



**PENERAPAN ALGORITME NEAREST CENTROID NEIGHBOR
CLASSIFIER BASED ON K LOCAL MEANS USING HARMONIC
MEAN DISTANCE (LMKHNCN) UNTUK KLASIFIKASI HASIL
KINERJA PEGAWAI NEGERI SIPIL
(STUDI KASUS PEGAWAI NEGERI SIPIL KOTA MALANG)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Adam Syarif Hidayatullah

NIM. 165150200111152



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
TAHUN
2020

PENGESAHAN

PENERAPAN ALGORITME NEAREST CENTROID NEIGHBOR CLASSIFIER BASED ON K
LOCAL MEANS USING HARMONIC MEAN DISTANCE (LMKHNCN) UNTUK
KLASIFIKASI HASIL KINERJA PEGAWAI NEGERI SIPIL
(STUDI KASUS PEGAWAI NEGERI SIPIL KOTA MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Adam Syarif Hidayatullah
NIM: 165150200111152

Skripsi ini telah diuji dan dinyatakan lulus pada
23 Juli 2020
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2



Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng.
NIP: 198406282019031006



Imam Cholissodin, S.Si., M. Kom.
NIK: 2012018507191001

Mengetahui

Ketua Jurusan Teknik Informatika



Acimad Basuki, S.T., M.MG., Ph.D.
NIP: 197411182003121002



PERNYATAAN ORISINALITAS

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

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, 18 Juni 2020



Adam Syarif Hidayatullah

NIM: 165150200111152



ABSTRAK

Adam Syarif Hidayatullah, Penerapan Algoritme *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) Untuk Klasifikasi Hasil Kinerja Pegawai Negeri Sipil

Pembimbing: Fitra Abdurrachman Bachtiar dan Imam Cholissodin

Keberhasilan sebuah perusahaan terjadi karena dapat mengelola sumber daya manusianya dengan baik begitu juga sebaliknya. Salah satu instansi yang mengelola sumber daya manusia menggunakan Manajemen Talenta adalah Badan Kepegawaian Daerah (BKD) kota Malang, dengan mengevaluasi pegawainya setiap tahunnya setelah pekerjaan selesai dilakukan. Hal ini menyebabkan hasil pekerjaan yang telah dilakukan tidak optimal, sehingga perlu identifikasi dini pegawai yang memiliki kinerja di bawah rata – rata sehingga dapat dievaluasi dan meminimalisir hasil pekerjaan yang tidak optimal dengan menggunakan teknik klasifikasi. Penelitian ini menggunakan teknik klasifikasi *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN). Metode ini merupakan metode modifikasi dari metode *K-Nearest Neighbor* (KNN) dan dibuktikan memiliki performa lebih baik dibandingkan dengan metode aslinya KNN. Dilakukan pengujian *F1-Score* dan akurasi menggunakan *K-Fold Cross Validation* untuk mengetahui persebaran akurasi dan juga pengujian mengenai pengaruh normalisasi karena tidak ada informasi normalisasi pada penelitian sebelumnya. Metode pada penelitian ini menghasilkan performa klasifikasi yang baik, dibuktikan bahwa hasil akurasi dan *F1-Score* oleh metode ini berturut – turut ialah mencapai 96,7% dan 97,4%

Kata kunci: *Klasifikasi Kinerja Pegawai, LMKHNCN, Data Mining*



ABSTRACT

Adam Syarif Hidayatullah, Implementation of Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN) for classification of employee performance

Supervisors: Fitra Abdurrachman Bachtiar and Imam Cholissodin

The success of company occurs because is manage human resources well and vice versa. One of institute that mange human resource using Talent Mangement is Malang city Badan Kepegawaian Daerah (BKD), which evaluates its employee annually after the work is completed. This can cause not optimal work result, so it necessary to early identification of employees who have performance below average performance so that can be evaluated and minimize not optimal result. This study is use classification technique Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN). This metode is modified base algorithm of K-Nearest Neighbor (KNN). This method is proven to have better performance compared to it original algorithm KNN. F1-Score and Accuracy using K-Fold Cross Validation to measure performance of this method and normalization testing due to no any information about that in previous study. The method in this study has produced good classification performance. The result of classification accuracy and F1-Score by this method reach 96,7% and 97,4%.

Keywords: Employee Performance Classification, LMKHNCN, Data Mining



DAFTAR ISI

PERSETUJUAN.....	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR KODE PROGRAM.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan masalah.....	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Kajian Pustaka.....	6
2.2 Kinerja.....	8
2.3 <i>Data Mining</i>	9
2.3.1 <i>Preprocessing</i>	9
2.3.1.1 Data Cleaning.....	9
2.3.1.2 Normalisasi.....	10
2.4 Klasifikasi.....	11
2.5 <i>K-Nearest Neighbor (KNN)</i>	11
2.6 <i>Local Mean K-Nearest Neighbor (LMKNN)</i>	12
2.7 <i>Nearest Centroid Neighbor (NCN)</i>	13
2.8 <i>K-Nearest Centroid Neighbor (KNCN)</i>	15
2.9 <i>Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN)</i>	15
2.10 <i>Confussion Matrix</i>	16



2.11 K-Fold Cross Validation	18
------------------------------------	----

BAB 3 METODOLOGI	19
-------------------------------	-----------

3.1 Tipe Penelitian	19
---------------------------	----

3.2 Strategi Penelitian	19
-------------------------------	----

3.3 Teknik Pengumpulan Data	19
-----------------------------------	----

3.4 Perancangan Algoritme	20
---------------------------------	----

3.5 Lokasi Penelitian	21
-----------------------------	----

3.6 Peralatan Pendukung	21
-------------------------------	----

3.7 Teknik Pengujian	21
----------------------------	----

BAB 4 PERANCANGAN	22
--------------------------------	-----------

4.1 Perancangan Algoritme	22
---------------------------------	----

4.1.1 Pre-processing	23
----------------------------	----

4.1.1.1 Membersihkan Nilai Kosong (Missing Value)	23
---	----

4.1.1.2 Menghapus Baris Yang Sama (Duplikat)	26
--	----

4.1.1.3 Normalisasi	26
---------------------------	----

4.1.2 Klasifikasi LMKHNCN	27
---------------------------------	----

4.1.2.1 Perhitungan Jarak Terdekat	29
--	----

4.1.2.2 Perhitungan K-Centroid Terdekat	30
---	----

4.1.2.3 Perhitungan K-Centroid Terdekat Setiap Kelas	31
--	----

4.1.2.4 Perhitungan Local Mean Vector	33
---	----

4.1.2.5 Perhitungan Harmonic Mean Distance	34
--	----

4.1.2.6 Menentukan Hasil Kelas Klasifikasi	35
--	----

4.2 Perhitungan Manual	36
------------------------------	----

4.2.1 Dataset	36
---------------------	----

4.2.2 Perhitungan Normalisasi	38
-------------------------------------	----

4.2.3 Mencari K-Centroid Terdekat (Set of KNCN)	39
---	----

4.2.4 Perhitungan Local Mean Vector	47
---	----

4.2.5 Perhitungan Harmonic Mean Distance	49
--	----

4.2.6 Menentukan Kelas	50
------------------------------	----

4.2.7 Perhitungan Akurasi	50
---------------------------------	----

4.3 Perancangan Pengujian Metode	51
--	----

4.3.1 Pengujian Sensitivitas K dan Normalisasi	51
--	----



BAB 5 IMPLEMENTASI	52
5.1 Implementasi Program	52
5.1.1 Fungsi Pendukung	52
5.1.1.1 Import Library	52
5.1.1.2 Identifikasi Kelas	53
5.1.1.3 Euclidean Distance	53
5.1.1.4 Deviasi Standar dan Rata – Rata	54
5.1.2 Fungsi Utama	55
5.1.2.1 Pre-Processing	55
5.1.2.1.1 Missing Value	55
5.1.2.1.2 Normalisasi	56
5.1.2.1.3 Random Dataset	56
5.1.2.1.4 Ekstraksi File	57
5.1.2.1.5 K-Fold Cross Validation	58
5.1.2.2 Klasifikasi LMKHNCN	59
5.1.2.2.1 Set of KNCN	59
5.1.2.2.2 Local Mean Vector	61
5.1.2.2.3 Harmonic Mean Distance	62
5.1.2.2.4 Menentukan Kelas	63
BAB 6 Hasil dan pembahasan	64
6.1 Pengujian Akurasi dan F1-Score	64
6.2 Perbandingan Performa Metode LMKHNCN dengan KNN	69
BAB 7 PENUTUP	73
7.1 Kesimpulan	73
7.2 Saran	74
DAFTAR REFERENSI	75
DAFTAR LAMPIRAN PENGAMBILAN DATASET PEGAWAI	78
DAFTAR LAMPIRAN DATASET	82
DAFTAR LAMPIRAN HASIL AKURASI	84
DAFTAR LAMPIRAN HASIL PENGUJIAN TERBAIK	88



DAFTAR TABEL

Tabel 2.1 <i>Confussion Matrix</i> (Putri, et al., 2015)	16
Tabel 2.2 <i>Confussion Matrix Multiple Classes</i> (Manliguez, 2016)	17
Tabel 3.1 <i>Fitur Dataset</i>	20
Tabel 3.2 Perangkat pendukung berupa <i>software</i>	21
Tabel 3.3 Perangkat pendukung berupa <i>hardware</i>	21
Tabel 4.1 Data Perhitungan Manual	36
Tabel 4.2 Data Uji	37
Tabel 4.3 Data Latih	37
Tabel 4.4 Nilai Maksimum dan Minimum Setiap Fitur	38
Tabel 4.5 Data Uji (Normalisasi)	39
Tabel 4.6 Data Latih (Normalisasi)	39
Tabel 4.7 Jarak <i>Centroid</i> Data Uji 1 ($k = 1$)	40
Tabel 4.8 Jarak <i>Centroid</i> Data Uji 2 ($k = 1$)	41
Tabel 4.9 Jarak <i>Centroid</i> Data Uji 3 ($k = 1$)	41
Tabel 4.10 Jarak <i>Centroid</i> Data Uji 4 ($k = 1$)	42
Tabel 4.11 <i>Centroid</i> Data Uji 1 ($k = 2$)	43
Tabel 4.12 <i>Centroid</i> Data Uji 2 ($k = 2$)	43
Tabel 4.13 <i>Centroid</i> Data Uji 3 ($k = 2$)	44
Tabel 4.14 <i>Centroid</i> Data Uji 4 ($k = 2$)	44
Tabel 4.15 Jarak <i>Centroid</i> Data Uji 1 ($k = 2$)	45
Tabel 4.16 Jarak <i>Centroid</i> Data Uji 2 ($k = 2$)	45
Tabel 4.17 Jarak <i>Centroid</i> Data Uji 3 ($k = 2$)	46
Tabel 4.18 Jarak <i>Centroid</i> Data Uji 4 ($k = 2$)	46
Tabel 4.19 <i>Set of K- Nearest Centroid</i>	47
Tabel 4.20 <i>Local Mean Vector</i> Data Uji 1	47
Tabel 4.21 <i>Local Mean Vector</i> Data Uji 2	48
Tabel 4.22 <i>Local Mean Vector</i> Data Uji 3	48
Tabel 4.23 <i>Local Mean Vector</i> Data Uji 4	48
Tabel 4.24 <i>Harmonic Mean Distance</i> Data Uji 1	49
Tabel 4.25 <i>Harmonic Mean Distance</i> Data Uji 2	49



Tabel 4.26 <i>Harmonic Mean Distance</i> Data Uji 3	50
Tabel 4.27 <i>Harmonic Mean Distance</i> Data Uji 4	50
Tabel 4.28 Hasil Klasifikasi	50
Tabel 4.29 Perancangan Pengujian Akurasi	51
Tabel 4.30 Perancangan Pengujian <i>F1-Score</i>	51
Tabel 6.1 Hasil Pengujian Tanpa Normalisasi	64
Tabel 6.2 Hasil Pengujian dengan Normalisasi	65
Tabel 6.3 <i>F1-Score</i> Tanpa Normalisasi	66
Tabel 6.4 <i>F1-Score</i> Menggunakan Normalisasi	67
Tabel 6.5 Akurasi Metode KNN	69
Tabel 6. 6 <i>F1-Score</i> Metode KNN	70



DAFTAR GAMBAR

Gambar 2.1 Alur Proses atau Fase <i>Data Mining</i>	9
Gambar 2.2 Ilustrasi Algoritme <i>K-Nearest Neighbor</i>	12
Gambar 2.3 Ilustrasi Algoritme KNN dengan LMKNN (<i>k</i> sama dengan PNNR).....	13
Gambar 2.4 Ilustrasi Algoritme <i>Nearest Centroid Neighbor</i>	14
Gambar 2.5 Ilustrasi <i>K-Fold Cross Validation</i> (<i>K</i> = 5).....	18
Gambar 3.1 Diagram Blok Perancangan Algoritme LMKHNCN	20
Gambar 4.1 Diagram Alir Perancangan Algoritme	22
Gambar 4.2 Diagram Alir <i>Preprocessing</i>	23
Gambar 4.3 Diagram Alir Membersihkan Nilai Kosong	25
Gambar 4.4 Diagram Alir Menghapus Duplikat	26
Gambar 4.5 Diagram Alir Normalisasi	27
Gambar 4.6 Diagram Alir Klasifikasi LMKHNCN	28
Gambar 4.7 Diagram Alir Perhitungan Jarak Terdekat	29
Gambar 4.8 Diagram Alir Perhitungan <i>K-Centroid</i> Terdekat	30
Gambar 4.9 Diagram Alir Perhitungan <i>K-Centroid</i> Terdekat Setiap Kelas	32
Gambar 4.10 Diagram Alir Perhitungan <i>Local Mean Vector</i>	33
Gambar 4.11 Diagram Alir Perhitungan <i>Harmonic Mean Distance</i>	34
Gambar 4.12 Diagram Alir Menentukan Kelas Klasifikasi	35
Gambar 6.1 Grafik Hasil Pengujian Akurasi Sebelum dan Sesudah Normalisasi ..	65
Gambar 6.2 Grafik Hasil Pengujian <i>Precision</i> dan <i>Recall</i>	68
Gambar 6.3 Grafik Hasil Pengujian <i>F1-Score</i>	68
Gambar 6.4 Grafik Hasil Pengujian Akurasi	70
Gambar 6.5 Grafik Hasil Pengujian <i>Precision</i>	71
Gambar 6. 6 Grafik Hasil Pengujian <i>Recall</i>	71
Gambar 6. 7 Grafik Hasil Pengujian <i>F1-Score</i>	72



DAFTAR KODE PROGRAM

Kode Program 5.1 <i>Import Library</i>	52
Kode Program 5.2 Identifikasi Kelas.....	53
Kode Program 5.3 <i>Euclidean Distance</i>	53
Kode Program 5.4 Deviasi Standar dan Rata – Rata.....	54
Kode Program 5.5 <i>Missing Value</i>	55
Kode Program 5.6 Normalisasi.....	56
Kode Program 5.7 <i>Random Dataset</i>	57
Kode Program 5.8 Ekstraksi File.....	57
Kode Program 5.9 <i>K-Fold Cross Validation</i>	58
Kode Program 5.10 1-NCN.....	59
Kode Program 5.11 N-NCN.....	60
Kode Program 5.12 <i>Local Mean Vector</i>	61
Kode Program 5.13 <i>Harmonic Mean Distance</i>	62
Kode Program 5.14 Menentukan Kelas.....	63



BAB 1 PENDAHULUAN

1.1 Latar belakang

Seorang karyawan yang bekerja disebuah perusahaan harus memiliki kriteria sesuai bidang yang ditekuni untuk membuat perusahaan tetap bertahan di pasar. Hal ini menjadi motivasi bagi perusahaan – perusahaan dalam memilih calon karyawan yang tepat dengan melakukan berbagai macam tes. Meskipun calon karyawan telah lolos dari tes, perusahaan tetap harus menjaga dan mengelola potensi karyawannya, atau disebut dengan Manajemen Talenta (*Talent Management*). Manajemen Talenta adalah tentang bagaimana mendefinisikan kebutuhan organisasi di masa depan dan tentang bagaimana mengambil tindakan dari kebutuhan kemampuan atau bakat tersebut (Jonason, 2016).

Hampir seluruh perusahaan menerapkan Manajemen Talenta. Manajemen Talenta dibuktikan memiliki pengaruh yang signifikan untuk mempertahankan karyawan – karyawan yang berpotensi dan bertalenta tinggi serta mampu berkerja secara efektif dan optimal untuk tetap bertahan di pasar. Hal ini berdasarkan penelitian yang menunjukkan bahwa variabel dari manajemen talenta terhadap variabel kinerja memiliki pengaruh yang signifikan sebesar 63% (Satria & Gita, 2017). Salah satu instansi pemerintahan yang menerapkan Manajemen Talenta ialah Badan Kepegawaian Daerah Kota Malang.

Hasil wawancara dengan bapak Bagus Winarno S. Kom. selaku kepala bagian data dan informasi, Badan Kepegawaian Daerah Kota Malang secara intensif melakukan proses Manajemen Talenta untuk meningkatkan kinerja pegawai dengan merujuk pada nilai hasil kinerja pegawai, yaitu pegawai dengan kinerja di bawah rata - rata akan dilakukan di evaluasi maupun di *training*. Tetapi, proses Manajemen Talenta menunggu proses penilaian kinerja selesai, setelah itu baru dapat dilakukan proses Manajemen Talenta, sehingga menyebabkan buah hasil pekerjaan pegawai yang telah dilakukan tidak optimal. Hal ini dibutuhkan identifikasi dini kinerja pegawai agar dapat dilakukan proses Manajemen Talenta segera mungkin, namun membutuhkan waktu yang cukup lama karena banyaknya pegawai sebanyak kurang lebih 9000 orang.

Sebelum melakukan identifikasi perlu untuk mengetahui faktor yang memengaruhi kinerja terlebih dahulu. Faktor tersebut dapat diketahui dari beberapa penelitian yang telah dilakukan, antara lain: faktor gaji dan masa kerja terhadap kinerja pegawai yang menunjukkan pengaruh terhadap kinerja sebesar 58,4% (Akbar, 2018), faktor kenaikan pangkat memiliki pengaruh terhadap kinerja sebesar 50,5% (Lena, et al., 2016), faktor kerja sama yang memiliki pengaruh sebesar 23,5% terhadap kinerja (Septiani, 2017), serta faktor disiplin, faktor penghargaan, dan faktor umur yang memiliki hubungan positif terhadap kinerja (Safitriani, 2018) (Saga & Tri, 2016) (Rochman, 2017). Selain itu, menurut Kepala Bagian (Kabag) Sub Bidang Data dan Informasi Badan Kepegawaian Daerah Kota Malang faktor lain yang memengaruhi kinerja yaitu faktor orientasi dan faktor integritas. Dari faktor – faktor yang ada, dapat dilakukan identifikasi



dini terhadap potensi kinerja pegawai yang berada di bawah rata – rata menggunakan teknik klasifikasi. Klasifikasi merupakan pendekatan Teknik secara *Supervised Learning* (Teknik pembelajaran dengan mengacu pada *dataset* yang ada) dalam memprediksi atau mengklasifikasikan hasil kelas (Aggarwal, 2015).

Penelitian yang terkait dengan objek yang sama yaitu klasifikasi kinerja pegawai antara lain, penelitian yang dilakukan oleh Dwi, dkk serta Raihan dan Permana yang berturut-turut mengklasifikasi kinerja pegawai universitas menggunakan metode *Neuro Fuzzy* dengan akurasi 89% (Dwi, et al., 2018) dan mengklasifikasi kinerja pegawai keamanan menggunakan metode *K-Nearest Neighbor* dengan akurasi 66,7% (Raihan & Permana, 2019). Selanjutnya, penelitian yang dilakukan oleh Setyorini dan Tri melakukan komparasi metode *K-Nearest Neighbor* (KNN), *Decision Tree*, dan *Naïve Bayes* (Setiyorini & Tri, 2018). Penelitian tersebut menunjukkan bahwa metode *K-Nearest Neighbor* (KNN) memiliki akurasi tertinggi. *Decision Tree*, dan *Naïve Bayes* pada kinerja siswa dengan metode KNN memiliki akurasi tertinggi. Setiyorini juga mencoba menambahkan *Information Gain* pada KNN untuk klasifikasi kinerja siswa dan hasilnya dapat menaikkan akurasi sebesar 74% menjadi 76,5% (Setiyorini & Tri, 2019).

Dari beberapa kajian tersebut metode modifikasi *K-Nearest Neighbor* (KNN) dipilih untuk pemodelan klasifikasi kinerja pegawai. Penelitian modifikasi KNN yang telah dilakukan oleh (Mehta, et al., 2018) berdasarkan motivasi kekurangan metode KNN pada penanganan *outlier* di mana data pada penelitian ini sudah dilakukan uji karakteristik dan menunjukan adanya *outlier* dan sensitivitas dari parameter *k* menghasilkan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan tingkat error yang rendah dan stabil walaupun parameter *K* diubah dari pengujian 26 *dataset* publik UCI dan KEEL.

Sehingga berdasarkan latar belakang di atas, penelitian ini mencoba menerapkan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) untuk mengklasifikasi kinerja Pegawai Negeri Sipil (PNS) Kota Malang dan juga menguji coba menormalisasikan dan tanpa menormalisasikan data dengan menggunakan metode normalisasi *min-max* karena metode normalisasi *min-max* dapat dikombinasi dengan perhitungan jarak *similarity matrix* untuk jarak fitur kategorikal yakni nilai 0 sampai dengan 1. Pada penelitian (Mehta, et al., 2018) tidak disebutkan tahapan *preprocessing* khususnya normalisasi. Data yang tidak dinormalisasi pada metode yang menggunakan perhitungan jarak, khususnya pada fitur yang memiliki jarak data yang sangat jauh akan mendominasi dari fitur – fitur yang memiliki jarak yang rendah, sehingga dapat mengarahkan pada kesalahan klasifikasi (*misclassified*) (Theodoridis & Koutrombas, 2003). Penelitian ini juga akan membandingkan performa klasifikasi modifikasi KNN (LMKHNCN) dengan metode aslinya yakni KNN.



1.2 Rumusan masalah

Bagian yang berisi tentang pertanyaan yang diambil dari inti atau pokok masalah yang dijabarkan pada latar belakang. Pertanyaan tersebut antara lain:

1. Adakah pengaruh yang signifikan parameter k , pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) untuk klasifikasi kinerja pegawai negeri sipil Kota Malang?
2. Bagaimana tingkat akurasi dan $F1$ -Score pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) sebelum dan sesudah normalisasi untuk klasifikasi kinerja pegawai negeri sipil Kota Malang?
3. Apakah hasil tingkat akurasi dan $F1$ -Score pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) lebih baik dibandingkan dengan metode KNN konvensional?

1.3 Tujuan

Bagian yang berisi tujuan yang akan dicapai penelitian ini dalam menyelesaikan permasalahan yang telah dituliskan pada bagian rumusan masalah. Adapun tujuan – tujuan tersebut antara lain:

1. Mengetahui pengaruh parameter k metode dalam penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dalam klasifikasi kinerja pegawai negeri sipil Kota Malang.
2. Menguji dan membandingkan tingkat akurasi dan $F1$ -Score metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) sebelum dan sesudah normalisasi untuk klasifikasi kinerja pegawai negeri sipil Kota Malang.
3. Menguji perbandingan tingkat akurasi dan $F1$ -Score metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan metode KNN konvensional.

1.4 Manfaat

Bagian yang berisi tentang manfaat yang akan didapatkan dari penelitian ini antara lain:

1. Untuk mengimplementasikan ilmu pengetahuan yang didapatkan oleh peneliti selama masa perkuliahan.
2. Mengetahui kontribusi penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dalam klasifikasi kinerja pegawai negeri sipil Kota Malang untuk penelitian selanjutnya.



3. Membantu pemerintah Kota Malang untuk mengevaluasi sistem peningkatan kinerja pegawai untuk mempertimbangkan faktor yang memengaruhi kinerja pegawai negeri sipil Kota Malang.

