

**PENERAPAN METODE *FUZZY K-NEAREST NEIGHBOR*
(FK-NN) UNTUK MENENTUKAN KUALITAS HASIL
RENDEMEN TANAMAN TEBU**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

RAHMI AMIRATUS SHOFA

NIM. 0910960013

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA DAN ILMU KOMPUTER

MALANG

2013

LEMBAR PERSETUJUAN

**PENERAPAN METODE *FUZZY K-NEAREST NEIGHBOR* (FK-NN)
UNTUK MENENTUKAN KUALITAS HASIL**

RENDEMEN TANAMAN TEBU

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

RAHMI AMIRATUS SHOFA

NIM. 0910960013

Skripsi ini telah disetujui oleh dosen pembimbing
pada tanggal 24 Desember 2013

Dosen Pembimbing I

Dosen Pembimbing II

Lailil Muflikhah, S.Kom, M.Sc

NIP.197411132005012001

Drs. Achmad Ridok, M.Kom

NIP. 19680825 199403 1 002

LEMBAR PENGESAHAN

**PENERAPAN METODE *FUZZY K-NEAREST NEIGHBOR* (FK-NN)
UNTUK MENENTUKAN KUALITAS HASIL**

RENDEMEN TANAMAN TEBU

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

RAHMI AMIRATUS SHOFA

NIM. 0910960013

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 24 Desember 2013

Penguji I

Penguji II

Suprpto, ST., MT.

Edy Santoso, S.Si., M.Kom

NIP. 19710727 196603 1001

NIP. 197404142003121004

Penguji III

Imam Cholissodin, S.Si., M.Kom.

NIP. 850719 16 1 1 0422

Mengetahui

Ketua Program Studi Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

**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, 2 Januari 2014

Mahasiswa,

RAHMI AMIRATUS SHOFA

NIM 0910960013

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **“Penerapan Metode Fuzzy K-Nearest Neighbor (FK-NN) untuk Menentukan Kualitas Hasil Rendemen Tanaman Tebu ”** ini dapat disusun dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Pada kesempatan ini penulis mengucapkan banyak terima kasih atas segala bantuan dan dedikasi moral maupun material dalam rangka penyusunan skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Lailil Muflikhah.S.Kom,M.Sc, selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
2. Drs.Achmad Ridok,M.kom, selaku pembimbing kedua yang telah banyak memberikan bimbingan dalam penulisan skripsi ini.
3. Drs. Marji, MT, selaku Ketua Program Studi Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, MT, selaku Dosen Penasehat Akademik.
5. Ir. Sutrisno, MT, selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
7. Segenap staf dan karyawan di Program Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Mama Diana, Bapak Sutrisno, Mas Fahmi, Mbak Widia dan juga seluruh keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.

9. Rekan-rekan Program Studi Informatika/Ilmu Komputer yang telah memberikan dukungannya kepada penulis.
10. Sahabat-sahabat saya khususnya yanita, ayu, melati, nanik, arintha, diantika, cahyo, aldi, mas indra, mamluatul, kicik yang telah banyak memberikan dukungan dan nasihat.
11. Keluarga kos kertoasri 71 yang telah banyak memberikan motivasi kepada saya.
12. Semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu

Penulis menyadari bahwa skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat penulis harapkan dari berbagai pihak demi penyempurnaan penulisan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, 25 November 2013

RAHMI AMIRATUS SHOFA

NIM. 0910960013

HALAMAN PERSEMBAHAN

Alhamdulillah, terima kasih kepada Allah SWT yang telah memudahkan penulis dalam proses pengerjaan skripsi ini. Puji syukur selalu penulis panjatkan kehadirat-Nya. Skripsi ini khusus penulis persembahkan untuk para pembaca yang bersedia membaca skripsi telah penulis susun, semoga setelah membaca skripsi ini pembaca mendapatkan ide dalam penyusunan skripsinya.

UNIVERSITAS BRAWIJAYA



ABSTRAK

Rahmi Amiratus Shofa. 2013. Penerapan Metode *Fuzzy k-Nearest Neighbor (FK-NN)* untuk Menentukan Kualitas Hasil Rendemen Tanaman Tebu. Skripsi Program Studi Teknik Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing : Lailil Muflikhah,S.Kom,M.Sc dan Drs.Achmad Ridok,M.kom.

Penelitian ini membahas mengenai penerapan metode *fuzzy k-nearest neighbor (fk-nn)* untuk menentukan kualitas hasil rendemen tanaman tebu. Bahan pokok pembuatan gula hampir sebagian besar berasal dari tanaman tebu. Kadar kandungan gula yang dihasilkan dari proses pengolahan nira tebu disuatu pabrik gula biasa disebut dengan rendemen. Algoritma *fuzzy k-nearest neighbor* dapat digunakan untuk mengklasifikasikan kualitas hasil rendemen tanaman tebu berdasarkan faktor-faktor yang berpengaruh (panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira). Pengujian ini dilakukan terhadap 4 macam data latih yaitu 100, 125, 150, dan 175. Pengujian akurasi juga dilakukan pada empat macam jumlah data uji yaitu sebanyak 50, 60, 70, 80 data Uji. Pengujian dilakukan pada tiga tahapan yaitu pengujian *k* untuk mengetahui pengaruhnya terhadap tingkat akurasi, pengujian pengaruh jumlah data latih terhadap tingkat akurasi, dan pengujian pengaruh jumlah data uji terhadap akurasi. Akurasi tertinggi yang didapatkan pada pengujian ini adalah sebesar 98%. Penelitian ini menunjukkan bahwa algoritma *fuzzy k-nearest neighbor* memiliki kinerja yang baik dalam klasifikasi kualitas hasil rendemen tanaman tebu.

Kata kunci : klasifikasi, rendemen, *fuzzy k-nearest neighbor*

ABSTRACT

Rahmi Amiratus Shofa. 2013. Implementation of Fuzzy k- Nearest Neighbor Algorithm to Determine Quality of Cane Rendemen. Minor Thesis Program of Study Information Technology / Computer Science, Program of Technology Information and Computer Science University of Brawijaya. Advisor : Lailil Muflikhah,S.Kom,M.Sc dan Drs.Achmad Ridok,M.kom.

This research discuss implementation of fuzzy k-nearest neighbor (fk-nn) algorithm to determine quality of cane rendemen. Cane is raw material of sugar. The quality value of cane that processed into sugar in sugar factory commonly called rendemen. The fuzzy k-nearest neighbor algorithm can be used to classify the quality of cane rendemen based the factors that influence (length of cane(m), diameter of cane(cm), segments cane, weight of cane(kg), % brix, HK, Nira). In this research, number of training data that used are 100, 125, 150 and 175. And also the number of test data that use are 50, 60, 70, 80. Experimental performed into three phase namely, k testing to determine its effect on the level of accuracy, testing the effect number of data training on the level of accuracy and testing the effect number of test data on the level of accuracy. The highest levels of accuracy reaches 98%. This experiment shows that the fuzzy k-nearest neighbor algorithm has a good performance in the classification of the quality of cane rendemen.

Keyword: classification, rendemen, fuzzy k-nearest neighbor

DAFTAR ISI

KATA PENGANTAR	i
HALAMAN PERSEMBAHAN	iii
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR SOURCE CODE	xi
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
 BAB II KAJIAN PUSTAKA DAN DASAR TEORI	 6
2.1 Kajian Pustaka	6
2.2 Tanaman Tebu(<i>Saccharum officinarum L</i>)	7
2.2.1 Definisi Tanaman Tebu	7
2.2.2 Rendemen Tanaman Tebu	7
2.2.3 Faktor yang Mempengaruhi Kualitas Rendemen	8
2.2.4 Pembagian Kelas Kualitas Rendemen Tebu	10
2.3 Logika Fuzzy	11
2.3.1 Pengertian Logika Fuzzy	11
2.3.2 Fuzzy Set(Himpunan Fuzzy)	12
2.3.3 Fungsi Keanggotaan (<i>Membership Function</i>)	12

2.4 Data Mining	14
2.4.1 Pengertian Data Mining	14
2.4.2 Klasifikasi	16
2.4.3 K-Nearest Neighbor (K-NN)	16
2.4.4 Proses K-Nearest Neighbor.....	17
2.5 Fuzzy K- Nearest Neighbor Classifier	18
2.5.1 Pengertian Fuzzy K-Nearest Neighbor	18
2.5.2 Algoritma Fuzzy K-Nearest Neighbor.....	20
2.6 Akurasi Hasil Pengujian.....	21
 BAB III METODE PENELITIAN	22
3.1 Studi Literatur.....	23
3.2 Pengumpulan Data	23
3.3 Deskripsi Sistem	24
3.4 Perancangan Sistem	24
3.4.1. Proses Normalisasi	26
3.4.2. Proses K -Nearest Neighbor	27
3.4.3 Transformasi Kedalam Bentuk Fuzzy	28
3.4.4 Fuzzy K-Nearest Neighbor.....	33
3.5 Perhitungan Manual	34
3.6 Perancangan Antarmuka	41
3.7 Perancangan Uji Coba.....	46
3.7.1 Uji Pengaruh Nilai K terhadap Tingkat Akurasi.....	47
3.7.2 Uji Pengaruh Jumlah Data Latih Terhadap Akurasi	47
3.7.3 Uji Pengaruh Jumlah Data Uji Terhadap Akurasi	48
 BAB IV PERANCANGAN IMPLEMENTASI.....	49
4.1 Lingkungan Implementasi	49
4.1.1 Lingkungan Implementasi Perangkat Keras	49
4.1.2 Lingkungan Implementasi Perangkat Lunak	49
4.2 Implementasi Program	50
4.2.1 Proses Penginputan Data Latih dan Data Uji.....	50

4.2.2 Proses Perhitungan Normalisasi.....	52
4.2.3 Proses Perhitungan Nilai Jarak <i>Euclidean</i>	54
4.2.4 Proses Sorting Data	56
4.2.5 Proses Pengambilan Data Berdasarkan Nilai <i>k</i>	57
4.2.6 Proses Perhitungan Nilai <i>Membership Function</i>	57
4.2.7 Proses Perhitungan <i>Fuzzy K-Nearest Neighbor</i>	59
4.3 Implementasi Antar Muka.....	60
4.3.1 Form Klasifikasi <i>Fuzzy K-Nearest Neighbor</i>	61
4.3.2 Form Proses <i>Fuzzy K-Nearest Neighbor</i>	62
4.3.3 Form Hasil FK-NN Data Uji.....	63
4.4.4 Form Akurasi	63
4.4.5 Form Akurasi (Berdasarkan Nilai <i>k</i>)	64
BAB V PENGUJIAN DAN ANALISIS.....	66
5.1 Implementasi Pengujian	66
5.1.1 Uji Pengaruh <i>k</i> Terhadap Tingkat Akurasi.....	66
5.1.2 Uji Pengaruh Jumlah Data Latih Terhadap Tingkat Akurasi	67
5.1.3 Uji Pengaruh Jumlah Data Uji Terhadap Tingkat Akurasi.....	68
5.2 Analisa Hasil.....	69
5.2.1 Analisa Hasil Pengaruh Nilai <i>k</i> Terhadap Tingkat Akurasi	69
5.2.2 Analisa Hasil Pengaruh Jumlah Data Latih Terhadap Akurasi ...	70
5.2.3 Analisa Hasil Pengaruh Jumlah Data Uji Terhadap Akurasi.....	71
BAB VI PENUTUP	73
6.1 Kesimpulan.....	73
6.2 Saran.....	74
DAFTAR PUSTAKA	75
LAMPIRAN	77
Lampiran 1. 175 Data Latih Klasifikasi Rendemen Tebu	77
Lampiran 2. 80 Data Uji Klasifikasi Rendemen Tebu.....	81
Lampiran 3. Uji Pengaruh nilai <i>k</i> pada 175 Data Latih dan 80 Data Uji.....	83

DAFTAR GAMBAR

Gambar 2.1 Representasi Kurva Bentuk Bahu.....	13
Gambar 2.2 Kurva Kelas Kualitas Rendemen Tanaman Tebu.....	14
Gambar 3.1 Diagram Alur Penelitian	22
Gambar 3.2 Proses Klasifikasi	25
Gambar 3.3 Alur Proses Normalisasi.....	26
Gambar 3.4 Proses K-Nearest Neighbor	27
Gambar 3.5 Pengambilan Data Berdasarkan k.....	28
Gambar 3.6 Transformasi Output kedalam Bentuk Fuzzy	29
Gambar 3.7 Proses perhitungan nilai <i>membership</i> buruk	30
Gambar 3.8 Proses perhitungan nilai <i>membership</i> baik.....	31
Gambar 3.9 Proses perhitungan nilai <i>membership</i> sangat baik.....	32
Gambar 3.10 Proses perhitungan Fuzzy K-Nearest Neighbor	33
Gambar 3.11 Rancangan Antarmuka Klasifikasi FK-NN.....	42
Gambar 3.12 Rancangan Antarmuka Proses FK-NN	43
Gambar 3.13 Rancangan Antarmuka Hasil FK-NN Data Uji.....	44
Gambar 3.14 Rancangan Antarmuka Akurasi Sistem	45
Gambar 3.15 Rancangan Antarmuka Akurasi (Berdasarkan Nilai k)	46
Gambar 4.1 Antarmuka Klasifikasi FK-NN	61
Gambar 4.2 Antarmuka Proses FK-NN	62
Gambar 4.3 Antarmuka Hasil FK-NN Data Uji.....	63
Gambar 4.4 Antarmuka Proses Akurasi Sistem	64
Gambar 4.5 Antarmuka Proses Akurasi (Berdasarkan Nilai k)	65
Gambar 5.1 Grafik Pengaruh Nilai k Terhadap Tingkat Akurasi.....	69
Gambar 5.2 Grafik Pengaruh Jumlah Data Latih terhadap Akurasi.....	70
Gambar 5.3 Grafik Pengaruh Jumlah Data Uji Terhadap Akurasi.....	71

DAFTAR TABEL

Tabel 2.1 Rentang Nilai Kelas Kualitas Rendemen Tanaman Tebu	10
Tabel 3.1 Data Pelatihan Sistem.....	34
Tabel 3.2 Data Uji Sistem	35
Tabel 3.3 Range Data Training	36
Tabel 3.4 Data Latih yang Telah di Normalisasi	36
Tabel 3.5 Data Uji yang Telah di Normalisasi	36
Tabel 3.6 Nilai Jarak Kedekatan Antara Data Uji dan 20 Data Latih	37
Tabel 3.7 Nilai Jarak Setelah di Urutkan	38
Tabel 3.8 Seleksi 5 Data dengan Nilai Jarak Terdekat	38
Tabel 3.9 Target dalam <i>Linguistic Membership</i>	39
Tabel 3.10 Nilai <i>Fuzzy</i> untuk Setiap Target.....	40
Tabel 3.11 Nilai <i>Membership</i> Akhir	41
Tabel 3.12 Rancangan Uji Pengaruh Nilai k Terhadap Akurasi Sistem.....	47
Tabel 3.13 Rancangan Uji Pengaruh Jumlah Data Latih Terhadap Akurasi	48
Tabel 3.14 Rancangan Uji Pengaruh Jumlah Data Uji Terhadap Akurasi.....	48
Tabel 5.1 Uji Pengaruh Nilai k Terhadap Akurasi	66
Tabel 5.2 Uji Pengaruh Jumlah Data Latih Terhadap Akurasi	68
Tabel 5.3 Uji Pengaruh Jumlah Data Uji Terhadap Akurasi.....	68

DAFTAR SOURCE CODE

Source Code 4.1 Pembacaan Data Latih	50
Source Code 4.2 Pembacaan Data Uji	51
Source Code 4.3 Perhitungan Normalisasi	52
Source Code 4.4 Perhitungan Jarak <i>Euclidean</i>	54
Source Code 4.5 Proses Sorting Data	56
Source Code 4.6 Pengambilan Data Berdasarkan k	57
Source Code 4.7 Proses Perhitungan Nilai <i>Membership Function</i>	58
Source Code 4.8 Proses Perhitungan <i>Fuzzy K-Nearest Neighbor</i>	59



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pembangunan pertanian di Indonesia tetap dianggap penting dari keseluruhan pembangunan ekonomi, apalagi semenjak sektor pertanian menjadi penyelamat perekonomian nasional dimasa krisis 1997–1999, dikarenakan pertumbuhan yang meningkat. Semakin pesatnya pertumbuhan penduduk menyebabkan kebutuhan akan pangan termasuk gula terus mengalami peningkatan permintaan [MAR-11]. Gula merupakan komoditas penting dalam pertanian dan merupakan salah satu kebutuhan pokok masyarakat. Bahan pokok pembuatan gula hampir sebagian besar berasal dari tanaman tebu.

Tebu (*Saccharum officinarum*) merupakan tanaman Graminae atau rumput-rumputan. Tanaman tebu merupakan golongan tanaman yang tumbuh di daerah beriklim sedang sampai panas, yaitu terletak di antara 40° LU dan 38° LS. Pada saat fase pertumbuhan, tanaman tebu membutuhkan banyak air akan tetapi setelah tua (6-8 bulan) dan pada saat proses pemasakan atau panen (12-14 bulan) tanaman tebu membutuhkan bulan kering dan ini sebaiknya tiba pada saat berakhirnya pertumbuhan vegetatif. Bila musim kering tiba sebelum pertumbuhan vegetatif berakhir, maka tanaman tebu yang tidak diairi akan mati sebelum mencapai tingkat masak, sebaliknya bila hujan turun terus-menerus maka pertumbuhan vegetatif tebu tetap giat, sehingga tidak mencapai kadar gula tertinggi [YUK-07].

Kadar kandungan gula yang dihasilkan dari proses pengolahan nira tebu disuatu pabrik gula biasa disebut dengan rendemen. Dalam konteks Indonesia, faktor rendemen menjadi sangat penting dikarenakan tebu yang dihasilkan oleh petani tidak langsung diolah melainkan dijual kepada pihak pabrik gula (PG). Petani menyerahkan hasil tebu kepada pihak pabrik gula untuk diolah menjadi gula. Perhitungan pembagian yang ditetapkan berdasarkan jumlah gula yang dihasilkan dengan sistem bagi hasil. Kurangnya pengetahuan petani dan kerumitan dalam pengukuran rendemen menimbulkan kecurigaan PG

memanipulasi rendemen gula. Dilain pihak, PG menilai mutu tebu kurang baik dan petani hanya mengejar bobot tebu saja.

Semakin pesatnya teknologi komputer memungkinkan untuk membantu peran seseorang dalam menentukan kualitas rendemen yang dihasilkan oleh tanaman tebu. Untuk penentuan kualitas rendemen yang dihasilkan oleh tanaman tebu dibutuhkan pengetahuan (knowledge) yang dirancang untuk pembangunan sistem perangkat lunak. Pengetahuan tersebut berupa cara bagaimana mengklasifikasikan kualitas hasil rendemen tebu berdasarkan faktor-faktor yang mempengaruhinya. Faktor-faktor yang digunakan adalah panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira.

Metode *K-Nearest Neighbor* (*K-NN*) merupakan metode klasifikasi data yang cara kerjanya bersifat relative lebih sederhana bila dibandingkan dengan metode klasifikasi pada data mining lainnya. Dimana metode ini melakukan klasifikasi kepada data baru yang masih belum diketahui masuk kedalam kelas mana, dengan menggunakan beberapa data dengan sejumlah k yang letaknya terdekat dengan data baru tersebut. *K-Nearest Neighbor* memiliki kelebihan, diantaranya adalah memiliki ketangguhan dalam mengelola data yang berukuran sangat besar dan memiliki banyak noise[RID-13]. Tetapi dibalik itu *K-NN* juga menyimpan kelemahan yaitu butuh komputasi yang tinggi akibat perlunya menghitung satu persatu data testing terhadap semua data training (tidak ada model yang dibentuk).

Metode *k-nearest neighbor* dapat digabungkan dengan metode *fuzzy*, dan metode ini biasa disebut dengan *fuzzy k-nearest neighbor*, dimana metode ini menggabungkan teknik *fuzzy* dengan teknik data mining [ZHA-09]. Pada penelitian ini digunakan metode *fuzzy* dengan gabungan metode *k-nearest neighbor* untuk mengklasifikasikan data .

Fuzzy k-nearest neighbor sebelumnya telah digunakan untuk memprediksi Iris oleh [KEL-85], Prediksi tingkat resiko penyakit Jantung Koroner, oleh [PUS-13], memprediksi Iris dan Vertebral Column yang mencapai akurasi sebesar 97% oleh [PRA-12] dan *Web Clasificaion Document* oleh [ZHA-09] . *Fuzzy k-nearest neighbor* memiliki dua keunggulan utama daripada algoritma *k-nearest neighbor*.

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.

Berdasarkan latar belakang yang telah diuraikan, maka skripsi ini diberi judul **“Penerapan metode *FUZZY K-NEAREST NEIGHBOR (FK-NN)* untuk menentukan kualitas hasil rendemen tanaman tebu”**. Data yang digunakan untuk klasifikasi kualitas hasil rendemen tebu berasal dari PG Semboro yang terdiri dari 7 atribut yaitu panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira.

1.2 Rumusan Masalah

Dari latar belakang di atas dapat diambil suatu rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasi metode *fuzzy k-nearest neighbor* untuk menentukan kualitas hasil rendemen tebu berdasarkan faktor – faktor yang berpengaruh (panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira)
2. Bagaimana pengaruh jumlah nilai k (tetangga) serta data latih terhadap tingkat akurasi hasil penentuan kualitas hasil rendemen tebu menggunakan metode *fuzzy k-nearest neighbor*.

1.3 Batasan Permasalahan

Pada skripsi ini, permasalahan dibatasi oleh hal-hal sebagai berikut:

1. Data yang digunakan untuk pengujian pada skripsi ini didapatkan dari data klasifikasi tebu studi kasus PG Semboro yang merupakan data analisa kemasakan tebu pada tahun 2012-2013.
2. Parameter yang digunakan untuk menentukan kualitas hasil rendemen tebu adalah panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira.

3. Terdapat tiga kategori hasil klasifikasi yaitu buruk, baik, dan sangat baik.
4. Metode klasifikasi data yang digunakan adalah metode *k-nearest neighbor*, yang nantinya digabungkan dengan metode *fuzzy* untuk memperoleh hasil output akhir.

1.4 Tujuan Penelitian

Tujuan dari penelitian untuk pembuatan skripsi ini adalah:

1. Mengimplementasi metode *fuzzy k-nearest neighbor* untuk menentukan kualitas hasil rendemen tebu berdasarkan faktor – faktor yang berpengaruh (panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira)
2. Mengetahui pengaruh sejumlah nilai *k* (tetangga) terhadap tingkat akurasi serta data latih metode *fuzzy k-nearest neighbor* untuk menentukan kualitas hasil rendemen tebu.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari skripsi ini yaitu dapat diklasifikasikannya kualitas hasil gula tebu menggunakan *fuzzy k-nearest Neighbor (fk-nn)*.

1.6 Sistematika Penulisan

Penyusunan skripsi ini berdasarkan sistematika penulisan sebagai berikut:

1. BAB I : PENDAHULUAN

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

2. BAB II : KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori dan bahan penelitian sebelumnya yang berkaitan dengan topik penulisan skripsi. Teori tersebut meliputi penjelasan mengenai tanaman tebu, metode *Fuzzy K-Nearest Neighbor* dan penelitian-penelitian sebelumnya yang berhubungan dengan tanaman tebu.

3. BAB III : METODE PENELITIAN

Bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari perencanaan, penelitian, analisis, perancangan, konstruksi, implementasi, dan penulisan laporan dengan studi kasus kualitas hasil rendemen tanaman tebu.

4. BAB IV : PERANCANGAN

Bab ini membahas spesifikasi perangkat lunak, perancangan system, perancangan antarmuka dengan studi kasus kualitas hasil rendemen tanaman tebu.

5. BAB V : PENGUJIAN DAN ANALISA

Bab ini berisi pembahasan dari implementasi penentuan kualitas hasil rendemen tanaman tebu menggunakan klasifikasi *Fuzzy K-Nearest Neighbor (FK-NN)* pada sistem dan hasil pengujian yang dilakukan.

6. BAB VI : PENUTUP

Bab ini berisi kesimpulan dari seluruh rangkaian penelitian serta saran-saran untuk kemungkinan dilakukan pengembangannya



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Penelitian ini membahas tentang identifikasi kualitas hasil rendemen tebu menggunakan metode *fuzzy k-nearest neighbor*(*fk-nn*). Penelitian serupa telah dilakukan oleh Maslikha Puspasari (2013) dengan memfokuskan prediksi tingkat resiko penyakit jantung koroner menggunakan metode *fuzzy k-nearest neighbor*.

Puspasari menggunakan metode ini untuk memprediksi tingkat resiko jantung koroner yang mana dipengaruhi oleh faktor usia, *LDL*(*low density lipoprotein*), kolesterol total, *HDL*(*high density lipoprotein*), trigleserida, tekanan darah sistolik. Berdasarkan faktor-faktor yang mempengaruhi tersebut,dilakukan proses klasifikasi untuk mengetahui resiko penyakit jantung koroner. Pada penelitian ini digunakan pembobotan atau *weighted* dalam penerapan metode *fuzzy k-nearest neighbor*[PUS-13].

Sedangkan penelitian ini berfokus pada penentuan kualitas hasil rendemen tebu. Adapun faktor-faktor yang mempengaruhi untuk menentukan kualitas hasil rendemen tebu adalah panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira. Faktor-faktor tersebut itulah yang nantinya digunakan dalam pengklasifikasian menggunakan metode *fuzzy k-nearest neighbor* untuk menentukan kualitas hasil rendemen tebu. Berbeda dengan penelitian sebelumnya, dalam proses penerapan metodenya tanpa menggunakan *weighted* atau pembobotan.

2.2 Tanaman Tebu (*Saccharum officinarum* L)

2.2.1 Definisi Tanaman Tebu

Tebu memiliki nama ilmiah *Saccharum officinarum* L .Tanaman ini merupakan jenis rumput-rumputan, termasuk kelas *Monocotyledonae*, ordo *Glumiflorae*, keluarga *Gramineae*. Terdapat lima spesies tebu yang diketahui yaitu *Saccharum spontaneum* (glagah), *Saccharum sinensis* (tebu Cina), *Saccharum barberry* (tebu India), *Saccharum robustum* (tebu Irian) dan *Saccharum officinarum* (tebu kunyah). Tanaman tebu termasuk golongan tanaman yang tumbuh di daerah beriklim sedang sampai panas, yaitu terletak di antara 40° LU dan 38° LS. Sifat morfologi yang dimiliki tebu antara lain memiliki bentuk batang konis, susunan antar ruas berbuku, dengan penampang melintang agak pipih, warna batang hijau kekuningan, batang memiliki lapisan lilin tipis, bentuk buku ruas konis terbalik 3-4 baris mata akar, daun memiliki warna hijau kekuningan, lebar daun sekitar 4-6 cm, daun melengkung kurang dari setengah panjang daun [AND-11].

Tanaman tebu merupakan bahan utama pembuatan gula sehingga bisa dikatakan bahwa tanaman tebu memiliki peranan penting dalam industri gula. Perindustrian gula di Indonesia dijalin atas dasar pola kerja sama antara petani tebu dengan pihak pabrik gula. Petani sebagai pemasok tebu, sedangkan pihak pabrik gula sebagai pengelola tebu sampai menjadi gula. Dalam industri gula pasti mengenal istilah rendemen. Rendemen merupakan kadar kandungan gula didalam tanaman tebu yang dinyatakan dalam bentuk persen.

