, yaitu untuk fold pertama dengan tingkat akurasi 61.1111 % dikarenakan pengujian tiap k berada pada kisaran antara 50 % - 83.3333 % dengan dominasi 50 % jadi untuk fold yang pertama memiliki akurasi 61.1111 %. Untuk fold kedua dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang kedua memiliki akurasi 100 %. Untuk fold ketiga dengan tingkat akurasi 96.2963 % dikarenakan pengujian tiap k berada pada kisaran antara 83.3333 % - 100 % dengan dominasi 100 % jadi untuk fold yang ketiga memiliki akurasi 96.2963 %. Untuk fold keempat dengan tingkat akurasi 22.2223 % dikarenakan pengujian tiap k berada pada kisaran antara 16.6667 % - 66.6667 % dengan dominasi 16.6667 % jadi untuk fold yang keempat memiliki akurasi 22.2223 %. Untuk fold kelima dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi

untuk fold yang kelima memiliki akurasi 100 %. Untuk fold keenam dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang keenam memiliki akurasi 100 %. Untuk fold ketujuh dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang ketujuh memiliki akurasi 100 %. Untuk fold kedelapan dengan tingkat akurasi 83.3333 % dikarenakan pengujian semua k menghasilkan nilai akurasi 83.3333 % jadi untuk fold yang kedelapan memiliki akurasi 83.3333 %. Untuk fold kesembilan dengan tingkat akurasi 96.2963 % dikarenakan pengujian tiap k berada pada kisaran antara 66.6667 % - 100 % dengan dominasi 100 % jadi untuk fold yang kesembilan memiliki akurasi 96.2963 %. Untuk fold yang terakhir dengan tingkat akurasi 100 % dikarenakan pengujian semua k menghasilkan nilai akurasi 100 % jadi untuk fold yang terakhir memiliki akurasi 100 %.



Gambar 5.8 Detil Akurasi untuk 10-Fold Cross Validation

BAB VI PENUTUP

6.1. Kesimpulan

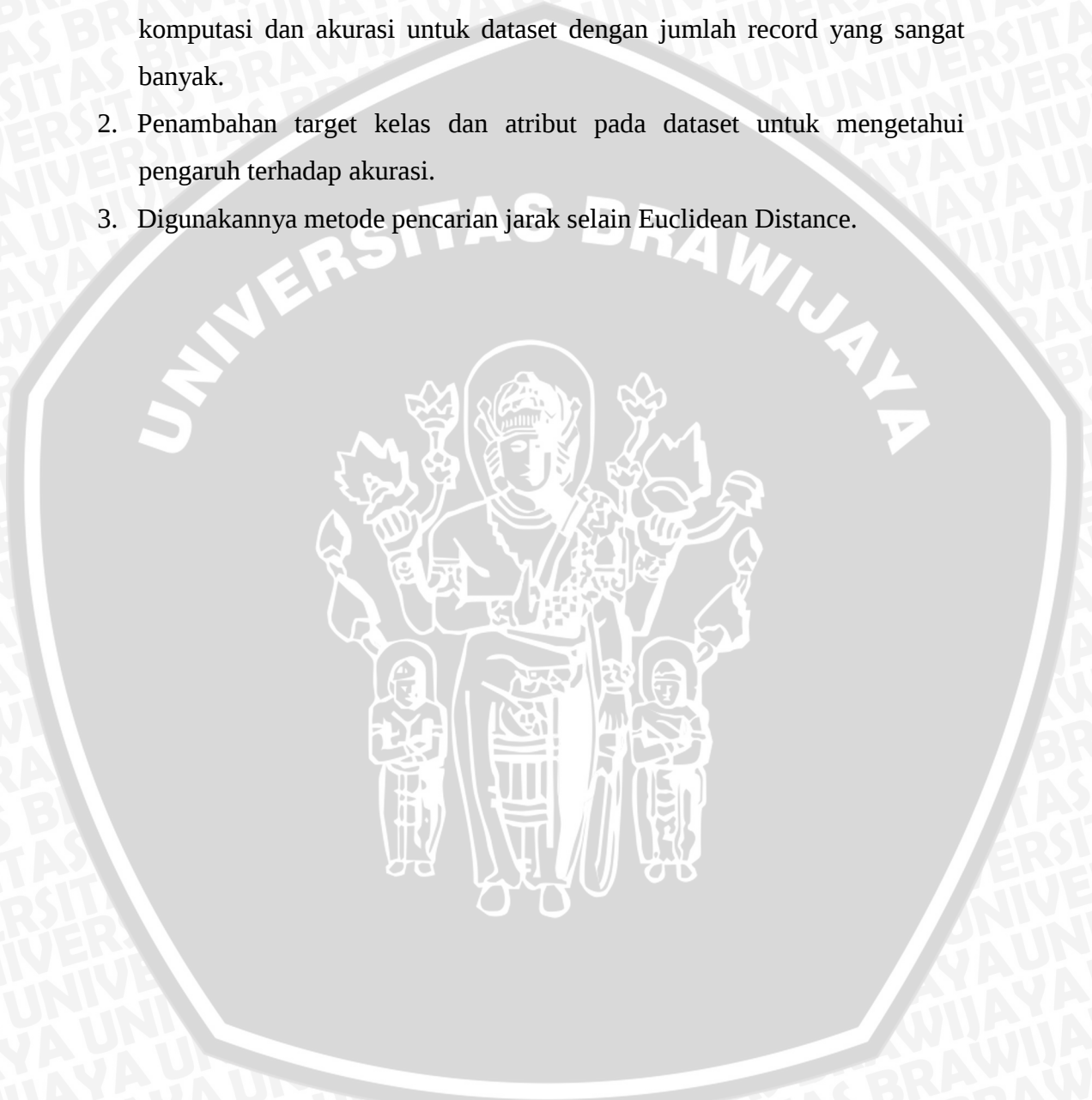
Dari hasil uji dan analisis yang telah dilakukan, dapat diambil kesimpulan antara lain :