1.5 Batasan masalah

Bagian ini menjelaskan batasan masalah yang ada pada penelitian ini. Batasan – Batasan tersebut antara lain:

1. Data yang digunakan pada penelitian ini merupakan data primer yang berasal dari *database* sistem informasi pegawai kota Malang sebanyak 6584 data.
2. Data dibagi atau diklasifikasikan menjadi 3 kelas yaitu kinerja “Amat Baik”, “Baik”, “Cukup” dan tidak memiliki label kelas “Kurang”.
3. Metode yang digunakan untuk klasifikasi kinerja pegawai negeri sipil Kota Malang adalah *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN).
4. Pemilihan fitur menggunakan penelitian terkait dengan pengaruh fitur – fitur yang ada dengan kinerja pegawai.

1.6 Sistematika pembahasan

Bagian yang berisi sistematika pembahasan terkait dengan penelitian yang dilakukan. Sistematika tersebut antara lain:

BAB 1 PENDAHULUAN

Bab ini mendeskripsikan tentang latar belakang masalah yang ada dengan menyampaikan solusinya, rumusan masalah, tujuan dari penelitian, manfaat dari penelitian, batasan masalah penelitian dan sistematika pembahasan pada klasifikasi kinerja pegawai.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini mendeskripsikan tentang teori – teori maupun penelitian sebelumnya yang berkaitan dengan klasifikasi kinerja pegawai dan solusi maupun latar belakang penelitian yang akan dilakukan, sebagai bahan rujukan dalam penelitian.

BAB 3 METODOLOGI

Bab ini menjelaskan tentang metode apa yang akan dilakukan untuk menyelesaikan masalah identifikasi dengan klasifikasi pada kinerja pegawai. Bab ini terdapat metodologi pengumpulan data, teknik penerapan dari metode yang dipakai, dan pembahasan hasil untuk penarikan kesimpulan.

BAB 4 PERANCANGAN

Bab ini menjelaskan tahapan penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) yang akan digambarkan melalui diagram dan manualisasi dari metode.



BAB 5 IMPLEMENTASI

Bab ini berisi implementasi metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dari perancangan yang telah dibangun pada bab perancangan.

BAB 6 HASIL DAN PEMBAHASAN

Bab ini menjelaskan cara dan metode pengujian yang digunakan untuk menganalisis hasil penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN).

BAB 7 PENUTUP

Bab ini mendeskripsikan penjelasan hasil penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) yang menjawab dari rumusan masalah yang telah dituliskan dan juga saran maupun evaluasi dari penerapan metode tersebut.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penilaian kinerja pada Badan Kepegawaian Daerah (BKD) Kota Malang diukur dengan cara penilaian dari atasan maupun pengawas yang bertanggung jawab untuk melakukan penilaian integritas, orientasi, kedisiplinan, dan kerja sama. Setelah melakukan penilaian, nilai akan dikirimkan ke BKD dan diproses pada sub bagian penilaian kinerja. Hasil kinerja tidak hanya berdasarkan penilaian yang telah diberikan pengawas, tetapi juga catatan, pelaporan dan hal-hal lain yang tidak bisa disebutkan karena dirahasiakan.

Hasil kinerja pegawai digunakan sebagai acuan dari proses pelatihan yang dilakukan bagi karyawan yang memiliki kinerja yang kurang baik. Karena proses pelatihan dilakukan setelah penilaian, hasil pekerjaan yang kurang baik tidak bisa dicegah. Proses identifikasi dini dibutuhkan guna mengidentifikasi pegawai yang memiliki kinerja yang kurang maupun cukup yang nantinya akan dilakukan proses pelatihan dan kinerja dapat ditingkatkan. Proses identifikasi kinerja dapat dilakukan dengan mengetahui faktor – faktor yang memengaruhi kinerja pegawai.

Faktor tersebut dapat diketahui dari beberapa penelitian yang telah dilakukan, antara lain: faktor gaji dan masa kerja terhadap kinerja pegawai yang menunjukkan pengaruh terhadap kinerja sebesar 58,4% (Akbar, 2018) faktor kenaikan pangkat memiliki pengaruh terhadap kinerja sebesar 50,5% (Lena, et al., 2016) faktor kerja sama yang memiliki pengaruh sebesar 23,5% terhadap kinerja (Septiani, 2017) serta faktor disiplin, faktor penghargaan, dan faktor umur yang memiliki hubungan positif terhadap kinerja (Safitriani, 2018) (Saga & Tri, 2016) (Rochman, 2017). Selain itu, menurut Kepala Bagian (Kabag) Sub Bidang Data dan Infomasi Badan Kepegawaian Daerah Kota Malang bapak Bagus Winarno S.Kom., faktor lain yang memengaruhi kinerja yaitu faktor orientasi dan faktor integritas.

Faktor – faktor tersebut diketahui dapat mempengaruhi kinerja pegawai juga nilai orientasi, integritas, disiplin, dan kerja sama juga mempengaruhi penilaian dari hasil kinerja. Sehingga faktor – faktor diatas dipilih untuk dijadikan fitur penentu identifikasi dini kinerja pada proses klasifikasi. Penelitian ini bertujuan mengidentifikasi dini kinerja pegawai negeri sipil pemerintah kota Malang untuk meminimalisir hasil buah pekerjaan yang kurang optimal dengan menggunakan teknik klasifikasi.

Dalam memilih teknik klasifikasi yang tepat dapat metujuk pada penelitian sebelumnya yang berkaitan. Penelitian terkait klasifikasi kinerja pegawai telah dilakukan oleh Dwi, dkk pada tahun 2018 (Dwi, et al., 2018). Metode klasifikasi yang digunakan ialah pendekatan *Neuro-Fuzzy* yaitu gabungan jaringan syaraf tiruan dengan metode logika fuzzy. Fitur yang digunakan didapatkan melalui wawancara ahli dengan latar belakang sarjana psikologi. Data yang dipakai sejumlah 171 data yang diperoleh dari unit bagian penilaian,



dengan 80 data sebagai data latih dan 91 data sebagai data uji. Metode pada penelitian ini menghasilkan akurasi sebesar 89%.

Pengambilan fitur yang didapatkan melalui wawancara dengan ahli juga dilakukan oleh (Raihan & Permana, 2019) yaitu klasifikasi kinerja pegawai keamanan. Penelitian ini menggunakan metode klasifikasi *K-Nearest Neighbor* (KNN). Pengambilan data diambil dari seorang ahli sebagai faktor penentu fitur dengan jumlah data sebanyak 130 data dan 10 fitur. Hasil dari perhitungan akurasi pada penelitian ini menggunakan 6 data uji dan 130 data latih yang menghasilkan nilai akurasi 66,7%.

Penelitian yang melakukan perbandingan metode KNN dengan metode lain yakni *Decision Tree*, dan *Naïve Bayes* terhadap kinerja siswa adalah penelitian yang dilakukan oleh Setiyorini dan Tri tahun 2018, dengan *dataset* yang digunakan adalah *dataset student performance* yang didapat dari UCI Repository dengan jumlah atribut sebanyak 30 dan 1 kelas (Setiyorini & Tri, 2018). Metodologi yang digunakan adalah mengklasifikasikan data menggunakan ketiga metode tersebut dan mengkomparasi akurasinya. Penelitian ini menunjukkan hasil dari metode *K-Nearest Neighbor* (KNN) lebih baik dibandingkan kedua metode lainnya.

Percobaan metode *K-Nearest Neighbor* (KNN) pada klasifikasi kinerja dilakukan lagi oleh (Setiyorini & Tri, 2019) dalam penerapan klasifikasi kinerja siswa dengan menambahkan metode tambahan yakni *Information Gain*. Metodologi yang dilakukan adalah dengan melakukan perbandingan metode biasa ditambahkan dengan *information gain* untuk memperingkat atribut yang paling berpengaruh dengan membuang atribut di bawah *threshold*. Teknik klasifikasi yang dilakukan dengan membagi *10-Fold Cross Validation* (data latih) dan data uji. Pengujian yang dilakukan dengan melihat performa akurasi, dengan akurasi rata – rata yang didapat yakni 74,068% dan 76,553% dengan menggunakan *information gain*.

Penelitian lain yang juga menggunakan tambahan metode yakni metode *Weighted Product* pada *K-Nearest Neighbor* (KNN) adalah penelitian yang dilakukan oleh (Nafi, et al., 2017) yaitu penerapan metode untuk penerimaan calon pegawai baru. Penelitian ini menghasilkan nilai akurasi dari hasil pengujian menggunakan *confusion matrix* dengan nilai K terbaik yaitu 3 bernilai 94% dengan nilai *precision* 100% dan *recall* 80%.

Berdasarkan rujukan – rujukan di atas, penelitian ini akan menggunakan metode *K-Nearest Neighbor* (KNN) dengan fitur ditentukan berdasarkan penelitian yang telah dilakukan. Tentunya penelitian ini memiliki pembeda yaitu pada studi kasus klasifikasi kinerja pegawai negeri sipil kota Malang. Karena metode *K-Nearest Neighbor* (KNN) memiliki kekurangan pada outlier dan juga sensitivitas pada parameter *k* (Suyanto, 2017), penelitian ini menggunakan metode modifikasi *K-Nearest Neighbor* (KNN) yaitu *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) yang telah dilakukan oleh (Mehta, et al., 2018), bahwa metode (LMKHNCN)



memiliki nilai error lebih rendah dan juga stabil dari 26 dataset yang berbeda dengan tipe data yang beragam.

Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN) merupakan salah satu metode modifikasi dari *K-Nearest Neighbor* (KNN), di mana metode *K-Nearest Neighbor* (KNN) biasa hanya melakukan perhitungan jarak terdekat yang mudah mengarahkan pada kesalahan klasifikasi (*misclassified*). Sedangkan pada metode (LMKHNCN) melakukan tambahan perhitungan dalam mempertimbangkan keputusan kelas klasifikasi. Disamping perhitungan jarak (*distance criterion*), tambahan perhitungan tersebut adalah *symmetric criterion* (Mehta, et al., 2018).

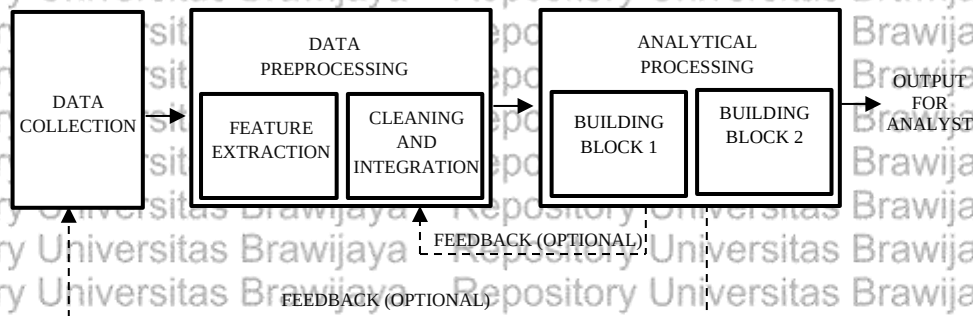
Penelitian ini merujuk pada topik kinerja pada penelitian sebelumnya yang telah diajarkan di atas dan komparasi metode terbaik yakni *K-Nearest Neighbor* (KNN) dengan tambahan modifikasi dari *K-Nearest Neighbor* (KNN) konvensional yaitu *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) yang belum pernah dilakukan pada topik klasifikasi kinerja pegawai negeri sipil kota Malang.

2.2 Kinerja

Kinerja menurut Kamus Besar Bahasa Indonesia (KBBI) merupakan sesuatu capaian, prestasi yang diperlihatkan, dan kemampuan kerja. Kinerja dapat dipahami juga sebagai hasil dari sebuah tanggung jawab yang telah diberikan. Menurut (Mangkunegara, 2000), kinerja merupakan hasil kerja yang dicapai oleh seorang pegawai secara kuantitas maupun kualitas dalam melaksanakan tugasnya sesuai dengan tanggung jawab yang diberikannya. Kinerja pegawai dalam sebuah perusahaan memiliki pengaruh besar dalam produktivitasnya. Kinerja pada Badan Kepegawaian Daerah Kota Malang salah satu parameter yang digunakan untuk mempertahankan pegawai yang memiliki semangat kerja dan kompetensi kerja yang tinggi.

2.3 Data Mining

Data mining merupakan proses pembelajaran tentang pengumpulan, pembersihan, pemrosesan, analisis dan pemerolehan manfaat dari data (Aggarwal, 2015). Ada beberapa fase atau bagian dalam melakukan proses *data mining*, proses tersebut dapat diilustrasikan pada Gambar 2.1.



Gambar 2.1 Alur Proses atau Fase Data Mining

Sumber: Aggarwal (2015)

Data Collection merupakan pengumpulan data yang mungkin membutuhkan teknik spesialisasi hardware seperti sensor maupun software seperti *web crawling engine*. Setelah proses pengumpulan data selesai, dilakukan *data preprocessing*, yaitu tidak siap untuk diproses. Dari berbagai kasus, data bersifat acak sehingga harus dilakukan penyatuan dan pembersihan data. Selanjutnya, adalah proses utama dari proses *data mining* dengan rancangan yang telah disusun dan metode yang sesuai untuk di analisis (Aggarwal, 2015).

2.3.1 Preprocessing

Basis data umumnya memiliki data yang berderau, data tidak lengkap (*missing value*), data yang tidak konsisten terlebih lagi data yang memiliki jumlah atau ukuran yang besar dan berasal dari sumber yang berbeda (Zaki & Meira, 2014). Untuk mengatasi masalah data yang berderau, tidak lengkap maupun data yang tidak konsisten, maka dapat dilakukan dengan proses pembersihan data (*data cleaning*) (Han, et al., 2012).

Missing value ditemukan pada data penelitian ini, ada beberapa pegawai yang memiliki data yang tidak lengkap. Hal ini akan memengaruhi proses klasifikasi, data yang memiliki *missing value* akan mengakibatkan kesalahan klasifikasi sehingga dapat mengurangi performa model klasifikasi (Suyanto, 2017)

2.3.1.1 Data Cleaning

Basis data dikatakan tidak bersih jika memiliki data yang berderau, data tidak lengkap (*missing value*), dan data yang tidak konsisten. Pembersihan data (*data cleaning*) dalam mengatasi nilai yang kosong (*missing value*) dapat dilakukan dengan berbagai cara antara lain (Han, et al., 2012):



1. Mengabaikan *tuple*, cara mengabaikan *tuple* merupakan cara yang efektif jika memiliki banyak nilai yang kosong. Namun sebaliknya, jika hanya sedikit yang memiliki nilai yang kosong maka nilai yang terisi tidak digunakan sama sekali.
2. Mengisi nilai kosong secara manual, hal ini akan memakan waktu yang cukup lama dan umumnya basis data dengan ukuran yang cukup besar memiliki nilai kosong yang cukup banyak.
3. Menggunakan konstanta global, mengisi nilai kosong dengan konstanta global dengan nilai tak dikenal atau menandai data yang kosong dengan nilai tertentu.
4. Menggunakan nilai tendensi sentral (rata – rata atau median), mengisi nilai kosong dengan rata – rata umumnya digunakan pada data yang memiliki distribusi normal atau penyebaran data normal, sedangkan median umumnya digunakan pada penyebaran data yang condong ke kanan maupun ke kiri.
5. Menggunakan rata – rata atau median dari setiap kelas yang sama untuk mengisi nilai yang kosong dalam kasus klasifikasi.
6. Menggunakan nilai yang paling mungkin untuk mengisi nilai yang kosong dengan teknik inferensi seperti *Naïve Bayes* atau *Decision Tree*.

Selanjutnya membersihkan data untuk data yang terdapat nilai duplikat. Data yang duplikat harus dideteksi dan dihilangkan dalam *dataset* karena untuk menghindari kesalahan klasifikasi (*misclassified*) dan data yang redundan (Han , et al., 2012). Penghapusan nilai duplikat pada penelitian ini menggunakan *library pandas* dengan cara kerja mencari dan menghapus salah satu data yang memiliki kesamaan nilai dari keseluruhan fitur.

2.3.1.2 Normalisasi

Normalisasi merupakan penyamaan jarak fitur data dari fitur satu ke fitur lainnya, karena perbedaan jarak data dapat memengaruhi hasil klasifikasi maupun klustering, data memiliki jarak yang jauh dapat mendominasi fitur data yang memiliki jarak yang kecil (Theodoridis & Koutrombas, 2003). Normalisasi data dapat menggunakan beberapa metode salah satunya normalisasi *Min-Max*.

Normalisasi *Min-Max* dapat membuat jarak seluruh fitur data menjadi seragam yakni 0 sampai 1. Berikut merupakan persamaan dari normalisasi *Min-Max* yang dapat dilihat pada Persamaan 2.1 (Zaki & Meira, 2014).

$$v' = \frac{v - \min}{\max - \min} \quad (2.1)$$

Keterangan:

v' : normalisasi

$\min, \max$: nilai minimal dan maksimal



2.4 Klasifikasi

Menurut (Suyanto, 2017), klasifikasi merupakan bagaimana cara mempelajari data dari sekumpulan banyak data yang nantinya akan menghasilkan aturan yang dapat mengklasifikasi atau mengenali data – data yang akan diuji atau data baru di mana data baru tersebut belum pernah dipelajari sebelumnya. Metode klasifikasi memiliki beberapa algoritme antara lain, *Support Vector Machine (SVM)*, *Naïve Bayes*, *Decision Tree*, *Nearest Neighbor Classifier*, dan masih banyak algoritme klasifikasi lainnya.

2.5 K-Nearest Neighbor (KNN)

Menurut (Aggarwal, 2015), algoritme *K-Nearest Neighbor (KNN)* merupakan algoritme klasifikasi pencarian data dalam k terdekat. Di mana jika data i memiliki sejumlah k tetangga terdekat, dan i didominasi oleh kelas x maka i merupakan kelas x . Sehingga, dengan kata lain metode ini membutuhkan nilai k dan perhitungan jarak didalamnya.

Perhitungan jarak merupakan langkah pertama dalam metode *K-Nearest Neighbor (KNN)*. Menghitung jarak terdekat antara data latih dengan data uji, umumnya menggunakan metode perhitungan jarak *Euclidean Distance*. Alternatif perhitungan lainnya adalah metode perhitungan jarak *Manhattan Distance* atau *city-block metric* (Witten & Frank, 2005). Berikut merupakan persamaan perhitungan jarak menggunakan metode *Euclidean Distance* yang dapat dilihat pada Persamaan (2.2).

$$x_1 = (x_{11}, x_{12}, \dots, x_{1n})$$

$$x_2 = (x_{21}, x_{22}, \dots, x_{2n})$$

$$d(x_1, x_2) = \sqrt{(x_{21} - x_{11})^2 + (x_{22} - x_{12})^2 + \dots + (x_{2n} - x_{1n})^2} \quad (2.2)$$

Keterangan:

x_1 dan x_2 : data ke-1 dan ke-2

$d(x_1, x_2)$: *Euclidian distance*

n : jumlah fitur

Perhitungan jarak atribut nominal tidak dapat digunakan metode perhitungan jarak *Euclidian distance*, perhitungan jarak nominal dapat menggunakan metode *similarity matrix*, di mana metode *similarity matrix* akan menghitung jarak atribut nominal dengan nilai 0 sampai 1 (Suyanto, 2017). Dapat dilihat pada Persamaan (2.3).

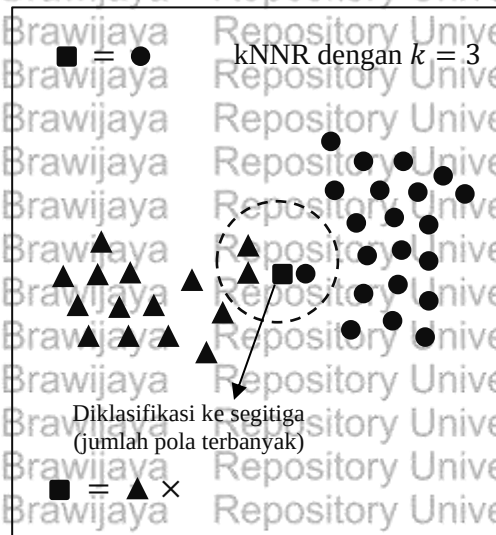
$$similarity = \frac{p-m}{p} \quad (2.3)$$

Keterangan:

p : jumlah atribut nominal

m : jumlah atribut yang sama

Setelah menghitung jarak sesuai dengan k yang ditentukan maka hitung jumlah tetangga terdekat dengan kelas terbanyak. Kelas yang merupakan tetangga terbanyak akan mengklasifikasikan data x menjadi kelas tersebut. Berikut merupakan ilustrasi dari algoritme *K-Nearest Neighbor* yang dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Algoritme *K-Nearest Neighbor*

Sumber: (Suyanto, 2017)

2.6 Local Mean K-Nearest Neighbor (LMKNN)

Local Mean K-Nearest Neighbor (LMKNN) merupakan metode modifikasi klasifikasi dari *K-Nearest Neighbor* (KNN). *Local Mean K-Nearest Neighbor* (LMKNN) dapat menurunkan atau mereduksi efek dari *outlier* yang ada karena teknik pemilihan tetangga terdekat mengambil data dari setiap kelas (Suyanto, 2017). Perubahan pada metode *Local Mean K-Nearest Neighbor* (LMKNN) ada pada parameter k , di mana parameter k pada *Local Mean K-Nearest Neighbor* (LMKNN) merupakan k jumlah tetangga terdekat pada setiap kelas sedangkan nilai k pada *K-Nearest Neighbor* (KNN) adalah jumlah tetangga terdekat untuk seluruh kelas (Syaliman, et al., 2017). Ilustrasi algoritme LMKNN dapat dilihat pada Gambar 2.3. Berikut merupakan langkah – langkah metode *Local Mean K-Nearest Neighbor* (LMKNN):

- (1) Menghitung jarak terdekat data uji ke setiap data latih sejumlah k dari setiap kelas menggunakan metode pengukuran jarak *Euclidean* pada Persamaan (2.2) atau juga bisa dengan menggunakan metode pengukuran jarak lainnya



(2) Menghitung *local mean vector* menggunakan Persamaan (2.4) dari setiap kelas terdekat

$$lm_i = \frac{1}{k} \sum_{j=1}^k d_j(.) \quad (2.4)$$

Keterangan:

lm_i : *local mean vector*

k : Jumlah tetangga dari setiap kelas

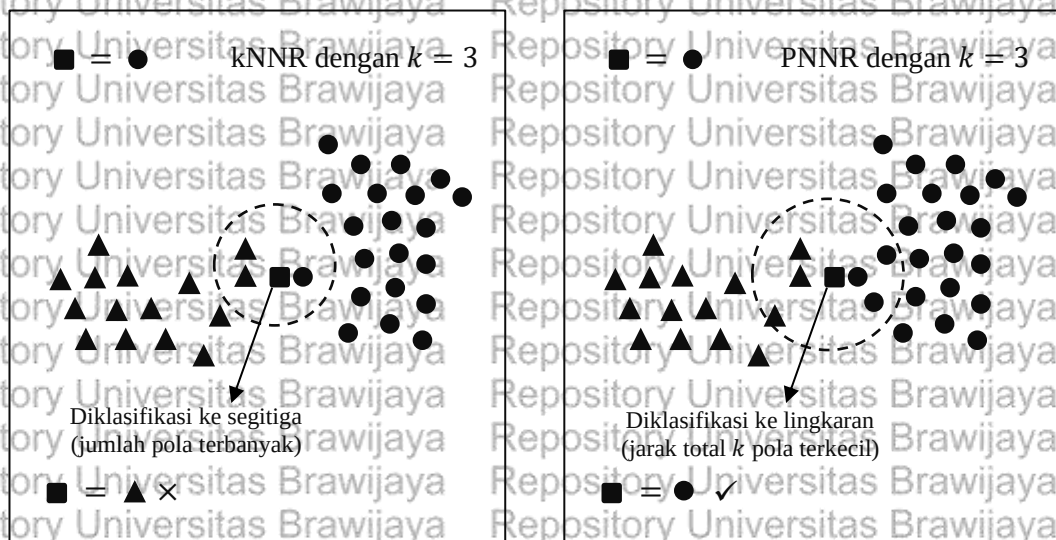
$d_j(.)$: *Euclidian distance* pada persamaan (2.2)

(3) Mencari nilai terkecil dari jarak rata-rata m_i menggunakan Persamaan (2.5) dan mengklasifikasikan x menjadi kelas i .

$$c_i = \argmin (lm_i) \quad (2.5)$$

Keterangan:

c_i : *centroid* ke- i



Gambar 2.3 Ilustrasi Algoritme KNN dengan LMKNN (k sama dengan PNNR)

Sumber: (Suyanto, 2017)

2.7 Nearest Centroid Neighbor (NCN)

Metode *Nearest Centroid Neighbor* (NCN) merupakan metode yang digagas dan diteliti oleh (Chaudhuri, 1996). Pada penelitiannya ia mencetus sebuah metode berdasarkan aturan ketetanggaan dari metode *K-Nearest Neighbor* (KNN), dengan motivasi memperbaiki sensitivitas parameter k dari metode *K-Nearest Neighbor* (KNN). Sama dengan *K-Nearest Neighbor* (KNN),



Nearest Centroid Neighbor (NCN) juga mencari jarak terdekat k tetangganya dengan data uji (*centroid*).