2.2.2 Rendemen Tanaman Tebu

Secara spesifik rendemen bisa dijelaskan sebagai berikut, bila dikatakan rendemen tebu 10% berarti bahwa dalam 100 kg tebu yang digiling di pabrik gula akan didapat sebanyak 10 kg gula. Ada tiga macam rendemen tebu yaitu rendemen contoh, rendemen sementara, dan rendemen efektif [LAT-07]. Untuk memperoleh nilai rendemen maka dilakukan upaya analisa rendemen individu.

Analisa rendemen individu merupakan bentuk penghargaan upaya yang dilakukan oleh pengelola tanaman tebu dimulai dari awal masa tanam sampai pengiriman tanaman tebu ke pabrik gula. Untuk memperoleh rendemen per individu, dilakukan analisa tanaman tebu dari masing-masing truk atau lori. Mekanisme pelaksanaan analisa rendemen individu sebagai berikut:

a) Selektor I

- Pengambilan sampel tebu sebanyak 2-3 lonjor
- Dipotong menjadi dua bagian
- Selanjutnya digiling menggunakan alat giling kecil diulang sebanyak dua kali
- Dilakukan analisa kandungan brix dengan alat Hand Brix Refraktometer dari hasil nira yang didapatkan
- Hasil nilai brix dituliskan ke SPAT (Surat perintah angkut tebu)
- Apabila angka brix memenuhi nilai standart yaitu 17 maka dapat dilakukan penimbangan dan penggilingan terhadap tanaman tebu.

b) Selektor II

- Diawali dengan mengamati mutu tebu diantaranya: kebersihan, kesegaran tebu, tebu cacahan, tebu blondolan tebu muda, tebu kering diatas meja tebu (Cane Table)
- Dari hasil pengamatan dituliskan ke SPAT kondisi tebu yang ada
- SPAT yang telah diisi hasil pengamatan brix dari selector I dan pengamatan mutu tebu dari selector II dikirimkan ke Laboratorium Analisa Rendemen Individu [BAS-10].

2.2.3 Faktor yang Mempengaruhi Kualitas Rendemen

Beberapa faktor yang mempengaruhi kualitas hasil rendemen diantaranya adalah ON farm (Kualitas tebu, Varietas, Kadar Nira Tebu, Manajemen Tebang Angkut), dan Off farm (Efisiensi Pabrik) [BAS-10]. Namun di dalam bab ini akan dibahas mengenai faktor-faktor yang berpengaruh terhadap kualitas rendemen tebu dan menjadi batasan masalah dari penelitian ini. Pengambilan faktor-faktor yang berpengaruh terhadap kualitas rendemen tebu pada penelitian ini sebagian

besar diambil dari nilai yang berpengaruh terhadap varietas dan beberapa hal yang mempengaruhi kualitas nira tebu. Analisis kualitas Nira meliputi %brix, %pol, PH, gula reduksi, Harkat Kemurnian (HK), dan nilai nira perahan pertama. Adapun faktor-faktor tersebut diantaranya:

a) Batang tebu

Adapun beberapa hal yang berkaitan dengan batang tebu adalah Panjang Batang Tebu (meter), Diameter Batang Tebu (cm), Berat Tebu (per meter). Pada pertumbuhan tebu yang baik akan didapat tinggi batang 3-5 meter pada setiap tanaman tebu [BAS-10]. Pertumbuhan tinggi batang tebu yang berbeda tersebut menyebabkan panjang batang tebu yang dihasilkan pada setiap tanaman memiliki ukuran yang berbeda pula, sedangkan diameter pada setiap batang tebu diukur dengan satuan baku cm (Centimeter).

b) Ruas Tebu (per batang)

Perbedaan varietas tebu menyebabkan perbedaan panjang ruas tebu. Panjang ruas tebu makin panjang dari pangkal sampai pertengahan batang tebu dan selanjutnya makin pendek ke ujung tanaman tebu. Apabila tanaman tebu kekurangan air menyebabkan panjang ruas menjadi pendek, sedangkan bila musim penghujan tiba atau bisa dikatakan dalam keadaan mendapat cukup air maka panjang ruas menjadi lebih panjang [BAS-10].

c) %Brix

%Brix merupakan zat padat kering terlarut dalam larutan (gr/100 gr larutan). Prosedur Analisa %Brix dilakukan dengan cara mengambil Nira dari gilingan pertama yang telah ditampung dalam timba secara kontinyu $\pm$ 2 liter, kemudian Nira yang terambil dimasukkan dalam tabung mol sampai penuh, dilanjutkan dengan memasukkan alat Brix Weger kedalam tabung mol dan didiamkankan selama $\pm$ 5 menit. Brix Weger diangkat kemudian di baca angka brix yang tertera dan sekaligus dibaca suhunya [BAS-10].

d) HK (Harkat Kemurnian)

Harkat kemurnian merupakan presentase %pol (jumlah gula yang terlarut dalam 100 gram larutan yang mempunyai kesamaan putaran optik dengan sukrosa murni) terhadap presentase brix [KUS-12].

e) Nira

Nira merupakan cairan yang keluar dari pohon ataupun batang penghasil nira, seperti tebu. Umumnya komposisi nira terdiri dari air, sukrosa, gula reduksi, bahan organik lain dan anorganik. Air dalam nira merupakan bagian yang terbesar yaitu antara 75-90%. Sukrosa merupakan bagian zat padat yang terbesar berkisar 12,30-17,40 %. Gula reduksi sebesar 0,50-1,00 % dan sisanya adalah senyawa organik serta anorganik. [APR-13]

2.2.4 Pembagian Kelas Kualitas Rendemen Tebu

Kualitas rendemen tebu terbagi dalam 3 kelas yaitu tingkat kualitas rendemen tanaman tebu buruk, kualitas rendemen tanaman tebu baik, dan kualitas rendemen tanaman tebu sangat baik. Untuk setiap penentuan kelas pada setiap kualitas rendemen tanaman tebu telah dikonsultasikan dengan pakar tanaman tebu yaitu Bapak Ir. Bambang Hadi S. Kelas kualitas rendemen tanaman tebu ditunjukkan pada tabel 2.1.

Tabel 2.1 Rentang nilai kelas kualitas rendemen tanaman tebu

No	Rentang nilai kelas (z)	Kelas kualitas rendemen tanaman tebu
1.	Kurang dari 4 ($z < 4$)	Buruk
2.	4-5 ($4 \leq z < 5$)	Baik
3.	Lebih dari dan sama dengan 5 ($z \geq 5$)	Sangat Baik

2.3 Logika fuzzy

2.3.1 Pengertian logika fuzzy

Logika *fuzzy* merupakan salah satu komponen pembentuk soft computing. Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai atau derajat keanggotaan atau *membership function* sebagai ciri utama dari penalaran dengan logika *fuzzy* tersebut [KUS-10].

Ada beberapa alasan mengapa orang menggunakan logika *fuzzy*, antara lain yaitu konsep logika *fuzzy* mudah dimengerti, logika *fuzzy* sangat fleksibel, logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat, logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks, logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan, logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional, dan logika *fuzzy* juga didasarkan pada bahasa alami. Dalam logika *fuzzy* dikenal berhingga keadaan dari nilai “0” sampai ke nilai “1”. Logika *fuzzy* tidak hanya mengenal dua keadaan tetapi juga mengenal sejumlah keadaan yang berkisar dari keadaan salah sampai keadaan benar [YUL-08].

Logika *fuzzy* memiliki beberapa karakteristik yaitu himpunan *fuzzy* dan fungsi keanggotaan. Pada logika Boolean, sebuah individu dipastikan menjadi anggota salah satu himpunan saja. Berbeda dengan logika Boolean, di dalam himpunan *fuzzy* bisa terdapat sebuah individu yang masuk ke dalam dua himpunan yang berbeda. Seberapa besar eksistensinya dapat dilihat dari nilai keanggotaannya [YUL-08].

2.3.2 Fuzzy set (Himpunan fuzzy)

Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki 2 kemungkinan, yaitu:

- a) satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- b) nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan

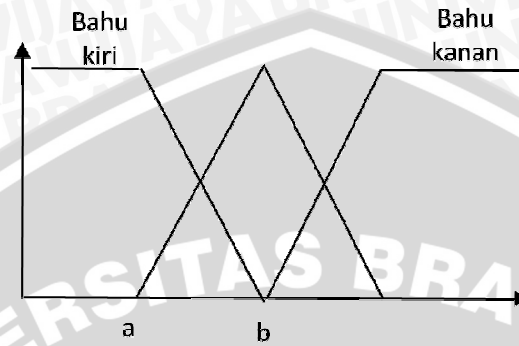
Kalau pada himpunan crisp, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A(x)=0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A(x)=1$ berarti x menjadi anggota penuh pada himpunan A .

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang [KUS-10].

2.3.3 Fungsi keanggotaan (membership function)

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang memetakan titik-titik input ke dalam nilai keanggotaannya (biasa disebut derajat keanggotaan) yang memiliki nilai interval antara 0 sampai satu. Salah satu cara yang digunakan untuk mendapatkan nilai keanggotaan adalah dengan pendekatan fungsi. Dalam penelitian ini fungsi keanggotaan yang dipakai adalah representasi kurva bentuk bahu. Representasi kurva bentuk bahu dapat digambarkan sebagai daerah yang terletak ditengah-tengah suatu *variable* yang direpresentasikan dalam bentuk segitiga, pada sisi kiri dan kanannya akan naik dan turun [KUS-10]. Kurva bentuk bahu digunakan untuk mengakhiri *variable* suatu daerah *fuzzy*. Bahu kiri

bergerak dari benar menuju salah, begitupula dengan bahu kanan bergerak dari salah menuju benar. Gambar 2.1 menunjukkan representasi kurva bentuk bahu:



Gambar 2.1 Representasi Kurva Bentuk Bahu

Fungsi keanggotaan representasi kurva bentuk bahu dituliskan pada persamaan (2-1) [HAF-08]:

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

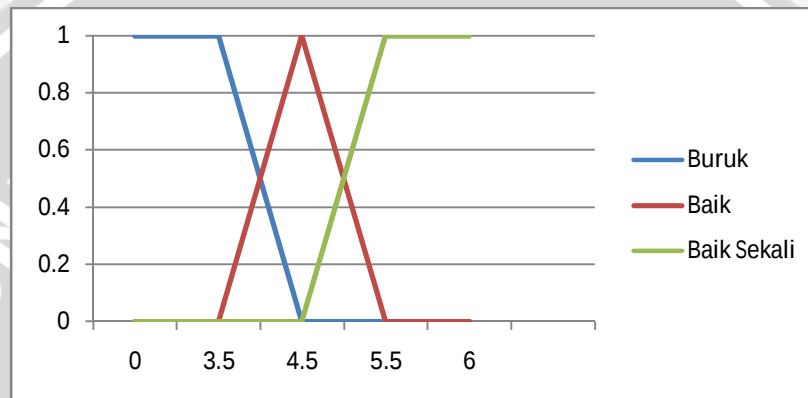
Pada penelitian ini tingkat kualitas rendemen dibagi dalam tiga kelas yaitu buruk (rendemen < 4), baik (4 ≤ rendemen < 5) dan sangat baik (rendemen ≥ 5). Berdasarkan rentang nilai tersebut akan dibentuk *membership function* untuk setiap kelas. Cara pembuatan *membership function* untuk kualitas rendemen tebu pada penelitian ini merujuk pada [PUS-13]. Nilai keanggotaan dari tingkat kualitas rendemen tebu pada penelitian ini dituliskan pada persamaan (2-2):

$$\mu_{\text{buruk}}(x) = \begin{cases} 1; & 0 \leq x < 3,5 \\ \frac{4,5-x}{4,5-3,5}; & 3,5 \leq x \leq 4,5 \\ 0; & x > 4,5 \end{cases}$$

$$\mu_{\text{baik}}(x) = \begin{cases} 0; & x \leq 3,5, x \geq 5,5 \\ \frac{x-3,5}{4,5-3,5}; & 3,5 \leq x < 4,5 \\ \frac{5,5-x}{5,5-4,5}; & 4,5 \leq x < 5,5 \end{cases} \quad (2-2)$$

$$\mu_{\text{sangat baik}}(x) = \begin{cases} 0 & ; & x \leq 4,5 \\ \frac{x - 4,5}{(5,5 - 4,5)} & ; & 4,5 \leq x < 5,5 \\ 1 & ; & x \geq 5,5 \end{cases}$$

Himpunan fuzzy tingkat kualitas rendemen untuk setiap kelasnya menggunakan kurva berbentuk representasi kurva bentuk bahu ditunjukkan oleh gambar 2.2:



Gambar 2.2 Kurva Kelas Kualitas Rendemen Tanaman Tebu

2.4 Data Mining

2.4.1 Pengertian Data Mining

Secara sederhana data mining bisa dikatakan sebagai penambangan atau proses penemuan informasi baru dengan cara mencari pola-pola atau aturan tertentu dari sejumlah data yang sangat besar. Data mining juga disebut sebagai serangkaian proses untuk menggali nilai tambah yang berupa pengetahuan dan selama ini tidak diketahui secara manual dari satu kumpulan data. Data mining biasanya juga disebut sebagai *knowledge discovery in database (KDD)*. KDD adalah kegiatan yang meliputi pemakaian data, pengumpulan historis untuk menemukan suatu pola dalam set data berukuran besar. Terdapat enam tahapan dalam melakukan proses data mining diantaranya adalah:

- Pembersihan data (*data cleaning*)

Pembersihan data merupakan proses menghilangkan noise, dan juga data yang tidak konsisten ataupun tidak relevan pada data yang diperoleh.

b) Integrasi data (*data integration*)

Proses penggabungan data dari berbagai data yang diperoleh kedalam data baru

c) Seleksi data(*data selection*)

Data yang diperoleh,seringkali tidak dipakai semuanya. Oleh karena itu hanya data yang sesuai untuk dianalisislah yang diambil

d) Transformasi data(*data transformation*)

Data diubah atau digabung kedalam format yang sesuai untuk diproses dalam data mining

e) Proses mining

Merupakan prosesutama yang dilakukan untuk memperoleh pengetahuan berharga dan tersembunyi.

f) Evaluasi pola (*pattern evaluation*)

Merupakan proses mengidentifikasi pola-pola menarik kedalam *knowledge base* yang ditemukan.

Data mining berkaitan dengan bidang ilmu-ilmu lain, seperti data warehousing , statistic, database system, machine learning, komputasi tingkat tinggi, information retrieval. Selain itu data mining juga didukung oleh berbagai disiplin ilmu lainnya seperti pengenalan pola, neural network, spatial data, analysis , image database, dan signal processing. Beberapa karakteristik data mining adalah sebagai berikut:

- a) Data mining merupakan proses yang berhubungan dengan penemuan sesuatu hal yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya
- b) Data mining digunakan pada data yang berukuran sangat besar. Penggunaan data yang besar biasanya bertujuan untuk proses perolehan hasil yang lebih dipercaya
- c) Penggunaan data mining bertujuan untuk proses pembuatan keputusan yang kritis, terutama dalam hal strategi

Mining sendiri berarti suatu usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material pokok.Oleh sebab itu data mining sebenarnya mempunyai akar yang panjang dari bidang ilmu seperti kecerdasan

buatan (*artificial intelligent*), machine learning, statistik dan database. Beberapa metode yang sering disebut-sebut adalah clustering, *classification*, *association*, *rules mining*, *neural network*, *genetic algorithm* [HUD-10].

2.4.2 Klasifikasi (*classification*)

Klasifikasi merupakan suatu teknik yang melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan cara memanipulasi data yang ada dan telah diklasifikasikan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasikan. Teknik ini juga menggunakan supervised induction untuk memanfaatkan kumpulan pengujian dari record yang terklasifikasikan untuk menentukan kelas-kelas tambahan. [KUS-07]

Terdapat beberapa metode klasifikasi yang sering digunakan dalam teknik data mining, diantaranya adalah metode *k-nearest-neighbor*, *naive bayes*, *decision tree*, *fuzzy*, *neural network* dan *support vector machine (SVM)*. Dibandingkan dengan metode klasifikasi lainnya yang hanya bias menetapkan satu kelas untuk sample masukan, metode *fuzzy* klasifikasi memiliki keunggulan yang lebih yaitu metode ini dapat menetapkan sebuah vector *fuzzy* yang menunjukkan kekuatan *relative* pada masing-masing kelas. [PUS-13].

2.4.3 K-Nearest Neighbor (K-NN)

Salah satu metode klasifikasi yang sering digunakan adalah metode *k-nearest neighbor*. Penggunaan *k-nearest neighbor* bertujuan untuk mengklasifikasikan objek baru berdasarkan atribut dan *training samples*. Algoritma *k-nearest neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Teknik algoritma *k-nearest neighbor* sangat mudah untuk diimplementasikan. Dalam hal ini jumlah data atau biasa disebut dengan tetangga terdekat ditentukan oleh user yang dinyatakan oleh k . Misalkan ditentukan $k=5$, maka setiap data uji dihitung jarak kedekatannya terhadap data training. Untuk

k=5 berarti data training yang dipilih adalah sebanyak lima yang jaraknya paling dekat terhadap data latih. Periksa output atau label dari masing-masing data yang didapat. Pengelompokan data uji atau data baru yang dimasukkan berdasarkan label data output yang memiliki frekuensi kemunculan terbanyak.[HAS-13]

2.4.4 Proses K-Nearest Neighbor

Langkah pertama untuk melakukan proses perhitungan jarak antara record data uji dan data latih perlu dilakukan proses *praprocessing* data pada setiap atribut terlebih dahulu. Proses *praprocessing* data tersebut berupa normalisasi data. Hal ini bertujuan untuk standarisasi semua atribut data yang digunakan dalam perhitungan. Proses normalisasi yang digunakan pada penelitian ini adalah *Min-Max normalization*. Perhitungan *Min-Max normalization* ditunjukkan pada persamaan (2-3):

$$V[i][j]' = \frac{(V[i][j] - \min[j])}{\text{Range}[j]} \quad (2-3)$$

Dimana:

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

$V[i][j]$ = Nilai atribut yang akan dinormalisasikan

$\text{Max}[j]$ = Nilai maximum dari suatu atribut j

$\text{Min}[j]$ = Nilai minimum dari suatu atribut j

$\text{Range}[j]$ = nilai dari $(\text{max}[j] - \text{min}[j])$

Selanjutnya dihitung jarak kedekatan antara data uji data latih terlebih dahulu. Untuk perhitungan jarak terdekat digunakan fungsi *Euclidean Distance* yang ditunjukkan pada persamaan (2-4):

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (2-4)$$

Dengan:

d_i = Jarak kedekatan

p = Dimensi data

i = Variabel data

x_1 = Data sample

x_2 = Data uji

x_1 dan x_2 merupakan dua record dengan n atribut. Perhitungan dengan menggunakan persamaan (2-4) bertujuan untuk mengetahui jarak antara data x_1 dan x_2 pada masing-masing record.

2.5 Fuzzy K-Nearest Neighbor Classifier

2.5.1 Pengertian Fuzzy K-Nearest Neighbor

Pada metode *k-nearest neighbor* prediksi yang diberikan dilakukan secara tegas pada data uji berdasarkan perbandingan k tetangga terdekat. Namun pada metode ini, prediksi juga dilakukan berdasarkan k tetangga terdekat tapi tidak secara tegas memprediksi kelas yang harus diikuti data uji dan dilakukan pemberian label kelas dengan memberikan nilai keanggotaan seperti pada teori himpunan *fuzzy*. Metode ini biasa disebut dengan *fuzzy k-nearest neighbor*. Algoritma *fuzzy k-nearest neighbor* (FK-NN) diperkenalkan oleh [KEL-85] dengan cara mengembangkan metode *k-nearest neighbor* yang digabungkan dengan teori himpunan *fuzzy* dalam hal pemberian definisi label kelas pada data yang digunakan [PRA-12].

Pada teori himpunan *fuzzy*, sebuah data dikatakan memiliki nilai keanggotaan pada setiap kelas. Hal ini bisa diartikan sebuah data dapat dimiliki oleh kelas yang berbeda dengan nilai derajat keanggotaan pada interval $[0,1]$. Teori himpunan *fuzzy* yang menggeneralisasi teori *k-nearest neighbor* klasik

dengan pendefinisian nilai keanggotaan sebuah data pada masing-masing kelas diperoleh dengan persamaan(2-5) [KEL-85] :

$$\mu_i(x) = \frac{\sum_{j=1}^k \mu_{ij} \left(\frac{1}{|x-x_j|^{2/m-1}} \right)}{\sum_{j=1}^k \left(\frac{1}{|x-x_j|^{2/m-1}} \right)} \quad (2-5)$$

Dimana μ_{ij} adalah nilai keanggotaan *fuzzy* pada contoh pengujian (x, x_j) yang merupakan satu dari contoh-contoh *fuzzy k-nearest neighbor*, k merupakan banyaknya nilai ketetanggaan terdekat yang diambil, j merupakan variable data untuk keanggotaan data latih, i merupakan variable data untuk keanggotaan data uji, sedangkan m merupakan berat kebalikan yang sebanding dengan jarak antara x_j dan x . Variabel (m) merupakan penentuan seberapa banyak pemberian bobot pada jarak saat menghitung kontribusi jarak kedekatan pada masing-masing tetangga dengan nilai keanggotaan. jika nilai m adalah dua, maka jarak kontribusi dari setiap titik tetangga(data latih) dibobotkan oleh nilai kebalikan dari jarak titik tetangga tersebut dengan titik yang sedang diklasifikasikan (data uji). ketika nilai m naik, titik-titik tetangga tersebut dibobotkan lebih merata dan efek dari jarak relatif dari titik yang sedang diklasifikasikan akan berkurang. ketika nilai m mendekati satu, semakin dekat tetangga maka akan dibobotkan lebih besar daripada tetangga yang lebih jauh (semakin besar nilai jarak maka semakin besar bobotnya), yang mana hal ini akan mempengaruhi pengurangan jumlah titik (tetangga) yang berkontribusi terhadap nilai keanggotaan dari titik yang sedang diklasifikasikan. hasil yang ditampilkan pada jurnal ini, menggunakan nilai $m=2$ tetapi perhatikan bahwa hampir tingkat kesalahan yang diperoleh pada data ini hampir sama dengan menggunakan nilai m yang beragam[KEL85].

Pada persamaan (2-5), nilai keanggotaan ditugaskan dari x dan dipengaruhi oleh kebalikan dari jarak tetangga terdekat juga keanggotaan kelas tersebut. Kebalikan dari jarak tetangga terdekat itu berfungsi sebagai bobot keanggotaan vektor. Pada data yang diujikan dapat diberikan nilai keanggotan kelas dengan beberapa cara diantaranya adalah dengan memberikan keanggotaan

lengkap di kelas yang sama dan non keanggotaan dikelas lainnya selain itu pemberiannilai keanggotaan pada setiap kelas juga bisa dilakukan dengan penetapan keanggotaan sampel berdasarkan jarak dari rata-rata kelas. Namun pada penelitian ini, penetapan nilai keanggotaan pada setiap kelas menggunakan metode representasi kurva bentuk bahu [PUS-13].