1. Data input numerik hasil ekstraksi fitur citra tekstur daun sebanyak sepuluh fitur, yaitu rata-rata intensitas, *smoothness*, entropy, 5 komponen momen invariant meliputi: momen 1, momen 2, momen 4, momen 6, momen 7, energi, kontras dalam file csv sesuai fold dimasukkan dalam sistem dan secara otomatis sistem yang akan menghitung akurasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) lalu menampilkannya dalam grafik.
2. Metode *Fuzzy K-Nearest Neighbor* (FK-NN) menghasilkan nilai akurasi antara 80 % – 90 % dengan predikat baik, yaitu sebesar 81,1574 %.
3. Nilai akurasi sebesar 81,1574 % maka metode *Fuzzy K-Nearest Neighbor* (FK-NN) lebih baik dari metode *K-Nearest Neighbor* (K-NN) dengan nilai akurasi sebesar 54,24 % dan *backpropagation* dengan nilai akurasi sebesar 65,19 % dalam penelitian klasifikasi dataset numerik jenis mangga berdasarkan tekstur daun pada penelitian sebelumnya.
4. Naik turunnya grafik di setiap fold dalam *K-Fold Cross Validation* disebabkan sebaran data latih banyak yang dekat atau jauh dengan sebaran data uji, jika sebaran data latih banyak yang dekat dengan sebaran data uji maka grafik di suatu fold akan naik dan sebaliknya. Maka dari itu teknik *K-Fold Cross Validation* digunakan untuk mencari nilai k terbaik sebelum digunakan untuk prediksi inputan data baru. Dalam penelitian ini k terbaik berada pada k=10.

6.2. Saran

Beberapa saran untuk pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah :

1. Menggunakan metode pengembangan yang dapat meningkatkan waktu komputasi dan akurasi untuk dataset dengan jumlah record yang sangat banyak.
2. Penambahan target kelas dan atribut pada dataset untuk mengetahui pengaruh terhadap akurasi.
3. Digunakannya metode pencarian jarak selain Euclidean Distance.