Berbeda dengan *K-Nearest Neighbor* (KNN), *Nearest Centroid Neighbor* (NCN) membuktikan data uji juga terletak simetris diantara data latih. Sehingga *Nearest Centroid Neighbor* (NCN) peduli terhadap dua kriteria, yaitu jarak dan tata letak data. Ilustrasi metode NCN dapat dilihat pada Gambar 2.4. Berikut ini merupakan langkah – langkah metode *Nearest Centroid Neighbor* (NCN) (Chaudhuri, 1996):

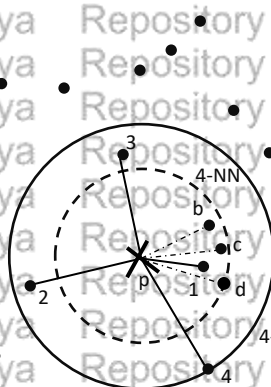
- (1) Mencari tetangga terdekat pertama pada data uji dengan data latih, menggunakan metode perhitungan jarak *Euclidian* pada Persamaan (2.2).
- (2) Mencari *centroid* terdekat pada data uji dari tetangga pertama p_1 dengan tetangga ke – k terdekat p_k . Perhitungan *centroid* dapat dilakukan dengan persamaan berikut:

$$c_k = \frac{1}{k} \sum_{j=1}^k p_j \quad (2.6)$$

Keterangan:

c_k : *centroid* ke – i

p_j : set point *centroid* terdekat j



Gambar 2.4 Ilustrasi Algoritme *Nearest Centroid Neighbor*

Sumber: (Lozano & Sanchez, 2014)



2.8 K-Nearest Centroid Neighbor (KNCN)

K-Nearest Centroid Neighbor (KNCN) juga merupakan metode modifikasi klasifikasi dari *K-Nearest Neighbor* (KNN). *K-Nearest Centroid Neighbor* (KNCN) berasal dari ide metode *Nearest Centroid Neighbor* (NCN) yang menghitung tidak hanya kedekatan dari sebuah objek tetapi juga menghitung kriteria simetrik (*symmetric criterion*). Di mana kriteria simetrik menghitung kedekatan kemiripan data di dalam *centroid* harus semirip mungkin diikuti dengan jaraknya (Li, et al., 2017).

Metode *K-Nearest Centroid Neighbor* (KNCN) mengadaptasi ide dari metode *Nearest Centroid Neighbor* (NCN) pada Persamaan (2.5) untuk melakukan perhitungan *centroid*. Berikut ini merupakan tahapan dari metode *K-Nearest Centroid Neighbor* (KNCN) (Li, et al., 2017):

- (1) Mencari set poin *Nearest Centroid Neighbor* sejumlah *k* yang sudah dibahas pada Bab 2.7.
- (2) Menghitung jumlah *centroid* terdekat terbanyak, mengklasifikasikan data uji menjadi kelas dengan *centroid* terdekat terbanyak.

2.9 Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN)

Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance (LMKHNCN) merupakan metode kombinasi antara *Local Mean Nearest Neighbor* (LMKNN) dan *K-Nearest Centroid Neighbor* (KNCN). Di mana *Local Mean Nearest Neighbor* (LMKNN) untuk mengatasi kekurangan *K-Nearest Neighbor* (KNN) pada bagian *outlier* dan *K-Nearest Centroid Neighbor* (KNCN) mempertimbangkan dua faktor yaitu kemiripan dan jarak yang telah dibahas pada Bab 2.8 (Mehta, et al., 2018). Dengan menggunakan metode pengukuran jarak menggunakan *harmonic distance* karena lebih konsisten pada vektor rata – rata *centroid* lokal yang membuat nilai *k* tidak terlalu sensitif dibandingkan dengan *K-Nearest Neighbor* (KNN) konvensional. Berikut merupakan persamaan metode pengukuran jarak *harmonic distance* yang dapat dilihat pada Persamaan (2.7).

$$HMDS_i = \frac{k}{\sum_{j=1}^k \frac{1}{d(x, c_k^{NCN})}} \quad (2.7)$$

Keterangan:

$HMDS_i$: *harmonic mean distance* ke – *i*

k : jumlah tetangga terdekat dari setiap kelas

$d(x, c_k^{NCN})$: *euclidian distance* seperti pada persamaan (2.2)



Tahapan dari metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) adalah sebagai berikut:

- (1) Mencari set poin dari *K-Nearest Centroid Neighbor* (KNCN) pada Bab 2.8
- (2) Menghitung *local mean vector* dari set poin (KNCN) dari setiap kelas.
- (3) Menghitung jarak *centroid* lokal dari setiap kelas dengan menggunakan *harmonic mean distance* pada Persamaan (2.7).
- (4) Mengklasifikasikan data uji x sebagai kelas c di mana kelas c yang memiliki nilai *harmonic mean distance* yang paling minimum (Mehta, et al., 2018).

2.10 Confussion Matrix

Melakukan evaluasi terhadap klasifikasi yang dibuat dapat menggunakan *confussion matrix*. *Confussion matrix* merupakan metode evaluasi untuk menguji sebuah *classifier* atau mengevaluasi model klasifikasi (Putri, et al., 2015). Terdapat empat istilah penting untuk memahami *confussion matrix*, berikut merupakan gambaran dari istilah – istilah penting tersebut yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Confussion Matrix (Putri, et al., 2015)

	Kelas 1 (Prediksi)	Kelas 2 (Prediksi)	Kelas 3 (Prediksi)
Kelas 1 (Aktual)	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$
Kelas 2 (Aktual)	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$
Kelas 3 (Aktual)	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$

- TP atau *True Positive* adalah jumlah hasil klasifikasi benar oleh model klasifikasi dan kelas aktual yang berlabel positif pada Persamaan (2.8).

$$TP = C_{1,1} + C_{2,2} + C_{3,3} \quad (2.8)$$

- TN atau *True Negative* adalah jumlah hasil klasifikasi benar oleh model klasifikasi dan kelas aktual yang berlabel negatif pada Persamaan (2.9).

$$TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) \quad (2.9)$$

- FP atau *False Positive* adalah jumlah hasil klasifikasi salah oleh model klasifikasi yang seharusnya berlabel negatif di kelas aktual pada Persamaan (2.10).

$$FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) \quad (2.10)$$



- FN atau *False Negative* adalah jumlah hasil klasifikasi salah oleh model klasifikasi yang seharusnya berlabel positif di kelas aktual pada Persamaan (2.11).

$$FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) \quad (2.11)$$

Untuk menghitung nilai yang digunakan untuk evaluasi dari model klasifikasi dari *confusion matrix* dari Tabel 2.1 adalah untuk kelas yang berjumlah 2 sedangkan untuk kelas yang berjumlah lebih dari 2 kelas memiliki konsep yang sama (Manlíguez, 2016). Selain itu, untuk menghitung akurasi, presisi, *recall* dan *specificity* pada kelas yang memiliki lebih dari dua kelas dapat dilihat pada Tabel 2.2.

Tabel 2.2 Confussion Matrix Multiple Classes (Manlíguez, 2016)

		Predicted Class			
		Class 1	Class 2	...	Class n
Actual Class	Class 1	x_{11}	x_{12}	...	x_{1n}
	Class 2	x_{21}	x_{22}	...	x_{2n}
	...	...	...	...	...
	Class n	x_{n1}	x_{n2}	...	x_{nn}

Dalam menghitung empat istilah penting yang disebutkan pada Tabel 2.1 dapat menggunakan total jumlah dari masing – masing istilah tersebut. Berikut merupakan persamaan dalam menghitung keempat istilah tersebut yang dapat dilihat pada Persamaan (2.12) sampai (2.15):

$$Total FN_i = \sum_{j=1, j \neq i}^n x_{ij} \quad (2.12)$$

$$Total FP_i = \sum_{j=1, j \neq i}^n x_{ji} \quad (2.13)$$

$$Total TN_i = \sum_{j=1, j \neq i}^n \sum_{k=1, k \neq i}^n x_{jk} \quad (2.14)$$

$$Total TP_{all} = \sum_{j=1, j \neq i}^n x_{jj} \quad (2.15)$$

Menghitung akurasi, *F1-Score*, presisi (P), *recall* (R) dan *specificity* (S) dalam *Confussion Matrix* dari setiap kelas i , masing – masing dapat dilihat pada Persamaan (2.19), (2.20), (2.16), (2.17), dan (2.18).

$$P_i = \frac{Total TP_i}{Total TP_i + Total FP_i} \quad (2.16)$$

$$R_i = \frac{Total TP_i}{Total TP_i + Total FN_i} \quad (2.17)$$

$$S_i = \frac{Total TP_i}{Total TN_i + Total FP_i} \quad (2.18)$$



$$Akurasi = \frac{Total\ TP_{all}}{n} \quad (2.19)$$

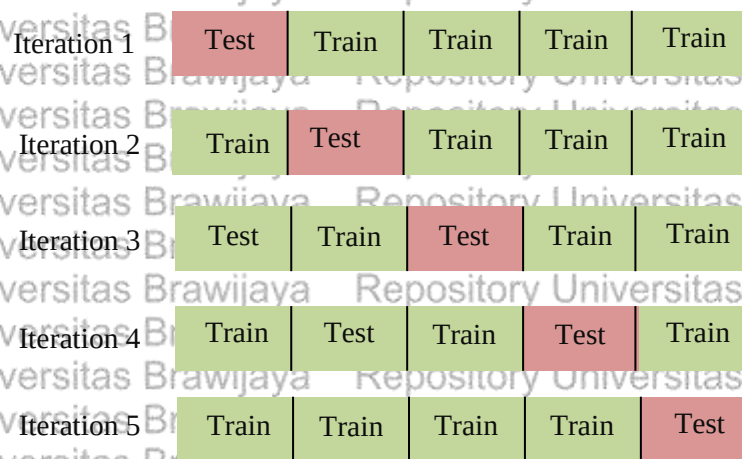
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.20)$$

Keterangan:

n : jumlah data uji

2.11 K-Fold Cross Validation

K-Fold Cross Validation merupakan salah satu metode evaluasi validasi data yang dibagi menjadi beberapa partisi (dua atau lebih) himpunan data D menjadi k fold (subhimpunan) yang saling bebas: $f_1, f_2, \dots, f_k$, sehingga masing-masing dari $fold$ berisi $\frac{1}{k}$ bagian data (Suyanto, 2017). Gambar 2.5 merupakan ilustrasi dari metode evaluasi *K-Fold Cross Validation*.



Gambar 2.5 Ilustrasi K-Fold Cross Validation (K = 5)

Sumber: towardsdatascience.com (2018)



BAB 3 METODOLOGI

3.1 Tipe Penelitian

Pada penelitian penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) klasifikasi kinerja pegawai negeri sipil kota Malang merupakan penelitian yang bersifat non-implementatif. Penelitian ini bersifat non-implementatif karena berfokus pada penyelesaian masalah suatu objek yang kemudian akan menghasilkan tinjauan ilmiah. Penelitian ini menggunakan tipe analitik karena membahas tentang hubungan antara objek yang diteliti dengan metode yang digunakan, yang nantinya akan dipilih menjadi dasar sebuah keputusan ataupun penelitian yang lebih lanjut.

3.2 Strategi Penelitian

Strategi penelitian yang dilakukan untuk menerapkan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) pada klasifikasi kinerja pegawai negeri sipil kota Malang dengan pengambilan data pada sistem informasi pegawai yang dibahas pada sub bab bagian teknik pengumpulan data Bab 3.3. Selanjutnya data akan dibagi dua yaitu data latih dan data uji yang dibagi bergantian sebanyak K pada *K-Fold Cross Validation*.

Setelah melakukan pembagian data yakni data latih dan data uji, akan dilanjutkan tahapan *preprocessing* pada data yang memiliki *missing value* dan *duplicate* setelah itu melakukan data akan dilakukan normalisasi dan tanpa normalisasi untuk pengujian akurasi dan *F1-Score* dengan pengujian beberapa parameter k pada LMKHNCN juga perbandingan dengan metode aslinya yakni KNN dengan menggunakan *library sklearn*. Diagram perancangan metode LMKHNCN dapat dilihat pada sub bab bagian perancangan algoritme Bab 3.4.

3.3 Teknik Pengumpulan Data

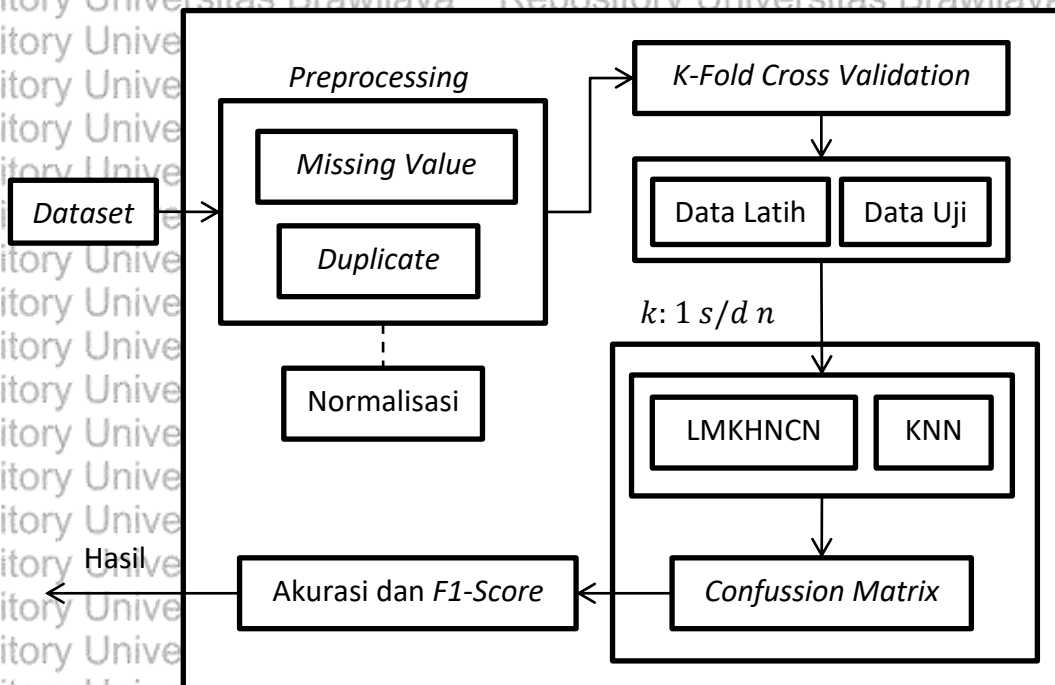
Teknik pengumpulan data pada penelitian penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) untuk klasifikasi kinerja pegawai negeri sipil Kota Malang dengan menggunakan data sekunder di dalam sistem informasi pegawai Kota Malang. Pengambilan data dilakukan dengan cara *Web Scrapping* menggunakan *Library Selenium* disetiap data pegawai pada sistem informasi pegawai Kota Malang dengan pengambilan fitur menggunakan penelitian sebelumnya juga pendapat ahli dari Badan Kepegawaian Daerah Kota Malang pak Bagus Winarno S. Kom sebanyak 6584 data pegawai. Berikut merupakan fitur (*input*) dan kelas (*ouput*) beserta tipe datanya yang dapat dilihat pada Tabel 3.1:

Tabel 3.1 Fitur Dataset

Fitur	Tipe Data
Masa Kerja	Numerik
Gaji	Numerik
Jumlah Kenaikan Pangkat	Numerik
Jumlah Penghargaan	Numerik
Orientasi	Numerik
Integritas	Numerik
Disiplin	Numerik
Kerjasama	Numerik
Kawin	Kategorikal / Nominal
Kinerja (Kelas)	Kategorikal / Nominal

3.4 Perancangan Algoritme

Berikut ini merupakan diagram tahapan – tahapan dari strategi penelitian yang dilakukan dalam klasifikasi pegawai negeri sipil Kota Malang yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Perancangan Algoritme LMKHNCN



3.5 Lokasi Penelitian

Penelitian ini akan dilakukan di Laboratorium Sistem Cerdas Fakultas Ilmu Komputer Universitas Brawijaya dengan estimasi lama penelitian sekitar empat bulan.

3.6 Peralatan Pendukung

Peralatan yang akan digunakan dalam penelitian penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) untuk klasifikasi kinerja pegawai negeri sipil kota Malang meliputi *software* dan *hardware* secara berturut dapat dilihat pada Tabel 3.2 dan Tabel 3.3

Tabel 3.2 Perangkat pendukung berupa *software*

<i>Software</i>	Spesifikasi
Sistem Operasi (<i>Operating System/OS</i>)	Windows 10 Pro 64-bit
Bahasa Pemrograman	Python 3.7
<i>Text Editor</i>	Visual Studio Code

Tabel 3.3 Perangkat pendukung berupa *hardware*

<i>Hardware</i>	Spesifikasi
Laptop	Tipe : MSI GL62 7QF Processor : Intel Core i7-7700HQ RAM : 12 Gigabyte Internal Storage : HDD 1 Terabyte GPU : NVIDIA GeForce GTX960M

3.7 Teknik Pengujian

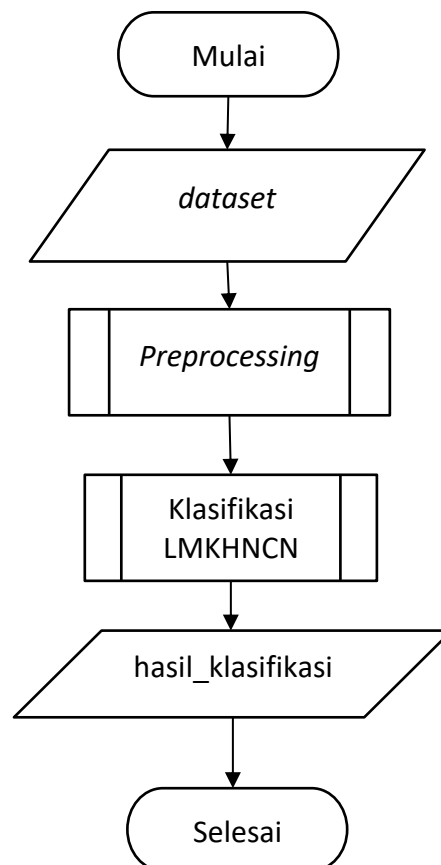
Pengujian akurasi dilakukan dengan menggunakan *Confussion Matrix* dan *10-Fold Cross Validation* agar keseluruhan performa model klasifikasi dapat diukur, di mana metode *10-Fold Cross Validation* akan mengevaluasi akurasi dari keseluruhan bagian data. Menguji nilai akurasi dan beberapa percobaan nilai parameter k pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN).

BAB 4 PERANCANGAN

Bab ini akan menjelaskan atau membahas terkait dengan tahapan maupun alur dari penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Mean Using Harmonic Mean Distance* (LMKHNCN) untuk klasifikasi kinerja pegawai negeri sipil Kota Malang dalam bentuk diagram alir dan juga menampilkan manualisasi dari penerapan metode ini dengan pengambilan beberapa sampel data.

4.1 Perancangan Algoritme

Perancangan algoritme ini berfungsi untuk menampilkan dan membahas alur maupun tahapan – tahapan proses dalam menerapkan metode *Nearest Centroid Neighbor Classifier Based on K Local Mean Using Harmonic Mean Distance* (LMKHNCN). Tahapan – tahapan proses metode ini akan direpresentasikan melalui diagram alir (*flow chart*) agar proses pengerjaan dapat dilakukan secara sistematis. Berikut merupakan diagram alir dari perancangan metode *Nearest Centroid Neighbor Classifier Based on K Local Mean Using Harmonic Mean Distance* (LMKHNCN) yang dapat dilihat pada Gambar 4.1.



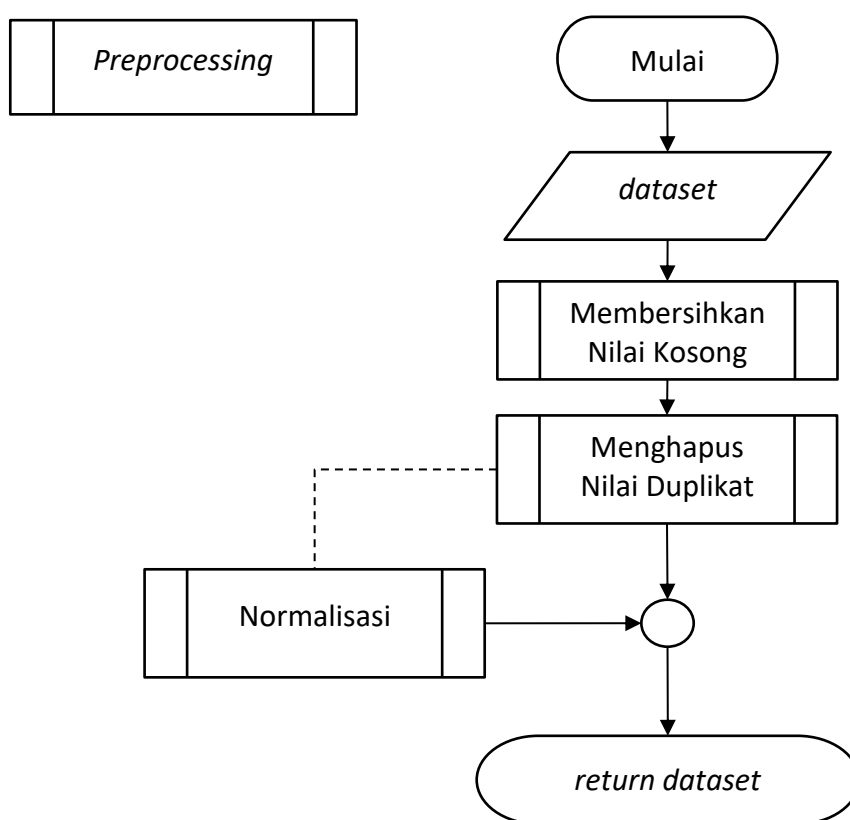
Gambar 4.1 Diagram Alir Perancangan Algoritme



4.1.1 Pre-processing

Preprocessing merupakan salah satu tahapan yang tidak kalah penting dengan tahapan – tahapan lainnya. Sebelum melakukan penambangan data (*data mining*) ada baiknya melakukan *preprocessing* dahulu, pembersihan (*missing value*, derau, pencilan, inkonsisten data dan lain – lain) data dan untuk meningkatkan kualitas data sehingga menghasilkan penambangan yang baik (Suyanto, 2017).

Setelah membersihkan data, tahapan selanjutnya adalah menghapus baris nilai yang sama (duplikat). Baris duplikat pada *dataset* dapat memengaruhi hasil klasifikasi menjadi *imbalance*. Diagram alir (*flow chart*) dari tahapan *preprocessing* yang dapat dilihat pada Gambar 4.2.