2.5.2 Algoritma *Fuzzy K-Nearest Neighbor*

Hal yang paling mendasar dalam algoritma *fuzzy k-nearest neighbor* adalah penetapan nilai keanggotaan pada setiap data yang telah diketahui nilai jaraknya menggunakan algoritma *k-nearest neighbor* dan juga keanggotaan tetangga mereka pada kelas-kelas yang memungkinkan dalam klasifikasi. Berikut ini merupakan contoh algoritma *fuzzy k-nearest neighbor*[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 ≤ 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 (2.7)
        Increment i
    END DO UNTIL
END
  
```

2.6 Akurasi Hasil Pengujian

Akurasi merupakan suatu perhitungan yang menyatakan seberapa dekat suatu hasil pengukuran terhadap angka sebenarnya. Semakin besar tingkat akurasi yang diperoleh oleh sistem maka semakin akurat hasil klasifikasi yang dilakukan. Pada penelitian ini tingkat akurasi dapat diperoleh dengan persamaan (2-6). Sedangkan untuk mencari prosentase dalam tingkat akurasi dapat diperoleh dengan persamaan (2-7) [PUS-13].

$$\text{Tingkat Akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \quad (2-6)$$

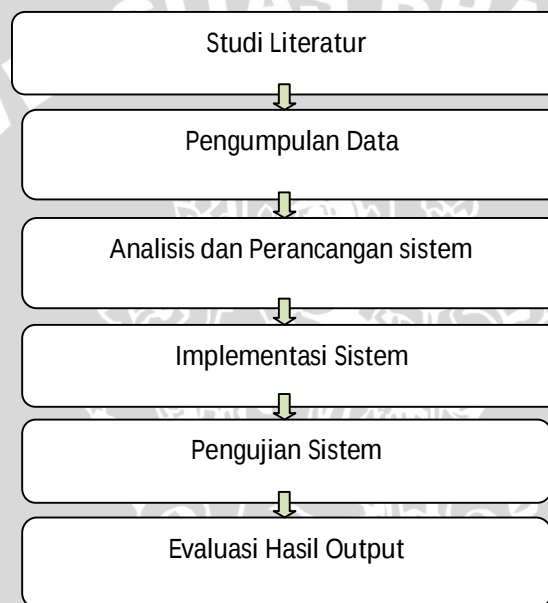
$$\text{Akurasi (\%)} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100 \quad (2-7)$$

Jumlah data uji benar merupakan jumlah record dari data yang uji dalam pengklasifikasian dan hasilnya sama dengan kelas sebenarnya. Jumlah total data uji merupakan jumlah seluruh record yang digunakan dalam klasifikasi (semua data uji).

BAB III

METODE PENELITIAN

Pada bab metode penelitian ini berisi tentang pembahasan metodologi yang digunakan dalam penelitian yang berjudul ” penerapan metode *fuzzy k-nearest neighbor* untuk menentukan kualitas rendemen tanaman tebu”. Secara umum tahapan dari penelitian ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Penjelasan langkah-langkah pada gambar 3.1 adalah sebagai berikut:

1. Mempelajari berbagai literatur tentang *fuzzy k-nearest neighbor* dan kualitas rendemen tanaman tebu.
2. Pengumpulan data set yang digunakan untuk data latih , yaitu data set kualitas rendemen tanaman tebu.
3. Melakukan analisa dan perancangan sistem menggunakan algoritma *fuzzy k-nearest neighbor* untuk menentukan kualitas rendemen tanaman tebu.
4. Tahap dari proses implementasi sistem merupakan bagian dari pembuatan sistem atau bias dikatakan pembangunan perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan (implementasi)

5. Proses pengujian perangkat lunak yang telah dibuat, untuk memastikan bahwa perangkat lunak yang dibangun telah berjalan sesuai dengan yang diharapkan .
6. Tahapan ini merupakan proses evaluasi data output yang dihasilkan oleh perangkat lunak, untuk memastikan ketepatan yang dihasilkan oleh sistem tersebut.

3.1 Studi Literatur

Studi literatur dalam sebuah penelitian untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan orang lain dan bagaimana orang mengerjakannya, kemudian seberapa berbeda penelitian yang akan kita lakukan. Penting karena untuk menghindari usaha yang sebenarnya sudah pernah dilakukan orang lain dan bisa digunakan pada penelitian kita untuk menghemat waktu, tenaga dan biaya. Penting juga untuk memberi arah penelitian selanjutnya yang perlu dilakukan untuk melanjutkan misi penelitian. Beberapa literatur yang dipelajari untuk dibandingkan dalam penyusunan sistem adalah literatur mengenai kualitas hasil rendemen tanaman tebu , teori himpunan *fuzzy*, metode *k-nearest neighbor* dan metode *fuzzy k-nearest neighbor* . Semua Literatur tersebut diperoleh dari buku, jurnal, dan browsing Internet serta literatur lain yang mendukung bab 2. Literatur- literatur tersebut digunakan sebagai suatu bahan kajian pada penelitian ini agar nantinya sistem memiliki tingkat keakuratan yang tepat.

3.2 Pengumpulan Data

Pada tahap pengumpulan Data ini, dataset yang digunakan didapatkan dari PG Semboro Jember, berupa panjang tebu (m), diameter tebu (cm), banyak ruas per tebu, berat per meter (kg), persentase (%) brix, harkat kemurnian (HK), dan nira juga rendemen tebu yang nantinya akan di tentukan kualitasnya. Data ini merupakan data antara tahun 2012 sampai 2013. Data latih yang digunakan terdiri dari empat macam yaitu 100, 125, 150, dan 175 data latih dan data uji yang digunakan adalah 50, 60, 70, 80 data uji.

3.3 Deskripsi Sistem

Sistem yang dibuat merupakan suatu perangkat lunak untuk mengklasifikasikan kualitas hasil rendemen tanaman tebu yang mengimplementasikan metode *fuzzy k-nearest neighbor*. Tujuan sistem ini adalah untuk mengetahui apakah kualitas hasil rendemen tebu yang dimiliki adalah buruk, baik, atau sangat baik. Pada awalnya, sistem ini sudah memiliki data faktor-faktor yang berpengaruh terhadap kualitas rendemen tanaman tebu yang nantinya digunakan untuk diklasifikasi menggunakan metode *fuzzy k-nearest neighbor*. Faktor yang berpengaruh terhadap .kualitas hasil rendemen tanaman tebu terdiri dari 7 atribut diantaranya adalah panjang, diameter, ruas, berat, %brix, harkat kemurnian, dan nira. Perangkat lunak ini juga menguji keakuratan hasil dari klasifikasi dataset kualitas hasil rendemen tanaman tebu terhadap data yang sebenarnya. Selain itu parameter yang diujikan berkaitan dengan nilai k (tetangga) yang berpengaruh terhadap tingkat akurasi

3.4 Perancangan sistem

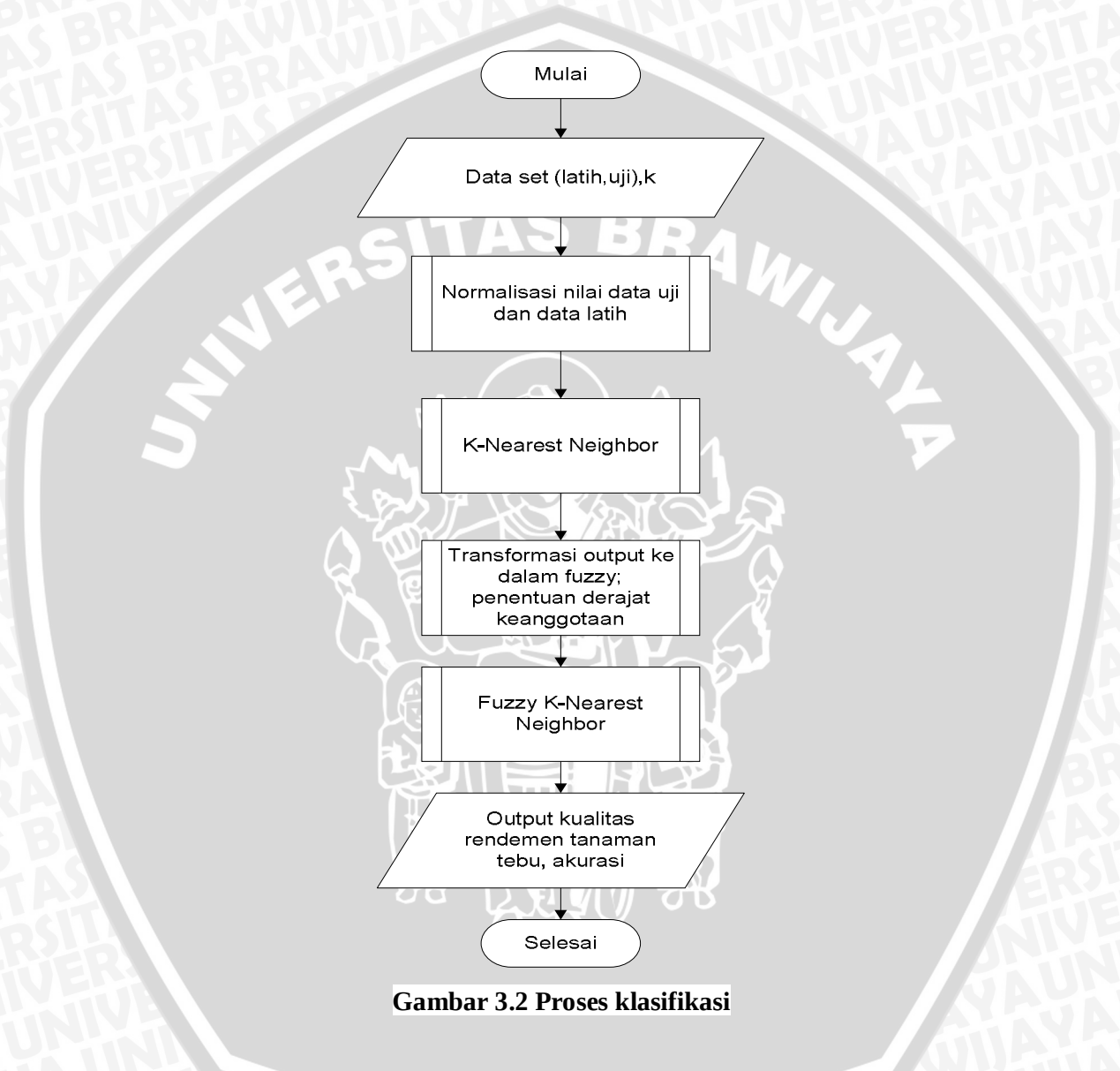
Pada subbab perancangan sistem ini membahas masalah proses-proses dalam membangun sebuah sistem. Sistem ini akan menentukan kualitas hasil rendemen tanaman tebu dari data uji yang dimiliki. Sistem akan mengklasifikasikan kualitas hasil rendemen tanaman tebu tersebut sesuai kriteria data uji yang dimasukkan. Klasifikasi dilakukan dengan menggunakan metode *fuzzy k-nearest neighbor*.

Secara singkat proses klasifikasi kualitas hasil rendemen tanaman tebu terdiri dari beberapa tahapan, diantaranya:

1. Inputkan data faktor-faktor yang mempengaruhi kualitas hasil rendemen tanaman tebu (panjang, diameter, ruas, berat, %brix, harkat kemurnian, dan nira), dan juga nilai k.
2. Dilakukan proses normalisasi terhadap semua data baik data uji maupun juga data latih yang sedang digunakan.
3. Dilakukan proses perhitungan *k-nearest neighbor* menggunakan persamaan *euclidean distance*
4. Transformasi kedalam bentuk *fuzzy*

5. Didapatkan hasil kualitas rendemen tanaman tebu yang dihitung menggunakan *fuzzy k-nearest neighbor*.

Tahapan proses klasifikasi dapat dilihat pada gambar 3.2 dibawah ini:

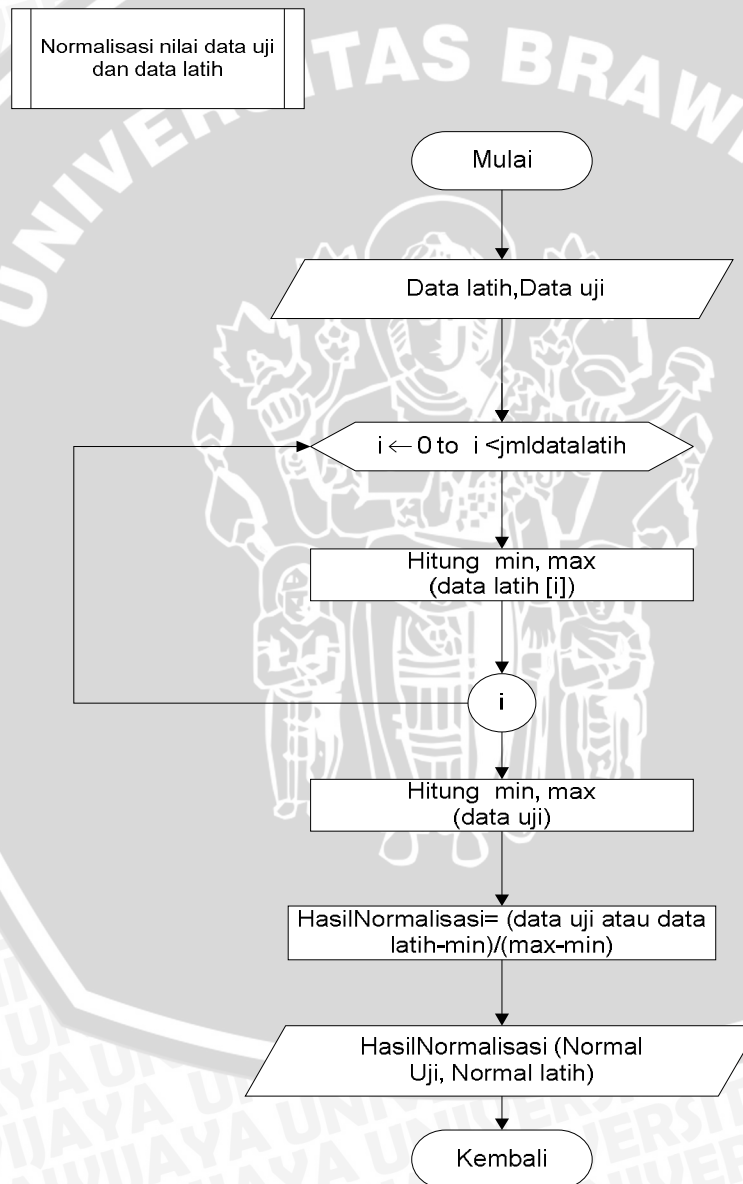


Gambar 3.2 Proses klasifikasi

Gambar 3.2 merupakan tahapan yang dilakukan untuk mengklasifikasikan kualitas rendemen tanaman tebu. Pada proses klasifikasi tersebut terdapat 5 proses utama yang dilakukan. Proses tersebut adalah normalisasi data, *k-nearest neighbor*, penentuan jarak baru, transformasi ke dalam bentuk *fuzzy*, dan *fuzzy k-nearest neighbor*.

3.4.1 Proses Normalisasi

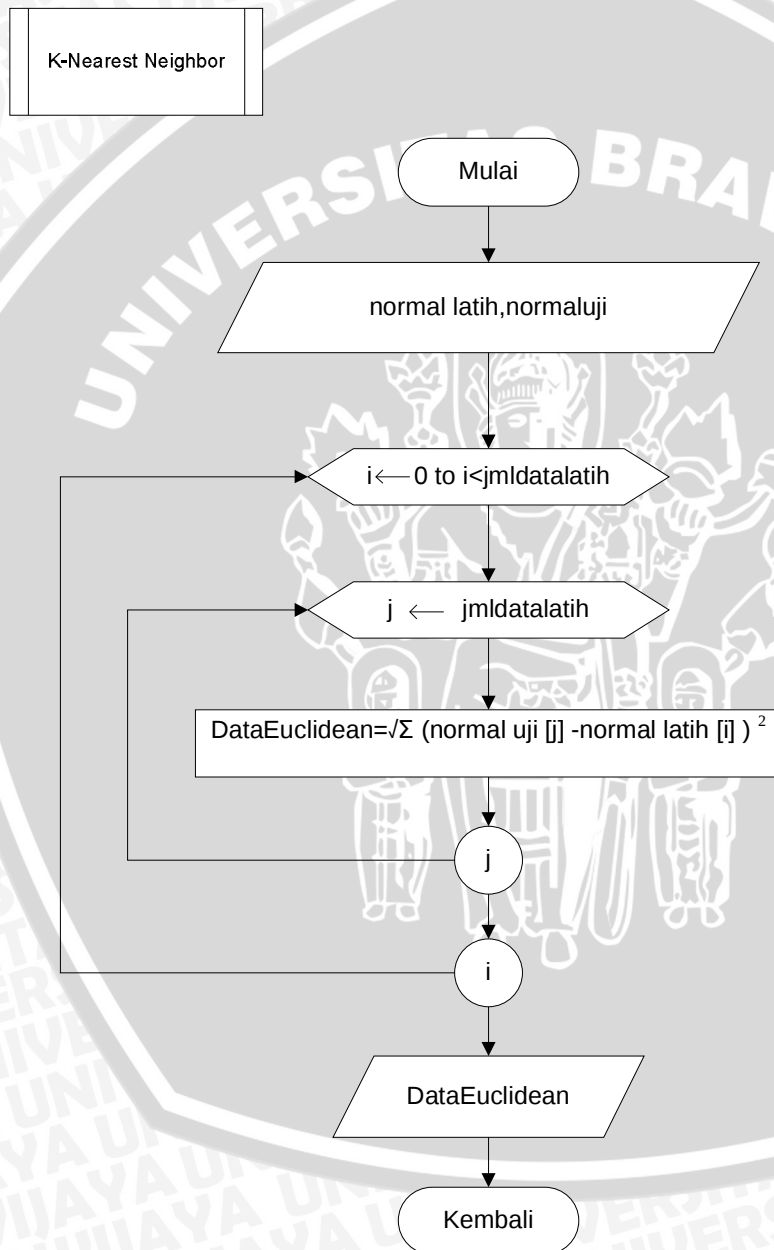
Proses ini merupakan proses *praprocessing* data dimana dilakukan proses transformasi berupa normalisasi terhadap nilai dari setiap atribut untuk memberlakukan skala pengaruh yang ada pada atribut terhadap hasil dan untuk mencari nilai normalisasi digunakan persamaan (2-3). Proses normalisasi yang digunakan adalah *min-max normalization*. Diagram alur proses normalisasi data ditunjukkan oleh gambar 3.3:



Gambar 3.3 Alur proses normalisasi

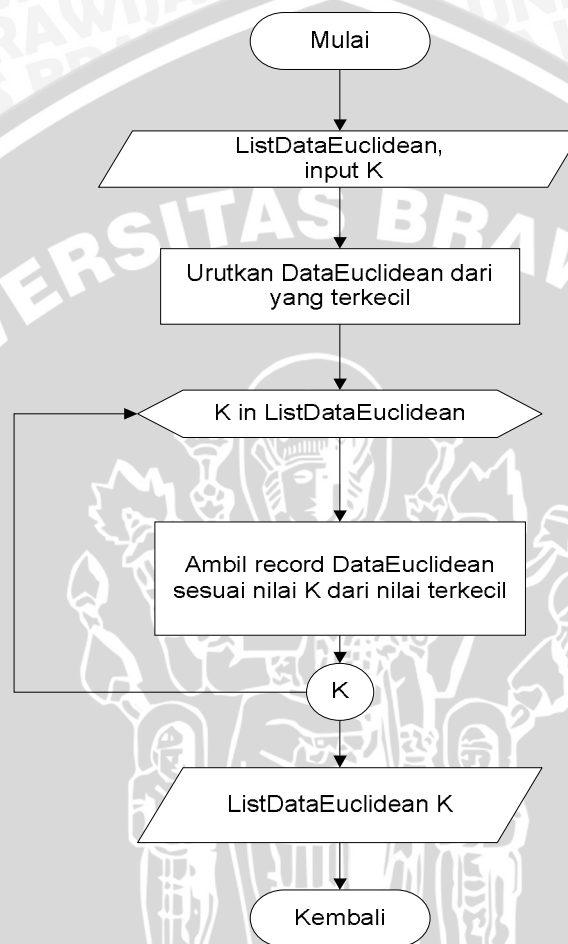
3.4.2 Proses K-Nearest Neighbor

Tahapan ini merupakan proses perhitungan nilai jarak kedekatan tetangga data uji terhadap data latih menggunakan *euclidean distance*. *Eclidean Distance* dijelaskan pada persamaan (2-4). Flowchart dari proses *k-nearest neighbor* ditunjukkan oleh gambar 3.4:



Gambar 3.4 Proses *k-nearest neighbor*

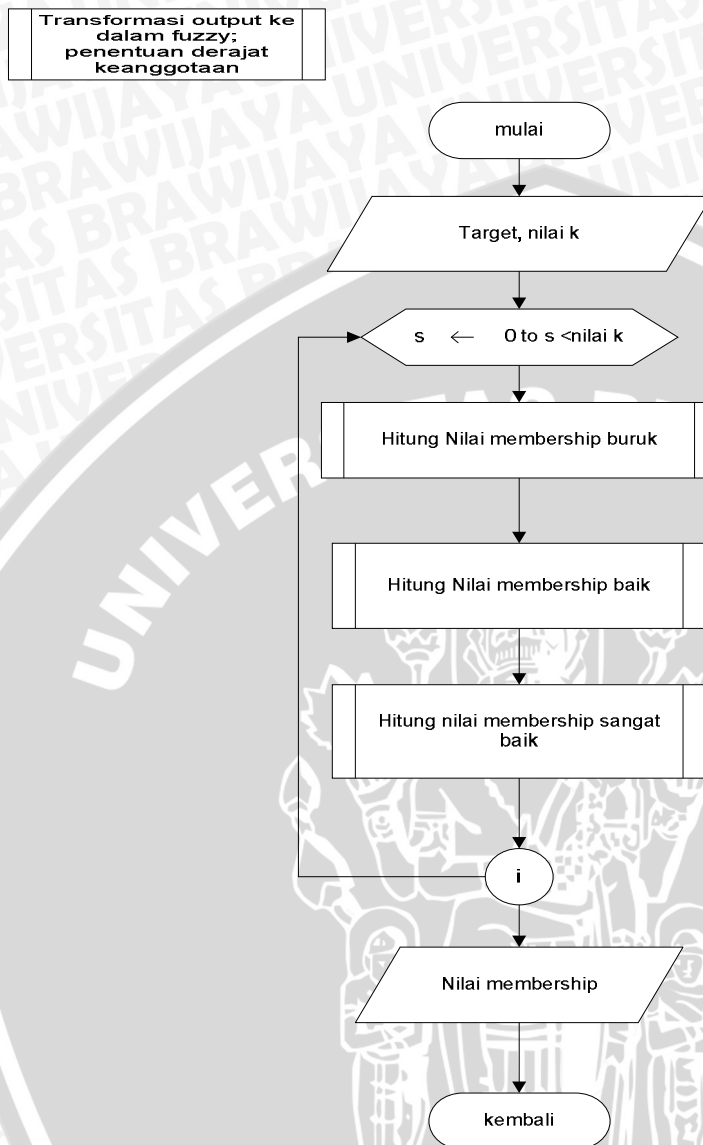
Setelah diketahui nilai jarak yang dihasilkan dari proses perhitungan menggunakan *EuclideanDistance* maka selanjutnya dicari data yang diproses berdasarkan jumlah k yang diinputkan. Proses pengambilan data yang digunakan berdasarkan nilai k yang diinputkan ditunjukkan oleh gambar 3.5:



Gambar 3.5 Pengambilan data berdasarkan K

3.4.3 Transformasi kedalam bentuk *fuzzy*

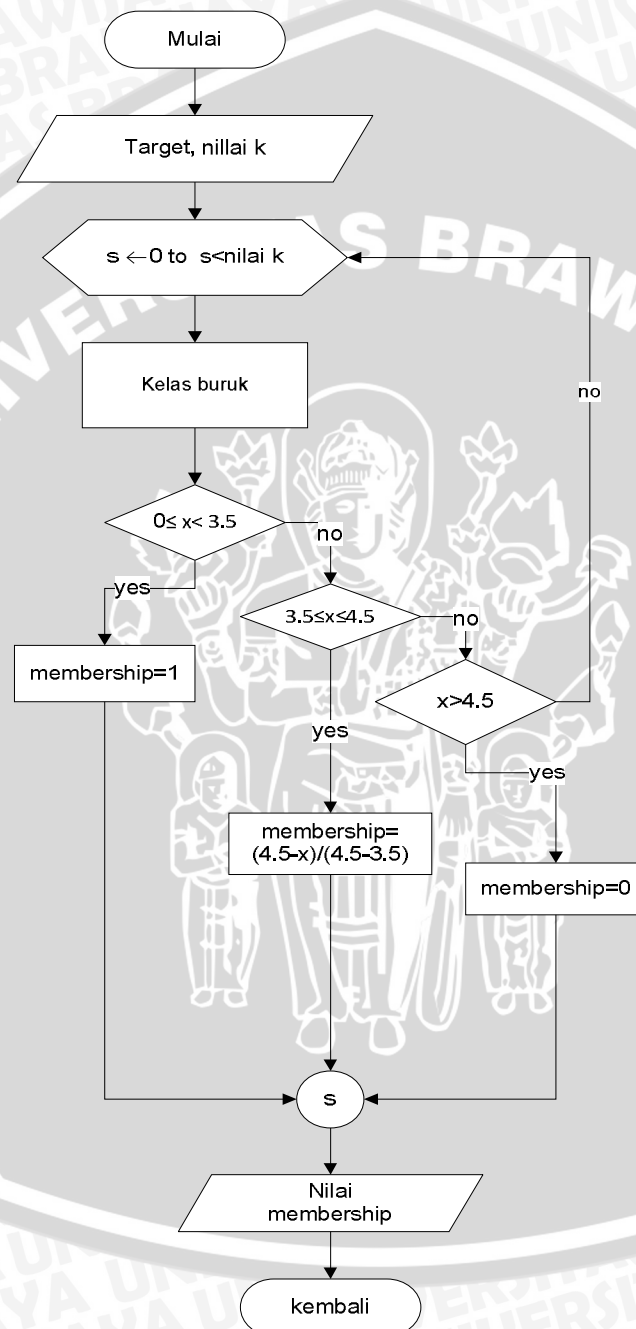
Transformasi kedalam bentuk *fuzzy* merupakan proses untuk mengubah nilai kuantitatif dari setiap atribut kualitas rendemen tanaman tebu menjadi derajat keanggotaan *fuzzy* beserta linguistiknya berdasarkan keanggotaan atribut. Fungsi keanggotaan ini nantinya akan digunakan pada proses *fuzzy k-nearest neighbor*. Flowchart dari pembentukan membership function kualitas rendemen tanaman tebu ditunjukkan oleh gambar 3.6:



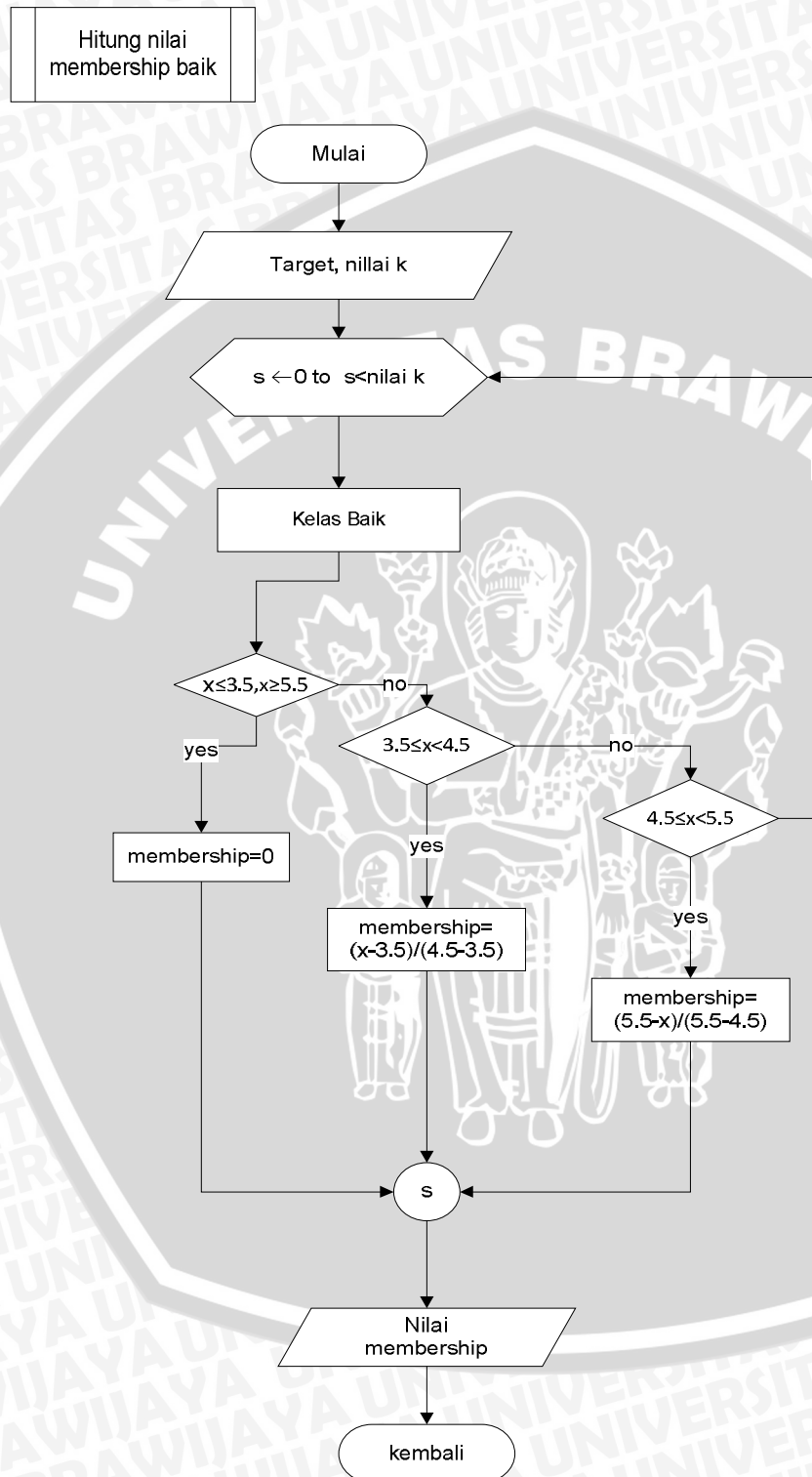
Gambar 3.6 Proses transformasi kedalam bentuk fuzzy

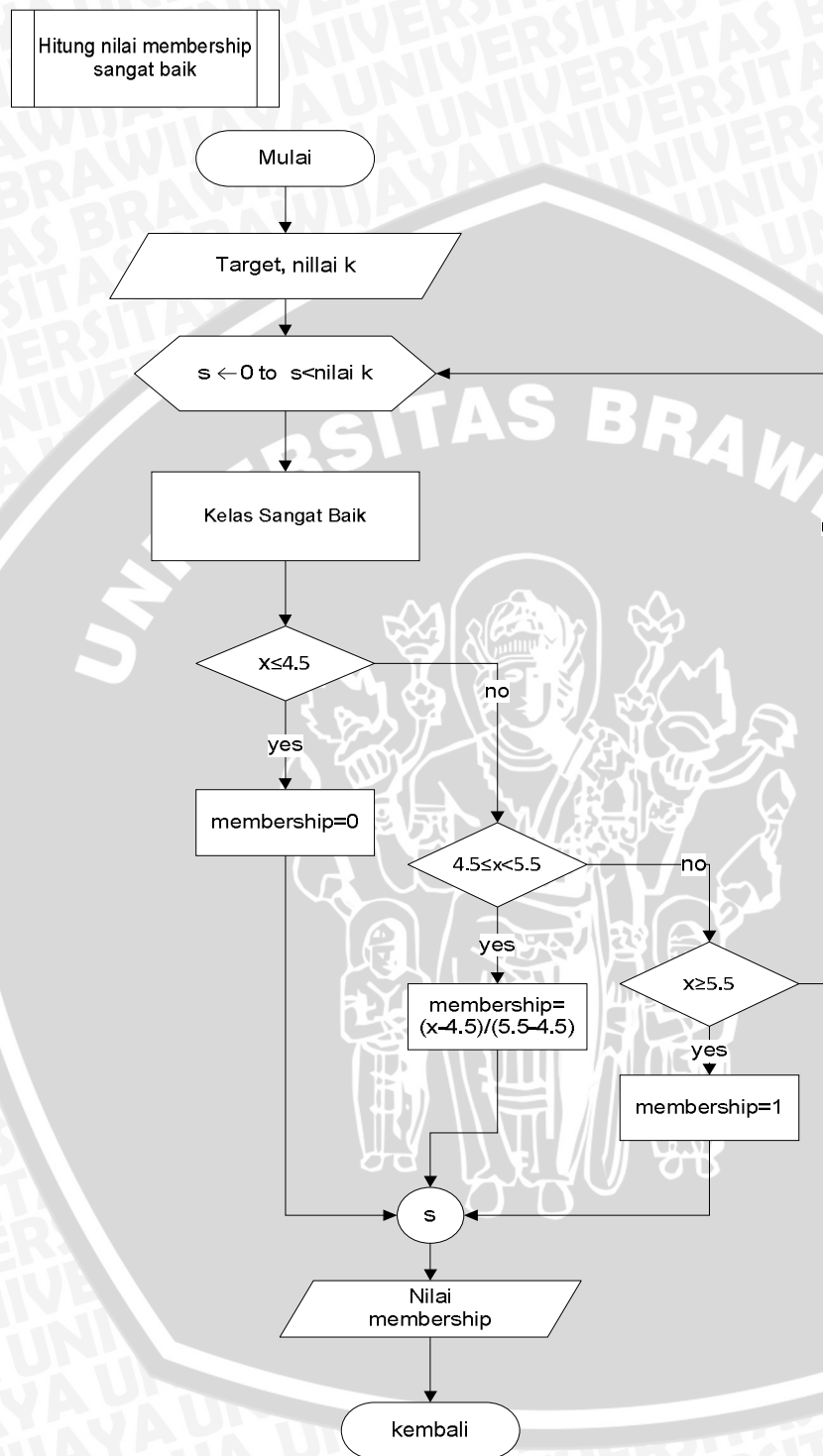
Proses transformasi ke dalam bentuk *fuzzy* ini terdiri dari tiga sub proses yaitu menghitung nilai *membership* buruk, *membership* baik, dan hitung nilai *membership* sangat baik. Flowchart dari perhitungan nilai *membership* rendemen tanaman tebu ditunjukkan oleh gambar 3.7, gambar 3.8, dan gambar 3.9:

Hitung Nilai
Membership buruk



Gambar 3.7 Proses perhitungan nilai *membership* buruk

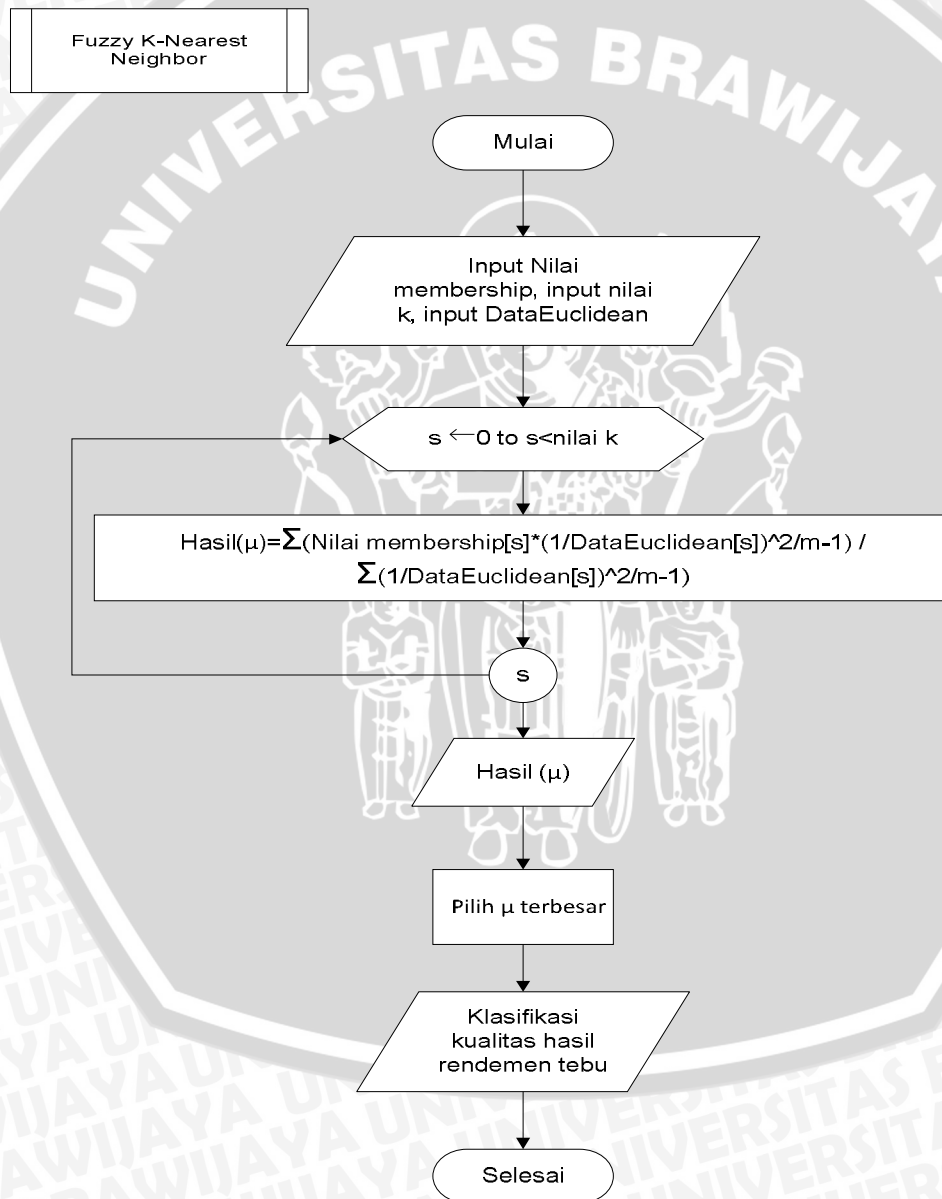
Gambar 3.8 Proses perhitungan nilai *membership* baik



Gambar 3.9 Proses perhitungan nilai *membership* sangat baik

3.4.4 Fuzzy K-Nearest Neighbor

Fuzzy k-nearest neighbor merupakan proses perhitungan nilai membership akhir dari kualitas rendemen tanaman tebu pada masing-masing kelas (buruk, baik, sangat baik). Kualitas rendemen tanaman tebu ditentukan oleh nilai membership akhir terbesar diantara ke tiga kelas tersebut. Adapun perhitungan *fuzzy k-nearest neighbor* didapatkan dari persamaan (2-5). Flowchat untuk proses *fuzzy k-nearest neighbor* ditunjukkan oleh gambar 3.10:



Gambar 3.10 Proses perhitungan *fuzzy k-nearest neighbor*

3.5 Perhitungan Manual

Pada subbab ini menampilkan perhitungan manual untuk proses klasifikasi kualitas hasil rendemen menggunakan metode *fuzzy k-nearest neighbor*. Data yang digunakan sebanyak 20 data latih dan 1 data uji. Berikut ini akan ditampilkan 20 data latih pada tabel 3.1 :

Tabel 3.1 data pelatihan sistem:

No	panjang	diameter	ruas/	berat/	rata-rata		Nilai	rendemen			rata-rata rendemen
	(mtr)	(cm)	btng	Mtr	%brix	HK		atas	Tengah	bawah	
1	2.30	2.82	16	0.64	11.40	64.0	5.66	0.65	3.99	4.87	3.17
2	2.22	2.82	16	0.60	10.90	63.7	5.36	0.47	3.81	4.77	3.017
3	2.68	2.60	17	0.67	11.80	65.2	6.05	0.63	2.92	3.72	2.423
4	2.00	2.92	17	0.68	12.00	63.7	5.90	0.72	3.62	4.52	2.953
5	1.78	2.54	13	0.66	12.50	65.9	6.53	0.67	4.06	4.89	3.207
6	2.46	2.74	23	0.51	12.60	66.5	6.69	1.18	4.73	5.49	3.8
7	2.22	2.44	13	0.59	12.60	65.0	6.43	1.20	4.60	5.34	3.713
8	2.46	2.56	17	0.69	14.00	65.7	7.28	1.31	5.67	6.29	4.423
9	2.20	2.80	18	0.58	15.40	67.0	8.29	1.52	5.72	6.33	4.523
10	2.50	2.82	21	0.68	15.70	67.5	8.56	1.66	5.83	6.63	4.707
11	2.10	2.90	19	0.53	14.60	68.8	8.22	1.77	5.60	6.69	4.687
12	1.98	2.62	14	0.59	15.20	64.1	7.56	1.39	5.31	6.02	4.24
13	2.20	2.64	18	0.58	14.10	67.4	7.66	1.50	5.47	6.20	4.39
14	1.72	2.40	15	0.68	14.00	67.2	7.57	1.10	5.14	5.78	12.02
15	1.92	2.78	19	0.63	14.00	66.7	7.47	1.22	4.26	5.49	10.97
16	2.00	2.51	15	0.49	12.80	65.4	6.60	0.87	4.90	5.94	11.71
17	1.98	2.84	14	0.61	14.60	68.7	8.20	1.78	5.94	6.91	14.63
18	2.21	2.53	15	0.67	14.20	67.8	7.80	0.85	3.77	4.19	8.81
19	1.88	2.42	16	0.67	16.20	69.7	9.33	2.59	6.52	7.31	5.473
20	1.70	2.60	15	0.64	13.10	66.6	6.97	1.24	4.96	5.67	11.87

Dari 20 data latih tersebut terdapat rata-rata rendemen yang didapatkan dari hasil perhitungan nilai rata-rata rendemen atas, rendemen tengah, dan juga rendemen bawah. Rata-rata rendemen inilah yang nantinya digunakan sebagai penentu kelas kualitas hasil rendemen atau disebut dengan target output. Seperti dijelaskan pada subbab 2.2.3, kelas kualitas rendemen tanaman tebu terbagi menjadi tiga jenis yaitu buruk, baik, dan sangat baik.

Tabel 3.2 Data uji sistem

No	Panjang	Diameter	ruas/	berat/	rata-rata		Nilai	rendemen			rata-rata rendemen
	(mtr)	(cm)	btng	Mtr	%brix	HK		atas	tengah	bawah	
1	1.68	2.60	16	0.67	15.60	69.2	8.88	?			

Langkah-langkah perhitungan untuk algoritma *fuzzy k-nearest neighbor* adalah sebagai berikut :

- a) Lakukan proses normalisasi untuk setiap atribut pada data training yang digunakan. Pada perhitungan normalisasi , nilai *range* didapat dari nilai *maximum* data training dikurangi nilai *minimum* data training. Pada tabel 3.4 dan tabel 3.5 menunjukkan hasil normalisasi dari data training yang berupa 20 data latih dan 1 data uji. Berikut ini adalah contoh perhitungan proses normalisasi nilai record pertama untuk atribut panjang (m).

$V[i][j] = 2.30$ (nilai record pertama untuk atribut panjang)

$\min[j] = 1.68$

$\max[j] = 2.68$

$\text{range}[j] = 1.00$

Setelah diketahui nilai *maximum* dan nilai *minimum* dari atribut panjang maka selanjutnya dilakukan perhitungan menggunakan persamaan (2-3)

$$V[i][j]' = \frac{(V[i][j] - \min[j])}{\text{Range}[j]}$$

$$\begin{aligned}
 V[i][j]' &= \frac{(2.30 - 1.68)}{1.00} \\
 &= 0.62
 \end{aligned}$$

Dari perhitungan normalisasi ini diperoleh nilai untuk record pertama atribut panjang adalah 0.62. Proses yang sama juga akan dilakukan untuk nilai

record pada seluruh atribut. Hasil dari perhitungan proses normalisasi ditunjukkan pada Tabel 3.4 dan 3.5.

Tabel 3.3 Range data training

	panjang	Diameter	Ruas	berat	%brix	HK	nira
Min	1.68	2.40	13	0.49	10.90	63.7	5.36
Max	2.68	2.92	23	0.69	16.20	69.7	9.33
Range	1.00	0.52	10	0.20	5.30	6.0	3.97

Tabel 3.4 Data latih yang telah di normalisasi

No	panjang	Diameter	Ruas	Berat	%brix	HK	nira
1	0.62	0.807692	0.3	0.75	0.09434	0.05	0.075567
2	0.54	0.807692	0.3	0.55	0	0	0
3	1.00	0.384615	0.4	0.9	0.169811	0.25	0.173804
4	0.32	1	0.4	0.95	0.207547	0	0.13602
5	0.10	0.269231	0	0.85	0.301887	0.366667	0.29471
6	0.78	0.653846	1	0.1	0.320755	0.466667	0.335013
7	0.54	0.076923	0	0.5	0.320755	0.216667	0.269521
8	0.78	0.307692	0.4	1	0.584906	0.333333	0.483627
9	0.52	0.769231	0.5	0.45	0.849057	0.55	0.738035
10	0.82	0.807692	0.8	0.95	0.90566	0.633333	0.806045
11	0.42	0.961538	0.6	0.2	0.698113	0.85	0.720403
12	0.30	0.423077	0.1	0.5	0.811321	0.066667	0.554156
13	0.52	0.461538	0.5	0.45	0.603774	0.616667	0.579345
14	0.04	0	0.2	0.95	0.584906	0.583333	0.556675
15	0.24	0.730769	0.6	0.7	0.584906	0.5	0.531486
16	0.32	0.211538	0.2	0	0.358491	0.283333	0.312343
17	0.30	0.846154	0.1	0.6	0.698113	0.833333	0.715365
18	0.53	0.25	0.2	0.9	0.622642	0.683333	0.61461
19	0.20	0.038462	0.3	0.9	1	1	1
20	0.02	0.384615	0.2	0.75	0.415094	0.483333	0.405542

Tabel 3.5 data uji yang telah di normalisasi

No	panjang	Diameter	Ruas	berat	%brix	HK	nira
21	0.00	0.384615	0.3	0.9	0.886792	0.916667	0.88665

b) Dilakukan pencarian jarak terdekat dari data uji dengan data latih. Pencarian jarak terdekat didapatkan dari formula *euclidean distance* pada persamaan (2-4). Berikut ini merupakan contoh proses perhitungan nilai jarak terdekat dari record pertama data latih dengan data uji.