DAFTAR PUSTAKA

- [AGR-08] Agromedia, Redaksi. 2008. *Menanam dan Membuahkan Mangga Dalam Pot*. PT Agromedia Pustaka. Jakarta
- [AGU-11] Agustin, Soffiana dan Eko Prasetyo. 2011. *Klasifikasi Jenis Pohon Mangga Gadung Dan Curut Berdasarkan Tesktur Daun*. Institut Teknologi Sepuluh November
- [AHM-05] Ahmad, U., 2005. *Pengolahan Citra Digital & Teknik Pemrogramannya, Edisi 1*, Graha Ilmu: Yogyakarta
- [AHM-10] Ahmad, U., 2010. *Pangan*, Vol. 19 No. 1, pp.71-80
- [BRO-03] Broto, Wisnu. 2003. *Mangga: Budi Daya, Pascapanen, dan Tata Niaganya*. PT Agromedia Pustaka. Jakarta
- [FAU-10] Fauzie, R. 2010. *Pengenalan Citra Wajah Menggunakan Algoritme VFI5 Dengan Praproses Principal Component Analysis*. Bogor. Institut Pertanian Bogor
- [GON-08] Gonzalez, R.C, Wood, R.E., 2008. *Digital Image Processing, 3rd Edition*, Pearson Prentice Hall: New Jersey
- [HAI-04] Hairyanov, Mohamad. 2004. *Data Mining Email (Online)*, (<http://budi.insan.co.id/courses/ec7010/dikmenjur-2004/hairyanov-report.pdf>, diakses tanggal 1 Oktober 2012).
- [HAN-00] Han, Jiawei dan Kamber, Micheline. 2000. *Data Mining: Concepts and Techniques*. San Fransisco : Morgan Kaufmann Publishers.

- [KAN-03] Kantardzic, Mehmed. 2003. *Data Mining: Concepts, Models, Methods and Algorithm*. New York : John Wiley and Sons.
- [KEL-85] Keller JM, Gray MR, Givens J.A. 1985. *A Fuzzy K-Nearest Neighbor Algorithm*. IEEE Transaction on Systems, Man, and Cybernetics. Vol 15. No 4:580-585.
- [KUS-07] Kusnawi. 2007. *Pengantar Solusi Data Mining (Online)*, (<http://p3m.amikom.ac.id/p3m/56%20-%20PENGANTAR%20SOLUSI%20DATA%20MINING.pdf>, diakses tanggal 1 Oktober 2012).
- [KUS-04] Kusumadewi. S dan H. Purnomo. 2004. *Aplikasi Logika Fuzzy Untuk Mendukung Keputusan*. Graha Ilmu, Yogyakarta.
- [LEU-07] Leung, K. Ming. 2007. *K-Nearest Neighbor Algorithm for Classification*. Department of Computer Science.
- [MAR-05] Marimin. 2005. *Teori dan aplikasi sistem pakar dalam teknologi manajerial*. IPB –Press, Bogor.
- [MOR-09] Moradian, Mehdi dan Baarani, Ahmad. 2009. *KNNBA: k-Nearest-Neighbor-Based-Association Algorithm*.
- [PRA-03] Pramudiono, Iko. 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. Ilmu Komputer.Com.
- [PRA-12] Prasetyo, Eko. 2012. *Fuzzy K-Nearest Neighbor In Every Class Untuk Klasifikasi Data*. Universitas Pembangunan Nasional Veteran Jawa Timur

- [PRA-10] Pratiwi. 2010. *Pengembangan Model Pengenalan Wajah Dengan Jarak Euclid Pada Ruana Eigen Dengan 2DPCA*. Bogor. Program Pascasarjana, Institut Pertanian Bogor
- [RID-10] Ridok, Achmad dan Muhammad Tanzil Furqon. 2010. *Pengelompokan Dokumen Berbahasa Indonesia Menggunakan Metode K-NN*. Malang. Universitas Brawijaya
- [RUK-97] Rukmana, R., 1997. *Mangga Budidaya dan Pasca panen*, Kanisius: Yogyakarta
- [SAN-11] Santika, Diaz D. 2011. *Pembuatan Model Prediksi Penjualan Pada PT. Kompas Gramedia Dengan Pendekatan Support Vector Regression*. Jakarta. Binus University
- [SAR-00] Sarkar, Manish dan Leong, Tze-Yun. 2000. *Application of K-Nearest Neighbors Algorithm on Breast Cancer Diagnosis Problem*. Singapore : The National University of Singapore.
- [TAN-06] Tan P, Steinbach M, Kumar V. 2006. *Introduction to Data Mining*. Pearson Education: Boston San Fransisco New York.
- [YUK-11] Yuke. 2011. <http://yuke113081041.wordpress.com/2011/10/06/bab-i/>. di akses pada tanggal 24 September 2012