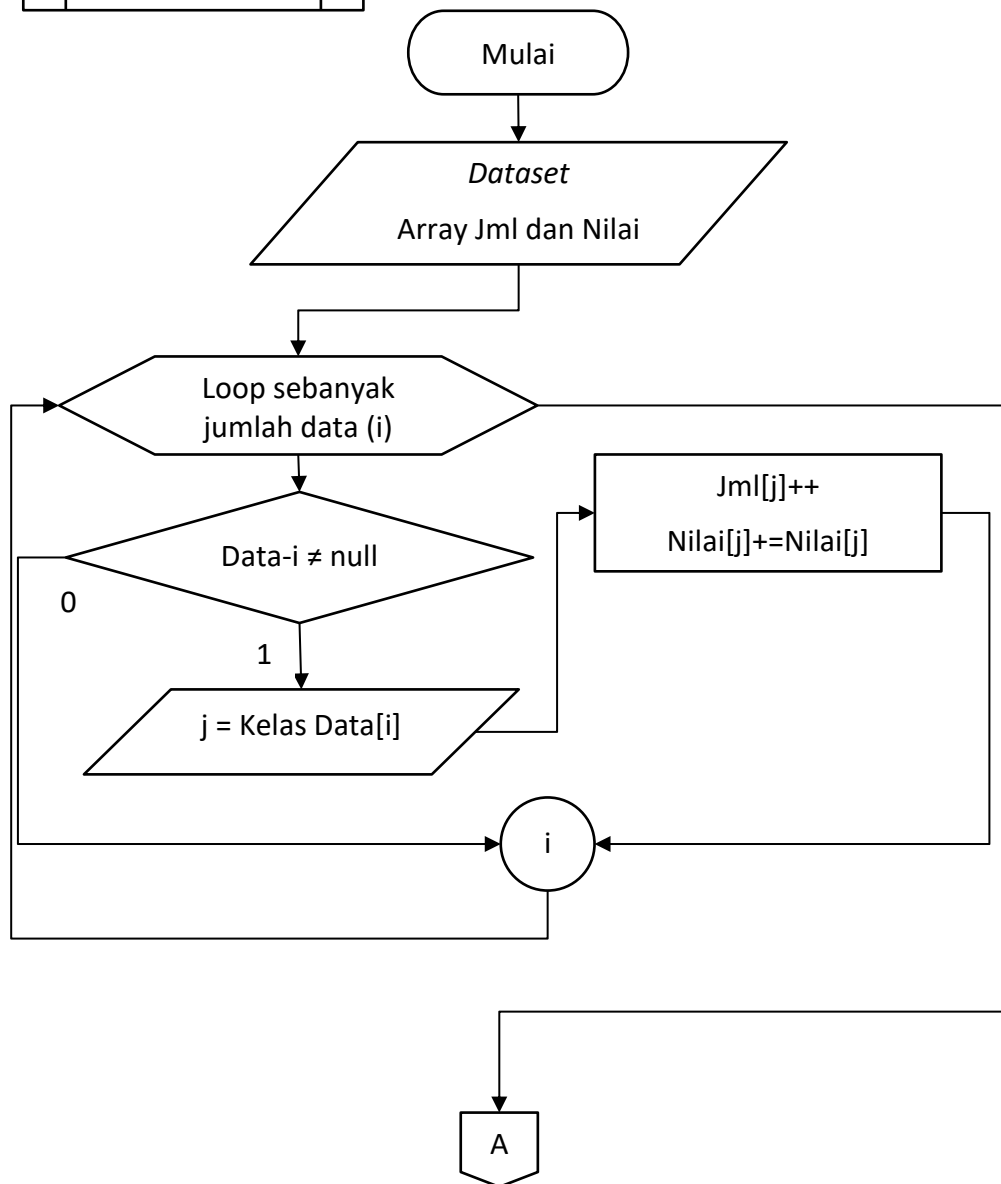


Gambar 4.2 Diagram Alir *Preprocessing*

4.1.1.1 Membersihkan Nilai Kosong (*Missing Value*)

Membersihkan nilai kosong dapat dilakukan dengan cara mengisi nilai kosong tersebut dengan rata – rata dari setiap kelas yang ada (kasus klasifikasi). Diagram alir (*flow chart*) untuk membersihkan nilai kosong dapat dilihat pada Gambar 4.3.

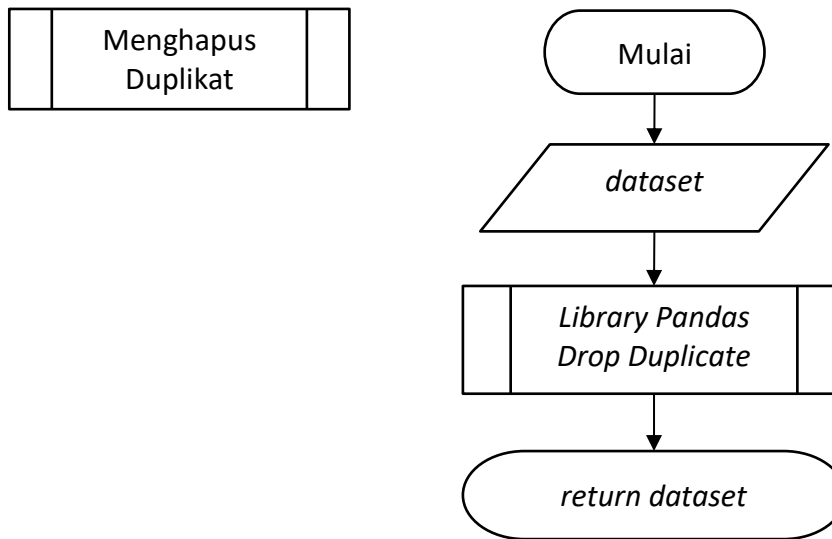
Membersihkan
Nilai Kosong





4.1.1.2 Menghapus Baris Yang Sama (Duplikat)

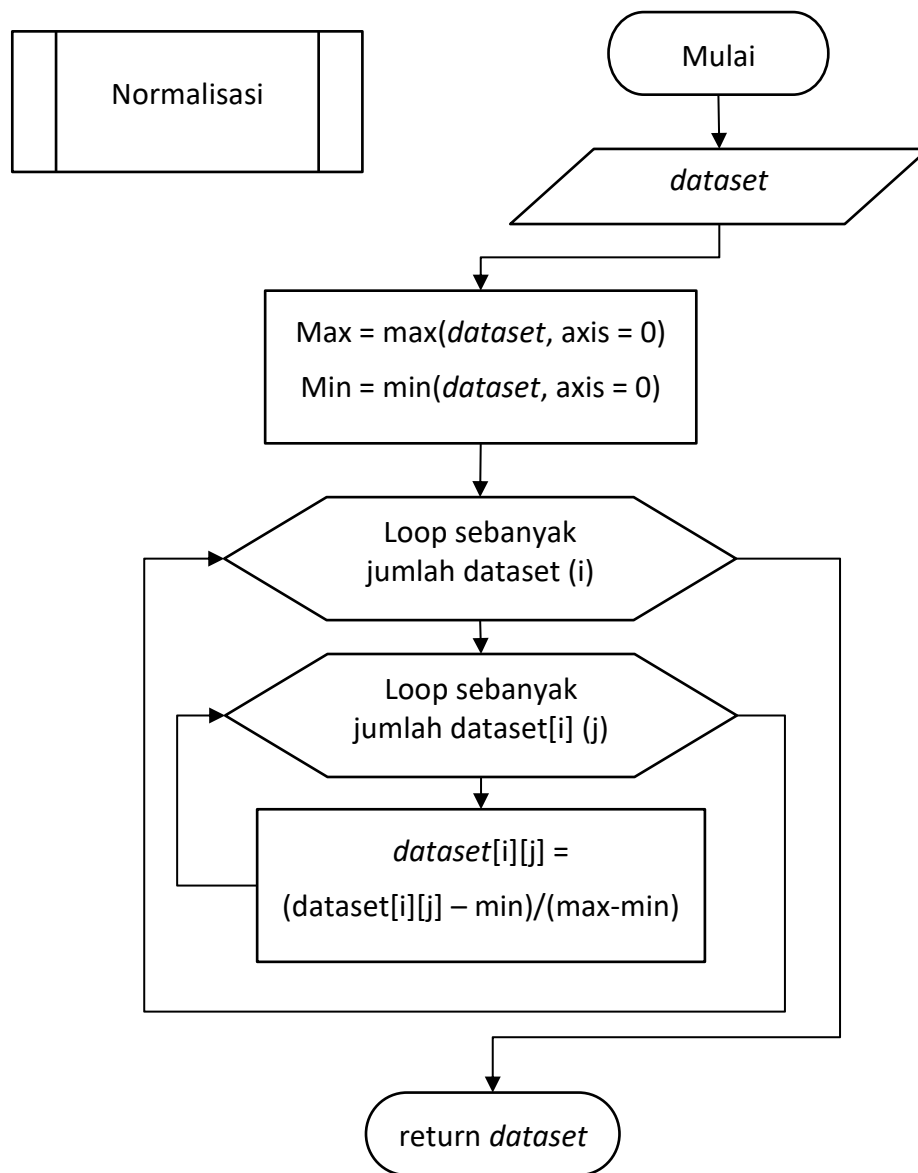
Tahapan *preprocessing* selanjutnya ialah menghapus baris dengan nilai yang sama, baris yang sama akan menyebabkan atau memengaruhi hasil dari model klasifikasi, terlebih lagi jika model klasifikasi berbasis *Nearest Neighbor* di mana jika tetangga terdekat yang terpilih adalah nilai yang duplikat yang akan menyebabkan data yang tidak seimbang (*imbalance*) di atas khususnya pada parameter k yang kecil. Diagram alir (*flow chart*) menghapus nilai duplikat menggunakan *library pandas* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir Menghapus Duplikat

4.1.1.3 Normalisasi

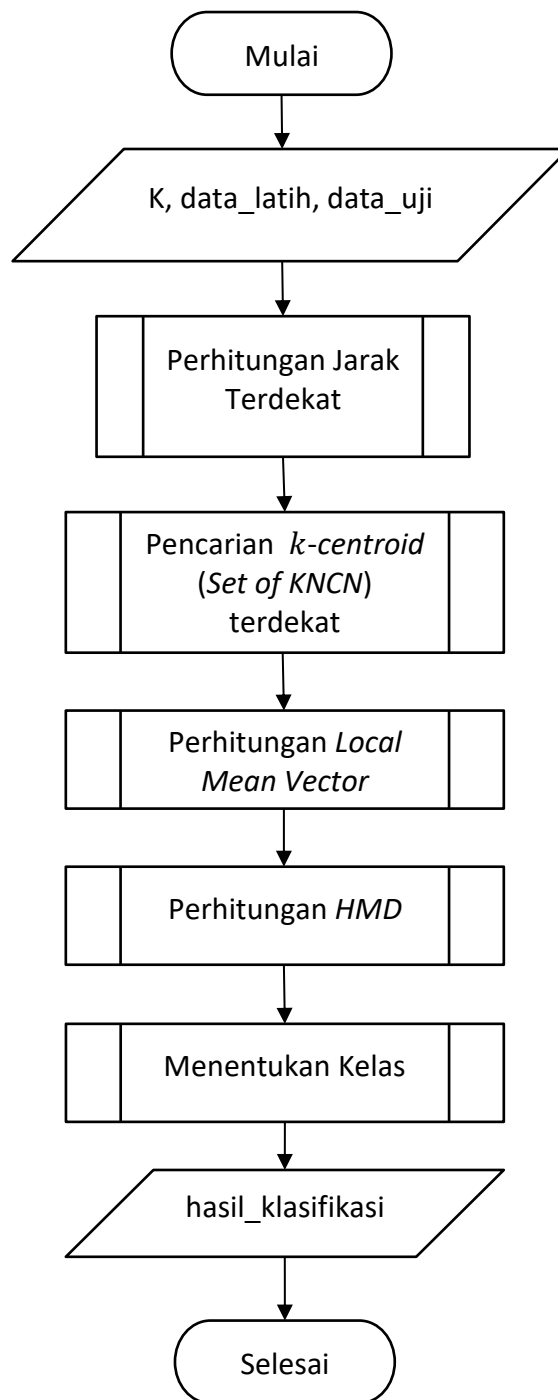
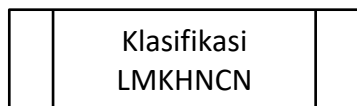
Tahapan normalisasi merupakan tahapan opsional pada pengujian normalisasi dengan menggunakan normalisasi *Min-Max*, untuk pengujian tanpa normalisasi dapat melewati tahapan ini. Tahapan normalisasi berguna untuk menyamakan jarak data dari 0 sampai 1 dari setiap fitur. Diagram alir (*flow chart*) normalisasi menggunakan metode normalisasi *Min-Max* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram Alir Normalisasi

4.1.2 Klasifikasi LMKHNCN

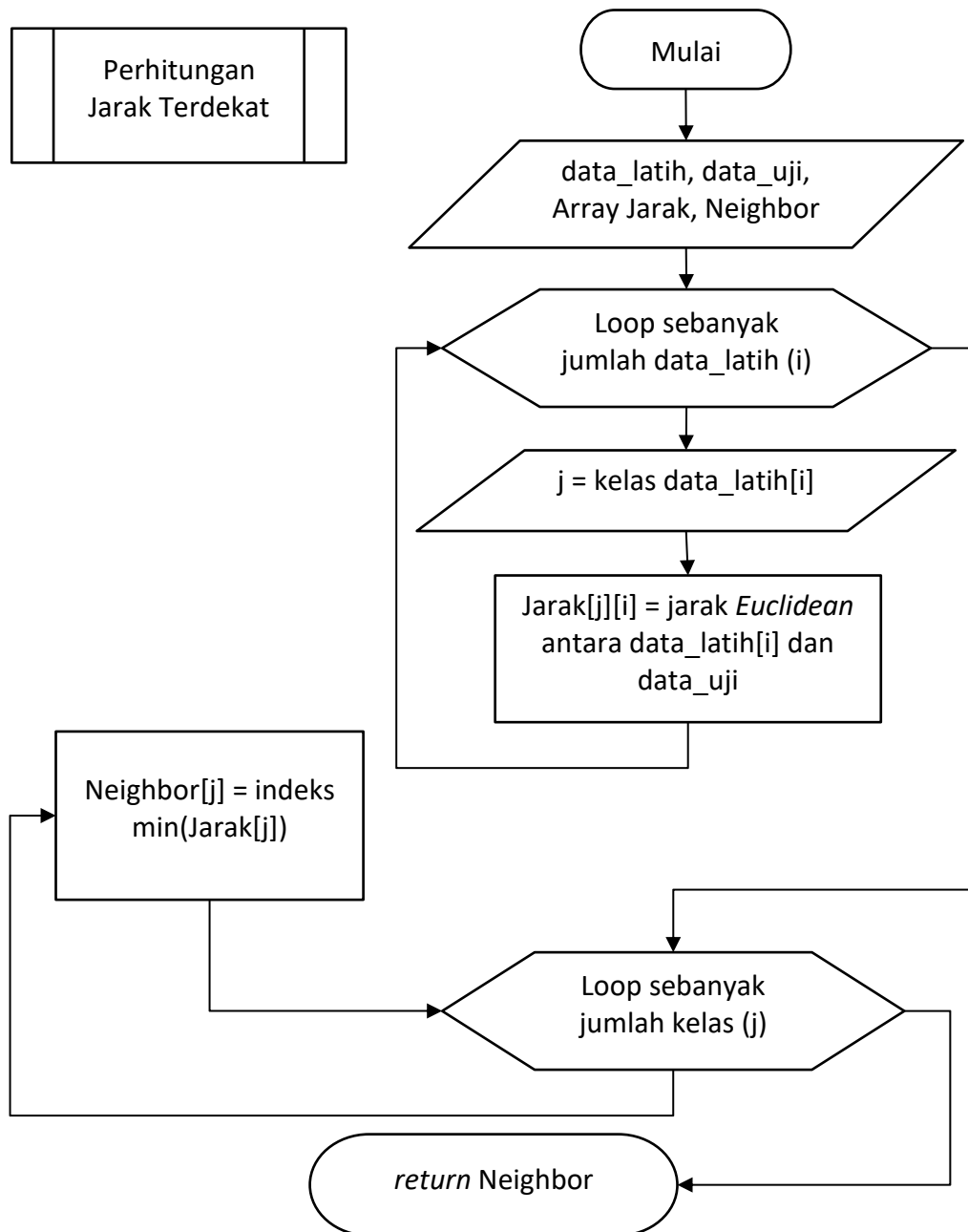
Klasifikasi metode *Nearest Centroid Neighbor Classifier Based on K Local Mean Using Harmonic Mean Distance* (LMKHNCN) akan dimulai dengan melakukan perhitungan jarak data uji terhadap seluruh data latih dengan mengambil jarak yang terdekat. Selanjutnya perhitungan pencarian *k-centroid* (*Set of KNCN*) terdekat diikuti dengan perhitungan *Local Mean Vector* dan *Harmonic Mean Distance* untuk melakukan klasifikasi. Diagram alir (*flow chart*) pada tahapan klasifikasi *Nearest Centroid Neighbor Classifier Based on K Local Mean Using Harmonic Mean Distance* (LMKHNCN) dapat dilihat pada Gambar 4.6.



Gambar 4.6 Diagram Alir Klasifikasi LMKHNCN

4.1.2.1 Perhitungan Jarak Terdekat

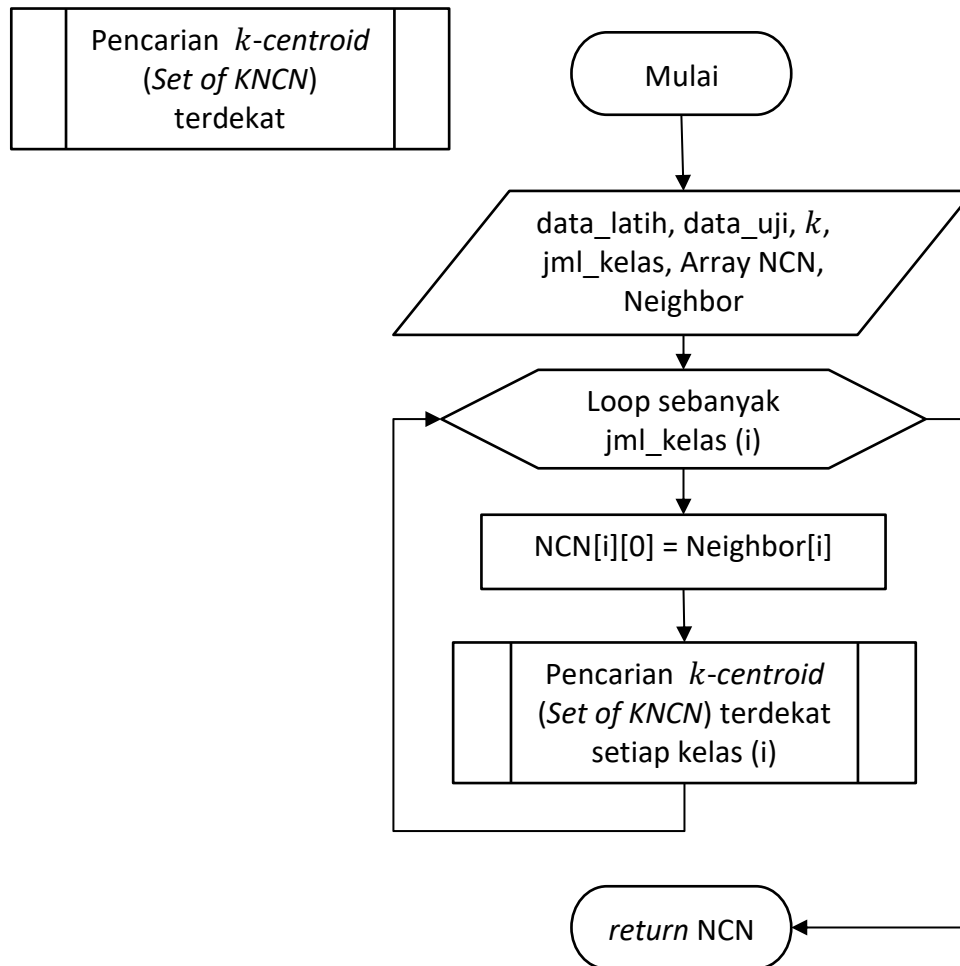
Perhitungan jarak terdekat untuk mengetahui tetangga terdekat pertama menggunakan metode perhitungan jarak *Euclidean Distance* pada Persamaan (2.2) dari setiap kelas agar dapat melakukan perhitungan pencarian *k-centroid* terdekat selanjutnya (*Set of KNCN*). Diagram alir (*flow chart*) perhitungan jarak terdekat dari setiap data latih ke data uji dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Perhitungan Jarak Terdekat

4.1.2.2 Perhitungan *K-Centroid* Terdekat

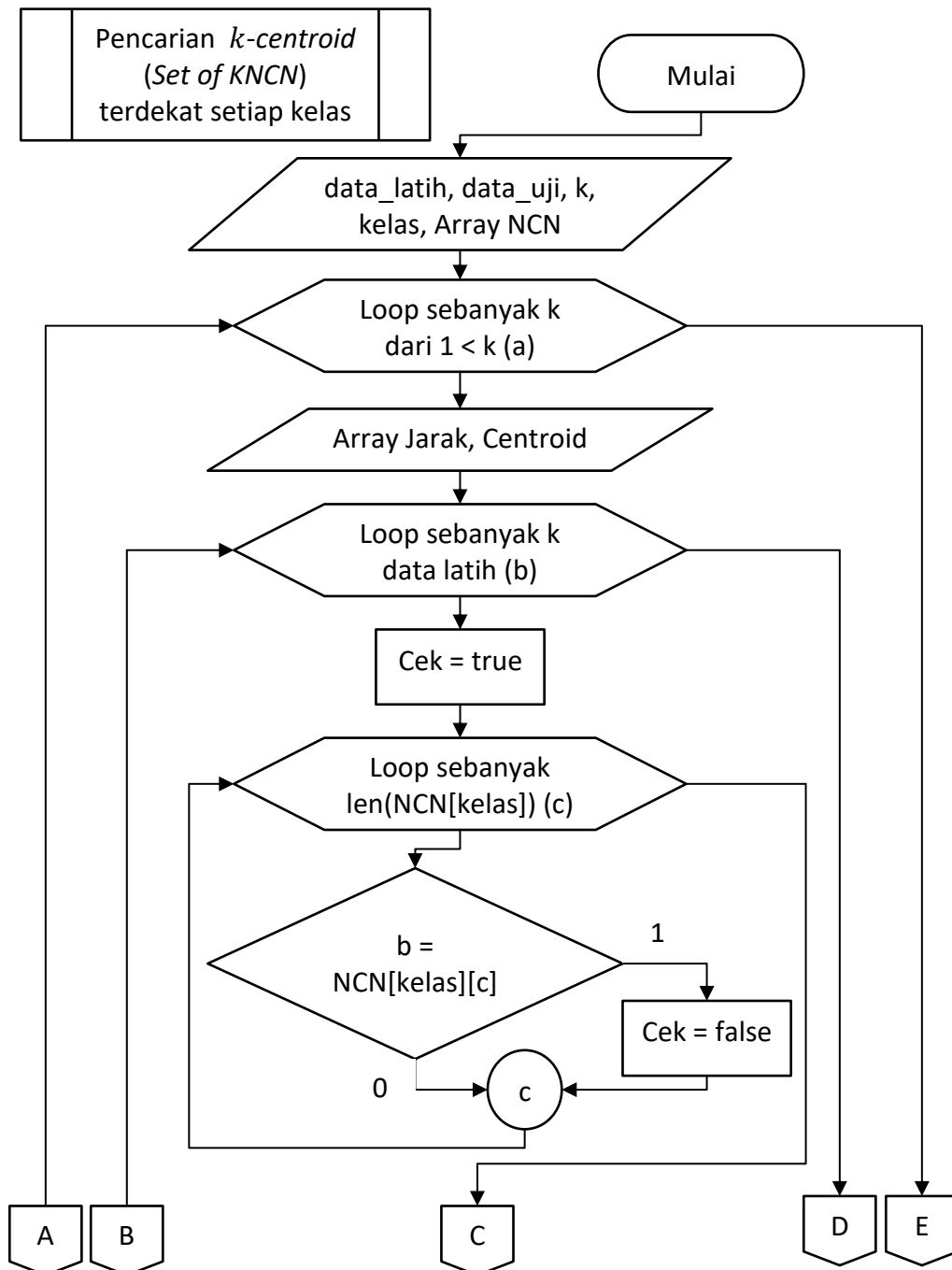
Perhitungan mencari *k-centroid* terdekat membutuhkan tetangga pertama terdekat. Diagram alir (*flow chart*) perhitungan mencari *k-centroid* terdekat dapat dilihat pada Gambar 4.8.