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

$$d_i = \sqrt{(0.00 - 0.62)^2 + (0.384615 - 0.807692)^2 + (0.3 - 0.3)^2 + (0.9 - 0.75)^2 + (0.886792 - 0.09434)^2 + (0.916667 - 0.05)^2 + (0.88665 - 0.075567)^2}$$

$$d_i = 1.619519223$$

c) Menghitung nilai jarak antara record data uji dan data latih lainnya. Hasil perhitungan nilai dari jarak terdekat dapat dilihat pada table 3.6.

Tabel 3.6 Nilai jarak kedekatan antara data uji dan 20 data latih

Data ke-	jarak	rank minimum distance	Target
1	1.619519223	18	3.17
2	1.733759128	20	3.017
3	1.573739592	17	2.423
4	1.535804394	16	2.953
5	1.053954775	9	3.207
6	1.624219078	19	3.8
7	1.351858282	14	3.713
8	1.108222832	13	4.423
9	0.904866695	8	4.523
10	1.091340895	12	4.707
11	1.075603344	11	4.687
12	1.063121503	10	4.24
13	0.885072353	7	4.39
14	0.687865624	2	12.02
15	0.83564975	6	10.97
16	1.400856344	15	11.71
17	0.710563494	3	14.63
18	0.712213025	4	8.81
19	0.438669727	1	5.473
20	0.821365994	5	11.87

d) Nilai jarak yang sudah ditemukan selanjutnya diurutkan berdasarkan nilai jarak yang paling dekat. Hasil pengurutan nilai jarak dapat dilihat pada tabel 3.7.

Tabel 3.7 Nilai jarak setelah diurutkan

Data ke-	Jarak	rank minimum distance	target
19	0.438669727	1	5.473
14	0.687865624	2	12.02
17	0.710563494	3	14.63
18	0.712213025	4	8.81
20	0.821365994	5	11.87
15	0.83564975	6	10.97
13	0.885072353	7	4.39
9	0.904866695	8	4.523
5	1.053954775	9	3.207
12	1.063121503	10	4.24
11	1.075603344	11	4.687
10	1.091340895	12	4.707
8	1.108222832	13	4.423
7	1.351858282	14	3.713
16	1.400856344	15	11.71
4	1.535804394	16	2.953
3	1.573739592	17	2.423
1	1.619519223	18	3.17
6	1.624219078	19	3.8
2	1.733759128	20	3.017

e) Apabila diinputkan nilai k sama dengan 5 ($k = 5$) maka data yang digunakan adalah jarak terdekat pada *rank* 1-5. Tabel 3.8 menunjukkan nilai jarak terdekat yang diambil pada rank data 1-5.

Tabel 3.8 seleksi 5 data dengan nilai jarak terdekat

Data ke-	Jarak	rank minimum distance	target
19	0.438669727	1	5.473
14	0.687865624	2	12.02
17	0.710563494	3	14.63
18	0.712213025	4	8.81
20	0.821365994	5	11.87

Didapatkan 5 nilai data jarak terdekat, dimana setiap data memiliki nilai target rendemen tanaman tebu.

f) Proses perhitungan *membership function* dilakukan sesuai dengan nilai awal target nilai data latih. Misal untuk data pada rank pertama dengan nilai target 5.473. Jadi $x=5.473$, akan dimasukkan kedalam semua linguistic. Berikut ini adalah contoh perhitungan nilai $x=5.473$ dalam nilai *membership function* yang ditunjukkan pada tabel 3.9 dan perhitungan nilai *fuzzy* untuk setiap target pada tabel 3.10

Tabel 3.9 Target dalam *linguistic membership*nya

Membership	Range	Perhitungan	Hasil
μ Buruk	$0 \leq x < 3,5$	1	-
	$3,5 \leq x \leq 4,5$	$\frac{4,5-x}{(4,5-3,5)}$	-
	$x > 4,5$	0	0
μ Baik	$x \leq 3,5, \quad x \geq 5,5$	0	-
	$3,5 \leq x < 4,5$	$\frac{x-3,5}{(4,5-3,5)}$	-
	$4,5 \leq x < 5,5$	$\frac{5,5-x}{(5,5-4,5)}$	0.027
μ Sangat Baik	$x \leq 4,5$	0	-
	$4,5 \leq x < 5,5$	$\frac{x-4,5}{(5,5-4,5)}$	0.973
	$x \geq 5,5$	1	-

Tabel 3.10 Nilai *fuzzy* untuk setiap target

Data ke-	Jarak	target	Buruk	Baik	Sangat Baik
19	0.438669727	5.473	0	0.027	0.973
14	0.687865624	12.02	0	0	1
17	0.710563494	14.63	0	0	1
18	0.712213025	8.81	0	0	1
20	0.821365994	11.87	0	0	1

g) Setelah diketahui nilai *membership* dari setiap target maka perhitungan dilanjutkan dengan proses *fuzzy k-nearest neighbor*. Perhitungan didapat dari persamaan (2-5). Nilai m yang digunakan pada perhitungan ini adalah 2. Berikut ini adalah contoh perhitungan nilai *membership* akhir.

$$\mu_i(x) = \frac{\sum_{j=1}^k \mu_{ij} \left(\frac{1}{|x-x_j|^{2/m-1}} \right)}{\sum_{j=1}^k \left(\frac{1}{|x-x_j|^{2/m-1}} \right)}$$

- Nilai *membership* akhir kualitas buruk (1)

$$\mu_i(1) = \frac{0 \left(\frac{1}{|0,438669727|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}{\left(\frac{1}{|0,438669727|^{2/2-1}} \right) + \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}$$

$$\mu_i(1) = 0$$

- Nilai *membership* akhir kualitas baik(2)

$$\mu_i(2) = \frac{0,027 \left(\frac{1}{|0,438669727|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + 0 \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}{\left(\frac{1}{|0,438669727|^{2/2-1}} \right) + \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}$$

$$\mu_i(2) = 0.011009538$$

- Nilai *membership* akhir kualitas sangat baik (3)

$$\mu_i(3) = \frac{0,973 \left(\frac{1}{|0,438669727|^{2/2-1}} \right) + 1 \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + 1 \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + 1 \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + 1 \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}{\left(\frac{1}{|0,438669727|^{2/2-1}} \right) + \left(\frac{1}{|0,687865624|^{2/2-1}} \right) + \left(\frac{1}{|0,710563494|^{2/2-1}} \right) + \left(\frac{1}{|0,712213025|^{2/2-1}} \right) + \left(\frac{1}{|0,821365994|^{2/2-1}} \right)}$$

$$\mu_i(3) = 0,988990462$$

Tabel 3.11 Nilai *membership* akhir

μ Buruk	0
μ Baik	0.011009538
μ Sangat Baik	0.988990462

h) Pada *membership* akhir setiap *linguistic* tersebut akan dicari nilai yang paling besar. Berdasarkan perhitungan diatas nilai terbesar didapat oleh kualitas rendemen sangat baik dengan nilai 0,988990462

i) Maka, apabila inputan tersebut dicari dengan $k=5$ akan didapat hasil dari data uji dengan target kualitas rendemen sangat baik yang bernilai FK-NN 0,988990462

3.6 Perancangan Antarmuka

Rancangan antarmuka yang digunakan pada sistem akan ditunjukkan pada terdiri dari lima bagian yaitu klasifikasi *fk-nn*, Proses *fk-nn*, Hasil *fk-nn* Data Uji, Akurasi, dan Akurasi (Berdasarkan nilai k).

Rancangan antarmuka untuk klasifikasi *fk-nn* ditunjukkan pada gambar 3.11 berikut ini:

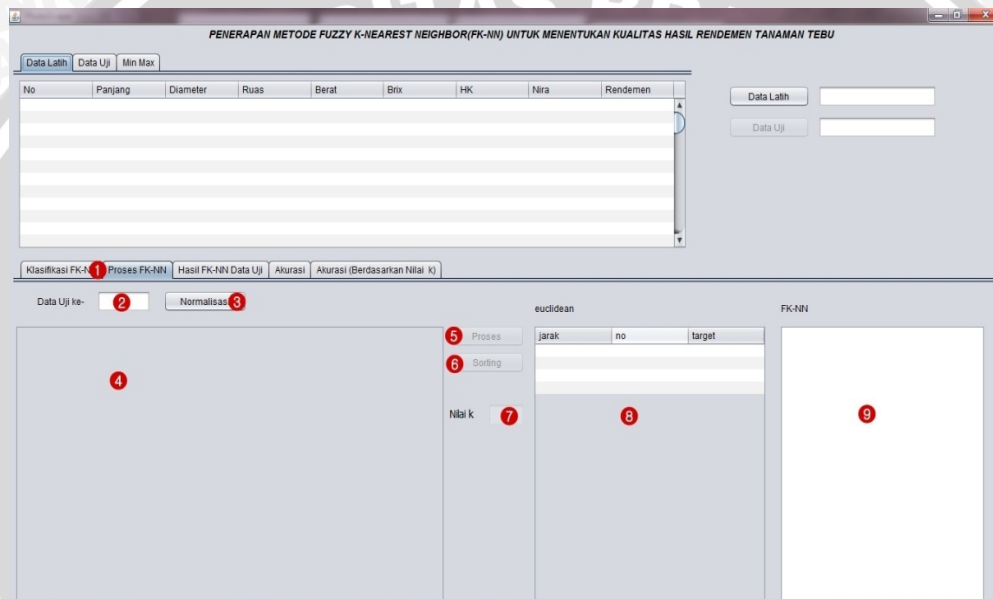
Gambar 3.11 Rancangan Antarmuka Klasifikasi FK-NN

Keterangan:

1. *JTabbedPane* data latih untuk menampilkan data latih yang digunakan
2. *JTabbedPane* data uji untuk menampilkan data uji yang digunakan
3. *JTabbedPane* max-min untuk menampilkan nilai minimum dan maximum data latih dan data uji
4. *JButton* data latih digunakan untuk mencari data latih yang akan diinputkan ke dalam sistem
5. *JButton* data uji digunakan untuk mencari data uji yang akan diinputkan ke dalam sistem
6. *JTabbedPane* menampilkan hasil klasifikasi *fk-nn*
7. *JTextField* digunakan untuk menginputkan nilai panjang, diameter, ruas, berat, brix, HK, nira
8. *JTextField* nilai k digunakan untuk menginputkan nilai yang diinginkan

9. *JButton* proses digunakan untuk memulai proses klasifikasi *fk-nn*
10. *JButton* clear digunakan untuk menghapus atau memulai proses pengujian yang baru
11. *JTextArea* *fk-nn* digunakan untuk menampilkan hasil *fk-nn* yang telah diproses

Rancangan antarmuka untuk proses *fk-nn* ditunjukkan pada gambar 3.12 berikut ini:



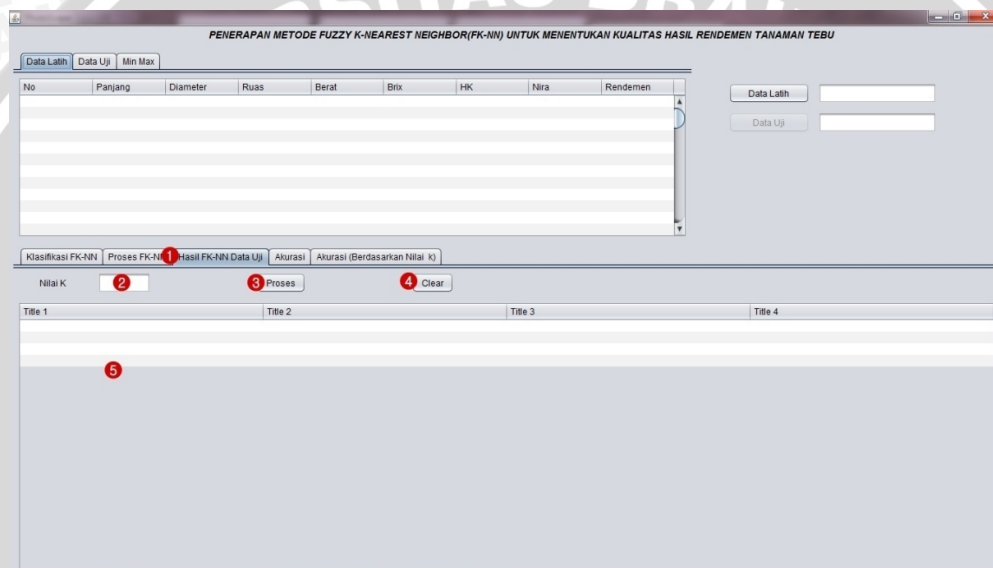
Gambar 3.12 Rancangan Antarmuka Proses FK-NN

Keterangan:

1. *JTabbedPane* menampilkan proses *fuzzy k-nearest neighbor*
2. *JTextField* data uji ke- digunakan untuk menginputkan pilihan data uji yang diinginkan
3. *JButton* normalisasi digunakan untuk memulai proses normalisasi data
4. *JTable* digunakan untuk menampilkan hasil normalisasi dari proses yang dilakukan
5. *JButton* proses digunakan untuk memulai proses pencarian jarak *euclidean*

6. *JButton* sorting digunakan untuk proses pengurutan jarak *euclidean*
7. *TextField* digunakan untuk menginputkan nilai *k* yang diinginkan
8. *JTable euclidean* digunakan untuk menampilkan hasil nilai jarak *euclidean*
9. *JScrollPane FK-NN* untuk menampilkan hasil klasifikasi *fuzzy k-nearest neighbor*

Rancangan antarmuka untuk hasil *fk-nn* data uji ditunjukkan pada gambar 3.13 berikut ini:

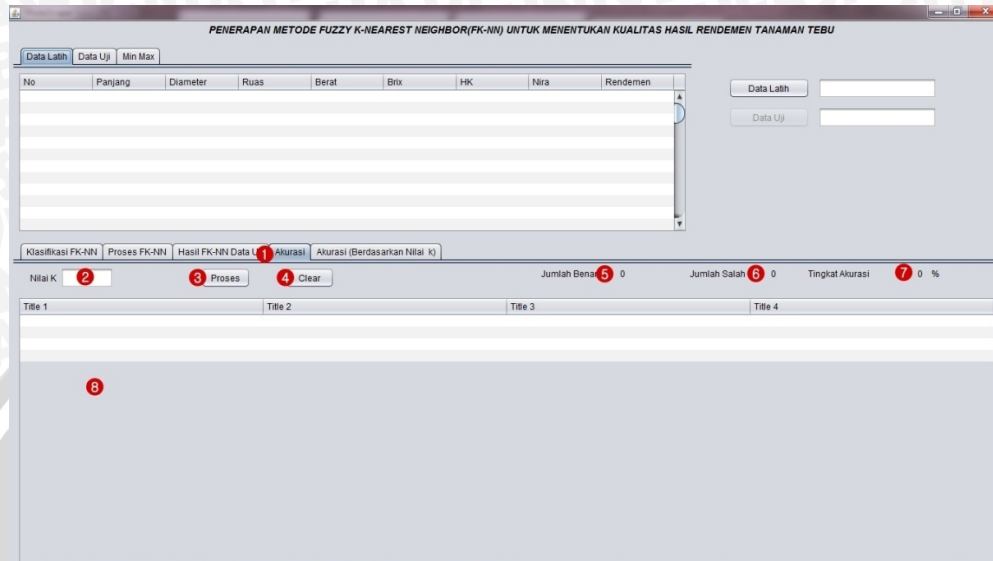


Gambar 3.13 Rancangan Antarmuka Hasil FK-NN Data Uji

Keterangan:

1. *JTabbedPane* hasil *fk-nn* data uji menampilkan proses *fk-nn* pada keseluruhan data uji
2. *TextField* digunakan untuk menginputkan nilai *k* yang diinginkan
3. *JButton* proses digunakan untuk memulai proses klasifikasi pada seluruh data uji
4. *JButton* clear digunakan untuk menghapus atau memulai proses pengujian yang baru
5. *JTable* digunakan untuk menampilkan hasil proses klasifikasi pada keseluruhan data uji.

Rancangan antarmuka untuk hasil akurasi ditunjukkan pada gambar 3.14 berikut ini:

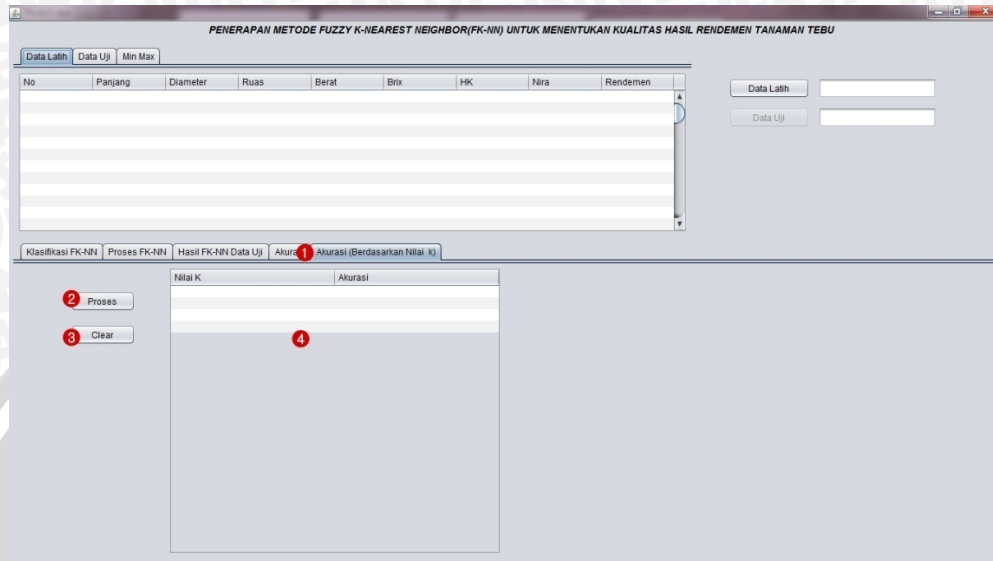


Gambar 3.14 Rancangan Antarmuka Akurasi Sistem

Keterangan:

1. *JTabbedPane* akurasi digunakan untuk menampilkan pencarian nilai akurasi dari sistem
2. *TextField* nilai k digunakan untuk menginputkan nilai k yang diinginkan
3. *Button* proses digunakan untuk memulai proses pencarian nilai akurasi
4. *Button* clear digunakan untuk menghapus atau memulai proses pengujian yang baru
5. *Label* jumlah benar untuk menampilkan berapa data uji yang benar
6. *Label* jumlah salah untuk menampilkan berapa data uji yang salah
7. *Label* akurasi digunakan untuk menampilkan hasil akurasi dari sistem
8. *Table* menampilkan hasil sistem dan hasil sebenarnya.

Rancangan antarmuka untuk hasil akurasi (berdasarkan nilai k) ditunjukkan pada gambar 3.15 berikut ini:



Gambar 3.15 Rancangan Antarmuka Akurasi (Berdasarkan Nilai k)

Keterangan:

1. *JTabbedPane* akurasi (berdasarkan nilai k) digunakan untuk menampilkan pencarian keseluruhan nilai akurasi dari sistem berdasarkan nilai k
2. *JButton* proses digunakan untuk memulai proses pencarian nilai akurasi (berdasarkan nilai k)
3. *JButton* clear digunakan untuk menghapus atau memulai proses pengujian yang baru
4. *JTable* digunakan untuk menampilkan hasil proses pencarian nilai akurasi (berdasarkan nilai k).

3.7 Perancangan Uji Coba

Setelah sistem telah selesai dibuat, maka dilanjutkan dengan proses pengujian sistem. Pengujian ini bertujuan untuk mengetahui tingkat akurasi dari hasil klasifikasi kualitas rendemen tanaman tebu menggunakan metode *fuzzy k-nearest neighbor*. Terdapat dua macam proses pengujian yaitu proses untuk mengetahui pengaruh nilai k terhadap tingkat akurasi, dan pengaruh data latih

terhadap tingkat akurasi. Nilai akurasi dapat dihitung dengan menggunakan persamaan (2-6) dan (2-7). Pengujian dan juga hasil analisis klasifikasi kualitas rendemen tanaman tebu dilakukan melalui perbandingan jumlah kesalahan antara klasifikasi kualitas rendemen tanaman tebu sistem dengan klasifikasi kualitas rendemen tanaman tebu yang dilakukan oleh pakar.

3.7.1 Rancangan Uji Pengaruh Nilai K Terhadap Tingkat Akurasi

Uji pengaruh nilai k adalah proses pengujian yang dilakukan terhadap data uji yang sama dengan beberapa nilai k yang berbeda. Dengan nilai k yang berbeda maka dapat mempengaruhi tingkat akurasi hasil keputusan untuk penentuan kelas kualitas rendemen tanaman tebu. Nilai m pada uji pengaruh nilai k dianggap konstan. Tabel 3.12 menggambarkan rancangan tabel yang digunakan untuk mencatat dari hasil pengujian.

Tabel Uji 3.12 Pengaruh Nilai k Terhadap Akurasi

Nilai k	Akurasi(%)
k=1	
k=2	
k=3	
k=4	
k=....	
k=175	

Keterangan:

1. Nilai k :Jumlah nilai k yang digunakan
2. Akurasi(%) : Nilai Akurasi Sistem

3.7.2 Rancangan Uji Pengaruh Jumlah Data Latih Terhadap Akurasi

Uji pengaruh jumlah data latih terhadap nilai akurasi adalah proses untuk mengetahui pengaruh banyaknya data latih yang digunakan terhadap tingkat akurasi sistem. Dalam proses pengujian ini nilai m yang digunakan tetap (konstan). Tabel 3.13 merupakan tabel untuk mencatat hasil nilai pengaruh jumlah data latih terhadap akurasi.

Tabel Uji 3.13 Pengaruh Jumlah Data Latih Terhadap Akurasi

Jumlah Data Latih	Akurasi(%)			
	k=9	k=10	k=11	k=12
100 Data Latih				
125 Data Latih				
150 Data Latih				
175 Data Latih				

Keterangan:

1. Jumlah Data Latih :Jumlah data latih yang digunakan
2. Akurasi(%) : Nilai Akurasi Sistem

3.7.2 Rancangan Uji Pengaruh Jumlah Data Uji Terhadap Akurasi

Uji pengaruh jumlah data latih terhadap nilai akurasi adalah proses untuk mengetahui pengaruh banyaknya data uji yang digunakan terhadap tingkat akurasi sistem. Dalam proses pengujian ini nilai m yang digunakan tetap (konstan). Tabel 3.14 merupakan tabel untuk mencatat hasil nilai pengaruh jumlah data latih terhadap akurasi.

Tabel Uji 3.14 Pengaruh Jumlah Data Uji Terhadap Akurasi

Jumlah Data Uji	Akurasi(%)			
	k=9	k=10	k=11	k=12
50 Data Uji				
60 Data Uji				
70 Data Uji				
80 Data Uji				

Keterangan:

1. Jumlah Data Uji :Jumlah data uji yang digunakan
2. Akurasi(%) : Nilai Akurasi Sistem

BAB IV

PERANCANGAN IMPLEMENTASI

4.1 Lingkungan Implementasi

Implementasi perangkat lunak ini merupakan aplikasi pemrograman yang didalamnya diterapkan metode *fuzzy k-nearest neighbor* untuk penentuan kualitas rendemen tanaman tebu. Ada beberapa aspek yang perlu diperhatikan dalam lingkungan implementasi yaitu dari segi perangkat keras (hardware) dan perangkat lunak (software).

4.1.1 Lingkungan Implementasi Perangkat Keras

Dalam pengembangan aplikasi perangkat lunak penerapan metode *fuzzy k-nearest neighbor* untuk menentukan kualitas rendemen tanaman tebu diperlukan aplikasi perangkat keras berupa sebuah laptop dengan spesifikasi sebagai berikut:

1. Processor Intel(R)Core™ i5-2410M CPU@2.30GHz
2. Memory 4 Gb
3. Hardisk 500 Gb

4.1.2 Lingkungan Implementasi Perangkat Lunak

Pengembangan aplikasi system penerapan metode *fuzzy k-nearest neighbor* untuk menentukan kualitas rendemen tanaman tebu juga memerlukan beberapa perangkat lunak. Adapun beberapa perangkat lunak yang digunakan adalah sebagai berikut:

1. System operasi yang digunakan adalah Windows 7 Ultimate
2. Aplikasi pembangun GUI dan code menggunakan NetBeans IDE 7.3.1
3. Bahasa pemrograman yang digunakan adalah java
4. Komponen java yang digunakan yaitu Java Development Kit (JDK) 1.7

4.2. Implementasi Program

Proses implementasi program ini dimulai dari proses penginputan data latih dan data uji, pencarian perhitungan nilai normalisasi, perhitungan nilai jarak euclidean, proses sorting data berdasarkan nilai yang terkecil, pengambilan data berdasarkan nilai k , perhitungan nilai membership function, perhitungan nilai $f-knn$.

4.2.1 Proses penginputan data latih dan data uji

Pada tahap ini dilakukan proses pembacaan data. Data yang dibaca terdiri dari dua jenis yaitu data latih yang digunakan sebagai data pelatihan, dan juga data uji yang digunakan sebagai data yang di cek kualitas hasil rendemen tanaman tebunya. Penyimpanan data latih maupun data uji menggunakan file txt yang kemudian dibaca menggunakan txt tersebut. Setelah terbaca maka selanjutnya data latih di pecah ke dalam delapan parameter yaitu panjang, diameter, ruas, berat brix, hk, nira dan rendemen, sedangkan didalam data uji ditambahkan parameter rata sehingga menjadi sembilan parameter. Setiap parameter dari data uji maupun data latih itu disimpan masing-masing kedalam array data. Tahapan proses pembacaan data latih maupun data uji ditunjukkan dalam *sourcecode* 4.1 dan *sourcecode* 4.2.