LAMPIRAN

Lampiran 1 Dataset

No.	Fitur										Target
	RTA	SMO	ENT	M1	M2	M4	M6	M7	ENE	KON	
1	22,9308	0,0104	3,5174	5,0157	10,2957	20,7524	26,0901	42,9884	0,2885	46,912	1
2	26,7442	0,0126	3,512	5,3106	11,0406	19,8087	25,4705	40,7678	0,2681	45,6608	0
3	21,8585	0,0101	3,4356	5,0127	10,2944	20,7491	26,1038	41,6225	0,3058	45,4907	1
4	31,7645	0,0191	3,2132	5,4151	11,1654	20,9969	27,427	41,6468	0,2994	55,6361	0
5	25,9265	0,0131	3,5267	5,2557	10,8722	22,2616	28,5474	44,76	0,2885	65,813	1
6	29,8966	0,013	3,6046	5,3714	11,2581	21,0658	26,8844	43,0076	0,2242	46,8415	0
7	24,2602	0,0129	3,2604	5,3013	10,9709	23,4557	30,0072	46,1636	0,3267	42,6226	1
8	29,186	0,0123	3,6001	5,3612	11,2564	21,1341	26,9547	43,0248	0,2223	45,285	0
9	22,1514	0,011	3,3673	5,1437	10,6013	21,5305	27,1572	42,7366	0,3252	37,6108	1
10	30,0384	0,0125	3,5232	5,3783	11,3044	20,3286	26,0532	41,9308	0,2153	43,3852	0
11	26,4007	0,0147	3,5961	5,3119	10,9793	22,6928	28,7302	44,9686	0,3038	66,0016	1
12	29,9325	0,0124	3,5354	5,397	11,3635	20,4353	26,1743	42,2122	0,2161	40,9603	0
13	30,7381	0,0185	3,5061	5,4946	11,3941	20,9706	26,7292	47,4832	0,2978	75,9769	1
14	31,1821	0,0133	3,8381	5,321	11,0843	21,513	27,3052	43,0113	0,2073	64,5236	0
15	27,9176	0,018	3,2303	5,4865	11,374	21,1735	26,9448	45,7487	0,3465	73,9455	1
16	31,8636	0,014	3,8424	5,364	11,1819	21,7538	27,5027	43,6303	0,2093	63,1219	0
17	32,6357	0,0161	3,6921	5,4356	11,2893	22,2441	33,9108	43,6082	0,2339	53,6529	1
18	26,7316	0,0114	3,603	5,3124	11,1195	23,7571	29,5838	48,2862	0,2418	58,346	0
19	34,0326	0,0171	3,7517	5,4777	11,3749	22,7801	30,1354	44,5894	0,229	56,1978	1
20	28,7469	0,0134	3,6849	5,3928	11,2746	23,44	31,7583	48,8011	0,2464	61,4687	0
21	31,9312	0,0154	3,6765	5,4126	11,2388	22,9342	29,1025	45,0863	0,2329	55,4962	1
22	29,3415	0,0132	3,6274	5,3961	11,3008	23,5487	32,2958	45,7782	0,2352	61,8965	0
23	31,3799	0,0156	3,6244	5,4179	11,2478	23,0439	28,8683	47,9436	0,2452	52,4196	1