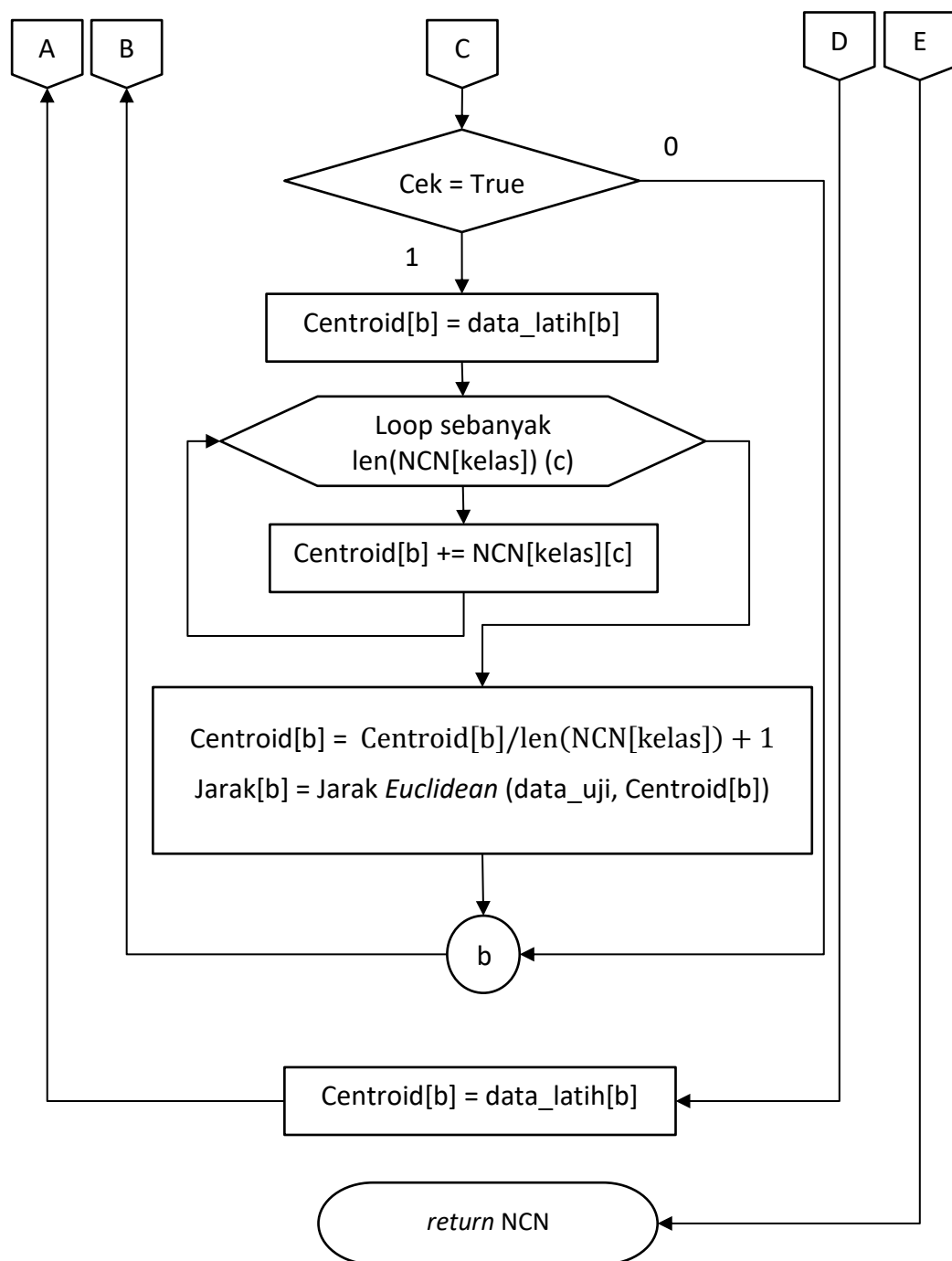


Gambar 4.8 Diagram Alir Perhitungan *K-Centroid* Terdekat

4.1.2.3 Perhitungan *K-Centroid* Terdekat Setiap Kelas

Tahapan ini melakukan pencarian *k-centroid* terdekat pada setiap kelas label dari *dataset*. Diagram alir (*flow chart*) untuk melakukan tahapan perhitungan pencarian *k-centroid* terdekat pada setiap kelas dapat dilihat pada Gambar 4.9.

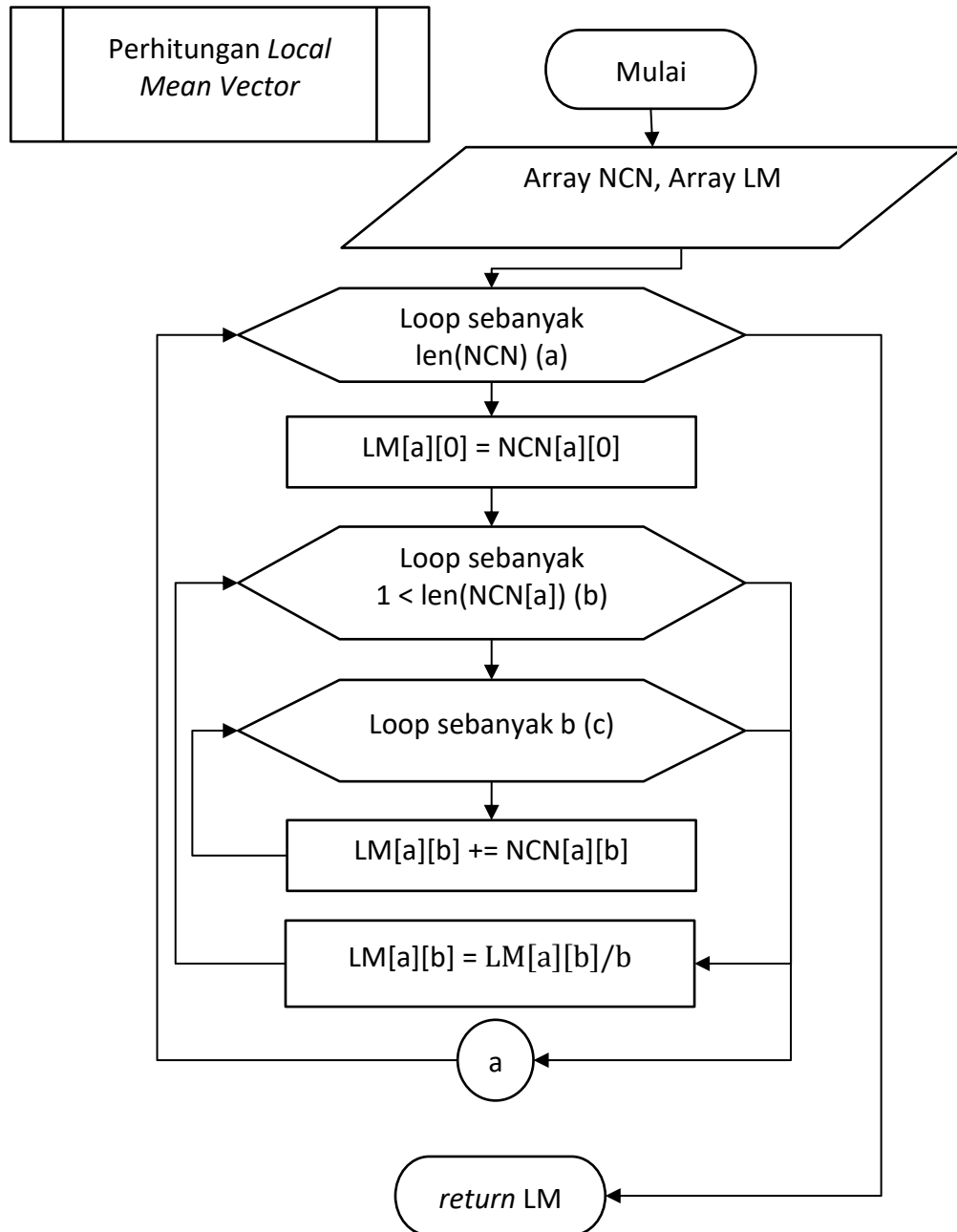




Gambar 4.9 Diagram Alir Perhitungan K-Centroid Terdekat Setiap Kelas

4.1.2.4 Perhitungan *Local Mean Vector*

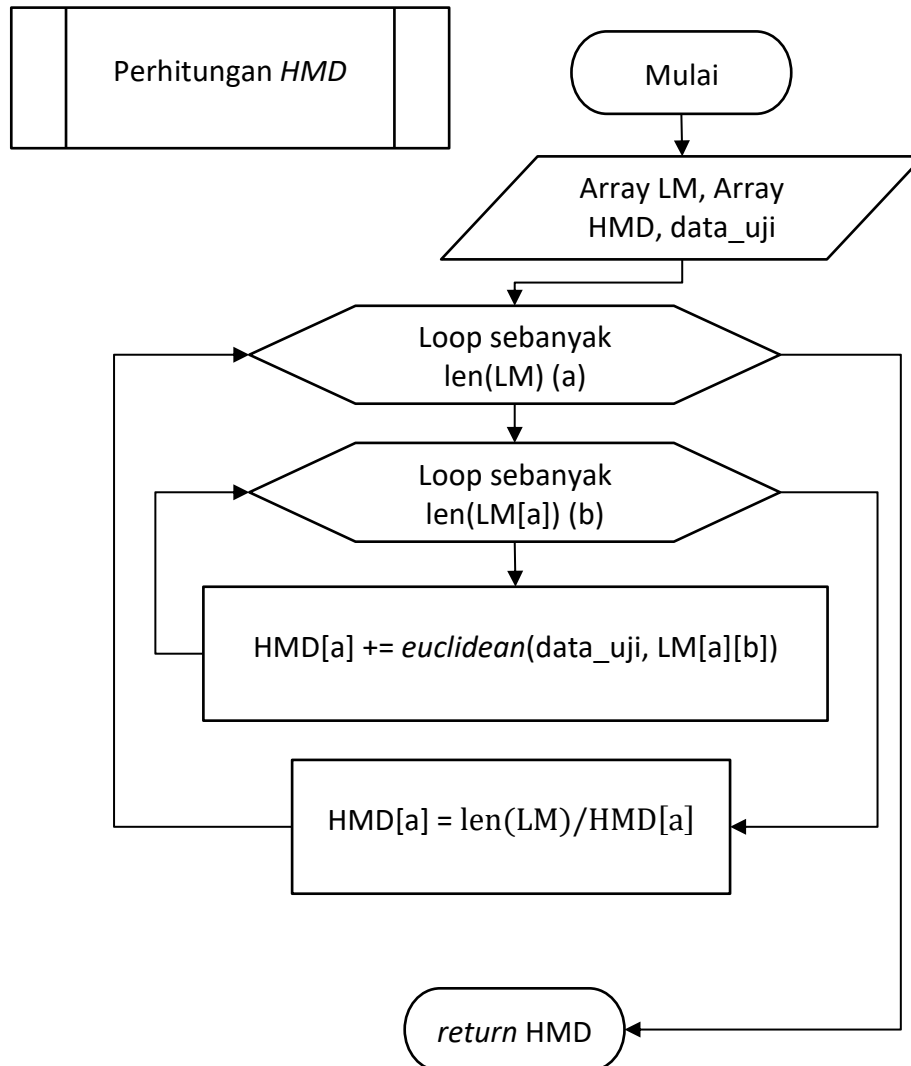
Perhitungan *Local Mean Vector* merupakan tahapan pencarian rata – rata vektor lokal dari set *K-NCN* yang sudah dicari untuk dijadikan pencarian jarak *centroid* terdekat dari data uji. Diagram alir (*flow chart*) dari tahapan perhitungan *Local Mean Vector* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Diagram Alir Perhitungan *Local Mean Vector*

4.1.2.5 Perhitungan *Harmonic Mean Distance*

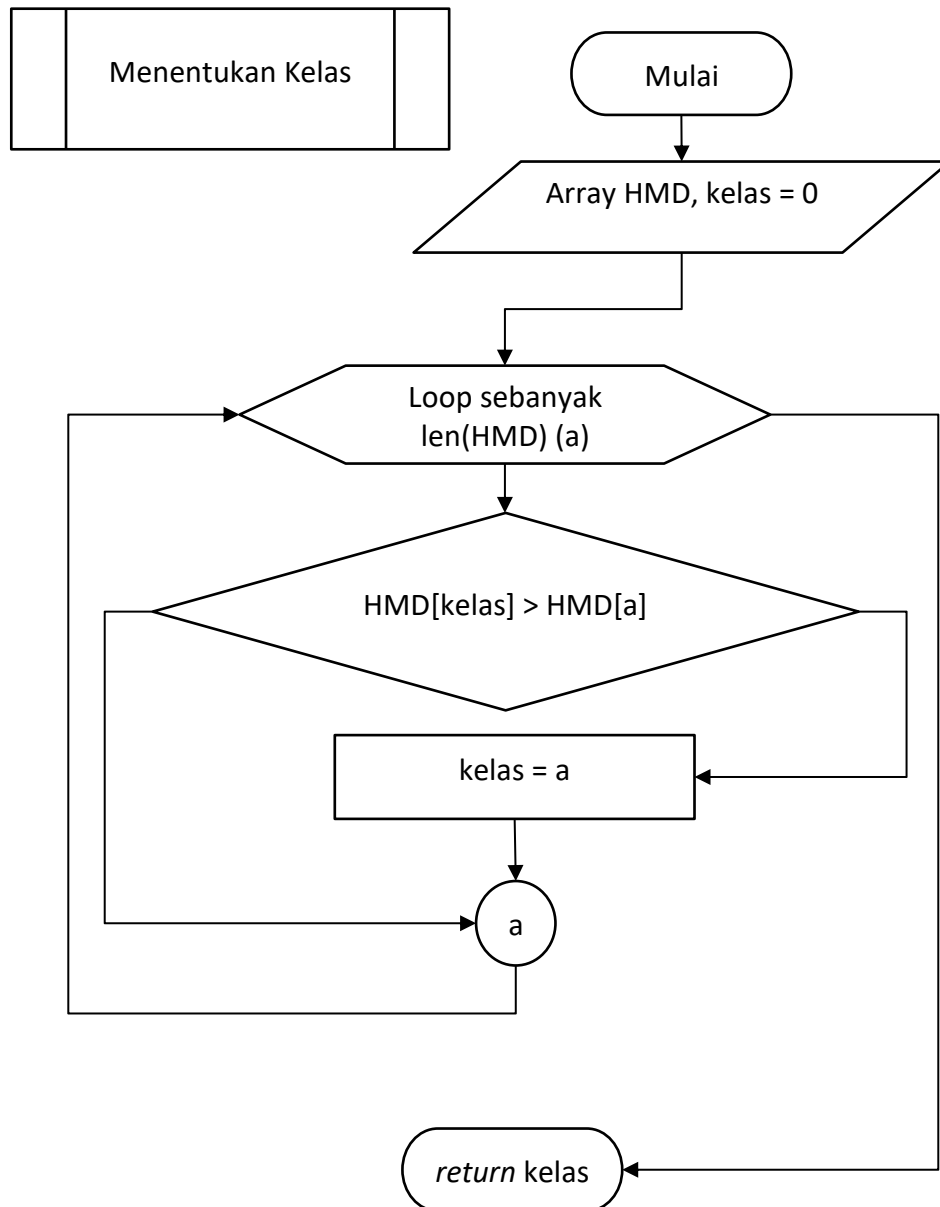
Tahapan perhitungan *Harmonic Mean Distance* adalah menghitung jarak rata – rata harmonik dari *Local Mean Vector* dengan data uji menggunakan metode perhitungan jarak *Euclidean* dari setiap kelas. Diagram alir (*flow chart*) dari tahapan *Harmonic Mean Distance* dapat dilihat pada Gambar 4.11



Gambar 4.11 Diagram Alir Perhitungan *Harmonic Mean Distance*

4.1.2.6 Menentukan Hasil Kelas Klasifikasi

Tahapan ini merupakan tahapan terakhir pada klasifikasi *Nearest Centroid Neighbor Classifier Based on K local Means Using Harmonic Mean Distance* (LMKHNCN), yaitu menentukan hasil kelas menggunakan rata-rata harmonik (*Harmonic Mean Distance*) terkecil dari setiap kelas. Diagram alir (*flow chart*) dari tahapan penentuan hasil klasifikasi dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram Alir Menentukan Hasil Kelas Klasifikasi



4.2 Perhitungan Manual

Tahapan perhitungan manual metode *Nearest Centroid Neighbor Classifier Based on K local Means Using Harmonic Mean Distance* (LMKHNCN) berfungsi untuk memperjelas dan membuktikan bahwa tahapan perancangan algoritme yang dirancang sudah sesuai. Perhitungan manual menggunakan dataset pegawai negeri sipil Kota Malang sejumlah 16 data acak dengan 12 data latih dan 4 data uji dengan 12 data latih dibagi rata menjadi 3 kelas (4 data latih setiap kelas).

4.2.1 Dataset

Seperti yang telah disebutkan pada Bab 4.2 dataset memiliki tiga kelas, ketiga kelas tersebut yakni kelas kinerja dengan kelas 1, 2 dan 3 berturut – turut adalah “Amat Baik (3)”, “Baik (2)”, “Cukup (1)” dan sembilan fitur yang digunakan untuk klasifikasi. Berikut merupakan 16 sampel data acak yang digunakan untuk perhitungan manual yang dapat dilihat pada Tabel 4.1.

Keterangan:

X1 = Masa Kerja, X2 = Gaji, X3 = Jumlah Kenaikan Pangkat,

X4 = Jumlah Penghargaan, X5 = Orientasi, X6 = Integritas, X7 = Disiplin

X8 = Kerja Sama, X9 = Kawin

Tabel 4.1 Data Perhitungan Manual

X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
8	998600	4	0	76	76	72	78	3	1
15	1974202	4	0	72	79	79	72	1	1
14	1841400	5	0	73	73	79	78	1	1
13	1455200	5	0	82	72	80	82	1	1
18	2918647	2	0	87	85	76	78	1	2
10	3116500	4	1	83	82	87	84	1	2
21	2178100	10	1	89	80	82	81	1	2
14	1919300	10	1	84	82	80	84	1	2
9	2623600	2	0	94	96	90	97	1	3
25	2207600	4	2	94	97	85	94	1	3
11	1823814	4	1	87	88	89	97	1	3
11	1823814	4	1	96	85	96	91	1	3
15	1974202	4	0	74	75	74	76	1	1



X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
18	2918647	2	0	88	89	87	78	1	2
27	1642600	9	1	79	79	89	75	1	2
11	1823814	2	3	95	88	93	96	1	3

Kata X pada tabel *dataset* di atas merupakan singkatan dari fitur, seperti X1 merupakan fitur yang pertama dan seterusnya. *Dataset* di atas akan dibagi menjadi dua bagian yaitu data latih dan data uji untuk melakukan klasifikasi. Pembagian data uji dan data latih dapat dilihat pada Tabel 4.2 dan 4.3.

Tabel 4.2 Data Uji

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	15	1974202	4	0	74	75	74	76	1	1
2	18	2918647	2	0	88	89	87	78	1	2
3	27	1642600	9	1	79	79	89	75	1	2
4	11	1823814	2	3	95	88	93	96	1	3

Tabel 4.3 Data Latih

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	8	998600	4	0	76	76	72	78	3	1
2	15	1974202	4	0	72	79	79	72	1	1
3	14	1841400	5	0	73	73	79	78	1	1
4	13	1455200	5	0	82	72	80	82	1	1
5	18	2918647	2	0	87	85	76	78	1	2
6	10	3116500	4	1	83	82	87	84	1	2
7	21	2178100	10	1	89	80	82	81	1	2
8	14	1919300	10	1	84	82	80	84	1	2
9	9	2623600	2	0	94	96	90	97	1	3
10	25	2207600	4	2	94	97	85	94	1	3
11	11	1823814	4	1	87	88	89	97	1	3
12	11	1823814	4	1	96	85	96	91	1	3



4.2.2 Perhitungan Normalisasi

Tahapan ini merupakan tahapan percobaan pada rumusan masalah. Tahapan ini dapat dilewati langsung jika pengujian tidak menggunakan tahapan normalisasi, tetapi untuk tahapan manualisasi normalisasi tetap digunakan agar dapat memperlihatkan alur kerja atau cara kerja normalisasi. Proses normalisasi dapat dilakukan menggunakan metode *Min-Max* pada Persamaan (2.1). Berikut merupakan tabel nilai maksimal dan minimal dari setiap fitur yang dapat dilihat pada Tabel 4.4.

Tabel 4.4 Nilai Maksimum dan Minimum Setiap Fitur

	Min	Max
X1	8	27
X2	Min	Max
	998600	3116500
X3	Min	Max
	2	10
X4	Min	Max
	0	3
X5	Min	Max
	72	96
X6	Min	Max
	72	97
X7	Min	Max
	72	96
X8	Min	Max
	72	97
X9	Min	Max
	1	3

Pada bagian Bab 2.3.1.2 normalisasi data telah dijelaskan bahwa untuk mendapat nilai normalisasi dibutuhkan nilai maksimal dan minimal. Setelah itu, akan dilanjutkan proses perhitungan normalisasinya seperti pada Persamaan (2.1). Setelah melakukan proses normalisasi akan dihasilkan nilai antara 0 sampai satu pada data uji dan data latih yang dapat dilihat pada Tabel 4.5 dan 4.6. Berikut merupakan contoh perhitungan normalisasi pada Tabel 4.2 nomor 1 fitur 1 (X_1).



$$X1' = \frac{X1 - \min}{\max - \min}$$

$$X1' = \frac{15 - 8}{27 - 8}$$

$$X1' = 0.368$$

Tabel 4.5 Data Uji (Normalisasi)

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.368	0.46	0.25	0	0.083	0.12	0.083	0.16	0	1
2	0.526	0.90	0	0	0.66	0.68	0.625	0.24	0	2
3	1	0.30	0.875	0.333	0.291	0.28	0.708	0.12	0	2
4	0.1578	0.38	0	1	0.958	0.64	0.875	0.96	0	3

Tabel 4.6 Data Latih (Normalisasi)

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0	0	0.25	0	0.166	0.16	0	0.24	1	1
2	0.3684	0.460	0.25	0	0	0.28	0.2917	0	0	1
3	0.3157	0.397	0.375	0	0.041	0.04	0.2917	0.24	0	1
4	0.2631	0.215	0.375	0	0.416	0	0.3333	0.4	0	1
5	0.5263	0.906	0	0	0.625	0.52	0.1667	0.24	0	2
6	0.1052	1	0.25	0.333	0.458	0.4	0.625	0.48	0	2
7	0.6842	0.556	1	0.333	0.708	0.32	0.4167	0.36	0	2
8	0.3157	0.434	1	0.333	0.5	0.4	0.3333	0.48	0	2
9	0.0526	0.767	0	0	0.916	0.96	0.75	1	0	3
10	0.8947	0.570	0.25	0.666	0.916	1	0.5417	0.88	0	3
11	0.1578	0.389	0.25	0.333	0.62	0.64	0.7083	1	0	3
12	0.1578	0.389	0.25	0.333	1	0.52	1	0.76	0	3

4.2.3 Mencari K-Centroid Terdekat (Set of KNCN)

Tahapan ini merupakan tahapan metode pencarian *centroid* sejumlah *k* terdekat dari data uji dengan data latih untuk manualisasi ini *k* bernilai 2. Tahapan ini dimulai dengan pencarian *centroid* terdekat pertama dilanjutkan dengan *centroid* terdekat *k*. Perhitungan pencarian *centroid* terdekat pertama tidak berbeda dengan pencarian tetangga terdekat pertama dari metode KNN

yang telah dibahas pada Bab 2.5 yaitu dengan melakukan perhitungan jarak point data uji dengan seluruh data latih.