```
void dataLatih(String bacateks) {
    try {

        File bacafile = new File(bacateks);
        File bacafile2 = new File(bacateks); //mengubah inputan string
        //menjadi sebuah file
        FileReader inputDokumen = new FileReader(bacafile); //baca
        //baris di txt
        FileReader inputDokumen2 = new FileReader
        (bacafile2); //membaca inputan sebuah dokumen
        BufferedReader br = new BufferedReader(inputDokumen);
        BufferedReader br2 = new BufferedReader(inputDokumen2);
        String Line, Line2;
        while ((Line2 = br2.readLine()) != null) {
            jumlah=jumlah+1;
        }
        panjang = new double[jumlah];
        diameter = new double[jumlah];
        ruas = new double[jumlah];
        berat = new double[jumlah];
        brix = new double[jumlah];
        HK = new double[jumlah];
        nira = new double[jumlah];
```

```

        rendemen = new double[jumlah];
        int baris = 0;
        while ((Line = br.readLine()) != null) {
            String[] split = Line.split("\\s");
            //pemecahan data ke dalam array berdasarkan masing-masing
            parameter

            panjang[baris] = Double.parseDouble(split[0]);
            diameter[baris] = Double.parseDouble(split[1]);
            ruas[baris] = Double.parseDouble(split[2]);
            berat[baris] = Double.parseDouble(split[3]);
            brix[baris] = Double.parseDouble(split[4]);
            HK[baris] = Double.parseDouble(split[5]);
            nira[baris] = Double.parseDouble(split[6]);
            rendemen[baris] = Double.parseDouble(split[7]);
            baris++;
        }
        System.out.println();
    } catch (IOException ex) {
    }
}

```

Sourcecode 4.1 Pembacaan Data Latih

```

void dataUji(String bacateks) {
    try {

        File bacafile = new File(bacateks);
        File bacafile2 = new File(bacateks); //mengubah inputan string
        mejadi sebuah file
        FileReader inputDokumen = new FileReader(bacafile); //baca
        baris di txt
        FileReader inputDokumen2 = new FileReader
        (bacafile2); //membaca inputan sebuah dokumen
        BufferedReader br = new BufferedReader(inputDokumen);
        BufferedReader br2 = new BufferedReader(inputDokumen2);
        String Line, Line2;
        while ((Line2 = br2.readLine()) != null) {
            jumlah=jumlah+1;
        }
        panjangUji = new double[jumlah];
        diameterUji = new double[jumlah];
        ruasUji = new double[jumlah];
        beratUji = new double[jumlah];
        brixUji = new double[jumlah];
        HKUji = new double[jumlah];
        niraUji = new double[jumlah];
        Rata = new double[jumlah];
        rendemen = new String[jumlah];
        int baris = 0;
        while ((Line = br.readLine()) != null) {
            String[] split = Line.split("\\s");
            //pemecahan data ke dalam array berdasarkan masing-masing
            parameter
            panjangUji[baris] = Double.parseDouble(split[0]);

```

```

        diameterUji[baris] = Double.parseDouble(split[1]);
        ruasUji[baris] = Double.parseDouble(split[2]);
        beratUji[baris] = Double.parseDouble(split[3]);
        brixUji[baris] = Double.parseDouble(split[4]);
        HKUji[baris] = Double.parseDouble(split[5]);
        niraUji[baris] = Double.parseDouble(split[6]);
        Rata[baris] = Double.parseDouble(split[7]);
        rendemen[baris] = split[8];
        // }

        baris++;
    }
} catch (IOException ex) {
}
}

```

Sourcecode 4.2 Pembacaan Data Uji

4.2.2 Proses perhitungan normalisasi

Normalisasi dilakukan menggunakan metode *min-max normalization*. Pada tahapan normalisasi ini diambil nilai *minimum* dan *maximum* pada setiap atribut atau parameter yang telah dihitung sebelumnya. Normalisasi dilakukan pada semua data latih yang digunakan dan salah satu data uji ke- yang telah dipilih. Tahapan perhitungan nilai normalisasi ditunjukkan pada *sourcecode 4.3*.

```

public void proses1(int datake){

    double panjang,hasilPanjang,diameter,hasilDiameter,
    ruas,hasilRuas,berat,hasilBerat,brix,hasilBrix,HK,hasilHK,nira,
    hasilNira;
    //mengambil nilai maximum dan minimum
    double maxPanjang = Double.parseDouble(tblModel9.
    getValueAt(datake, 1).toString());
    double minPanjang = Double.parseDouble(tblModel9.
    getValueAt(datake, 2).toString());

    double maxDiameter = Double.parseDouble(tblModel9.
    getValueAt(datake, 3).toString());
    double minDiameter = Double.parseDouble(tblModel9.
    getValueAt(datake, 4).toString());

    double maxRuas= Double.parseDouble(tblModel9. getValueAt(datake,
    5).toString());
    double minRuas = Double.parseDouble(tblModel9. getValueAt(datake,
    6).toString());

    double maxBerat = Double.parseDouble(tblModel9. getValueAt(datake,
    7).toString());
    double minBerat = Double.parseDouble(tblModel9. getValueAt(datake,
    8).toString());
}

```

```
double maxBrix = Double.parseDouble(tblModel9. getValueAt(datake,
9).toString());
double minBrix = Double.parseDouble(tblModel9. getValueAt(datake,
10).toString());

double maxHK = Double.parseDouble(tblModel9. getValueAt(datake,
11).toString());
double minHK = Double.parseDouble(tblModel9. getValueAt(datake,
12).toString());

double maxNira = Double.parseDouble(tblModel9. getValueAt(datake,
13).toString());
double minNira = Double.parseDouble(tblModel9. getValueAt(datake,
14).toString());

//normalisasi data latih
for (int i = 0; i < jmldatalatih; i++) {
    panjang = Double.parseDouble(tblModel7.getValueAt(i,
1).toString());
    diameter= Double.parseDouble(tblModel7.getValueAt(i,
2).toString());
    ruas=Double.parseDouble(tblModel7.getValueAt(i,
3).toString());
    berat= Double.parseDouble(tblModel7.getValueAt(i, 4)
.toString());
    brix=Double.parseDouble(tblModel7.getValueAt(i,
5).toString());
    HK=Double.parseDouble(tblModel7.getValueAt(i,6).
toString());
    nira=Double.parseDouble(tblModel7.getValueAt(i,
7).toString());

    hasilPanjang = (panjang - minPanjang)/(maxPanjang -
minPanjang);
    hasilDiameter = (diameter - minDiameter)/(maxDiameter -
minDiameter);
    hasilRuas=(ruas-minRuas)/(maxRuas-minRuas);
    hasilBerat=(berat-minBerat)/(maxBerat-minBerat);
    hasilBrix=(brix-minBrix)/(maxBrix-minBerat);
    hasilHK=(HK-minHK)/(maxHK-minHK);
    hasilNira=(nira-minNira)/(maxNira-minNira);

    data4[0]=String.valueOf(i+1);
    data4[1]=String.valueOf(hasilPanjang);
    data4[2]=String.valueOf(hasilDiameter);
    data4[3]=String.valueOf(hasilRuas);
    data4[4]=String.valueOf(hasilBerat);
    data4[5]=String.valueOf(hasilBrix);
    data4[6]=String.valueOf(hasilHK);
    data4[7]=String.valueOf(hasilNira);
    tblModel4.addRow(data4);
}
```

```
//normalisasi data ujike-
panjang = Double.parseDouble(tblModel8.getValueAt (datake,
1).toString());
diameter= Double.parseDouble(tblModel8.getValueAt (datake,
2).toString());
ruas=Double.parseDouble(tblModel8.getValueAt(datake,
3).toString());
berat= Double.parseDouble(tblModel8.getValueAt(datake,
4).toString());
brix=Double.parseDouble(tblModel8.getValueAt(datake,5).
toString());
HK=Double.parseDouble(tblModel8.getValueAt(datake,6).toStri
ng());
nira=Double.parseDouble(tblModel8.getValueAt(datake,7).
toString());

hasilPanjang= (panjang - minPanjang)/(maxPanjang -
minPanjang);
hasilDiameter= (diameter - minDiameter)/(maxDiameter -
minDiameter);
hasilRuas=(ruas-minRuas)/(maxRuas-minRuas);
hasilBerat=(berat-minBerat)/(maxBerat-minBerat);
hasilBrix=(brix-minBrix)/(maxBrix-minBrix);
hasilHK=(HK-minHK)/(maxHK-minHK);
hasilNira=(nira-minNira)/(maxNira-minNira);

data4[0]=String.valueOf(tblModel4.getRowCount()+1);
data4[1]=String.valueOf(hasilPanjang);
data4[2]=String.valueOf(hasilDiameter);
data4[3]=String.valueOf(hasilRuas);
data4[4]=String.valueOf(hasilBerat);
data4[5]=String.valueOf(hasilBrix);
data4[6]=String.valueOf(hasilHK);
data4[7]=String.valueOf(hasilNira);
tblModel4.addRow(data4);
}
```

Sourcecode 4.3 Perhitungan Normalisasi

4.2.3 Proses perhitungan nilai jarak *euclidean*

Tahapan proses perhitungan jarak *euclidean* diawali dengan pengambilan data uji terpilih yang telah dinormalisasi pada tblnormalisasi(terletak setelah data latih).Selanjutnya diambil juga semua data latih yang telah dinormalisasi pada tblnormalisasi. Perhitungan hasil jarak *euclidean* didapat dari formula yang telah dijelaskan pada subbab 2.4.4.Tahapan proses perhitungan nilai jarak *euclidean* ditunjukkan pada *sourcecode* 4.4.

```
public void proses2(){

    double normalPanjang,
    normalDiameter,normalRuas,normalBerat,normalBrix,normalHK,normalNi
    ra,normalPanjangUji,normalDiameterUji,normalRuasUji,normalBeratUji
    ,normalBrixUji,normalHKUji,normalNiraUji,hasilJarak= 0;

    //pengambilan data uji ke- yang sudah di normalisasi
    int j=jmldatalatih;
    normalPanjangUji = Double.parseDouble(tblModel4. getValueAt(j,
    1).toString());
    normalDiameterUji = Double.parseDouble(tblModel4. getValueAt(j,
    2).toString());
    normalRuasUji = Double.parseDouble(tblModel4. getValueAt(j,
    3).toString());
    normalBeratUji = Double.parseDouble(tblModel4. getValueAt(j,
    4).toString());
    normalBrixUji = Double.parseDouble(tblModel4. getValueAt(j,
    5).toString());
    normalHKUji = Double.parseDouble(tblModel4. getValueAt(j,
    6).toString());
    normalNiraUji = Double.parseDouble(tblModel4. getValueAt(j,
    7).toString());

    //pengambilan data latih yang sudah dinormalisasi
    for (int i = 0; i < jmldatalatih; i++) {
        normalPanjang = Double.parseDouble(tblModel4.
        getValueAt(i, 1).toString());
        normalDiameter= Double.parseDouble(tblModel4.
        getValueAt(i, 2).toString());
        normalRuas=Double.parseDouble(tblModel4.
        getValueAt(i, 3).toString());
        normalBerat= Double.parseDouble(tblModel4.
        getValueAt(i, 4).toString());
        normalBrix=Double.parseDouble(tblModel4.
        getValueAt(i, 5).toString());
        normalHK=Double.parseDouble(tblModel4.
        getValueAt(i,6).toString());
        normalNira=Double.parseDouble(tblModel4.
        getValueAt(i, 7).toString());

        //perhitungan jarak euclidean distance
        hasilJarak=Math.sqrt(Math.pow(normalPanjangUji-normalPanjang,2)
        +Math.pow(normalDiameterUji-normalDiameter,2)
        +Math.pow(normalRuasUji-normalRuas,2)+ Math.pow(normalBeratUji-
        normalBerat,2)+Math.pow(normal BrixUji-normalBrix,2)
        +Math.pow(normalHKUji-normalHK,2) +Math.pow(normalNiraUji-
        normalNira,2));
        data5[0]=String.valueOf(hasilJarak);
        data5[1]=String.valueOf(i+1);
        data5[2]=tblModel7.getValueAt(i, 8).toString();
        tblModel5.addRow(data5); } }
```

Sourcecode 4.4 Perhitungan jarak euclidean

4.2.4 Proses Sorting Data

Tahapan sorting dilakukan dengan proses *bubble sort*. Pada awalnya dicari nilai jarak(dis) yang paling kecil atau minimum. Setelah itu dibandingkan dengan nilai jarak sebelumnya. Bila nilai jarak berikutnya bernilai minimum maka akan ditukar dengan jarak sebelumnya. Proses itu berlangsung terus hingga data terurut, dimulai dari jarak terkecil hingga terbesar. Sedangkan untuk nilai target(tar) dan nilai urut data(num) ditukar bila nilai jarak(dis) tersebut ditukar. Tahapan proses sorting data ditunjukkan pada *sourcecode* 4.5.