24	28,3529	0,0132	3,5706	5,4094	11,3367	24,3451	30,6306	48,1419	0,2511	62,9698	0
25	27,9487	0,0149	3,1779	5,4533	11,3738	21,1046	26,7936	44,814	0,2987	47,1112	1
26	32,8221	0,0156	3,7852	5,5173	11,5596	21,6605	27,5898	44,0239	0,2217	68,3875	0
27	29,704	0,0156	3,4396	5,4471	11,373	22,0218	27,7131	46,1071	0,2722	73,2776	1
28	33,4647	0,0153	3,8466	5,5244	11,5918	21,7392	27,6525	44,2706	0,2074	65,4779	0
29	30,1586	0,0165	3,2758	5,4947	11,4474	21,7318	27,5095	45,766	0,2855	48,5624	1
30	33,8383	0,0156	3,8677	5,5295	11,5959	21,681	27,6063	44,2342	0,2069	69,1944	0
31	30,3328	0,0159	3,4121	5,51	11,4994	21,7591	27,5463	46,4551	0,2706	53,0058	1
32	33,4208	0,0156	3,8452	5,5306	11,5986	21,753	27,6848	44,2779	0,2128	68,653	0
33	36,3493	0,0218	3,5343	5,8069	12,2852	22,8607	29,0412	47,7408	0,261	62,4773	1
34	32,5496	0,0158	3,5935	5,4015	11,1916	21,2392	27,249	42,8177	0,2321	72,1401	0
35	37,027	0,0222	3,5208	5,8172	12,3067	23,1994	29,4193	47,5247	0,2574	63,4462	1
36	31,3359	0,0138	3,6672	5,3463	11,0873	21,1085	27,0275	42,6598	0,2151	70,1101	0
37	35,5074	0,0223	3,3934	5,8222	12,3151	23,1075	29,3006	47,6737	0,2815	61,0906	1
38	33,0079	0,0158	3,6762	5,4128	11,2234	21,5474	27,697	43,1419	0,2238	75,2294	0
39	33,8422	0,021	3,3648	5,7678	12,1696	22,5781	28,8225	46,1383	0,2895	57,2624	1
40	33,0212	0,0153	3,6986	5,4074	11,2113	21,6648	27,785	43,4036	0,216	71,0647	0
41	30,3104	0,0166	3,4706	5,456	11,2992	21,5149	27,3216	44,2937	0,2805	52,0211	1
42	28,3778	0,0109	3,5533	5,2621	10,9682	20,3656	26,0351	41,417	0,2059	60,7486	0
43	29,7622	0,0167	3,3821	5,4524	11,2859	21,827	27,7229	44,4432	0,2923	52,3289	1
44	30,7768	0,0115	3,7104	5,3203	11,1108	21,3824	27,3508	42,5751	0,1816	70,368	0
45	31,405	0,0186	3,3846	5,5088	11,3972	22,4403	28,4759	45,2711	0,2937	56,0111	1
46	30,9377	0,0112	3,7929	5,3175	11,1123	21,6944	27,8371	42,8845	0,1737	70,0519	0
47	35,7292	0,0196	3,7924	5,5175	11,4263	22,6851	29,1374	45,213	0,2364	79,7554	1
48	31,5011	0,0108	3,8728	5,31	11,1034	21,4415	27,4029	42,6441	0,1573	68,8046	0
49	29,1729	0,0155	3,4826	5,5427	11,6998	22,9311	29,0055	44,8459	0,2786	67,7212	1
50	35,2274	0,0153	4,265	5,4728	11,5021	23,216	29,9284	45,8194	0,1738	83,7451	0
51	33,9193	0,023	3,3328	5,7097	11,9975	21,7963	28,5323	44,7507	0,3116	84,3237	1
52	34,4043	0,0144	4,2666	5,4251	11,3855	23,2463	29,6159	46,1905	0,1704	86,0139	0
53	29,0943	0,0156	3,5126	5,5885	11,8344	24,5504	30,7731	48,3301	0,2784	69,3236	1

54	27,2064	0,0103	3,7283	5,3438	11,2981	22,0147	29,7404	43,1424	0,205	68,9388	0
55	29,0359	0,0148	3,4111	5,5716	11,8278	24,376	30,9883	47,6693	0,2721	78,6404	1
56	26,4059	0,0102	3,664	5,3492	11,3047	21,3572	27,5919	42,5218	0,2204	57,0943	0
57	30,0713	0,0158	3,5678	5,5993	11,8586	24,2964	30,4569	48,0587	0,2659	69,6455	1
58	27,9246	0,01	3,9025	5,3477	11,3233	21,2033	27,3976	42,338	0,185	63,55	0
59	30,6568	0,0159	3,5851	5,6038	11,87	23,9468	30,0445	47,65	0,2584	68,567	1
60	30,2173	0,0111	4,0013	5,4143	11,4539	21,3308	27,4956	42,5375	0,1755	59,8338	0