Tetangga terdekat pertama digunakan sebagai *centroid* pertama yang di mana hal ini menyebabkan pencarian *centroid* sejumlah k untuk k bernilai 1 metode KNCN sama seperti KNN. Tabel 4.7 sampai 4.10 berturut – turut merupakan perhitungan jarak terdekat dari data uji pertama sampai data uji keempat menggunakan metode perhitungan jarak *Euclidean* pada Persamaan (2.2), jarak terdekat akan ditandai dengan warna abu - abu. Karena metode KNCN pada kasus ini menggunakan *Local Mean Based* yang telah dibahas pada Bab 2.6 maka pencarian *centroid* terdekat dicari pada setiap kelas, sehingga tetangga pertama yang didapat sejumlah kelas pada *dataset*. Berikut merupakan contoh perhitungan jarak data latih pertama pada Tabel 4.6 dan data uji pertama Tabel 4.5.

$$\text{jarak} = \sqrt{(0 - 0.368)^2 + (0 - 0.46)^2 + (0.25 - 0.25)^2 + \dots + (1 - 0)^2}$$

$$\text{jarak} = 1.170392083$$

Tabel 4.7 Jarak Centroid Data Uji 1 (k = 1)

Data Latih	Kelas	Jarak
1	1	1.170392083
2	1	0.318664749
3	1	0.283312256
4	1	0.576514007
5	2	0.867776287
6	2	1.042158235
7	2	1.168049157
8	2	1.045927483
9	3	1.675210398
10	3	1.711990434
11	3	1.349259932
12	3	1.536564396

Tabel 4.8 Jarak Centroid Data Uji 2 ($k = 1$)

Data Latih	Kelas	Jarak
1	1	1.752832465
2	1	1.02929297
3	1	1.164043081
4	1	1.150179652
5	2	0.48724281
6	2	0.734184171
7	2	1.203097482
8	2	1.275511334
9	3	0.988892975
10	3	1.15621754
11	3	1.079099403
12	3	1.06033536

Tabel 4.9 Jarak Centroid Data Uji 3 ($k = 1$)

Data Latih	Kelas	Jarak
1	1	1.772105725
2	1	1.094633798
3	1	1.070596655
4	1	1.10676411
5	2	1.393068538
6	2	1.361694949
7	2	0.705095057
8	2	0.91042067
9	3	1.902091863
10	3	1.448581176
11	3	1.456776724
12	3	1.470065599



Tabel 4.10 Jarak *Centroid* Data Uji 4 ($k = 1$)

Data Latih	Kelas	Jarak
1	1	2.092942788
2	1	1.849795673
3	1	1.795711628
4	1	1.578608002
5	2	1.596436994
6	2	1.217684887
7	2	1.576199664
8	2	1.504287237
9	3	1.129179827
10	3	0.999115566
11	3	0.804632421
12	3	0.760726991

Setelah melakukan pencarian *centroid* terdekat pertama atau tetangga terdekat pertama dari setiap kelas pada data latih akan dicari *centroid* terdekat kedua sampai k . Perhitungan *centroid* terdekat kedua sampai k berbeda dengan pencarian *centroid* pertama sebelumnya hanya dengan menghitung jarak antara data uji dengan data latih, tetapi perhitungan *centroid* kedua sampai k bergantung pada k sebelumnya dan menghitung nilai *centroid* dengan Persamaan (2.6). Sebagai contoh untuk $k = 2$ maka nilai *centroid* kedua adalah perhitungan *centroid* dengan Persamaan (2.6) antara *centroid* pertama dengan tetangga kedua, untuk $k = 3$ maka nilai *centroid* ketiga adalah perhitungan *centroid* dengan Persamaan (2.6) antara *centroid* pertama, kedua dan tetangga ketiga.

Tabel 4.11 sampai 4.14 merupakan tabel hasil dari perhitungan *centroid* untuk $k = 2$ dari setiap data uji. Menghilangkan data latih yang sudah menjadi *centroid* sebelumnya (*centroid* pertama) agar tidak terhitung menjadi *centroid* lagi. Berikut merupakan contoh perhitungan *centroid* kedua ($k = 2$) menggunakan *centroid* pertama kelas 1 (data latih 4) pada Tabel 4.6 dan data uji pertama Tabel 4.5.

$$X1_{centroid_2} = \frac{(0 + 0.3157)}{2} = 0.157$$

**Tabel 4.11 Centroid Data Uji 1 ($k = 2$)**

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.157	0.198	0.312	0	0.104	0.1	0.145	0.24	0.5	1
2	0.342	0.429	0.312	0	0.020	0.16	0.291	0.12	0	1
4	0.289	0.306	0.375	0	0.229	0.02	0.312	0.32	0	1
6	0.315	0.953	0.125	0.166	0.541	0.46	0.395	0.36	0	2
7	0.605	0.731	0.5	0.166	0.666	0.42	0.291	0.3	0	2
8	0.421	0.670	0.5	0.166	0.562	0.46	0.25	0.36	0	2
9	0.105	0.578	0.125	0.166	0.770	0.8	0.729	1	0	3
10	0.526	0.480	0.25	0.5	0.770	0.82	0.625	0.94	0	3
12	0.157	0.389	0.25	0.333	0.812	0.58	0.854	0.88	0	3

Tabel 4.12 Centroid Data Uji 2 ($k = 2$)

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.184	0.230	0.25	0	0.083	0.22	0.146	0.12	0.5	1
3	0.342	0.429	0.312	0	0.020	0.16	0.292	0.12	0	1
4	0.315	0.338	0.312	0	0.208	0.14	0.313	0.2	0	1
6	0.315	0.953	0.125	0.166	0.541	0.46	0.396	0.36	0	2
7	0.605	0.731	0.5	0.166	0.666	0.42	0.292	0.3	0	2
8	0.421	0.670	0.5	0.166	0.562	0.46	0.25	0.36	0	2
10	0.473	0.669	0.125	0.333	0.916	0.98	0.646	0.94	0	3
11	0.105	0.578	0.125	0.166	0.770	0.8	0.729	1	0	3
12	0.105	0.578	0.125	0.166	0.958	0.74	0.875	0.88	0	3

Tabel 4.13 Centroid Data Uji 3 ($k = 2$)

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.157	0.198	0.312	0	0.104	0.1	0.145	0.24	0.5	1
2	0.342	0.429	0.312	0	0.020	0.16	0.291	0.12	0	1
4	0.289	0.306	0.375	0	0.229	0.02	0.312	0.32	0	1
5	0.605	0.731	0.5	0.166	0.666	0.42	0.291	0.3	0	2
6	0.394	0.778	0.625	0.333	0.583	0.36	0.520	0.42	0	2
8	0.5	0.495	1	0.333	0.604	0.36	0.375	0.42	0	2
9	0.473	0.669	0.125	0.333	0.916	0.98	0.648	0.94	0	3
11	0.526	0.480	0.25	0.5	0.770	0.82	0.625	0.94	0	3
12	0.526	0.480	0.25	0.5	0.958	0.76	0.770	0.82	0	3

Tabel 4.14 Centroid Data Uji 4 ($k = 2$)

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.131	0.107	0.312	0	0.291	0.08	0.167	0.32	0.5	1
2	0.315	0.338	0.312	0	0.208	0.14	0.313	0.2	0	1
3	0.289	0.306	0.375	0	0.229	0.02	0.313	0.32	0	1
5	0.315	0.953	0.125	0.166	0.541	0.46	0.396	0.36	0	2
7	0.394	0.778	0.625	0.333	0.583	0.36	0.521	0.42	0	2
8	0.210	0.717	0.625	0.333	0.479	0.4	0.479	0.48	0	2
9	0.105	0.578	0.125	0.166	0.958	0.74	0.875	0.88	0	3
10	0.526	0.480	0.25	0.5	0.958	0.76	0.771	0.82	0	3
11	0.157	0.389	0.25	0.333	0.812	0.58	0.854	0.88	0	3

Langkah selanjutnya adalah melakukan perhitungan jarak *centroid* terdekat dari setiap kelas untuk dijadikan sebagai *centroid* kedua terdekat dari setiap kelas. Jarak *Euclidean* antara *centroid* kedua dari data latih dapat dilihat pada Tabel 4.15 sampai 4.18. Jarak terkecil akan ditandai dengan warna abu - abu pada tabel dan akan dijadikan *centroid* kedua terdekat dari kelasnya masing – masing.

Tabel 4.15 Jarak *Centroid* Data Uji 1 ($k = 2$)

Data Latih	Kelas	Jarak
1	1	0.614688369
2	1	0.236834888
4	1	0.393600556
6	2	0.867292929
7	2	0.844482079
8	2	0.741706689
9	3	1.477934526
10	3	1.462796578
12	3	1.419979804

Tabel 4.16 Jarak *Centroid* Data Uji 2 ($k = 2$)

Data Latih	Kelas	Jarak
1	1	1.297204241
3	1	1.082776243
4	1	1.032498182
6	2	0.46994496
7	2	0.704902317
8	2	0.747526708
10	3	0.910415693
11	3	0.970596085
12	3	0.942934152

Tabel 4.17 Jarak *Centroid* Data Uji 3 ($k = 2$)

Data Latih	Kelas	Jarak
1	1	1.340174619
2	1	1.06648226
4	1	1.064967736
5	2	0.934538506
6	2	0.933011424
8	2	0.779460773
9	3	1.590488294
11	3	1.369072658
12	3	1.357411323

Tabel 4.18 Jarak *Centroid* Data Uji 4 ($k = 2$)

Data Latih	Kelas	Jarak
1	1	1.759876263
2	1	1.682779694
3	1	1.675379365
5	2	1.359527885
7	2	1.295532771
8	2	1.272606406
9	3	0.874580435
10	3	0.70802125
11	3	0.733924951

Karena pada tahapan manualisasi ini nilai dari parameter k adalah 2 maka proses perhitungan untuk mencari *centroid* terdekat selesai. Sehingga didapat kumpulan dari *centroid* terdekat (*Set of KNCN*) sebanyak k yaitu 2 dengan anggota dari *Set of KNCN* adalah data latih yang terpilih, dapat dilihat pada Tabel 4.19.

Tabel 4.19 Set of K - Nearest Centroid

Data Uji	Data Latih					
	Kelas 1		Kelas 2		Kelas 3	
	$k = 1$	$k = 2$	$k = 1$	$k = 2$	$k = 1$	$k = 2$
1	3	2	5	8	11	12
2	2	4	5	6	9	10
3	3	4	7	8	10	12
4	4	3	6	8	12	10

4.2.4 Perhitungan Local Mean Vector

Setelah mendapatkan *Set of KNCN* maka dapat dilakukan tahapan perhitungan *Local Mean Vector* untuk merubah *Set of KNCN* yang didapat menjadi rata – rata poin lokal. Menghitung *Local Mean Vector* menggunakan Persamaan (2.4). Hasil dari perhitungan *Local Mean Vector* dari setiap data uji 1 sampai 4 yang dapat dilihat pada Tabel 4.20 sampai 4.23. Berikut merupakan contoh perhitungan *local mean vector* kelas yang pertama dari data latih yang terpilih menjadi *centroid* yaitu data latih ketiga dan kedua pada Tabel 4.6 dan data uji pertama pada Tabel 4.5.

$$(k = 1), X1_{lm} = \frac{0.315}{1} = 0.315$$

$$(k = 2), X1_{lm} = \frac{0.315 + 0.368}{2} = 0.342$$

Tabel 4.20 Local Mean Vector Data Uji 1

K	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.315	0.397	0.375	0	0.041	0.04	0.291	0.24	0	1
2	0.342	0.429	0.3125	0	0.020	0.16	0.291	0.12	0	1
1	0.526	0.906	0	0	0.625	0.52	0.166	0.24	0	2
2	0.421	0.670	0.5	0.166	0.562	0.46	0.25	0.36	0	2
1	0.157	0.389	0.25	0.333	0.625	0.64	0.708	1	0	3
2	0.157	0.389	0.25	0.333	0.812	0.58	0.854	0.88	0	3

**Tabel 4.21 Local Mean Vector Data Uji 2**

K	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.368	0.460	0.25	0	0	0.28	0.291	0	0	1
2	0.315	0.338	0.3125	0	0.208	0.14	0.312	0.2	0	1
1	0.526	0.906	0	0	0.625	0.52	0.166	0.24	0	2
2	0.315	0.953	0.125	0.166	0.541	0.46	0.395	0.36	0	2
1	0.052	0.767	0	0	0.916	0.96	0.75	1	0	3
2	0.473	0.669	0.125	0.333	0.916	0.98	0.645	0.94	0	3

Tabel 4.22 Local Mean Vector Data Uji 3

K	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.315	0.397	0.375	0	0.041	0.04	0.291	0.24	0	1
2	0.289	0.306	0.375	0	0.229	0.02	0.312	0.32	0	1
1	0.684	0.556	1	0.333	0.708	0.32	0.416	0.36	0	2
2	0.5	0.495	1	0.333	0.604	0.36	0.375	0.42	0	2
1	0.894	0.570	0.25	0.666	0.916	1	0.541	0.88	0	3
2	0.526	0.480	0.25	0.5	0.958	0.76	0.770	0.82	0	3

Tabel 4.23 Local Mean Vector Data Uji 4

K	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1	0.263	0.215	0.375	0	0.416	0	0.333	0.4	0	1
2	0.289	0.306	0.375	0	0.229	0.02	0.3125	0.32	0	1
1	0.105	1	0.25	0.333	0.458	0.4	0.625	0.48	0	2
2	0.210	0.717	0.625	0.333	0.479	0.4	0.479	0.48	0	2
1	0.157	0.389	0.25	0.333	1	0.52	1	0.76	0	3
2	0.526	0.480	0.25	0.5	0.958	0.76	0.770	0.82	0	3

4.2.5 Perhitungan *Harmonic Mean Distance*

Setelah menghitung *Local Mean Vector* dari *Set of KNCN* langkah berikutnya adalah menghitung *Harmonic Mean Distance* menggunakan Persamaan (2.7). *Harmonic Mean Distance* dilakukan dengan menghitung jumlah dari jarak *Local Mean Vector* dari setiap kelas dengan data uji sehingga hasil dari *Harmonic Mean Distance* nanti akan dipakai untuk menentukan kelas. Hasil dari perhitungan *Harmonic Mean Distance* dari setiap data uji 1 sampai 4 berturut – turut dapat dilihat pada Tabel 4.24 sampai Tabel 4.27. Hasil *Harmonic Mean Distance* yang terkecil akan ditandai dengan warna abu - abu untuk menentukan kelas nantinya. Berikut merupakan contoh perhitungan *Harmonic Mean Distance* dari *Local Mean Vector* pada kelas yang pertama pada Tabel 4.20.

$$\begin{aligned} euclidean_1 &= \sqrt{(0.315 - 0.368)^2 + (0.397 - 0.46)^2 + \dots + (0 - 0)^2} \\ euclidean_1 &= 0.283 \\ euclidean_2 &= \sqrt{(0.342 - 0.368)^2 + (0.429 - 0.46)^2 + \dots + (0 - 0)^2} \\ euclidean_2 &= 0.236 \\ harmonic\ mean\ distance &= \frac{2}{\left(\frac{1}{0.283} + \frac{1}{0.236}\right)} = 0.257997097 \end{aligned}$$

Tabel 4.24 *Harmonic Mean Distance* Data Uji 1

Kelas	HMD
1	0.257997097
2	0.799804019
3	1.383716858

Tabel 4.25 *Harmonic Mean Distance* Data Uji 2

Kelas	HMD
1	1.030893085
2	0.478437586
3	0.948033038



Tabel 4.26 Harmonic Mean Distance Data Uji 3

Kelas	HMD
1	1.067774777
2	0.740415318
3	1.401515144

Tabel 4.27 Harmonic Mean Distance Data Uji 4

Kelas	HMD
1	1.625554726
2	1.24454002
3	0.733428453

4.2.6 Menentukan Kelas

Menentukan kelas pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dapat dilakukan dengan menggunakan hasil dari *Harmonic Mean Distance* yang terkecil. Sehingga jika didapat dari Tabel 4.24 sampai 4.27 didapatkan hasil kelas klasifikasi sebagai berikut yang dapat dilihat pada Tabel 4.28.

Tabel 4.28 Hasil Klasifikasi

Data Uji	Prediksi	Aktual
1	1	1
2	2	2
3	2	2
4	3	3

4.2.7 Perhitungan Akurasi

Mengevaluasi hasil model klasifikasi dalam perhitungan manualisasi ini digunakan nilai akurasi. Hasil akurasi dari model klasifikasi LMKHNCN dapat menggunakan Persamaan (2.19).

$$akurasi = \frac{4}{4} \times 100\%$$

$$akurasi = 100\%$$



4.3 Perancangan Pengujian Metode

Tahapan perancangan pengujian Metode LMKHNCN bertujuan untuk melihat sensitivitas parameter k dan akurasi dari setiap pengujian parameter k dengan tambahan pengujian *dataset* yang dinormalisasi dan tidak dinormalisasi. Sehingga didapat parameter k yang optimal pada kasus ini dan terlihat sensitivitas dari parameter k yang diubah terhadap akurasinya.

4.3.1 Pengujian Sensitivitas K dan Normalisasi

Pengujian sensitivitas parameter k digunakan nilai parameter k dari 1 sampai 15 dengan *10-Fold Cross Validation*. Pemilihan k merujuk pada deviasi standar data rendah yang artinya persebaran data yang rapat, sehingga dalam pemilihan parameter k untuk deviasi standar data yang rendah digunakan parameter k yang kecil (Suyanto, 2017). Berikut dapat dilihat pada Tabel 4.29 yang merupakan tabel perancangan pengujian sensitivitas parameter k dan pengujian perbandingan normalisasi dan tanpa normalisasi. Tabel akan disertai dengan deviasi standar karena evaluasi model klasifikasi pengujian digunakan *10-Fold Cross Validation*.

Tabel 4.29 Perancangan Pengujian Akurasi

K	Akurasi	Deviasi Standar
1		
2		
...		
15		

Juga dapat dilihat pada Tabel 4.30 merupakan pengujian *F1-Score*, berguna untuk mengevaluasi data yang sifatnya tidak seimbang juga untuk analisa serta membandingkan dengan hasil akurasi dari model klasifikasi.

Tabel 4.30 Perancangan Pengujian *F1-Score*

K	Precision	Recall	F1-Score
1			
2			
...			
15			



BAB 5 IMPLEMENTASI

5.1 Implementasi Program

Bab ini akan menjelaskan tentang hasil implementasi program yang telah dirancang pada Bab 4. Program ini dibuat dengan Bahasa pemrograman python dengan kode editor *jupyter notebook*. Implementasi program dibuat menggunakan fungsi atau *method* yang terdapat dua jenis fungsi atau *method*. Fungsi pertama ialah fungsi pendukung seperti fungsi perhitungan jarak, fungsi perhitungan jumlah kelas dan sebagainya yang akan dijelaskan setelah ini. Fungsi kedua yakni fungsi utama, fungsi yang terbentuk dari hasil perancangan pada Bab 4 yaitu fungsi metode yang digunakan (LMKHNCN).

5.1.1 Fungsi Pendukung

Fungsi pendukung merupakan fungsi selain fungsi utama dari perancangan yang telah dilakukan pada Bab 4. Fungsi ini bertujuan agar tidak mengulang penulisan program dan memudahkan pembacaan pada fungsi utama.

5.1.1.1 Import Library

Import library berguna untuk mengambil fungsi dasar yang telah dibuat oleh ahli seperti *numpy* untuk perhitungan matematis, *pandas* untuk menyimpan dan mengolah data, *math* sama seperti *numpy* hanya saja lebih mendasar, *random* untuk memproses sebuah angka acak, dan *dill* untuk menyimpan *session* agar tidak mengulang program yang membutuhkan waktu yang cukup lama. Fungsi ini juga berguna untuk menghemat waktu dibandingkan dengan menulis ulang program yang digunakan, berikut merupakan kode program dari *import library* yang dapat dilihat pada Kode Program 5.1.

Kode Program 5.1 Import Library

Import Library	
1	import numpy as np
2	import pandas as pd
3	import math
4	import random
5	import dill
6	try:
7	dill.load_session('Pengujian.db')

Penjelasan Kode Program 5.1:

1. Baris ke – 1 sampai 5 merupakan pemanggilan *library* yang digunakan yakni *numpy*, *pandas*, *math*, *random* dan *dill*.
2. Baris ke – 6 dan 7 merupakan *error catching* jika *file library* tidak ditemukan.



5.1.1.2 Identifikasi Kelas

Fungsi pendukung ini berguna untuk mengidentifikasi kelas di mana nilai masukan merupakan *array dataset* yang akan menghasilkan jumlah kelasnya dan *array* dari kelas tersebut. Berikut merupakan kode program dari identifikasi kelas yang dapat dilihat pada Kode Program 5.2.

Kode Program 5.2 Identifikasi Kelas

```

Identifikasi kelas
1  def count_label(data):
2      distinct = [data[0][-1]]
3      for i in range(len(data)):
4          check = True
5          for j in range(len(distinct)):
6              if distinct[j] == data[i][-1]:
7                  check = False
8          if check:
9              distinct.append(data[i][-1])
10     return np.array(distinct).astype(int), len(distinct)
    
```

Penjelasan Kode Program 5.2:

1. Baris ke – 1 dan 2 inialisasi fungsi dengan parameter data dan variabel yang akan dikembalikan.
2. Baris ke – 3 sampai 9 merupakan pengulangan pengecekan nilai yang sama jika nilai variabel *check* adalah *False* maka variabel kelas akan dimasukkan kedalam variabel *distinct*.
3. Baris ke – 10 merupakan nilai kembalian yakni *array distinct* dan panjang *array distinct*.

5.1.1.3 Euclidean Distance

Fungsi pendukung ini merupakan fungsi perhitungan jarak menggunakan metode pengukuran jarak *Euclidean* yang terdapat pada Persamaan (2.2). Fungsi ini memiliki nilai masukan *array1* dan *array2* yang mengeluarkan jarak antara kedua titik data. Berikut merupakan kode program dari metode *Euclidean Distance* yang dapat dilihat pada Kode Program 5.3.

Kode Program 5.3 Euclidean Distance

```

Euclidean Distance
1  def euclidian(X, Y, categories = []):
2      euc = 0
3      for i in range(len(X)):
4          if not i in categories: ## similarity matrix
5              euc += math.pow((X[i]-Y[i]), 2)
6      euc = math.sqrt(euc)
7      if len(categories) != 0:
    
```




```

8      p = len(categories)
9      m = 0
10     for i in range(len(categories)):
11         if X[categories[i]] == Y[categories[i]]:
12             m += 1
13     cat_dist = (p-m)/p
14     euc += cat_dist
15     return euc

```

Penjelasan Kode Program 5.3:

1. Baris ke – 1 dan 2 merupakan inisialisasi fungsi parameter *array* X dan Y ditambah nilai *default array* kosong variabel untuk dikembalikan.
2. Baris ke – 3 sampai 5 merupakan pengulangan pengecekan variabel yang merupakan nilai kategorikal atau tidak, jika tidak maka dihitung jarak menggunakan *Euclidian distance* jika ya dihitung menggunakan *similarity matrix*.
3. Baris ke – 6 perhitungan nilai *Euclidian distance*.
4. Baris ke – 7 sampai 14 merupakan perhitungan *similarity matrix* yaitu fitur yang bertipe kategorikal.
5. Baris ke – 15 merupakan pengembalian nilai jarak *Euclidian* ditambah dengan nilai *similarity matrix*.

5.1.1.4 Deviasi Standar dan Rata – Rata

Fungsi pendukung ini bertujuan untuk menghitung deviasi standar dan rata – rata dari sebuah nilai masukkan berupa *array*. Fungsi ini nantinya akan digunakan untuk analisa hasil akurasi dari performa model klasifikasi LMKHNCN. Berikut merupakan kode program dari perhitungan deviasi standar dan rata – rata yang dapat dilihat pada Kode Program 5.4.

Kode Program 5.4 Deviasi Standar dan Rata – Rata

```

Deviasi Standar dan Rata – Rata
1      def evaluation(arr_acc):
2          mean = np.mean(arr_acc)
3          varian = sum((arr_acc - mean) * (arr_acc - mean)) /
4                  len(arr_acc)
5          std = math.sqrt(varian)
6          return mean, std

```

Penjelasan Kode Program 5.4:

1. Baris ke – 1 merupakan inisialisasi fungsi parameter *array* akurasi.
2. Baris ke – 2 sampai 5 merupakan perhitungan berurut – turut pengembalian nilai rata – rata, varian dan deviasi standar.



5.1.2 Fungsi Utama

Fungsi utama merupakan fungsi yang sudah dirancang pada Bab 4. Fungsi ini akan menjelaskan hasil implementasi dari metode yang telah dirancang dari *pre-processing* sampai penentuan kelas klasifikasi. Di bawah ini akan diperlihatkan kode program dan kegunaan dari fungsi utama yang telah diimplementasikan.

5.1.2.1 Pre-Processing

Fungsi ini merupakan tahapan dari fungsi utama yang terlebih dahulu diproses yang nantinya berfungsi untuk membersihkan dan memanipulasi *dataset* sebelum diproses pada model klasifikasi.