```
Public void proses3(){  
    int bilangan=tblModel5.getRowCount();  
    double [] dis = new double[bilangan];  
    double [] tar = new double[bilangan];  
    int[] num=new int[bilangan];  
  
    for(int a=0;a<tblModel5.getRowCount();a++){  
  
        dis[a]=Double.parseDouble(tblModel5.getValueAt(a,0).  
            toString());  
        tar[a]=Double.parseDouble(tblModel5.getValueAt(a,2).  
            toString());  
        num[a]=Integer.parseInt(tblModel5.getValueAt(a,1).  
            toString());  
    }  
    double    min = 0;  
    int        tempat = 0;  
    double    temp = 0;  
    double    temp2 = 0;  
    int        temp3=0;  
  
    // mencari nilai jarak paling minimum  
    for(int i=0; i<bilangan; i++)  
    {  
        min = dis[i];  
        for(int j=i; j<bilangan; j++)  
        {  
            if(dis[j]<=min)  
            {  
                min = dis[j];  
                tempat=j;  
            }  
        }  
  
        temp = dis[i];  
        temp2=tar[i];  
        temp3=num[i];
```

```
//proses penukaran data berdasarkan nilai yang terkecil
dis[i] = dis[tempat] ;
tar[i]=tar[tempat];
num[i]=num[tempat];
dis[tempat] = temp;
tar[tempat]=temp2;
num[tempat]=temp3;
}
for(int b=0;b<dis.length;b++){
    data[0]=String.valueOf(dis[b]);
    data[1]=String.valueOf(num[b]);
    data[2]=String.valueOf(tar[b]);
    tblModel.addRow(data);
}
}
```

Sourcecode 4.5 Proses Sorting Data

4.2.5 Proses pengambilan data berdasarkan nilai k

Tahapan pengambilan data sesuai dengan nilai k yang diinputkan ,dilakukan setelah data di sorting atau diurutkan.Data yang diambil untuk diolah sesuai dengan nilai k yang telah diinputkan.Proses pengambilan data berdasarkan nilai k ditunjukkan pada *sourcecode 4.6*.

```
int jml=nilk;
for(int i=0;i<jml;i++){
    data[0]=tblModel.getValueAt(i, 0).toString();
    data[1]=tblModel.getValueAt(i, 1).toString();
    data[2]=tblModel.getValueAt(i, 2).toString();
    tblModel2.addRow(data);
}
```

Sourcecode 4.6 pengambilan data berdasarkan k

4.2.6 Proses perhitungan nilai *membership function*

Perhitungan nilai *membership function* yang digunakan diproses pada class *membership*. Pada kelas ini ,*membership function* terbagi menjadi tiga yaitu mburuk,mbaik,msangat baik. Pada setiap perhitungan masing-masing *membership* itu terdapat tiga seleksi.Pada *membership* buruk, seleksi pertama jika nilai lebih besar sama dengan 0 dan nilai lebih kecil 3,5 maka nilai tersebut adalah 1,seleksi kedua jika nilai lebih besar sama dengan 3,5 dan nilai lebih kecil sama

dengan 4,5 maka nilai itu akan dihitung untuk mendapatkan nilainya,seleksi ketiga lebih besar 4,5 maka nilai tersebut adalah 0.

Pada *membership* baik,seleksi pertama jika nilai lebih kecil sama dengan 3,5 dan lebih besar sama dengan 5,5 maka nilai tersebut 0,seleksi kedua jika nilai lebih besar sama dengan 3,5 dan lebih kecil dari 4,5 maka nilai itu akan dihitung untuk mendapatkan nilainya,seleksi ketiga jika nilai lebih besar sama dengan 4,5 dan nilai kurang dari 5,5 maka nilai tersebut akan dihitung untuk mendapatkan nilainya.

Pada *membership* sangat baik,seleksi pertama jika nilai lebih kecil sama dengan 4,5 maka nilai tersebut adalah 0,seleksi kedua jika nilai lebih besar sama dengan 4,5 dan kurang dari 5,5 maka nilai tersebut akan dihitung untuk mendapatkan nilainya,seleksi ketiga jika nilai lebih besar sama dengan 5,5 maka nilai tersebut adalah 1.*Sourcecode* perhitungan nilai *membership function* ditunjukkan pada *sourcecode* 4.7.

```
public class membership {
    private double mburuk;
    private double mbaik;
    private double msangatbaik;

    public membership(){
    }

    //perhitungan membership buruk
    public void buruk(double nil){
        if(nil>=0 && nil<3.5){
            mburuk=1;}
        else if(nil>=3.5 && nil<=4.5){
            mburuk=(4.5-nil)/(4.5-3.5);}
        else if(nil>4.5){
            mburuk=0; }
    }

    // perhitungan membership baik
    public void baik(double nil){
        if(nil<=3.5){
            mbaik=0;}
        else if(nil>=3.5 && nil<4.5){
            mbaik=(nil-3.5)/(4.5-3.5);
        }
        else if(nil>=4.5 && nil<5.5){
            mbaik=(5.5-nil)/(5.5-4.5);
        }
        else if(nil>=5.5){
            mbaik=0;
        }
    }
}
```

```
//perhitungan membership sangat baik
public void sangatbaik(double nil){
    if(nil<=4.5){
        msangatbaik=0;}
    else if(nil>=4.5 && nil<5.5){
        msangatbaik=(nil-4.5)/(5.5-4.5);}
    else if(nil>=5.5){
        msangatbaik=1; }
    }
}
```

Sourcecode 4.7 Proses perhitungan nilai membership function

4.2.7 Proses perhitungan fuzzy k-nearest neighbor

Pada proses tahapan perhitungan fuzzy k-nearest neighbor dilakukan berdasarkan masing-masing nilai membership yaitu baik, buruk dan sangat baik. Perhitungan nilai membership akhir ini berdasarkan persamaan (2-7). Dan nilai membership akhir yang diambil adalah nilai membership bernilai paling maksimal berdasarkan perhitungan tersebut. Tahapan proses perhitungan nilai fuzzy k-nearest neighbor ditunjukkan oleh sourcecode 4.8.

```
membership mem=new membership();//inisialisasi baru untuk
memanggil kelas membership

pangkat=2/Double.valueOf(2-1);//perhitungan nilai pangkat fk-nn
dimana nilai m=2

for(int s=0;s<jml;s++){

    double nilai=Double.parseDouble(tblModel2.
    getValueAt(s, 2).toString());

    //perhitungan fk-nn buruk
    mem.buruk(nilai);
    data3[0]=String.valueOf(mem.getMburuk());
    tmbhdibagiburuk=mem.getMburuk()*(1/(Math.pow(Double.
    parseDouble(tblModel2.getValueAt(s, 0).
    toString()), pangkat)));
    tmbhpembagiburuk=(1/(Math.pow(Double.parseDouble(tblModel2.
    getValueAt(s, 0).toString()), pangkat)));
    dibagiburuk=dibagiburuk+tmbhdibagiburuk;
    pembagiburuk=pembagiburuk+tmbhpembagiburuk;

    //perhitungan fk-nn baik
    mem.baik(nilai);
    data3[1]=String.valueOf(mem.getMbaik());
    tmbhdibagibaik=mem.getMbaik()*(1/(Math.pow(Double.
    parseDouble(tblModel2.getValueAt(s, 0).toString()),
    pangkat )));
    tmbhpembagibaik=(1/(Math.pow(Double.parseDouble(tblMode
    l2.getValueAt(s, 0).toString()), pangkat)));
}
```

```

dibagibaik=dibagibaik+tmbhdibagibaik;
pembagibaik=pembagibaik+tmbhpembagibaik;

//perhitungan fk-nn sangat baik
mem.sangatbaik(nilai);
tmbhdibagisangatbaik=mem.getMangatbaik()*(1/(Math.pow(Double.parseDouble(tblModel2.getValueAt(s, 0).toString()), pangkat)));
tmbhpembagisangatbaik=(1/(Math.pow(Double.parseDouble(tblModel2.getValueAt(s, 0).toString()), pangkat)));
dibagisangatbaik=dibagisangatbaik+tmbhdibagisangatbaik;
pembagisangatbaik=pembagisangatbaik+tmbhpembagisangatbaik;
}
//perhitungan final fk-nn
hasilfinalburuk=dibagiburuk/pembagiburuk;
hasilfinalbaik=dibagibaik/pembagibaik;
hasilfinalsangatbaik=dibagisangatbaik/pembagisangat baik;

```

Sourcecode 4.8 Proses perhitungan fuzzy k-nearest neighbor

4.3 Implementasi Antarmuka

Implementasi antarmuka pada aplikasi system penentuan kualitas rendemen tanaman tebu terdiri dari lima bagian utama yaitu:

1. Form klasifikasi *FK-NN*

Form ini bertujuan untuk menentukan kualitas rendemen tanaman tebu menggunakan metode *fuzzy k-nearest neighbor* dengan cara menginputkan data-data yang merupakan atribut yang diperlukan.

2. Form proses *FK-NN*

Form ini bertujuan untuk menentukan kualitas rendemen tanaman tebu dengan menginputkan data uji yang diinginkan

3. Form hasil *FK-NN* data uji

Form ini bertujuan untuk mengetahui kualitas rendemen tanaman tebu pada keseluruhan data uji

4. Form akurasi

Form ini bertujuan untuk mengetahui jumlah data uji yang benar dan yang salah dalam proses pengujian, juga untuk mengetahui akurasi dari *dataset* rendemen yang dimasukkan.

5. Form akurasi(berdasarkan nilai k)

Form ini bertujuan untuk mengetahui nilai keseluruhan akurasi berdsarkan nilai k sebanyak data latih yang digunakan

4.3.1 Form Klasifikasi *Fuzzy K-Nearest Neighbor*

Form *fuzzy k-nearest neighbor* ini, digunakan untuk mengetahui kualitas hasil rendemen tanaman tebu berdasarkan atribut-atribut yang telah diinputkan. Terdapat sembilan atribut yang diinputkan untuk untuk diketahui kualitas hasil rendemen tanaman tebunya. Ada enam atribut yang merupakan faktor-faktor yang berpengaruh terhadap kualitas rendemen tanaman tebu yaitu panjang, diameter, ruas, berat, brix, hk, nira. Dan satu atribut yang digunakan dalam proses perhitungan *fuzzy k-nearest neighbor* yaitu nilai k . Semua data yang diinputkan pada proses *fuzzy k-nearest neighbor* ini tidak boleh *missing value* dikarenakan jika salah satu parameter atau data tidak terisi maka akan muncul peringatan dan system tidak dapat dilanjutkan prosesnya. Setelah semua input telah dimasukkan maka data akan diproses. Kemudian hasil dari proses tersebut akan ditampilkan pada *jScrollPane FK-NN*. Antarmuka dari proses form klasifikasi *FK-NN* ditunjukkan pada gambar 4.1:

The screenshot displays the 'PENERAPAN METODE FUZZY K-NEAREST NEIGHBOR (FK-NN) UNTUK MENENTUKAN KUALITAS HASIL RENDEMEN TANAMAN TEBU' application. It features a data table with 12 rows of attributes (No, Panjang, Diameter, Ruas, Berat, Brix, HK, Nira, Rendemen) and a processing window on the right. The processing window shows the results of the fuzzy k-nearest neighbor calculation for a specific input set.

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	2.28	2.74	19.0	0.53	13.1	66.7	7.0	3.91
2	2.2	2.72	16.0	0.59	12.4	64.7	6.27	3.823333333
3	2.46	2.94	15.0	0.53	13.3	65.5	6.87	3.806666667
4	2.28	2.66	16.0	0.56	13.4	66.7	7.16	3.936666667
5	2.0	2.54	15.0	0.68	13.1	66.6	6.97	3.903333333
6	2.22	2.84	16.0	0.62	13.0	65.2	6.67	3.833333333
7	2.2	2.66	15.0	0.6	13.6	66.4	7.2	3.966666667
8	2.36	2.72	16.0	0.68	13.3	65.4	6.86	3.923333333
9	2.32	2.84	16.0	0.61	13.0	65.2	6.67	3.943333333
10	2.18	2.84	16.0	0.62	13.1	66.7	7.0	3.943333333
11	2.2	2.8	15.0	0.53	13.3	66.7	7.1	3.983333333
12	2.02	2.62	14.0	0.67	13.2	66.9	6.61	3.886666667

The processing window shows the following results:

```

DIBAGI=====
buruk      = 1.4198014760
baik       = 5.8247068060
sangat baik = 2.5089104116
PEMBAGI=====
buruk      = 9.8534186935
baik       = 9.8534186935
sangat baik = 9.8534186935
NILAI MEMBERSHIP AKHIR=====
uburuk     = 0.1440922710
ubaik      = 0.5911356239
usangat baik = 0.2647721053
FK-NN=====
kualitas rendemen baik dengan nilai 0.5911356239
  
```

Gambar 4.1 Antarmuka Klasifikasi FK-NN

4.3.2 Form Proses Fuzzy K-Nearest Neighbor

Form *fuzzy k-nearest neighbor* ini, digunakan untuk mengecek kualitas rendemen tanaman tebu dari data uji yang diinginkan. Langkah awal, pengecekan kualitas rendemen tanaman tebu ini adalah dengan memasukkan data uji yang dipilih. Kedua, dilakukan proses normalisasi dari data latih maupun data uji yang digunakan. Ketiga, dilanjutkan dengan perhitungan nilai *euclidean distance* yang merupakan nilai jarak kedekatan antara data latih dan data uji yang sudah dipilih. Keempat, dilakukan proses sorting data pada jarak yang sudah dihitung sebelumnya dan juga nilai target rendemen yang sudah dimiliki mulai awal oleh data training. Kelima, memilih data training yang digunakan sesuai dengan nilai k yang telah diinputkan. Penginputan nilai k ini menandai system telah diproses sampai akhir, yang secara otomatis akan menampilkan hasil nilai *FK-NN*, dan kelas kualitas rendemen tanaman tebu dari data uji yang telah dipilih pada *jScrollPane FK-NN*. Semua parameter atau data yang diinputkan pada proses *fuzzy k-nearest neighbor* ini tidak boleh *missing value* dikarenakan jika salah satu parameter atau data tidak terisi maka akan muncul peringatan dan system tidak dapat dilanjutkan prosesnya. Antarmuka dari form proses *fuzzy k-nearest neighbor* ditunjukkan pada gambar 4.2:

PENERAPAN METODE FUZZY K-NEAREST NEIGHBOR (FK-NN) UNTUK MENENTUKAN KUALITAS HASIL RENDEMEN TANAMAN TEBU

Data Lath Data Uji Min Max

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	2.05	2.52	17.0	0.6	12.6	62.7	6.02	buruk
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	buruk
3	2.0	2.61	16.0	0.54	11.2	64.2	5.59	buruk
4	2.24	2.57	17.0	0.7	13.0	64.2	6.49	buruk
5	2.2	2.58	15.0	0.55	13.0	65.4	6.7	buruk
6	2.0	2.58	15.0	0.62	13.0	63.5	6.3	buruk
7	1.75	2.46	15.0	0.67	12.0	65.9	6.27	buruk
8	2.18	2.74	17.0	0.59	12.7	65.4	6.54	buruk
9	2.28	2.67	15.0	0.55	12.2	66.2	6.42	buruk
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	buruk
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	buruk

Klasifikasi FK-NN Proses FK-NN Hasil FK-NN Data Uji Akurasi Akurasi (Berdasarkan Nilai k)

Data Uji ke 7 Normalisasi

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	0.654054	0.506849	0.571428	0.708333	0.092272	0.279561	0.233221	
2	0.610810	0.493150	0.357142	0.541666	0.051903	0.110169	0.110738	
3	0.751351	0.643835	0.285714	0.291666	0.103806	0.177966	0.211409	
4	0.654054	0.450254	0.357142	0.416666	0.109873	0.279561	0.290967	
5	0.502702	0.369863	0.285714	0.916666	0.092272	0.271186	0.228167	
6	0.621621	0.643835	0.357142	0.666666	0.086505	0.152542	0.177852	
7	0.610810	0.450254	0.285714	0.583333	0.121107	0.254237	0.266778	
8	0.697287	0.493150	0.357142	0.916666	0.103806	0.169491	0.209731	
9	0.675675	0.575342	0.357142	0.625	0.086505	0.152542	0.177852	
10	0.600000	0.575342	0.357142	0.666666	0.092272	0.279561	0.233221	
11	0.610810	0.547945	0.285714	0.291666	0.103806	0.279561	0.249999	
12	0.613813	0.420657	0.214285	0.875000	0.098039	0.286610	0.167786	
13	0.729729	0.205479	0.428571	0.583333	0.080738	0.271186	0.211409	
14	0.621621	0.342465	0.571428	0.583333	0.086505	0.211894	0.196308	
15	0.621621	0.501643	0.285714	0.916666	0.08039	0.279561	0.239532	
16	0.689189	0.450254	0.571428	0.333333	0.103806	0.277288	0.233221	
17	0.481081	0.397260	0.357142	0.499999	0.074971	0.194915	0.174496	
18	0.524324	0.301369	0.214285	0.416666	0.109573	0.211894	0.233221	
19	0.524324	0.534246	0.214285	0.458333	0.051903	0.169491	0.130872	
20	0.535135	0.575342	0.285714	0.458333	0.086505	0.169491	0.162895	
21	0.578378	0.493150	0.357142	0.333333	0.080738	0.237288	0.196664	

Proses Sorting

Euclidean

jarak	no	target
0.1779277507	56	3.133333333
0.1976066867	5	3.903333333
0.2182850610	12	3.986666667

FK-NN

jarak = 57.6408281364
 baik = 20.5427531423
 sangat baik = 0.0

PEMBAGIAN

buruk = 78.1835812786
 baik = 78.1835812786
 sangat baik = 78.1835812786

NILAI MEMBERSHIP AKHIR

buruk = 0.7372497805
 uburuk = 0.2627502194
 usangat baik = 0.0

FK-NN

kualitas rendemen buruk dengan nilai 0.7372497

Gambar 4.2 Antarmuka Proses Fuzzy K-Nearest Neighbor

4.3.3 Form Hasil FK-NN Data Uji

Pada form hasil *fk-nn* data uji ,menampilkan keseluruhan hasil klasifikasi dari data uji yang digunakan. Langkah awal yang harus dilakukan untuk menampilkan hasil tersebut adalah dengan menginputkan nilai *k* yang diinginkan untuk proses klasifikasi yang dilakukan.Selanjutnya klik tombol proses untuk memulai klasifikasi.Hasil dari klasifikasi seluruh data uji tersebut akan ditampilkan pada *jTable* . Nilai *k* yang diinputkan pada proses ini tidak boleh *missing value* dikarenakan bisa menyebabkan proses terhenti dan akan muncul peringatan. Antarmuka form hasil *fk-nn* ditunjukkan pada gambar 4.3:

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	2.06	2.52	17.0	0.6	12.6	62.7	6.02	buruk
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	buruk
3	2.0	2.61	16.0	0.54	11.2	64.2	5.59	buruk
4	2.24	2.57	17.0	0.7	13.0	64.2	6.49	buruk
5	2.2	2.58	16.0	0.56	13.0	65.4	6.7	buruk
6	2.08	2.58	16.0	0.62	13.0	63.5	6.35	buruk
7	1.76	2.46	15.0	0.67	12.0	65.9	6.27	buruk
8	2.18	2.74	17.0	0.59	12.7	65.4	6.54	buruk
9	2.28	2.87	16.0	0.56	13.2	66.4	8.98	buruk
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	buruk
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	buruk

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Hasil FKNN	FKNN
1	2.06	2.52	17.0	0.6	12.6	62.7	6.02	0.7458629939166561	buruk
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	0.7610565180827719	buruk
3	2.0	2.61	16.0	0.54	11.2	64.2	5.59	0.7754219084224078	buruk
4	2.24	2.57	17.0	0.7	13.0	64.2	6.49	0.6119880413271408	buruk
5	2.2	2.58	16.0	0.56	13.0	65.4	6.7	0.5495204467520453	buruk
6	2.08	2.58	16.0	0.62	13.0	63.5	6.35	0.6913673749246657	buruk
7	1.76	2.46	15.0	0.67	12.0	65.9	6.27	0.622561727011833	buruk
8	2.18	2.74	17.0	0.59	12.7	65.4	6.54	0.5676354214643801	buruk
9	2.28	2.87	16.0	0.56	13.2	66.4	8.98	0.5829998486320263	baik
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	0.6919793842213487	buruk
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	0.6303027365694486	buruk
12	2.06	2.4	16.0	0.56	11.5	63.7	5.65	0.7887115709361232	buruk
13	2.02	2.62	17.0	0.61	12.7	62.2	5.98	0.7686797122463818	buruk
14	2.02	2.74	16.0	0.6	13.2	63.8	6.52	0.6430116504752759	buruk
15	2.34	2.9	22.0	0.67	12.7	65.8	6.62	0.4993996905239892	baik
16	1.8	2.46	15.0	0.62	13.2	64.5	6.63	0.586519053553951	buruk
17	2.24	2.82	16.0	0.66	12.8	64.4	6.42	0.6366582762945975	buruk
18	1.9	2.51	16.0	0.68	14.3	64.9	7.27	0.48958668387088...	baik
19	2.06	2.82	16.0	0.66	14.0	66.2	7.38	0.5957592764469655	baik
20	2.34	2.6	19.0	0.51	14.4	68.0	7.95	0.5168779999995131	baik

Gambar 4.3 Antarmuka Hasil FK-NN Data Uji

4.3.4 Form Akurasi

Pada form proses akurasi dilakukan dengan cara memasukkan jumlah nilai *k* yang digunakan untuk mengetahui jumlah data uji yang benar dan data uji yang salah yang telah diproses,juga nilai akurasi dari aplikasi program ini.Didalamnya terdapat tombol proses yang digunakan untuk memulai proses penentuan nilai akurasi system dan juga tombol clear untuk untuk mengkosongkan data untuk memulai proses pengujian yang baru.Parameter yang diinputkan pada proses akurasi ini tidak boleh *missing value* dikarenakan jika salah satu parameter tidak

terisi maka akan muncul peringatan dan system tidak dapat dilanjutkan prosesnya. Antarmuka form proses akurasi ditunjukkan pada Gambar 4.4:

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	2.06	2.52	17.0	0.6	12.6	62.7	6.02	buruk
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	buruk
3	2.0	2.61	16.0	0.54	11.2	64.2	5.59	buruk
4	2.24	2.57	17.0	0.7	13.0	64.2	6.49	buruk
5	2.2	2.58	16.0	0.56	13.0	65.4	6.7	buruk
6	2.08	2.58	16.0	0.62	13.0	63.5	6.35	buruk
7	1.76	2.46	15.0	0.57	12.0	65.9	6.27	buruk
8	2.16	2.74	17.0	0.59	12.7	65.4	6.54	buruk
9	2.28	2.87	16.0	0.56	13.2	66.4	6.98	buruk
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	buruk
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	buruk
12	2.06	2.4	16.0	0.56	11.5	63.7	5.65	buruk

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Sebenarnya	FKNN	Akurasi
1	2.06	2.52	17.0	0.6	12.6	62.7	6.02	buruk	buruk	benar
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	buruk	buruk	benar
3	2.0	2.61	16.0	0.54	11.2	64.2	5.59	buruk	buruk	benar
4	2.24	2.57	17.0	0.7	13.0	64.2	6.49	buruk	buruk	benar
5	2.2	2.58	16.0	0.56	13.0	65.4	6.7	buruk	buruk	benar
6	2.08	2.58	16.0	0.62	13.0	63.5	6.35	buruk	buruk	benar
7	1.76	2.46	15.0	0.57	12.0	65.9	6.27	buruk	buruk	benar
8	2.16	2.74	17.0	0.59	12.7	65.4	6.54	buruk	buruk	benar
9	2.28	2.87	16.0	0.56	13.2	66.4	6.98	buruk	baik	salah
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	buruk	buruk	benar
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	buruk	buruk	benar
12	2.06	2.4	16.0	0.56	11.5	63.7	5.65	buruk	buruk	benar
13	2.3	2.4	17.0	0.49	11.5	65.7	5.98	buruk	buruk	benar
14	2.02	2.62	17.0	0.61	12.7	62.2	5.98	buruk	buruk	benar
15	2.02	2.74	16.0	0.6	13.2	63.9	6.52	buruk	buruk	benar
16	2.34	2.9	22.0	0.67	12.7	65.8	6.62	buruk	baik	salah
17	1.8	2.46	15.0	0.62	13.2	64.5	6.63	buruk	buruk	benar
18	2.24	2.82	16.0	0.66	12.8	64.4	6.42	buruk	buruk	benar
19	1.9	2.51	16.0	0.66	14.3	64.9	7.27	baik	baik	benar
20	2.06	2.82	18.0	0.66	14.0	66.2	7.38	baik	baik	benar
21	2.34	2.6	19.0	0.51	14.4	68.0	7.95	baik	baik	benar

Gambar 4.4 Antarmuka Proses Akurasi

Pada gambar 4.4 nilai k yang diinputkan adalah 23. Setelah ditekan tombol proses maka aplikasi program ini akan menampilkan perbandingan nilai kelas kualitas rendemen yang sebenarnya dengan nilai kelas kualitas rendemen tanaman tebu yang dihitung dengan menggunakan proses *fuzzy k-nearest neighbor*. Pada gambar bertanda hijau ditampilkan nilai akumulasi dari jumlah benar (48 data uji) dan jumlah salah (2 data uji) dari hasil perbandingan nilai kelas kualitas rendemen tanaman tebu tersebut dan juga nilai akurasi dari aplikasi program ini sebesar 96%.

4.3.5 Form Akurasi (Berdasarkan Nilai k)

Pada form ini sama dengan form sebelumnya yaitu bertujuan untuk menampilkan akurasi dari system, namun pada form ini menampilkan keseluruhan akurasi yang didapatkan sebanyak nilai k. Nilai k maksimum adalah sebanyak data latih yang digunakan. Untuk memulai proses hanya dengan mengklik tombol proses, dan selanjutnya akan dihasilkan beberapa akurasi berdasarkan nilai k sesuai dengan banyaknya data latih. Hasil tersebut akan ditampilkan pada *jTable*

yang sudah tersedia. Antarmuka form akurasi (berdasarkan nilai k) ditunjukkan pada gambar 4.5

PENERAPAN METODE FUZZY K-NEAREST NEIGHBOR (FK-NN) UNTUK MENENTUKAN KUALITAS HASIL RENDEMEN TANAMAN TEBU

Data Latih Data Uji Min Max

No	Panjang	Diameter	Ruas	Berat	Brix	HK	Nira	Rendemen
1	2.06	2.52	17.0	0.6	12.6	62.7	6.02	buruk
2	2.14	2.58	20.0	0.6	11.3	63.0	5.45	buruk
3	2.0	2.61	18.0	0.54	11.2	64.2	5.59	buruk
4	2.24	2.57	17.0	0.7	13.0	64.2	6.48	buruk
5	2.2	2.58	16.0	0.56	13.0	65.4	6.7	buruk
6	2.08	2.58	16.0	0.62	13.0	63.5	6.35	buruk
7	1.75	2.46	15.0	0.57	12.0	65.9	6.27	buruk
8	2.16	2.74	17.0	0.59	12.7	65.4	6.54	buruk
9	2.28	2.87	16.0	0.56	13.2	66.4	6.98	buruk
10	2.1	2.78	15.0	0.55	12.2	66.2	6.42	buruk
11	1.94	2.42	16.0	0.61	12.5	65.5	6.47	buruk
12	2.06	2.4	16.0	0.66	11.6	63.7	6.66	buruk

Klasifikasi FK-NN Proses FK-NN Hasil FK-NN Data Uji Akurasi Akurasi (Berdasarkan Nilai k)

Proses Clear

Nilai K	Akurasi
1	60.0
2	56.00000000000001
3	66.0
4	66.0
5	64.0
6	68.0
7	68.0
8	68.0
9	68.0
10	66.0
11	68.0
12	68.0
13	70.0
14	70.0
15	70.0
16	70.0
17	70.0
18	72.0
19	82.0
20	88.0
21	92.0
22	96.0
23	98.0

Gambar 4.5 Antarmuka Proses Akurasi (Berdasarkan Nilai k)

Pada gambar 4.5 Setelah tombol proses di klik maka akan tampil nilai akurasi system keseluruhan sesuai nilai k yang bergerak dari satu sampai sebanyak data latih yang digunakan.

BAB V

PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis hasil penentuan kualitas rendemen tanaman tebu menggunakan metode *fuzzy k-nearest neighbor*.

5.1 Implementasi Pengujian

Pada pengujian sistem penentuan kualitas hasil rendemen tanaman tebu ini terdapat dua macam data yang digunakan yaitu sebagai data uji dan data latih. Data uji yang digunakan terbagi menjadi empat yaitu sebanyak 50, 60, 70, 80 data uji. Sedangkan data latih yang digunakan terbagi menjadi empat yaitu 100, 125, 150, dan 175. Pembagian jumlah data latih dilakukan secara seimbang berdasarkan kelas penentuan rendemen tanaman tebu (buruk, baik, sangat baik).

Pengujian sistem terdiri dari tiga macam pengujian yang dilakukan yaitu uji pengaruh nilai k terhadap tingkat akurasi, uji pengaruh jumlah data latih terhadap tingkat akurasi dan uji pengaruh jumlah data uji terhadap tingkat akurasi.

5.1.1 Uji pengaruh nilai k terhadap tingkat akurasi

Pada pengujian pengaruh nilai k terhadap tingkat akurasi, digunakan nilai k dalam rentang 1 sampai sebanyak 175. Jumlah data latih dan data uji terbaik yang digunakan yaitu 175 data latih dan 80 data uji. Tabel 5.1 menunjukkan beberapa nilai uji pengaruh nilai k terhadap tingkat akurasi.

Tabel 5.1 Uji pengaruh nilai k terhadap tingkat akurasi

Nilai k	Akuras(%)
1	79
10	94
20	94
21	94
22	94
23	95
24	96
25	96
26	97
27	97

28	96
29	96
30	96
40	97
50	97
60	97
70	98
80	97
90	97
100	96
110	96
120	96
130	96
140	96
150	96
151	96
152	96
153	96
154	96
155	96
156	96
157	96
158	96
170	96
171	96
172	96
173	96
174	96
175	96

Dari pengujian menggunakan 175 data latih dan 80 data uji dapat diketahui bahwa nilai rata-rata akurasi (%) terendah terdapat pada nilai $k=1$ yaitu sebesar 79 %. Sedangkan untuk rata-rata nilai akurasi tertinggi terdapat pada nilai $k=70$ yaitu sebesar 98 %

5.1.2 Uji pengaruh jumlah data latih terhadap tingkat akurasi

Pada uji pengaruh jumlah data latih ini digunakan empat data latih yang berbeda jumlahnya yaitu 100, 125, 150, dan 175 data latih. Sedangkan data uji bersifat tetap yaitu sebanyak 80 data uji yang merupakan data uji terbaik. Nilai k

yang digunakan adalah beberapa nilai k yang nilai akurasi sudah berada dalam keadaan stabil yaitu pada $k=9$, $k=10$, $k=11$, $k=12$. Tabel 5.2 menunjukkan beberapa nilai uji pengaruh jumlah data latih terhadap tingkat akurasi.

Tabel 5.2 Uji pengaruh Jumlah Data Latih Terhadap Tingkat Akurasi

Jumlah data latih	Akurasi(%)			
	k=9	k=10	k=11	k=12
100	88	89	88	89
125	89	90	89	89
150	93	92	93	93
175	95	94	94	94

Pada uji pengaruh jumlah data latih terhadap tingkat akurasi dapat diketahui bahwa rata-rata akurasi tertinggi yang dihasilkan adalah sebanyak 95% yang terjadi pada saat data latih berjumlah 175 dan terendah sebesar 88% pada saat data latih berjumlah 100 data latih.

5.1.3 Uji pengaruh jumlah data uji terhadap tingkat akurasi

Pada uji pengaruh jumlah data uji ini digunakan empat data uji yang berbeda jumlahnya yaitu 50, 60, 70, dan 80 data uji. Sedangkan data latih bersifat tetap yaitu sebanyak 175 data latih. Nilai k yang digunakan adalah beberapa nilai k yang nilai akurasi sudah berada dalam keadaan stabil yaitu pada 9, 10, 11, 12. Tabel 5.3 menunjukkan beberapa nilai uji pengaruh jumlah data uji terhadap tingkat akurasi.

Tabel 5.3 Uji pengaruh Jumlah Data Latih Terhadap Tingkat Akurasi

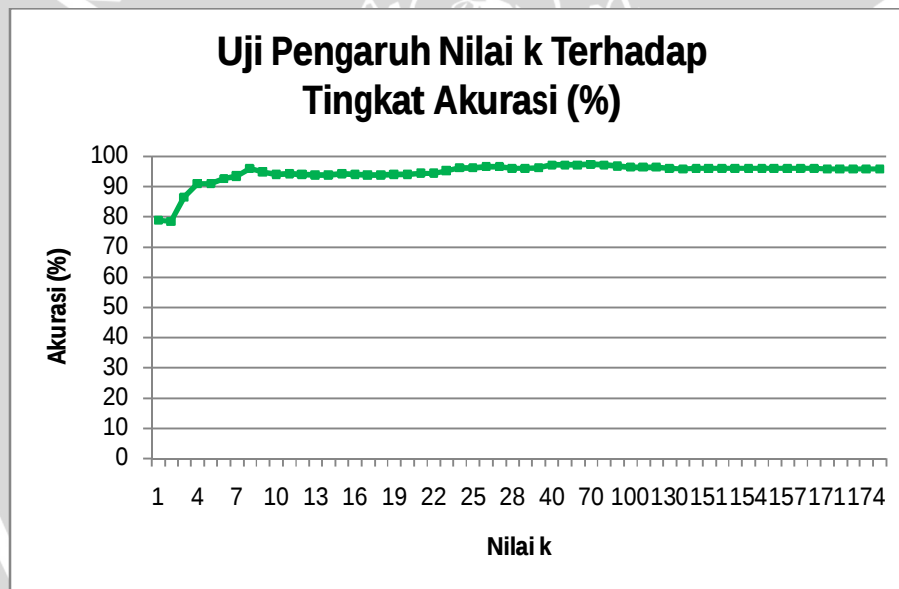
Jumlah data uji	Akurasi(%)			
	k=9	k=10	k=11	k=12
50	89	91	91	93
60	92	92	92	93
70	94	94	94	93
80	94	95	95	95

Pada uji pengaruh jumlah data uji ini akurasi terbesar yang didapatkan adalah sebanyak 95% yaitu ketika data uji berjumlah sebanyak 80 dan akurasi terendah sebesar 89% pada saat data uji berjumlah 50 data uji.

5.2 Analisa Hasil

5.2.1 Analisa Hasil Pengaruh Nilai k Terhadap Tingkat Akurasi

Pada analisa hasil pengaruh nilai k terhadap tingkat akurasi ini digunakan data latih dan data uji yang terbaik yaitu dengan jumlah 175 data latih dan 80 data uji. Nilai k yang digunakan dimulai dari $k=1$ sampai $k=175$ atau dibatasi sebanyak jumlah data latih. Grafik pengaruh jumlah k terhadap tingkat akurasi pada beberapa data latih yang berbeda dapat dilihat pada gambar grafik 5.1



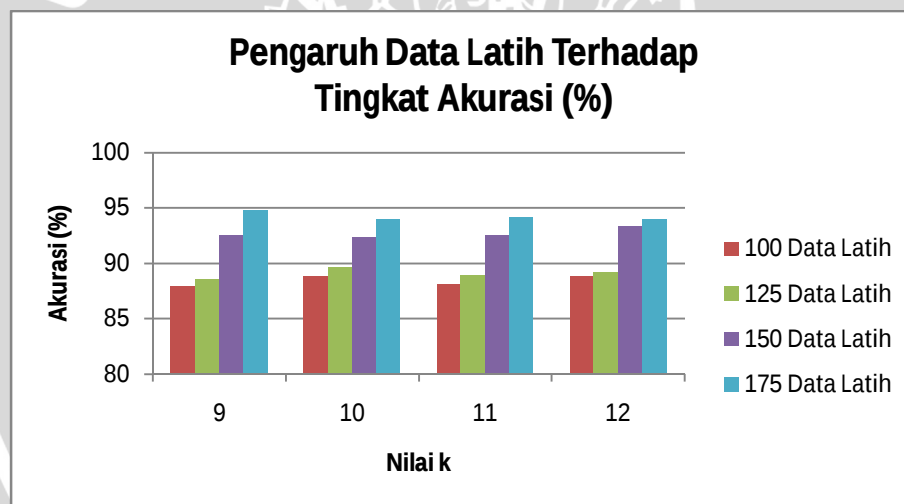
Gambar 5.1 grafik pengaruh nilai k terhadap tingkat akurasi

Pada gambar grafik 5.1 dapat terlihat bahwa pada semua data latih grafik terus mengalami kenaikan pada rentang nilai $k=1$ sampai $k=8$ dan cenderung stabil saat nilai $k>8$. Hal ini dikarenakan dengan semakin banyaknya k maka semakin banyak juga keanggotaan kelas yang mendekati kelas prediksi. Dapat diartikan juga bahwa semakin banyak nilai k maka semakin memperkecil nilai *noise* yang ada sehingga rata-rata nilai akurasi yang didapat semakin stabil. Tingkat *noise* pada sistem dapat diartikan sebagai tingkat pengaruh data

yang sangat besar pada saat klasifikasi dilakukan. Misalkan, data latih x memiliki kategori kelas baik, dilakukan pengujian dengan menggunakan nilai $k=1$ dan didapatkan kelas terdekat memiliki kategori sangat baik. Hal tersebut yang merupakan *noise* yang sangat besar dikarenakan nilai yang diambil tidak memperhitungkan pengaruh keanggotaan yang lainnya juga.

5.2.2 Analisa Hasil Pengaruh Jumlah Data Latih Terhadap Tingkat Akurasi

Berdasarkan pengujian yang telah dilakukan, terlihat bahwa jumlah data latih berpengaruh pada besar nilai akurasi yang dihasilkan. Pada pengujian pengaruh jumlah data latih ini digunakan beberapa nilai k yang nilai akurasinya sudah berada dalam keadaan stabil yaitu pada $k=9$, $k=10$, $k=11$, $k=12$, juga digunakan data uji terbaik yang berjumlah 80 data uji. Grafik hasil rata-rata akurasi untuk setiap jumlah data latih terlihat pada gambar 5.2



Gambar 5.2 grafik pengaruh jumlah data latih terhadap nilai akurasi

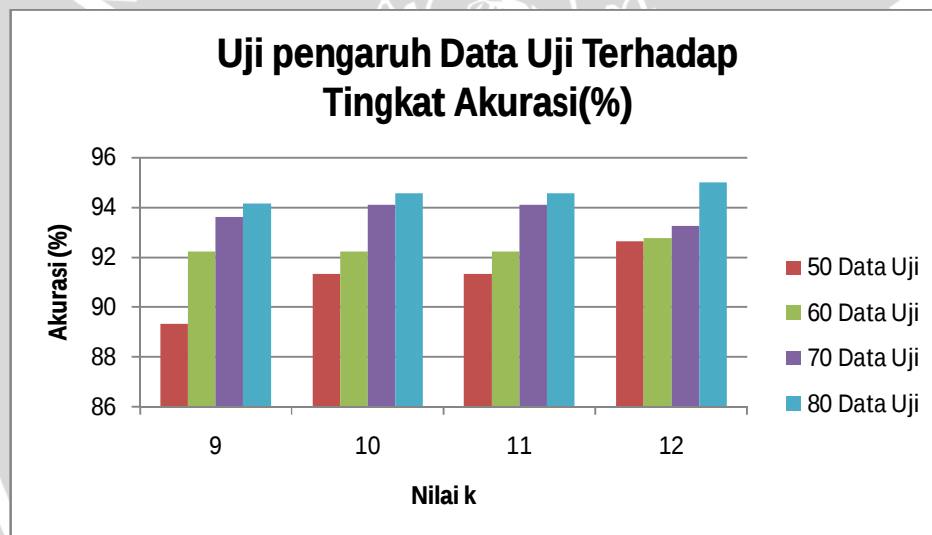
Pada gambar grafik 5.2 dapat terlihat bahwa nilai akurasi yang didapatkan berbanding lurus dengan semakin banyaknya jumlah data latih yang digunakan pada pengujian.

Dari grafik-grafik diatas dapat disimpulkan bahwa, semakin banyak jumlah data latih yang digunakan maka akan didapatkan nilai akurasi tertinggi yang lebih baik. Karena dengan semakin banyaknya data latih, maka semakin banyak data yang mendukung keanggotaan kelas klasifikasi. Apabila data latih

yang digunakan 100 data latih , maka kemungkinan kelas klasifikasi yang sesuai dengan data uji hanya berdasarkan keanggotaan kelas terdekat dari sebanyak 100 data tersebut. Namun dengan bertambahnya data latih sampai mencapai jumlah 175 data, maka kemungkinan kelas klasifikasi dari keanggotaan kelas terdekat mencapai 175 data latih tersebut.

5.2.3 Analisa Hasil Pengaruh Jumlah Data Uji Terhadap Tingkat Akurasi

Pada pengujian ini dilakukan dengan menggunakan jumlah data uji yang berbeda-beda dapat terlihat bahwa banyaknya data uji berpengaruh terhadap tingkat akurasi yang didapatkan. Pada pengujian ini digunakan empat nilai k sebagai acuan yaitu k=9, k=10, k=11, dan k=12, juga digunakan data latih terbaik yang berjumlah 175 data latih.