5.1.2.1.1 Missing Value

Fungsi *missing value* merupakan tahapan untuk membersihkan data yang kosong pada setiap kelasnya. Fungsi ini memiliki masukan *array dataset* yang memiliki data yang kosong didalamnya dan akan mengeluarkan *dataset* yang telah terisi dengan menggunakan rata-rata dari setiap kelasnya yang telah dibahas pada Bab 2.3.1.1. Berikut merupakan kode program dari fungsi normalisasi yang dapat dilihat pada Kode Program 5.5.

Kode Program 5.5 Missing Value

```

Missing Value
1  def handleMissValue(data):
2      filled_data = np.array([[0 for i in
3          range(len(data[0]))] for j in range(len(data))])
4      cLabel = count_label(data)[1]
5      arr_data = [[] for j in range(cLabel)]
6      mean = []
7      for i in range(len(data)):
8          arr_data[int(data[i][-1])-1].append(data[i])
9      counter = 0
10     for i in range(len(arr_data)):
11         mean.append(np.nanmean(arr_data[i], axis=0))
12         for j in range(len(arr_data[i])):
13             for k in range(len(arr_data[i][j])):
14                 if np.isnan(arr_data[i][j][k]):
15                     arr_data[i][j][k] = mean[i][k]
16                     filled_data[counter] = arr_data[i][j]
17                     counter += 1
18     return filled_data

```

Penjelasan Kode Program 5.5:

1. Baris ke – 1 sampai 5 merupakan inisialisasi fungsi parameter *array* data dan inisialisasi variabel *filled data* untuk dikembalikan, *cLabel* untuk menghitung jumlah kelas *arr_data* data yang asli dan *mean* untuk rata – rata.



2. Baris ke – 6 dan 7 pengulangan mengisi variabel dengan data yang asli dari setiap kelas untuk dikalkulasi rata – rata dari setiap kelas.

3. Baris ke – 9 sampai 16 merupakan pengulangan mengisi variabel *nan* atau kosong diisi dengan rata – rata dari setiap kelas.

4. Baris ke – 17 pengembalian nilai data yang telah diisi.

5.1.2.1.2 Normalisasi

Tahapan ini merupakan tahapan opsional yang dapat dilewati jika pengujian tidak menggunakan normalisasi. Fungsi normalisasi digunakan untuk menyetarakan jarak antar fitur yang telah dibahas pada Bab 2.3.1.2. Berikut merupakan kode program dari fungsi normalisasi yang dapat dilihat pada Kode Program 5.6.

Kode Program 5.6 Normalisasi

```

Normalisasi
1 def normalize(data):
2     data = np.array(data).astype(float)
3     max_value = np.max(data, axis=0)
4     min_value = np.min(data, axis=0)
5     for i in range(len(data)):
6         for j in range(len(data[0])-1):
7             data[i][j] = (data[i][j] - min_value[j]) /
8                 (max_value[j] - min_value[j])
9     return data
    
```

Penjelasan Kode Program 5.6:

1. Baris ke – 1 sampai 4 merupakan inisialisasi fungsi dengan parameter *array* data dan inisialisasi variabel *data* yaitu data asli, *max_value* yaitu nilai maksimal dari data setiap fitur dan *min_value* nilai minimal dari data setiap fitur.

2. Baris ke – 5 sampai 7 merupakan pengulangan untuk menghitung nilai normalisasi data dari setiap fitur.

3. Baris ke – 8 pengembalian nilai data yang telah dinormalisasi.

5.1.2.1.3 Random Dataset

Fungsi ini merupakan tahapan dari *preprocessing* di mana sebelum melakukan proses klasifikasi fungsi ini memiliki masukan *array dataset* yang belum teracak dan mengeluarkan *array dataset* acak. Berikut merupakan kode program dari fungsi *random dataset* yang dapat dilihat pada Kode Program 5.7.



Kode Program 5.7 Random Dataset

```
Random Dataset
1 def randomize(arr):
2     for i in range(len(arr)-1, 0, -1):
3         j = random.randint(0, i+1)
4         arr[i], arr[j] = arr[j], arr[i]
5     return arr
```

Penjelasan Kode Program 5.7:

1. Baris ke – 1 inialisasi fungsi dengan parameter *array* data.
2. Baris ke – 2 sampai 4 merupakan pengulangan untuk melakukan pengacakan data dengan menukar data urutan teratas ke bagian bawah secara acak dengan *library* random.
3. Baris ke – 5 merupakan pengembalian nilai *array* data yang telah diacak.

5.1.2.1.4 Ekstraksi File

Fungsi ini merupakan fungsi untuk memanggil dari fungsi – fungsi *preprocessing* sebelumnya atau bisa disebut pengaturan sebelum melakukan proses klasifikasi. Fungsi ini memiliki parameter yang berguna untuk mengatur *dataset*. Parameter pertama yakni *array dataset*, kedua yaitu *boolean* untuk mengatur normalisasi atau tidaknya dan terakhir *boolean* acak data untuk mengatur data di acak atau tidak. Berikut merupakan kode program dari fungsi ekstraksi file yang dapat dilihat pada Kode Program 5.8.

Kode Program 5.8 Ekstraksi File

```
Ekstraksi File
1 def extraction(file, is_normal = True, is_random=True):
2     data = pd.read_excel(file)
3     data = handleMissValue(np.array(data))
4     data = pd.DataFrame(data)
5     data.drop_duplicates()
6     data = np.array(data)
7     if is_normal:
8         data = normalize(data)
9     if is_random:
10        data = randomize(data)
11    return data
```

Penjelasan Kode Program 5.8:

1. Baris ke – 1 inialisasi fungsi dengan parameter *file* yaitu nama file, *is_normal* merupakan normalisasi atau tidak dan *is_random* merupakan data diacak atau tidak.



2. Baris ke – 2 sampai 5 berturut – turut meruapkan proses pembacaan file excel, mengisi nilai kosong, menyimpan data pada *dataframe* dan menghilangkan nilai duplikat dari data.

3. Baris ke – 7 sampai 10 merupakan proses normalisasi dan pengacakan data.

4. Baris ke – 11 merupakan pengembalian data dalam bentuk *array*.

5.1.2.1.5 K-Fold Cross Validation

Fungsi ini merupakan fungsi untuk membagi *dataset* menjadi 2 bagian yaitu data latih dan data uji dengan membagikan secara sama rata sejumlah K seperti yang sudah dijelaskan pada Bab 2.11. Berikut merupakan kode program dari fungsi *K-Fold Cross Validation* yang dapat dilihat pada Kode Program 5.9.

Kode Program 5.9 K-Fold Cross Validation

```
K-Fold Cross Validation
1 def crossValidation(data, cross_val, cross_index,
2 label_index):
3     split_test = int(len(data) * (1 / cross_val))
4     change_index = (cross_index - 1) * split_test
5     for i in range(split_test):
6         j = (i + change_index)
7         data[i], data[j] = data[j], data[i]
8     data_train_feature =
9     pd.DataFrame(data[split_test:]).drop([label_index],
10 axis=1)
11     data_train_label =
12     pd.DataFrame(pd.DataFrame(data[split_test:]),
13 columns=[label_index]).astype(np.int)
14     data_test_feature =
15     pd.DataFrame(data[0:split_test]).drop([label_index],
16 axis=1)
17     data_test_label =
18     pd.DataFrame(pd.DataFrame(data[0:split_test]),
19 columns=[label_index]).astype(np.int)
20     data_train_feature = np.array(data_train_feature)
21     data_train_label = np.array(data_train_label)
22     data_test_feature = np.array(data_test_feature)
23     data_test_label = np.array(data_test_label)
24     return data, data_train_feature, data_train_label,
25 data_test_feature, data_test_label
```

Penjelasan Kode Program 5.9:

1. Baris ke – 1 inialisasi fungsi dengan paramater *data* yaitu data asli, *cross_val* yaitu jumlah pembagian data, *cross_index* yaitu indeks *k* pada *K-Fold Cross Validation* dan *label_index* merupakan kelas dari data.



2. Baris ke – 2 dan 3 merupakan variabel nilai untuk pembagian data.
3. Baris ke – 4 sampai 6 merupakan pengulangan untuk menukar data latih dan data uji sesuai dengan nilai pada Baris 2 dan 3.
4. Baris ke – 7 sampai 14 merupakan pembagian fitur data latih, kelas data latih, fitur data uji dan kelas data uji pada *dataframe*.
5. Baris ke – 15 pengembalian data yang telah dibagi sebelumnya.

5.1.2.2 Klasifikasi LMKHNCN

Tahapan klasifikasi metode LMKHNCN memiliki beberapa langkah yang telah dijelaskan pada Bab 2.9. Setelah ini akan dijelaskan hasil implementasi kode program akan dijelaskan sesuai dengan langkah – langkah dan perancangan pada Bab 4.

5.1.2.2.1 Set of KNCN

Tahapan implementasi program Set of KNCN memiliki 2 bagian seperti yang telah dijelaskan pada Bab 4. Bagian pertama adalah pencarian *centroid* pertama terdekat dan bagian kedua adalah pencarian *centroid* terdekat ke – *n*. Berikut merupakan kode program dari fungsi Set of KNCN bagian pertama dan kedua berturut – turut yang dapat dilihat pada Kode Program 5.10 dan 5.11.

Kode Program 5.10 1-NCN

Algoritma: 1-NCN

```

1 def closest(Xx, Xx_label, Y, cLabel):
2     distance = [[None for i in range(len(Xx))] for j in
3                 range(cLabel)]
4     min_distance_value = [0 for i in range(cLabel)]
5     min_distance_index = [0 for i in range(cLabel)]
6     for i in range(len(Xx)):
7         distance[((Xx_label[i][0])-1)][i] =
8             euclidian(Xx[i], Y)
9         for i in range(cLabel):
10            min_distance_value[i] = min(x for x in
11                                       distance[i] if x is not None)
12            min_distance_index[i] = distance[i].index(min(x
13                                                         for x in distance[i] if x is not None))
14     return np.array(distance), min_distance_index,
15            min_distance_value

```

Penjelasan Kode Program 5.10:

1. Baris ke – 1 inialisasi fungsi dengan parameter *XX* yaitu fitur data latih, *XX_label* yaitu kelas data latih, *Y* yaitu data uji dan *cLabel* merupakan jumlah kelas.



2. Baris ke – 2 sampai 4 berurut – turut merupakan inisialisasi variabel *distance* untuk menyimpan jarak data latih ke data uji, *min_distance_value* dan *min_distance_index* untuk menyimpak nilai minimum dan indeks minimum jarak setiap data latih ke data uji.
3. Baris ke – 6 dan 7 merupakan pengulangan perhitungan jarak setiap data latih ke data uji.
4. Baris ke – 7 sampai 10 merupakan penyimpanan nilai jarak.
5. Baris ke – 11 merupakan pengembalian nilai jarak, dan indeks maupun nilai minimum jarak.

Kode Program 5.11 N-NCN

Algoritma: N-NCN

```

1  def kncn(k, Xx, Xx_label, Y, cLabel, close_dist):
2      ncn = np.array([[0 for i in range(k)] for j in
3                      range(cLabel)])
4      for i in range(cLabel):
5          ncn[i][0] = close_dist[i]
6      for i in range(1, k):
7          distance = [None for i in range(len(Xx))] for j
8                      in range(cLabel)]
9          min_distance_value = [0 for i in range(cLabel)]
10         min_distance_index = [0 for i in range(cLabel)]
11         for j in range(len(Xx)):
12             label = Xx_label[j][0] - 1
13             check = True
14             for a in range(i):
15                 if ncn[label][a] == j:
16                     check = False
17             if check:
18                 add = 0 + Xx[j]
19                 for a in range(i):
20                     indeks_ncn = ncn[label][a]
21                     add += Xx[indeks_ncn]
22                 centroid = add / (i+1)
23                 distance[label][j] = euclidian(Y,
24                                                 centroid)
25         for j in range(cLabel):
26             min_distance_value[j] = min(x for x in
27                                         distance[j] if x is not None)
28             min_distance_index[j] =
29                 distance[j].index(min(x for x in distance[j]
30                                     if x is not None))
31             ncn[j][i] = min_distance_index[j]
32     return ncn

```




Penjelasan Kode Program 5.11:

1. Baris ke – 1 inialisasi fungsi dengan parameter k nilai K pada metode LMKHNCN, XX yaitu fitur data latih, XX_label yaitu kelas data latih, Y yaitu data uji dan $close_dist$ merupakan *array* data latih yang terdekat dengan data uji.
2. Baris ke – 2 inialisasi variabel untuk menyimpan nilai NCN (*centroid* terdekat).
3. Baris ke – 3 dan 4 mengisi nilai *centroid* terdekat pertama pada variabel.
4. Baris ke – 5 sampai 24 merupakan perhitungan algoritme NCN pada setiap k dicari nilai *centroid* terdekat.
5. Baris ke – 25 pengembalian nilai berisi *array* NCN (*centroid* terdekat)

5.1.2.2.2 Local Mean Vector

Tahapan implementasi *Local Mean Vector* memiliki masukan dari keluaran tahapan *Set of KNCN* agar dapat dijalankan. Tahapan fungsi implementasi *Local Mean Vector* akan menghasilkan nilai titik *centroid* dari *Set of KNCN* seperti yang telah dijelaskan pada Bab 2.9. Berikut merupakan kode program dari fungsi *Local Mean Vector* yang dapat dilihat pada Kode Program 5.12.

Kode Program 5.12 Local Mean Vector

Algoritma: *Local Mean Vector*

```

1  def localMeanVector(Xx, ncn):
2      lm_ncn = [[0 for i in range(len(ncn[0]))] for j in
3              range(len(ncn))]
4      for i in range(len(ncn)):
5          lm_ncn[i][0] = Xx[ncn[i][0]]
6          for j in range(1, len(ncn[0])):
7              add = 0 + lm_ncn[i][0]
8              for a in range(j):
9                  add += Xx[ncn[i][a + 1]]
10             lm_ncn[i][j] = add / (j+1)
11     return np.array(lm_ncn)

```

Penjelasan Kode Program 5.12:

1. Baris ke – 1 inialisasi fungsi dengan parameter XX yaitu fitur data latih dan ncn yaitu *array* NCN (*centroid* terdekat).
2. Baris ke – 2 inialisasi variable untuk menyimpan nilai *local mean vector*.
3. Baris ke 3 sampai 9 merupakan pengulangan untuk menghitung nilai *local mean vector* dari *array* NCN dan disimpan variabel pada Baris ke – 2.



4. Baris ke – 10 merupakan pengembalian nilai *local mean vector* yang telah dihitung.

5.1.2.2.3 Harmonic Mean Distance

Tahapan implementasi *Harmonic Mean Distance* memiliki masukan dari keluaran tahapan *Local Mean Vector* agar dapat dijalankan. Tahapan fungsi implementasi *Harmonic Mean Distance* akan menghasilkan nilai rata – rata harmonik dari jarak *euclidean* data uji dengan *centroid* seperti yang telah dijelaskan pada Bab 2.9. Berikut merupakan kode program dari fungsi *Harmonic Mean Distance* yang dapat dilihat pada Kode Program 5.13.

Kode Program 5.13 Harmonic Mean Distance

```

Algoritma: Harmonic Mean Distance
1  def harmonicMean(lm_ncn, Y):
2      k = len(lm_ncn[0])
3      hm_ncn = []
4      for i in range(len(lm_ncn)):
5          add = 0
6          for j in range(len(lm_ncn[0])):
7              distance = euclidian(Y, lm_ncn[i][j])
8              if distance == 0:
9                  add += 0
10             else:
11                 add += (1/distance)
12         if add == 0:
13             harmonic_mean = 0
14         else:
15             harmonic_mean = (k/add)
16         hm_ncn.append(harmonic_mean)
17     return hm_ncn
    
```

Penjelasan Kode Program 5.13:

1. Baris ke – 1 inisialisasi fungsi dengan parameter *lm_ncn* yaitu nilai *local mean* dan *Y* merupakan fitur data uji.
2. Baris ke – 2 dan 3 berturut – turut merupakan inisialisasi variabel *k* yaitu untuk menyimpan nilai *k* dan *hm_ncn* untuk menyimpan nilai *harmonic mean*.
3. Baris ke – 4 sampai 16 merupakan perhitungan nilai *harmonic mean distance* dari *centroid* terdekat yang telah didapatkan.
4. Baris ke – 17 merupakan pengembalian variabel yang berisi nilai *harmonic mean distance*.



5.1.2.2.4 Menentukan Kelas

Tahapan implementasi Menentukan kelas merupakan tahapan terakhir pada metode LMKHNCN yang memiliki masukan dari keluaran tahapan *Harmonic Mean Distance* agar dapat menentukan kelas yang telah dijelaskan pada Bab 2.9. Berikut merupakan kode program dari tahapan menentukan kelas klasifikasi yang dapat dilihat pada Kode Program 5.14.

Kode Program 5.14 Menentukan Kelas

Algoritma: Menentukan Kelas

```
1 def decision(hm_ncn, Y_label):
2     min_distance_index = hm_ncn.index(min(x for x in
3         hm_ncn))
4     return (min_distance_index + 1), Y_label[0]
```

Penjelasan Kode Program 5.14:

1. Baris ke – 1 inialisasi fungsi dengan parameter *hm_ncn* yaitu nilai *harmonic mean* dan *Y_label* merupakan kelas data uji.
2. Baris ke – 2 merupakan menyimpan nilai terkecil dari array *hm_ncn*.
3. Baris ke – 3 merupakan pengembalian nilai kelas yaitu nilai kelas prediksi dan kelas aslinya.



BAB 6 HASIL DAN PEMBAHASAN

6.1 Pengujian Akurasi dan F1-Score

Pada tahap ini akan dilakukan pengujian sensitivitas parameter k dan pengujian perbandingan performa klasifikasi data setelah normalisasi dan tanpa normalisasi pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN). Pengujian ini dilakukan agar mengetahui apakah parameter k sensitif jika diubah nilainya dan mengetahui pengaruh performa klasifikasi untuk data menggunakan normalisasi dan tanpa normalisasi. Hasil pengujian akurasi untuk sensitivitas parameter k tanpa normalisasi dapat dilihat pada Tabel 6.1 dan hasil pengujian dengan normalisasi pada Tabel 6.2. Perhitungan akurasi adalah hasil akumulasi rata-rata dari 10-Fold Cross Validation diikuti dengan penyebaran rata-rata (deviasi standar).

Tabel 6.1 Hasil Pengujian Tanpa Normalisasi

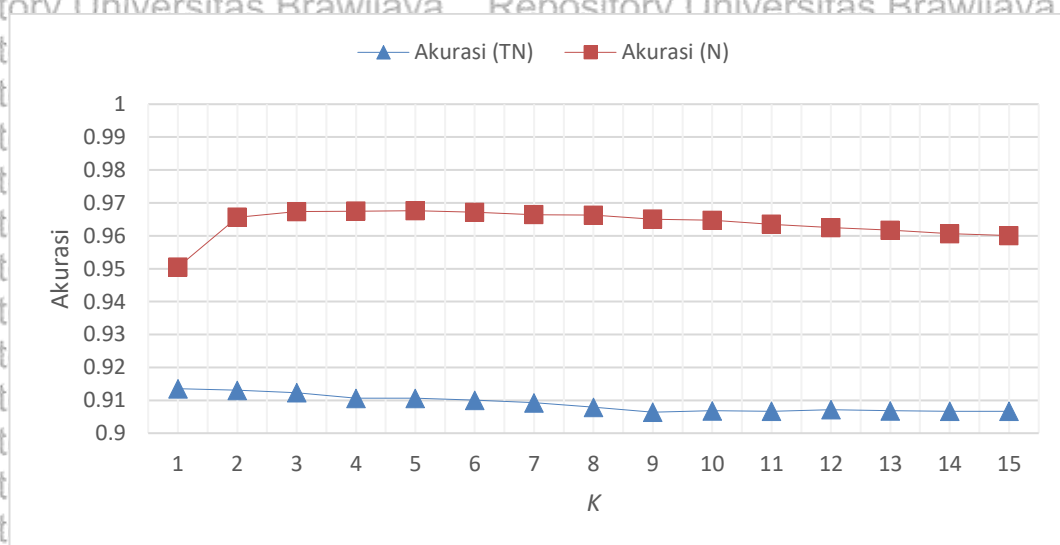
K	Akurasi	Deviasi Standar
1	0.913	0.0074
2	0.913	0.0053
3	0.912	0.0076
4	0.910	0.0073
5	0.910	0.0058
6	0.910	0.0059
7	0.909	0.0063
8	0.907	0.0061
9	0.906	0.0075
10	0.906	0.0074
11	0.906	0.0069
12	0.907	0.0066
13	0.906	0.0066
14	0.906	0.0064
15	0.906	0.0070



Tabel 6.2 Hasil Pengujian dengan Normalisasi

K	Akurasi	Deviasi Standar
1	0.950	0.0070
2	0.965	0.0039
3	0.967	0.0054
4	0.967	0.0045
5	0.967	0.0042
6	0.967	0.0037
7	0.966	0.0041
8	0.966	0.0050
9	0.965	0.0046
10	0.964	0.0046
11	0.963	0.0048
12	0.962	0.0054
13	0.961	0.0059
14	0.960	0.0058
15	0.960	0.0060

Hasil pengujian perbandingan akurasi metode LMKHNCN dari Tabel 6.1 dan pada Tabel 6.2 jika divisualisasikan dapat dilihat pada Gambar 6.1. (N) merupakan hasil dengan normalisasi dan (TN) merupakan hasil dari Tanpa Normalisasi.



Gambar 6.1 Grafik Hasil Pengujian Akurasi Sebelum dan Sesudah Normalisasi



Pada Gambar 6.1 di atas terlihat bahwa akurasi pada data yang dinormalisasi secara keseluruhan memiliki akurasi yang lebih tinggi dibandingkan dengan data yang tidak dinormalisasi. Hal ini dikarenakan normalisasi pada tahapan *preprocessing* akan menyamakan jarak setiap fiturnya sehingga fitur yang memiliki jarak yang jauh tidak mendominasi yang mengakibatkan kesalahan klasifikasi.

Selanjutnya pengujian performa dengan menggunakan *F1-Score*, pengujian *F1-Score* lebih baik digunakan untuk pengukuran performa dari data yang tidak seimbang (Suyanto, 2017). Hasil dari pengujian *F1-Score* tanpa normalisasi dan menggunakan normalisasi berturut – turut dapat dilihat pada Tabel 6.3 dan 6.4. Tabel tersebut akan terdapat perhitungan *Precision* dan *Recall* karena perhitungan tersebut digunakan untuk menghitung *F1-Score* seperti yang telah dibahas pada Bab 2.10.