Gambar 5.3 grafik pengaruh jumlah data uji terhadap nilai akurasi

Pada grafik 5.3 dapat terlihat bahwa hampir seluruh grafik menunjukkan bahwa dengan semakin bertambahnya data uji maka nilai akurasi yang didapatkan semakin bagus.

Dari grafik-grafik diatas dapat disimpulkan bahwa semakin banyak data uji sebenarnya berpeluang semakin banyak kesalahan dalam pengklasifikasian data uji, namun berpeluang juga semakin banyak data uji yang benar dalam pengklasifikasiannya. Ketika semakin banyak data uji yang benar dalam pengklasifikasiannya maka semakin besar pula nilai akurasinya. Dari penelitian

ini, ternyata semakin besar nilai data uji maka semakin besar nilai akurasi yang didapatkan. Hal ini dapat dikarenakan banyaknya data latih yang digunakan itu adalah yang terbaik dari pengujian data latih sebelumnya, jumlah data latih yang digunakan dalam pengujian masih lebih banyak jumlahnya daripada jumlah data ujinya dan kemungkinan pada pengujian data uji tersebut semua sebaran data yang digunakan representatif dengan data yang digunakan sebagai data latih. Representatif dapat diartikan variasi data yang diujikan sudah banyak diwakili pada data latihnya.

UNIVERSITAS BRAWIJAYA



BAB VI

PENUTUP

6.1 Kesimpulan

Setelah melakukan penelitian tentang penerapan metode *fuzzy k-nearest neighbor (fk-nn)* untuk menentukan kualitas rendemen tanaman tebu. Maka dapat disimpulkan:

6. Metode *fuzzy k-nearest neighbor* dapat diimplementasikan untuk klasifikasi kualitas rendemen tanaman tebu. Secara singkat proses klasifikasi kualitas hasil rendemen tanaman tebu yaitu data faktor-faktor yang mempengaruhi kualitas hasil rendemen tanaman tebu (panjang, diameter, ruas, berat, %brix, harkat kemurnian, dan nira) dan nilai k diinputkan kedalam system. Pada awalnya Data uji dan data latih yang sedang digunakan diproses menggunakan *min-max normalization*. Data uji dan data latih yang telah dinormalisasi dihitung menggunakan persamaan *euclidean distance*. Nilai target kualitas hasil rendemen tanaman tebu yang digunakan ditransformasikan kedalam bentuk *fuzzy*. Pada akhirnya didapatkan hasil kualitas rendemen tanaman tebu yang dihitung menggunakan persamaan *fuzzy k-nearest neighbor*.

7. Metode *fuzzy k-nearest neighbor* memiliki akurasi yang baik untuk data klasifikasi kualitas rendemen tanaman tebu. Akurasi tertinggi yang diperoleh adalah 98%. Akurasi 98% ini didapatkan pada pengujian yang menggunakan data latih 175 data latih dan 80 data uji. Hal ini didapatkan pada saat nilai $k=70$. Jumlah nilai k sangat berpengaruh pada tingkat akurasi yang didapatkan, dimana semakin banyak nilai k maka semakin memperkecil nilai *noise* yang ada sehingga nilai akurasi yang didapat semakin stabil.

6.2 Saran

Beberapa saran yang dapat diberikan penulis untuk pengembangan lebih lanjut:

1. Pada penelitian selanjutnya dapat dilakukan dengan menggunakan nilai m yang berbeda-beda untuk mengetahui bagaimana tingkat pengaruh nilai m terhadap data kualitas rendemen tanaman tebu yang sedang diklasifikasikan.
2. Pada penelitian selanjutnya juga dapat dilakukan pengujian pengaruh data uji dan data latih yang berpasangan dan diambil dengan prosentase.



DAFTAR PUSTAKA

- [AND-11] Andaka, Ginanjar. 2011. *Hidrolis Ampas Tebu menjadi Furfural dengan Katalisator Asam Sulfa*.Yogyakarta
- [APR-13] Apriwinda. 2013. *Studi Fermentasi Nira Batang Sorgum Manis*. Makassar
- [BAS-10] Basri, Irwan. 2010. *Panduan Teknik Budidaya Tebu*.PTPN XI. Surabaya
- [HAF-08] Hafsa, Heru Cahya Rustamaji, Yulia Inayati. 2008. *Sistem Pendukung Keputusan Pemilihan Jurusan di SMU dengan Logika Fuzzy*. Seminar nasional Informatika.Yogyakarta
- [HAS-13] Hasan, Indra. 2013. *Penerapan Algoritma K-Nearest Neighbor untuk Prediksi Potensi Calon Kreditur di XYZ Finance*
- [HER-08] Hermaduanty, Ninki dan Sri Kusumadewi. 2008. *Sistem Pendukung Keputusan Berbasis Sms Untuk Menentukan Status Gizi Dengan Metode K-Nearest Neighbor* . Yogyakarta.
- [HUD-10] Huda, Nuqson Masykur. 2010. *Aplikasi Data Mining untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa*
- [KEL-85] Keller, M. James, Michael R Gray, James A. Givens. 1985. *A Fuzzy K-Nearest Neighbor*. *IEEE Transactions On Sistem, Man And Cybernetics*, Vol. SMC-15 NO 4
- [KUS-07] Kusnawi. 2007. *Pengantar Solusi Data Mining*. Seminar Nasional Teknologi. Yogyakarta.
- [KUS-10] Kusumadewi, Sri dan Hari Purnomo. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Jakarta.:Penerbit Graha Ilmu

- [KUS-12] Kuspratomo, Diyanto Aries, Burhan, Muhammad Fakhry. 2012. *Pengaruh Varietas Tebu, Potongan dan Penundaan Giling Terhadap Kualitas Nira Tebu*. Madura
- [LAT-07] Latifah, luluk , DRs.H.Mohc.Agus Krisno Budiyo. 2007. *Insertion Fitase Gen Plant to Sugar Cane (Saccharum officinarum L.) using Agrobacterium tumefaciens* . Malang
- [MAR-11] Marliani, Vitta Puspita. 2011. *Analisa Kandungan Hara N dan P serta klorofil tebu transgenic IPB 1 yang ditanam di kebun percobaan PG Djatiroto Jawa , Timur*
- [PRA-12] Prasetyo, Eko. 2012. *Fuzzy K –Nearest Neighbor in Every Class untuk Klasifikasi Data*. Seminar Nasional Teknik Informatika.
- [PUS-13] Puspasari, Maslikha. 2013. *Prediksi Tingkat Risiko Penyakit Jantung Koroner (PJK) Menggunakan Metode Fuzzy K-Nearest Neighbor (FK-NN)*. Skripsi. Malang
- [RID-13] Ridok, Ahmad dan Widia Nur Diana. 2013. *Pengklasifikasian Dokumen Teks Berita Berbahasa Indonesia menggunakan Improved K-NN*. Malang
- [YUK-07] Yukamgo, Edo dan Nasih Widya Yuono. 2007. *Peran Silikon Sebagai Unsur Bermanfaat pada Tanaman Tebu*. UGM
- [YUL-08] Yulianto J.P, Sri, Indrastanti R.W, Martha Oktiani. 2008. *Aplikasi Pendukung Keputusan Dengan Menggunakan Logika Fuzzy*. Universitas Kristen Satya Wacana.
- [ZHA-09] Zhang, Juan, Yi Niu, dan Huabei Nie. 2009. "Web Document Classification Based on Fuzzy KNN Algorithm". International Conference on Computational Intelligence and Security.

Lampiran 1. 175 Data Latih Klasifikasi Rendemen Tebu

No	Panjang (m)	Diameter (cm)	Banyak Ruas /Batang	Berat/m	%Brix	HK	Nira	Rendemen
1	2.28	2.74	19	0.63	13.1	66.7	7	3.91
2	2.2	2.72	16	0.59	12.4	64.7	6.27	3.623333
3	2.46	2.94	15	0.53	13.3	65.5	6.87	3.806667
4	2.28	2.66	16	0.56	13.4	66.7	7.16	3.936667
5	2	2.54	15	0.68	13.1	66.6	6.97	3.903333
6	2.22	2.94	16	0.62	13	65.2	6.67	3.833333
7	2.2	2.66	15	0.6	13.6	66.4	7.2	3.966667
8	2.36	2.72	16	0.68	13.3	65.4	6.86	3.923333
9	2.32	2.84	16	0.61	13	65.2	6.67	3.943333
10	2.18	2.84	16	0.62	13.1	66.7	7	3.943333
11	2.2	2.8	15	0.53	13.3	66.7	7.1	3.993333
12	2.02	2.62	14	0.67	13.2	66.9	6.61	3.986667
13	2.42	2.3	17	0.6	12.9	66.6	6.87	3.936667
14	2.22	2.5	19	0.6	13	65.9	6.78	3.76
15	2.22	2.82	15	0.68	13.2	66.7	7.04	3.836667
16	2.16	2.66	19	0.54	13.3	66.2	7	3.963333
17	1.96	2.58	16	0.58	12.8	65.7	6.65	3.753333
18	2.04	2.44	14	0.56	13.4	65.9	7	3.776667
19	2.04	2.78	14	0.57	12.4	65.4	6.39	3.496667
20	2.06	2.84	15	0.57	13	65.4	6.7	2.186667
21	2.14	2.72	16	0.54	12.9	66.2	6.8	3.833333
22	2.02	2.58	18	0.56	13.2	65	6.73	3.69
23	2.24	2.66	15	0.53	13	65.7	6.75	3.78
24	2.04	2.66	15	0.56	13.7	66.8	7.33	3.97
25	2.16	2.74	15	0.53	12.7	66	6.65	3.583333
26	2.1	2.74	15	0.59	11.8	67	6.35	3.71
27	2.16	2.66	14	0.6	12.3	64.1	6.11	3.57
28	2.12	2.4	17	0.59	13.1	64.2	6.53	3.93
29	2.38	3.26	20	0.6	12.8	64.2	6.39	3.726667
30	2.3	2.58	19	0.58	13.3	66.9	7.14	3.826667
31	1.84	2.8	14	0.66	13	64	6.45	3.616667
32	1.92	2.5	18	0.64	12	67	6.46	3.3
33	2.04	3.08	16	0.58	12	65.9	6.27	3.26
34	2.1	2.8	16	0.56	13.2	67.2	7.14	3.963333
35	2.12	3.06	16	0.64	12.8	67.4	6.95	3.693333
36	2.1	3	18	0.64	12	64.4	6.02	3.33
37	2	3.04	15	0.69	12.4	67.7	6.79	3.806667

38	2.34	2.26	16	0.59	11.5	64.2	5.73	3.326667
39	2.18	2.3	16	0.59	11.9	64	5.91	3.57
40	1.46	3	12	0.69	14.1	67.3	7.65	3.896667
41	1.7	3.46	14	0.69	12.7	65.4	6.54	3.733333
42	2.06	3	13	0.68	13	66.6	6.92	3.23
43	2.2	2.4	16	0.69	12	63.4	5.85	2.503333
44	2.32	2.44	16	0.56	12.7	65.7	6.6	3.71
45	2.3	2.48	17	0.46	12.6	64.3	6.3	3.9
46	2.08	2.66	15	0.56	11.5	63.4	5.61	2.91
47	2	2.64	14	0.63	13.4	64.6	6.76	3.736667
48	1.92	2.22	19	0.49	13.2	65.3	6.79	3.976667
49	2.4	2.28	15	0.5	13.6	65.7	7.06	3.873333
50	2.02	2.32	16	0.6	12.9	64	6.4	3.633333
51	2	2.32	18	0.57	13	65	6.63	3.773333
52	2.02	2.42	16	0.65	14	67.5	7.63	3.693333
53	2.1	2.56	15	0.64	13.1	66.7	7	3.746667
54	1.82	3	18	0.6	14.2	66.3	7.12	3.793333
55	2.5	2.5	16	0.53	12.4	65.9	6.48	3.576667
56	1.92	2.6	16	0.69	13.3	65.7	6.54	3.133333
57	2.2	2.76	15	0.57	13	65.4	6.7	3.853333
58	2.1	2.58	18	0.53	11.8	64.2	5.88	3.176667
59	1.97	2.75	18	0.63	13.9	67.7	7.61	4.336667
60	2.22	2.88	19	0.63	13.8	67.4	7.5	4.203333
61	2.18	2.19	18	0.65	14.2	67.4	7.72	4.433333
62	2.08	2.93	17	0.64	13.8	68.2	7.65	4.233333
63	2.09	2.73	17	0.61	13.8	67	7.43	4.25
64	2.13	2.73	16	0.67	14.3	67.2	7.73	4.473333
65	2.08	2.81	16	0.67	13.5	66.4	7.14	4.01
66	2.32	2.9	16	0.59	14.3	68.4	8.3	4.48
67	2.14	2.6	16	0.52	13.2	66.7	7.04	4.353333
68	2.2	2.84	16	0.67	13.7	66.4	7.26	4.1
69	2.24	2.8	21	0.68	13.8	66.4	7.3	4.12
70	2.22	2.74	15	0.62	13.6	67.4	7.4	4.216667
71	2.31	2.7	17	0.6	13.6	67.4	7.76	4.156667
72	2.12	2.82	16	0.63	13.6	67.2	7.36	4.14
73	2.5	2.74	16	0.56	13.4	66.6	7.13	4.006667
74	2.32	2.76	20	0.62	13.2	66.9	7.08	4.02
75	2	2.64	16	0.69	13.4	68.3	7.45	4.186667
76	2.59	2.85	19	0.66	15	67.6	8.2	4.723333
77	2.48	2.77	25	0.52	13.6	67.9	7.48	4.143333
78	2.3	2.54	22	0.64	14.1	66	7.39	4.31
79	2.12	2.7	15	0.6	14.4	66.7	7.68	4.373333

80	1.88	2.82	17	0.63	14.4	68.3	8.01	4.293333
81	2.12	2.79	18	0.6	13.7	66	7.17	4.206667
82	2	2.78	17	0.67	14.2	67.3	7.79	4.3
83	1.92	2.8	16	0.61	14.2	68.2	7.87	4.37
84	1.82	2.42	14	0.6	15	67.4	8.15	4.663333
85	2.34	2.54	17	0.56	14.2	67.8	7.8	4.516667
86	2.04	2.52	16	0.58	14.5	66	7.6	4.223333
87	2.34	2.24	18	0.57	13.6	67.4	7.38	4.313333
88	2.32	2.2	16	0.55	14.1	66	7.39	4.413333
89	2.3	2.86	17	0.48	14.8	67.8	8.12	4.383333
90	2	2.4	16	0.52	13.4	67.5	7.3	4.1
91	1.96	2.4	18	0.52	13.1	65.9	6.84	4.516667
92	2	2	14	0.52	12	65.9	6.27	4.086667
93	1.76	2.98	14	0.69	14.3	65.8	7.44	4.06
94	2.1	2.56	16	0.51	12.5	65.2	6.41	4.126667
95	2.34	2.68	15	0.53	13.9	67	7.47	4.36
96	1.56	2.26	11	0.57	15	67.7	8.21	4.863333
97	1.72	2.38	13	0.55	14	67.1	7.55	4.206667
98	2.06	2.74	16	0.6	14.7	67.3	7.97	4.59
99	1.76	2.78	16	0.68	14.5	67.7	7.67	4.52
100	2	2.8	19	0.5	14	68.7	7.87	4.75
101	1.86	2.5	16	0.58	14.7	68.1	8.13	4.52
102	2.5	2.48	15	0.49	13.8	67.1	7.44	4.426667
103	2.1	2.3	13	0.6	14.3	67	7.69	4.293333
104	2.28	2.46	14	0.57	13.7	65.9	7.16	4.083333
105	1.98	2.26	15	0.57	13.8	66.7	7.36	4.21
106	2.18	2.16	15	0.55	13.9	66	7.28	4.28
107	1.82	2.42	14	0.61	13.7	65.2	7.02	4.043333
108	1.96	2.54	14	0.55	14.3	68.1	7.97	4.63
109	2.12	2.64	18	0.58	14	67.7	7.67	4.2
110	2.36	2.52	20	0.55	15.2	68.5	8.49	4.826667
111	2.26	2.64	19	0.63	14.3	68.2	7.93	4.6
112	1.78	2.16	14	0.58	14.2	67.4	7.72	4.46
113	2.04	2.05	14	0.7	13	67.8	7.13	4.13
114	2.04	2.72	17	0.69	13.7	69.2	7.79	4.613333
115	2	2.5	15	0.66	13.9	67.8	7.63	4.306667
116	2.8	2.5	16	0.69	15	68	8.28	4.506
117	2.32	2.71	17	0.52	14.4	71.3	8.62	5.266667
118	2.1	2.74	19	0.64	14.8	70.7	8.72	5.136667
119	1.94	2.43	16	0.65	14.7	72	8.93	5.406667
120	2.1	2.74	17	0.57	15.9	72.4	9.75	6.013333
121	2.72	2.94	20	0.63	16.2	72.7	10.01	6.046667

122	2.92	2.82	23	0.61	16.3	73	10.14	6.203333
123	2.22	2.72	17	0.58	16	73.4	10.04	6.253333
124	2.34	2.76	22	0.57	16.1	72.5	9.9	6.23
125	2.48	2.98	19	0.63	16.2	73.8	10.25	6.36
126	2.48	2.78	19	0.58	16.8	73.3	10.51	6.59
127	2.54	2.7	17	0.67	16.8	73.2	10.5	6.566667
128	2.74	2.94	21	0.62	16.7	74	10.62	6.543333
129	2.48	2.9	19	0.63	16.4	74.2	10.48	6.546667
130	2.56	2.8	19	0.61	17	74.8	11.01	6.966667
131	2.32	2.84	20	0.6	16	73.5	10.06	6.173333
132	2.28	3	19	0.64	16.7	73.7	10.55	6.46
133	2.18	3.14	19	0.62	17.3	75	11.25	7.053333
134	2.38	3.08	22	0.63	17.6	75.2	11.48	7.15
135	2.42	3.02	22	0.63	17.8	75	11.57	7.026667
136	2.1	2.9	22	0.63	17.3	74	11	7.026667
137	2.4	3.12	22	0.6	17.3	74.3	11.07	7.02
138	2.28	2.88	22	0.64	16.3	74.2	10.41	7.546667
139	2.6	2.82	23	0.63	17.2	75.1	11.21	7.023333
140	2.42	2.86	23	0.64	17.5	75	11.38	7.186667
141	2.64	2.9	24	0.64	17.7	75.1	11.53	7.11
142	2.14	2.64	19	0.68	15	69	8.49	5.17
143	2.38	2.58	19	0.55	15.5	69.5	8.88	5.11
144	2.2	2.82	21	0.67	15.5	70.2	9.03	5.38
145	1.8	2.26	15	0.69	15.7	69.5	8.99	5.14
146	1.72	2.72	16	0.65	15.3	70	8.87	5.28
147	2.52	2.7	20	0.61	16.2	69.9	9.37	5.3
148	1.96	2.84	14	0.66	15.3	69.2	8.71	5.1
149	2.24	2.64	18	0.53	16.2	70.7	9.55	5.63
150	2.71	2.76	22	0.56	15.1	69.2	8.59	5.07
151	2.86	2.4	19	0.47	15.6	69.2	8.87	5.13
152	1.96	2.54	14	0.58	15.4	69.3	8.78	5.07
153	1.07	2.9	17	0.64	14.6	69.5	8.37	5.08
154	1.94	2.62	15	0.58	15.7	68.2	8.71	5.06
155	2.46	3.02	14	0.69	12.1	68.4	6.75	5.15
156	1.95	2.66	16	0.59	15.1	69.2	8.59	5.14
157	1.78	2.86	15	0.48	14.6	68.3	8.12	5.22
158	2.5	2.8	18	0.49	13.1	65.7	6.81	5.2
159	2.04	2.92	18	0.58	16	69	9.06	5.32
160	2.16	2.78	15	0.6	15.4	68.2	8.42	5.01
161	2.08	2.9	16	0.61	16.2	69	9.17	5.23
162	2.01	2.9	16	0.64	13.4	67.9	7.38	12.84
163	2.26	3.02	20	0.69	13.4	67.1	7.23	11.01

164	2.06	2.62	16	0.58	14	66.2	7.46	12.36
165	2.2	2.54	15	0.6	14	67.2	7.57	11.95
166	2.1	2.54	16	0.55	14.3	67.2	7.73	13.25
167	1.92	2.78	19	0.63	14	66.7	7.47	10.97
168	2	2.51	15	0.49	12.8	65.4	6.6	11.71
169	2.21	2.53	15	0.67	14.2	67.8	7.8	8.81
170	2.2	2.82	21	0.67	15.5	70.2	9.03	5.38
171	1.07	2.9	17	0.64	14.6	69.5	8.37	5.076667
172	1.8	2.54	15	0.55	16.3	69	9.23	5.013333
173	2.46	3.02	14	0.69	12.1	68.4	6.75	5.153333
174	1.96	2.54	14	0.58	15.4	69.3	8.78	5.073333
175	2.24	2.64	18	0.53	16.2	70.7	9.55	5.626667

Lampiran 2. 80 Data Uji Klasifikasi Rendemen Tebu

No	Panjang (m)	Diameter (cm)	Banyak Ruas /Batang	Berat/m	%Brix	HK	Nira	Rendemen	Kualitas
1	2.06	2.52	17	0.6	12.6	62.7	6.02	3.406667	buruk
2	2.14	2.58	20	0.6	11.3	63	5.45	3.05	buruk
3	2	2.61	16	0.54	11.2	64.2	5.59	3.416667	buruk
4	2.24	2.57	17	0.7	13	64.2	6.49	3.85	buruk
5	2.2	2.58	16	0.56	13	65.4	6.7	3.76	buruk
6	2.08	2.58	16	0.62	13	63.5	6.35	3.546667	buruk
7	1.76	2.46	15	0.67	12	65.9	6.27	3.353333	buruk
8	2.18	2.74	17	0.59	12.7	65.4	6.54	3.793333	buruk
9	2.18	2.46	14	0.54	12.2	65.2	6.25	3.603333	buruk
10	2.1	2.78	15	0.55	12.2	66.2	6.42	3.703333	buruk
11	1.94	2.42	16	0.61	12.5	65.5	6.47	3.536667	buruk
12	2.06	2.4	16	0.56	11.5	63.7	5.65	3.253333	buruk
13	2.3	2.4	17	0.49	11.5	65.7	5.98	3.356667	buruk
14	2.02	2.62	17	0.61	12.7	62.2	5.98	3.303333	buruk
15	2.02	2.74	16	0.6	13.2	63.8	6.52	3.663333	buruk
16	2.22	2.84	16	0.55	12.2	66	6.39	3.583333	buruk
17	1.8	2.46	15	0.62	13.2	64.5	6.63	3.81	buruk
18	2.24	2.82	16	0.66	12.8	64.4	6.42	3.73	buruk
19	1.92	2.82	15	0.59	12.8	65.4	6.6	3.836667	buruk
20	2	2.48	17	0.63	12.8	64.9	6.51	3.716667	buruk
21	2.24	2.8	16	0.56	13.3	65.9	6.94	3.936667	buruk
22	1.92	2.65	14	0.61	12.7	64.2	6.29	3.623333	buruk

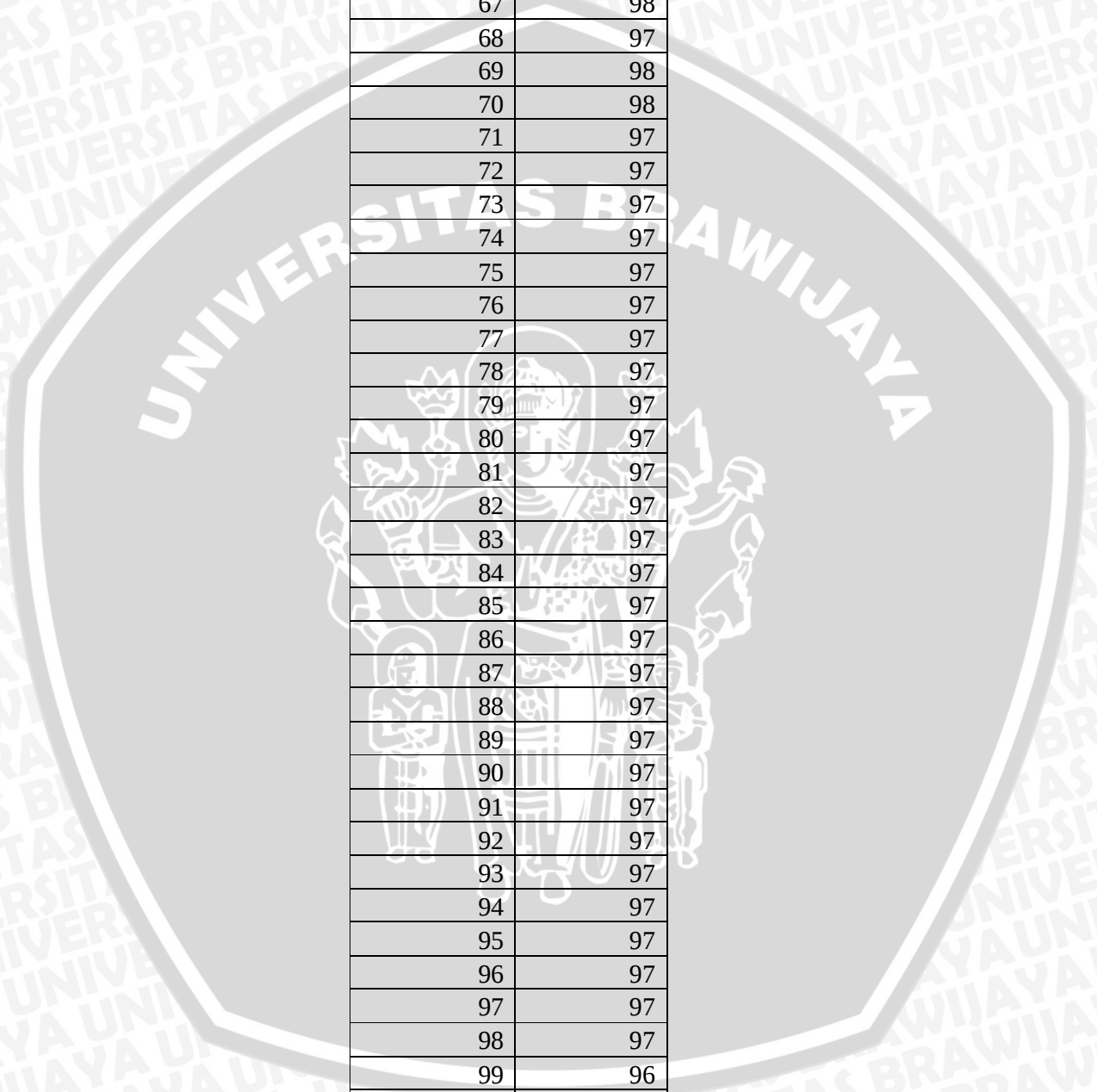
23	1.82	2.3	12	0.65	13.7	66.4	7.26	3.953333	buruk
24	1.92	2.62	17	0.6	12.7	64.8	6.44	3.673333	buruk
25	1.86	2.5	15	0.63	13	65	6.63	3.893333	buruk
26	2.1	2.32	15	0.65	13.4	64.9	6.83	3.983333	buruk
27	1.8	2.3	13	0.66	12	65.5	6.2	3.626667	buruk
28	1.84	2.22	13	0.63	13.6	64.2	6.78	3.893333	buruk
29	1.9	2.51	16	0.68	14.3	64.9	7.27	4.18	baik
30	2.06	2.82	18	0.66	14	66.2	7.38	4.346667	baik
31	2.34	2.6	19	0.51	14.4	68	7.95	4.733333	baik
32	2.54	2.8	16	0.53	14	67.8	7.69	4.124667	baik
33	2.16	2.82	14	0.56	13.8	67.4	7.5	4.193333	baik
34	2.2	2.62	18	0.6	14.1	68.1	7.8	4.593333	baik
35	2.24	2.5	21	0.59	14.2	67.9	7.82	4.44	baik
36	1.76	2.72	21	0.7	13.2	67.2	7.14	4.206667	baik
37	2.22	2.78	19	0.62	14.5	69.8	8.37	4.833333	baik
38	2.1	2.68	17	0.55	15	68.2	8.32	4.79	baik
39	2.1	2.5	18	0.6	14.6	68	8.06	4.566667	baik
40	1.98	2.58	14	0.65	13.7	67.1	7.39	4.326667	baik
41	2.24	2.48	19	0.62	14.1	64.8	7.16	4.263333	baik
42	1.8	2.71	16	0.59	14.3	65.7	7.43	4.08	baik
43	2.14	2.68	19	0.63	14.2	64.8	7.2	4.24	baik
44	1.98	2.6	18	0.64	14.3	66	7.5	4.306667	baik
45	2.12	2.5	17	0.65	14	65	7.14	4.156667	baik
46	1.98	2.58	14	0.65	13.7	67.1	7.39	4.326667	baik
47	2.02	2.54	20	0.64	14.3	65.2	7.33	4.343333	baik
48	2.08	2.56	18	0.64	16	64.9	8.13	4.653333	baik
49	1.92	2.46	15	0.61	14.6	67.9	8.03	4.253333	baik
50	2.12	2.6	18	0.58	14	66.2	7.38	4.133333	baik
51	2.08	2.52	15	0.54	14.9	66.7	7.96	4.023333	baik
52	2.5	2.54	15	0.66	13.9	67.7	7.61	4.15	baik
53	2.31	2.64	16	0.66	13.9	67	7.45	4.426667	baik
54	2.34	2.92	19	0.62	14.4	66.9	7.72	4.38	baik
55	2.08	2.54	19	0.69	15.5	69	8.77	4.956667	baik
56	2.29	2.78	15	0.63	15.3	68	8.44	4.763333	baik
57	2.3	2.82	19	0.67	13.2	65.1	6.75	4.016667	baik
58	2.57	2.58	19	0.64	13.4	66.8	7.17	4.126667	baik
59	2.63	2.98	24	0.61	18	75.3	11.77	7.396667	SangatBaik
60	2.62	2.92	26	0.62	17.7	74.9	11.48	7.14	SangatBaik
61	2.78	2.9	25	0.62	17.7	75.1	11.53	7.223333	SangatBaik
62	2.4	2.86	22	0.63	16.2	72.9	10.05	5.973333	SangatBaik
63	2.56	2.96	23	0.61	16.5	73.8	10.45	6.213333	SangatBaik
64	2.52	2.86	24	0.62	16.3	72.8	10.1	6.103333	SangatBaik

65	2.52	2.82	24	0.63	16.4	72.8	10.16	6.006667	SangatBaik
66	2.58	2.94	24	0.64	16	73.7	10.11	6.3	SangatBaik
67	2.64	2.94	25	0.64	17	74.2	10.85	6.726667	SangatBaik
68	2.42	2.9	21	0.65	16.4	73.4	10.3	6.306667	SangatBaik
69	2.52	2.88	25	0.64	16.7	74	10.62	6.506667	SangatBaik
70	1.88	2.42	16	0.67	16.2	69.7	9.33	5.473333	SangatBaik
71	2.6	2.88	22	0.61	16.5	74.1	10.52	6.446667	SangatBaik
72	2.42	2.92	22	0.63	15.7	74	9.99	6.046667	SangatBaik
73	2.48	2.9	23	0.64	16	74	10.18	6.416667	SangatBaik
74	2.3	2.9	22	0.6	16.5	74.2	10.54	6.346667	SangatBaik
75	2.6	2.92	23	0.64	16.9	73.5	10.63	6.573333	SangatBaik
76	2.56	2.86	22	0.63	16	74	10.2	6.31	SangatBaik
77	2.54	2.9	23	0.62	16.7	73	10.39	6.563333	SangatBaik
78	2.3	2.82	23	0.63	17.2	75	11.18	7.08	SangatBaik
79	2.6	2.84	23	0.6	17	74.4	10.91	6.926667	SangatBaik
80	2.44	2.84	21	0.63	16	73.4	10.04	6.383333	SangatBaik

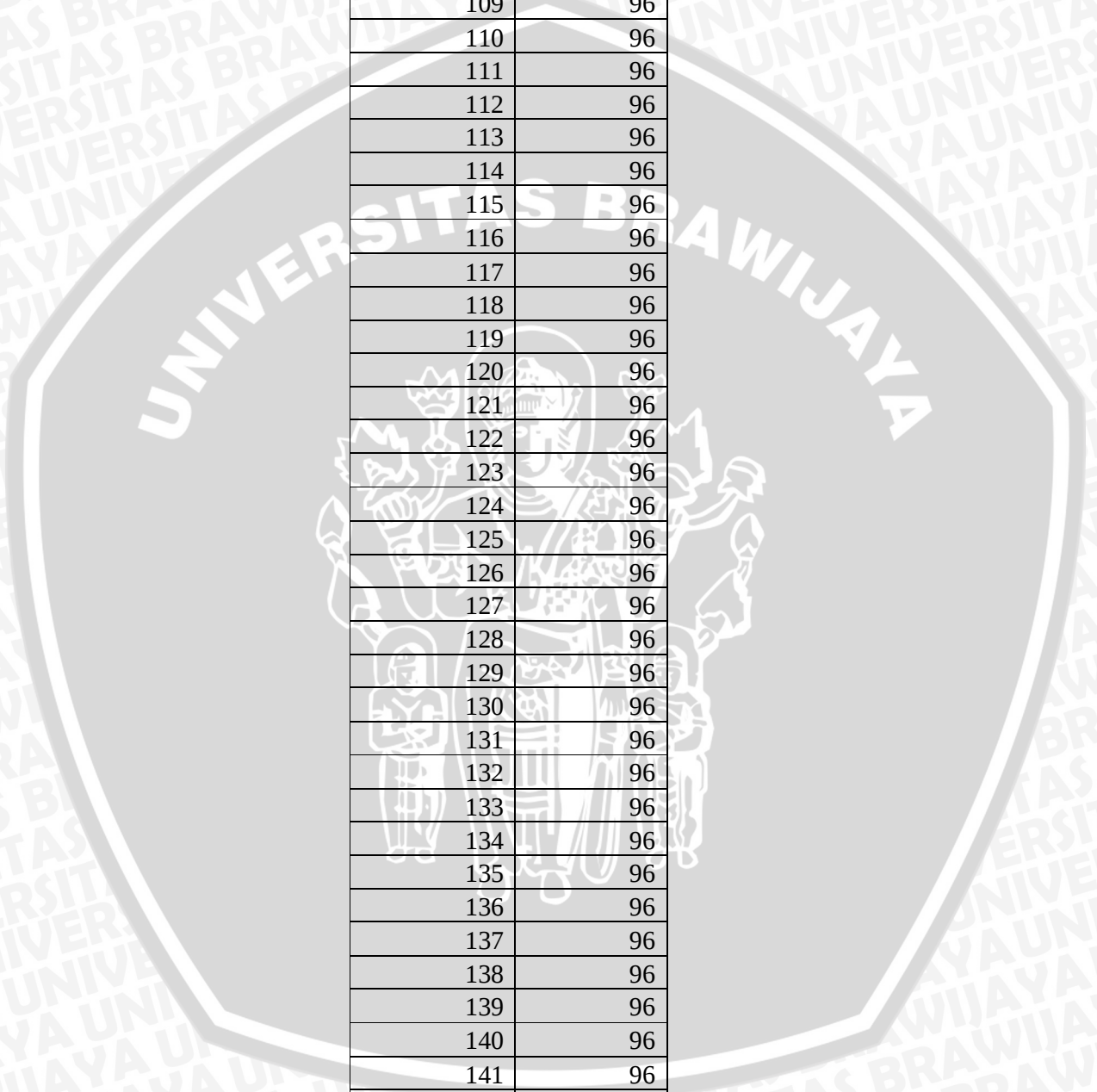
Lampiran 3. Uji Nilai k Menggunakan 175 Data Latih dan 80 Data Uji

Nilai k	Akurasi (%)
1	79
2	79
3	86
4	91
5	91
6	93
7	93
8	96
9	95
10	94
11	94
12	94
13	94
14	94
15	94
16	94
17	94
18	94
19	94
20	94

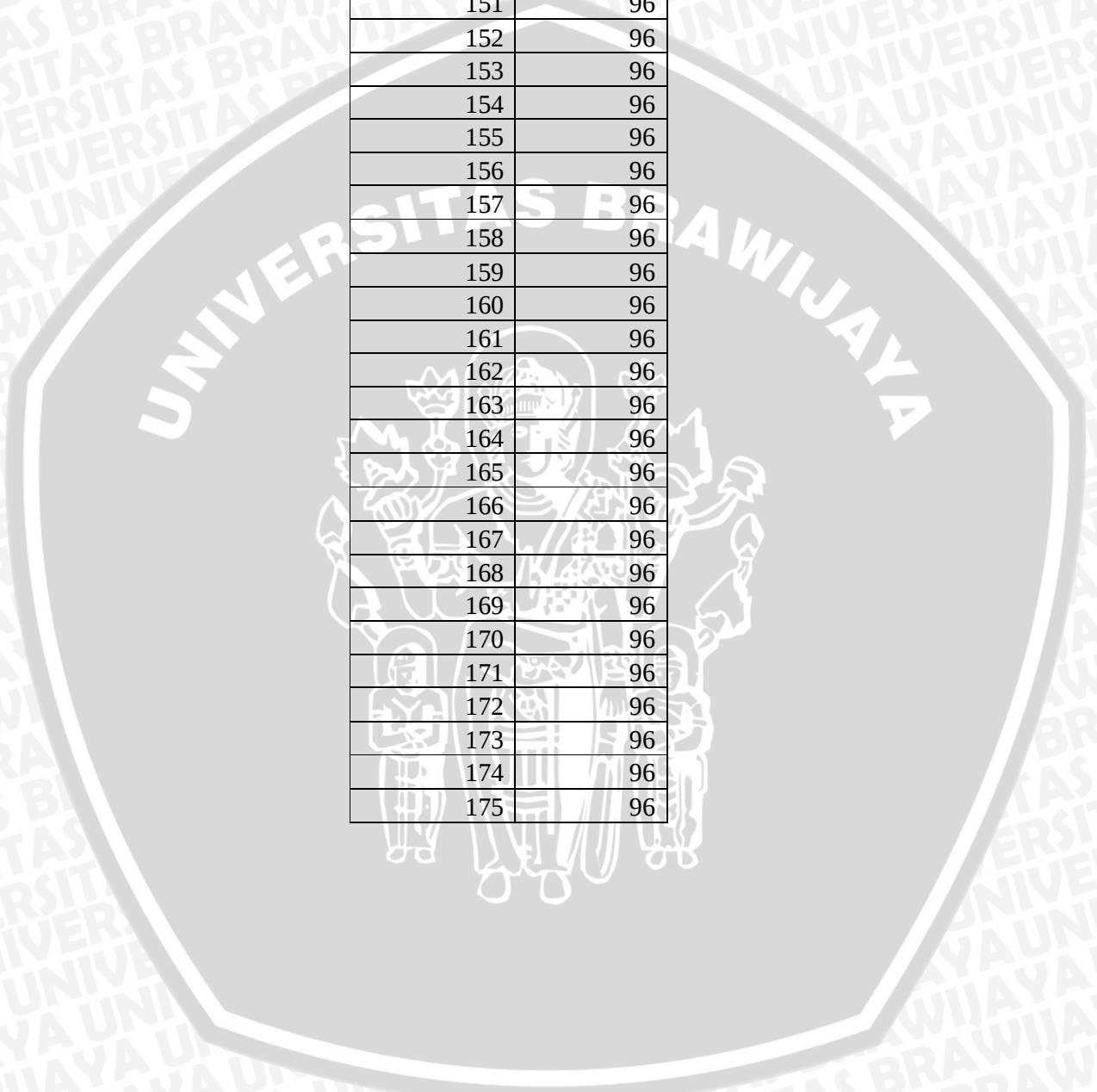
21	94
22	94
23	95
24	96
25	96
26	97
27	97
28	96
29	96
30	96
31	97
32	97
33	97
34	98
35	98
36	97
37	97
38	97
39	97
40	97
41	98
42	98
43	98
44	98
45	97
46	97
47	97
48	97
49	97
50	97
51	98
52	97
53	98
54	98
55	98
56	98
57	98
58	98
59	98
60	97
61	98
62	98



63	97
64	97
65	97
66	97
67	98
68	97
69	98
70	98
71	97
72	97
73	97
74	97
75	97
76	97
77	97
78	97
79	97
80	97
81	97
82	97
83	97
84	97
85	97
86	97
87	97
88	97
89	97
90	97
91	97
92	97
93	97
94	97
95	97
96	97
97	97
98	97
99	96
100	96
101	96
102	96
103	96
104	96



105	96
106	96
107	96
108	96
109	96
110	96
111	96
112	96
113	96
114	96
115	96
116	96
117	96
118	96
119	96
120	96
121	96
122	96
123	96
124	96
125	96
126	96
127	96
128	96
129	96
130	96
131	96
132	96
133	96
134	96
135	96
136	96
137	96
138	96
139	96
140	96
141	96
142	96
143	96
144	96
145	96
146	96



147	96
148	96
149	96
150	96
151	96
152	96
153	96
154	96
155	96
156	96
157	96
158	96
159	96
160	96
161	96
162	96
163	96
164	96
165	96
166	96
167	96
168	96
169	96
170	96
171	96
172	96
173	96
174	96
175	96