Tabel 6.3 *F1-Score* Tanpa Normalisasi

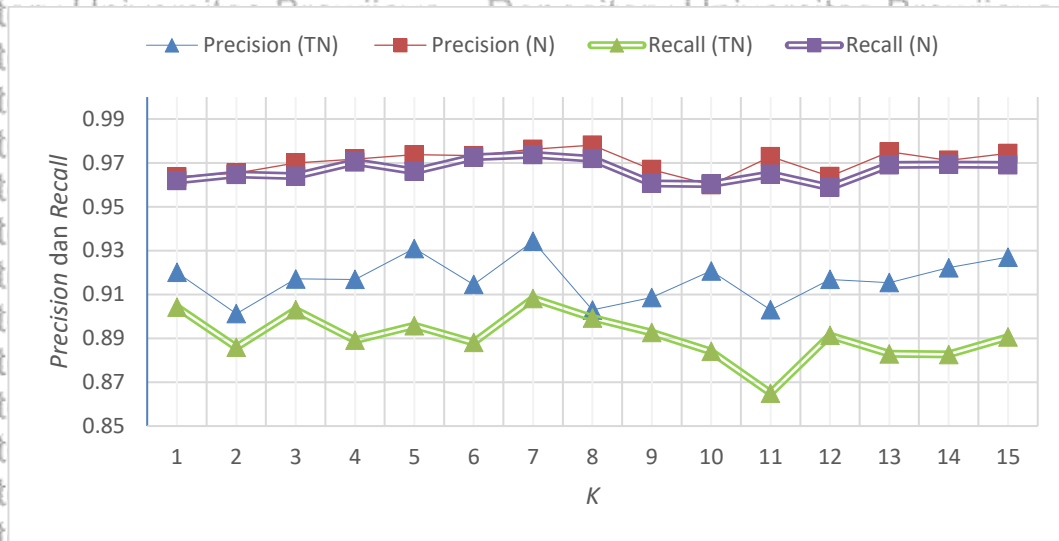
<i>K</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	0.920	0.904	0.911
2	0.901	0.885	0.893
3	0.917	0.902	0.909
4	0.916	0.889	0.901
5	0.930	0.895	0.911
6	0.914	0.888	0.899
7	0.934	0.908	0.920
8	0.902	0.899	0.901
9	0.908	0.892	0.900
10	0.920	0.884	0.900
11	0.902	0.865	0.881
12	0.916	0.891	0.902
13	0.915	0.882	0.897
14	0.922	0.882	0.900
15	0.927	0.890	0.906



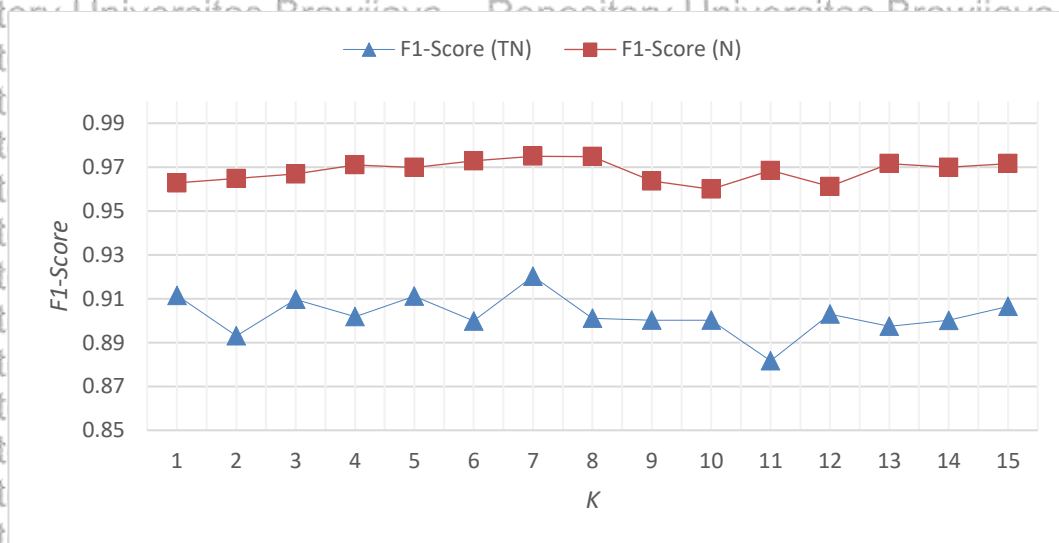
Tabel 6.4 F1-Score Menggunakan Normalisasi

K	Precision	Recall	F1-Score
1	0.963	0.961	0.962
2	0.965	0.964	0.964
3	0.969	0.963	0.966
4	0.971	0.970	0.971
5	0.973	0.966	0.969
6	0.973	0.972	0.972
7	0.976	0.973	0.974
8	0.978	0.971	0.974
9	0.966	0.960	0.963
10	0.959	0.960	0.959
11	0.972	0.964	0.968
12	0.963	0.958	0.961
13	0.975	0.969	0.971
14	0.971	0.969	0.969
15	0.974	0.969	0.971

Hasil pengujian *F1-Score* metode LMKHNCN dari Tabel 6.3 dan pada Tabel 6.4 jika divisualisasikan dapat dilihat pada Gambar 6.2 dan 6.3. (N) merupakan hasil dengan normalisasi dan (TN) merupakan hasil dari Tanpa Normalisasi.



Gambar 6.2 Grafik Hasil Pengujian *Precision* dan *Recall*



Gambar 6.3 Grafik Hasil Pengujian *F1-Score*

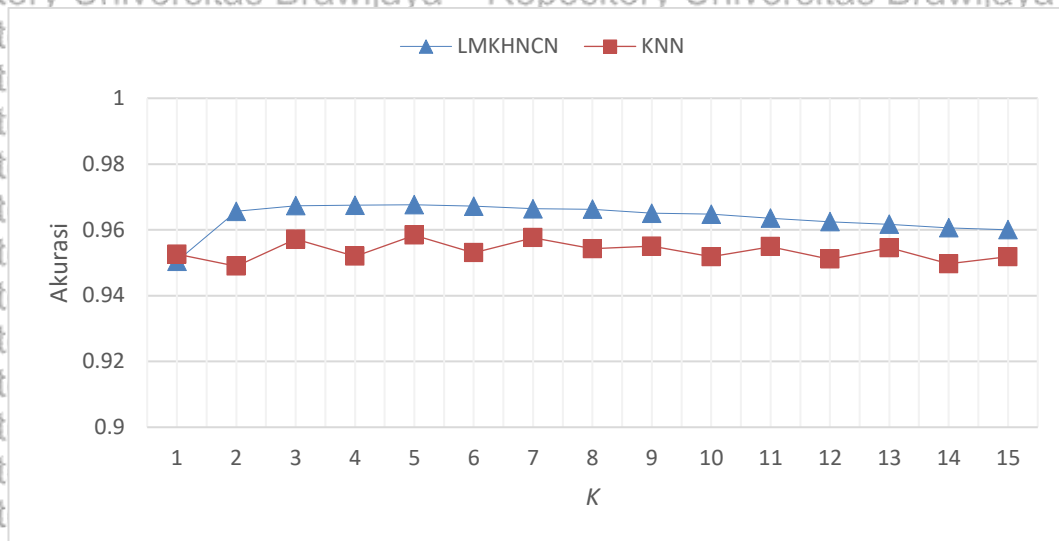
Pada Gambar 6.2 dan 6.3 di atas terlihat bahwa nilai *precision*, *recall*, *F1-Score* pada data yang dinormalisasi secara keseluruhan juga memiliki akurasi yang lebih tinggi dibandingkan dengan data yang tidak dinormalisasi. Hal ini dikarenakan normalisasi pada tahapan preprocessing akan menyamakan jarak setiap fiturnya sehingga fitur yang memiliki jarak yang jauh tidak mendominasi yang mengakibatkan kesalahan klasifikasi.

6.2 Perbandingan Performa Metode LMKHNCN dengan KNN

Pada tahap ini akan dilakukan pengujian perbandingan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan metode *K-Nearest Neighbor* yaitu metode aslinya. Pengujian ini dilakukan agar mengetahui apakah metode LMKHNCN dapat lebih baik mengklasifikasikan dibandingkan dengan metode aslinya yaitu KNN. Hasil pengujian akurasi dapat dilihat pada Tabel 6.5 dan hasil pengujian *F1-Score* pada tabel 6.6. Perhitungan akurasi adalah hasil akumulasi rata – rata dari *10-Fold Cross Validation* diikuti dengan penyebaran rata – rata (deviasi standar).

Tabel 6.5 Akurasi Metode KNN

K	Akurasi	Deviasi Standar
1	0.952	0.0050
2	0.948	0.0073
3	0.957	0.0063
4	0.951	0.0084
5	0.958	0.0076
6	0.953	0.0096
7	0.957	0.0076
8	0.954	0.0081
9	0.955	0.0084
10	0.951	0.0094
11	0.954	0.0091
12	0.951	0.0086
13	0.954	0.0063
14	0.949	0.0079
15	0.951	0.0063



Gambar 6.4 Grafik Hasil Pengujian Akurasi

Pada Gambar 6.4 di atas terlihat bahwa akurasi pada metode LMKHNCN hampir keseluruhan memiliki akurasi yang lebih tinggi dibandingkan dengan metode KNN. Hal ini dikarenakan proses dalam metode LMKHNCN tidak hanya mempertimbangkan kedekatan data juga mempertimbangkan tata letak data sehingga performa dari klasifikasi meningkat dibandingkan dengan metode aslinya KNN.

Selanjutnya pengujian performa dengan menggunakan *F1-Score* dapat dilihat pada Tabel 6.6.

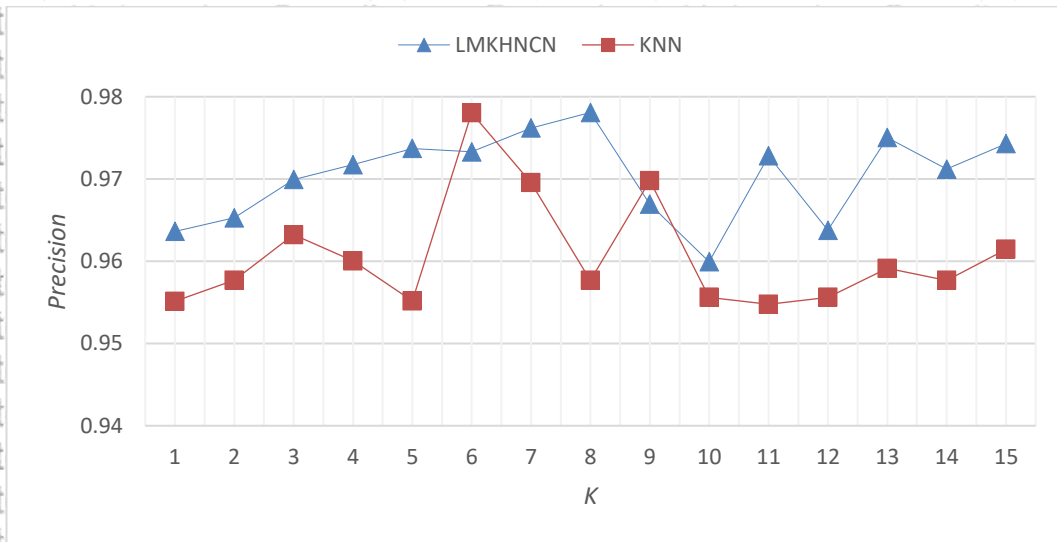
Tabel 6. 6 *F1-Score* Metode KNN

K	Precision	Recall	F1-Score
1	0.955	0.953	0.954
2	0.957	0.951	0.953
3	0.963	0.960	0.961
4	0.960	0.944	0.951
5	0.955	0.967	0.961
6	0.978	0.972	0.975
7	0.969	0.967	0.967
8	0.957	0.949	0.953
9	0.969	0.966	0.967
10	0.955	0.957	0.955
11	0.954	0.952	0.952
12	0.955	0.946	0.949

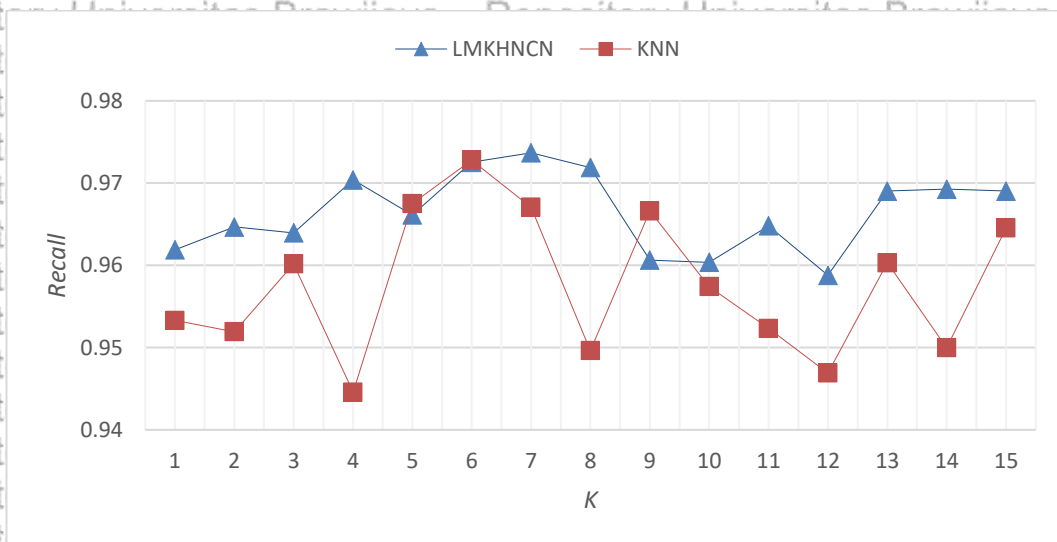


13	0.959	0.960	0.959
14	0.957	0.949	0.952
15	0.961	0.964	0.962

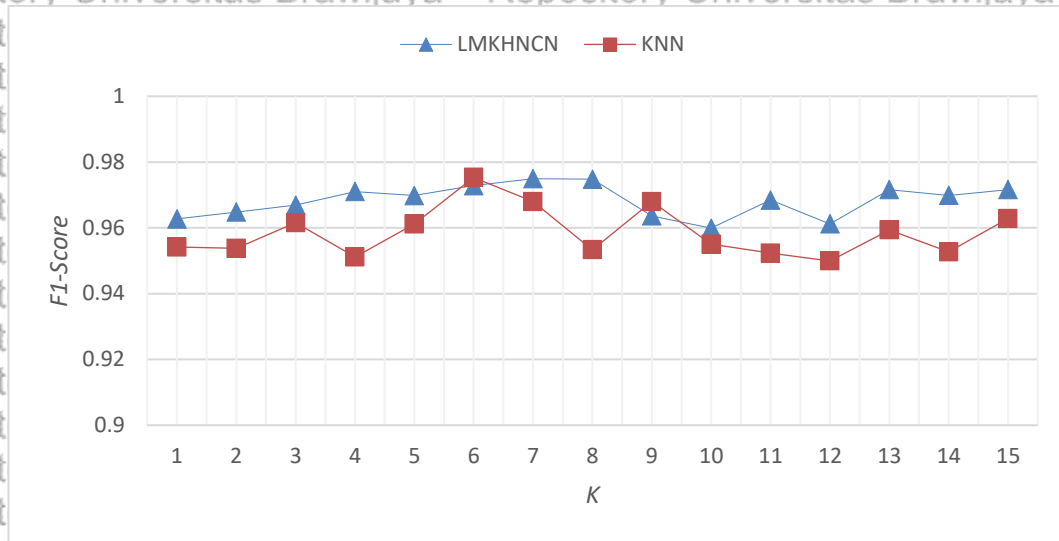
Hasil pengujian perbandingan *F1-Score* metode LMKHNCN dengan KNN dari Tabel 6.6 jika divisualisasikan dapat dilihat pada Gambar 6.5, 6.6 dan 6.7 berturut – turut merupakan *precision*, *recall* dan *F1-Score*.



Gambar 6.5 Grafik Hasil Pengujian *Precision*



Gambar 6. 6 Grafik Hasil Pengujian *Recall*



Gambar 6. 7 Grafik Hasil Pengujian *F1-Score*

Terlihat pada Gambar 6.5, 6.6, dan 6.7 bahwa metode LMKHNCN sebagian besar nilai *precision*, *recall* dan *F1-Score* dari setiap nilai *K* memiliki nilai yang lebih tinggi dibandingkan metode KNN. Tetapi, ada beberapa nilai *K* yang diungguli oleh metode KNN yaitu pada *K* bernilai 6 dan 9. Hal ini dikarenakan proses dalam metode LMKHNCN tidak hanya mempertimbangkan kedekatan data juga mempertimbangkan tata letak data sehingga performa dari klasifikasi meningkat dibandingkan dengan metode aslinya KNN. Dalam beberapa kasus pada *K* bernilai 6 dan 9 tetapi metode KNN lebih unggul tetapi terlihat metode LMKHNCN lebih mendominasi.



BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian penerapan metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) pada klasifikasi kinerja pegawai negeri sipil pemerintah kota malang sebagai berikut:

1. Penggunaan normalisasi pada saat *preprocessing* berpengaruh positif terhadap performa akurasi maupun *F1-Score* khususnya pada metode yang menggunakan perhitungan jarak pada algoritmenya. Hal ini dibuktikan pada Bab 6.1 pada Gambar 6.1 dan 6.2 performa akurasi dan *F1-Score* pada data yang menggunakan normalisasi secara keseluruhan nilai *K* memiliki performa yang lebih baik dibandingkan dengan data tanpa menggunakan normalisasi. Nilai akurasi dan *F1-Score* tertinggi tanpa normalisasi berturut – turut yakni 91,3% dan 91,1% sedangkan data yang menggunakan normalisasi nilai akurasi dan nilai *F1-Score* tertinggi berturut – turut adalah 96,7% dan 97,4%.
2. Kestabilan performa akurasi dan *F1-Score* pada model klasifikasi metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan metode aslinya *K-Nearest Neighbor* (KNN) pada perubahan parameter *K* pada kasus ini keduanya terlihat stabil. Terlihat pada Bab 6.2 pada Gambar 6.4 dan 6.7 perubahan nilai akurasi dan *F1-Score* pada keduanya tidak terlalu berubah secara signifikan (tidak berbeda jauh) setiap perubahan pada parameter *K*. Hal ini terjadi karena data pada kasus ini yang terdistribusi secara merata, sehingga performa model klasifikasi yang tinggi juga stabil.
3. Perbandingan performa akurasi pada model klasifikasi metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan metode aslinya *K-Nearest Neighbor* (KNN), metode LMKHNCN secara keseluruhan lebih baik dibandingkan dengan metode aslinya dengan nilai akurasi tertinggi berturut- turut 97,4% dan 95,7%. Sedangkan pada performa *F1-Score* model klasifikasi metode LMKHNCN sebagian besar nilai parameter *K*, tetapi metode KNN mendapatkan nilai *F1-Score* tertinggi pada nilai *K* sama dengan 6 dengan nilai *F1-Score* 97,5% dan *F1-Score* tertinggi LMKHNCN adalah 97,4% pada nilai *K* sama dengan 7. Hal ini dibuktikan pada Bab 6.2 Gambar 6.7.



7.2 Saran

Berdasarkan hasil dari pengujian dan analisis dari penelitian ini dari berbagai kelebihan tetap saja memiliki kekurangan, sehingga diperlukan saran yang diperlukan untuk memperbaiki kekurangan yang ada untuk penelitian selanjutnya. Saran yang diberikan untuk penelitian selanjutnya adalah:

1. Perbandingan pengujian sensitivitas parameter K pada metode *Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance* (LMKHNCN) dengan metode aslinya *K-Nearest Neighbor* (KNN) pada kasus ini masih belum terlihat karena data yang terdistribusi secara merata. Sehingga, disarankan untuk mencoba pengujian ini pada kasus lainnya untuk memperjelas apakah metode LMKHNCN lebih stabil dibandingkan dengan metode KNN.
2. Perlu dilakukan penelitian menggunakan model klasifikasi *Neural Network* karena *Neural Network* berbasis *data driven*, yang di mana data pada kasus ini tergolong banyak yakni kurang lebih 6000 data.



DAFTAR REFERENSI

- Aggarwal, C., 2015. *Data Mining: The Text Book*. Switzerland: Springer International Publishing.
- Akbar, M., 2018. Pengaruh Manajemen Talenta Terhadap Kinerja Karyawan melalui Retensi Karyawan pada Pt. Infomedia Nusantara di Bandung. [Skripsi] (Fakultas Ekonomi dan Bisnis, Manajemen, Universitas Pasundan, Bandung).
- Aula, H., 2019. Pengaruh Manajemen Talenta Terhadap Kinerja Karyawan melalui Retensi Karyawan Pada Pd. Bpr Bank Daerah Lamongan. *Jurnal Ilmu Manajemen*, 7(2).
- Chakure, A., 2019. *Towards Data Science*. [Online] Available at: <https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-bd375d14eec7> [Diakses 20 January 2020].
- Chaudhuri, 1996. A New Definition of Neighborhood of a Point in Multidimensional Space. *Pattern Recognition Letters*, Volume 17, pp. 11-17.
- Dwi, R., Budi, P. & Yudanintyas, E., 2018. Klasifikasi Kinerja Pegawai Universitas X dengan Pendekatan Neuro Fuzzy. *Jurnal EECCIS*, 12(1).
- Han, J., Micheline, K. & Pei, J., 2012. *Data Mining: Concept and Technique*. USA : Elsevier Inc.
- Putri, IR., Cholissodin, I., & Setiawan, BD., 2015, 'Optimasi Metode Adaptive Fuzzy K-Nearest Neighbor Dengan Particle Swarm Optimization Untuk Klasifikasi Status Sosial Ekonomi Keluarga', DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 3
- Jonason, K., 2016. *Strategic Talent Management*. Denmark: DK-2900 Hellerup.
- Lena, R., Dengo, S. & Ruru, J., 2016. Pengaruh Sistem Kenaikan Pangkat Terhadap Motivasi Kerja Pegawai Negeri Sipil di Kantor Pertahanan Kota Sorong. *Jurnal Administrasi Publik*, 40(3).
- Li, P., Gou, J. & Young, H., 2017. *The Distance-Weighted K-Nearest Centroid Neighbor Classification*. *Journal of Information Hiding and Multimedia Signal Processing*, 8(3).
- Lozano, M. & Sanchez, S., 2014. *Reducing Training Sets by NCN-based Exploratory Procedures*. LNCS, Volume 2652.
- Mangkunegara, P., 2000. *Manajemen Sumber Daya Manusia Perusahaan*. Bandung: Remaja Rosdakarya.
- Manliguez, C., 2016. *Generalized Confusion Matrix for Multiple Classes*. DOI: 10.13140/RG.2.2.31150.51523..



Mehta, S., Shen, X. & Gou, J., 2018. *A New Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance*. *Information*, 9(9), pp. 1-16.

Nafi, N., I. & Darma, B., 2017. Penerapan Metode K-Nearest Neighbor (KKN) dan Metode Weighted Product (WP) dalam Penerimaan Calon Guru dan Karyawan Tata Usaha Baru Berwawasan Teknologi (Studi Kasus: Sekolah Menengah Kejuruan Muhammadiyah 2 Kediri). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(5), pp. 378-385.

Raihan, M. & Permana, A., 2019. Penerapan Metode K-Nearest Neighbor untuk Klasifikasi Kinerja Satpam Berbasis Web. *Jurnal Teknologi dan Manajemen Informatika*, 5(1).

Rochman, 2017. Pengaruh Faktor Lingkungan Kerja, Usia Masa Pensiun, dan Tingkat Stress Kerja Terhadap Kinerja Pegawai di Perseroan Terbatas (Pt) Duta Ananda Textile. [Tesis] (Program Pasca Sarjana, Universitas Sebelas Maret, Surakarta).

Safitriani, 2018. Pengaruh Displin Kerja Terhadap Kinerja Pegawai pada Dinas Pendidikan Pemuda dan Olahraga di Kab. Jenepono. [Skripsi] (Fakultas Ekonomi dan Bisnis Islam, Manajemen, Universitas UIN Alauddin, Makassar).

Saga, A. & Tri, D., 2016. Pengaruh Penghargaan dan Motivasi Terhadap Kinerja Karyawan (Studi pada Divisi Penjualan PT. United Motors Center Suzuki Ahmad Yani, Surabaya). *Jurnal Ekonomi Bisnis dan Kewirausahaan*, 5(2).

Satria, N. & Gita, H., 2017. Pengaruh Manajemen Talenta terhadap Kinerja Karyawan Lintasarta Kota Jakarta. *Jurnal Manajemen Indonesia*, 17(3).

Septiani, D., 2017. *Pengaruh Kerjasama Tim Terhadap Kinerja Karyawan Pt Bio Farma*. [Skripsi] (Fakultas Komunikasi Bisnis, Administrasi, Bisnis, Universitas Telkom, Bandung).

Setiyorini, T. & Tri, R., 2018. Komparasi Metode Decision Tree, Naive Bayes dan K-Nearest Neighbor pada Klasifikasi Kinerja Siswa. *Jurnal TECHNO Nusa Mandiri*, 15(2).

Setiyorini, T. & Tri, R., 2019. Penerapan Metode K-Nearest Neighbor dan Information Gain pada Klasifikasi Kinerja Siswa. *Jurnal Ilmu Pengetahuan dan Teknologi komputer*, 5(1).

Shaikh, R., 2018. *Towards Data Science*. [Online] Available at: <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85> [Diakses 8 January 2020].

Suyanto, 2017. *Data Mining: untuk Klasifikasi dan Klasteriasi Data*. Bandung: Informatika.



Syaliman, U., Nababan, E. & Sitompul, O., 2017. *Improving The Accuracy Of K-Nearest Neighbor Using Local Mean Based And Distance Weight*. *Journal of Physics*, Volume 978, pp. 1-6.

Theodoridis, S. & Koutrombas, K., 2003. *Pattern Recognition Second Edition*. USA: Elsevier.

Witten, I. & Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Technique*. San Fransisco: Elsevier, inc.

Zaki, M. & Meira, W., 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. USA: Cambridge University Press.



DAFTAR LAMPIRAN PENGAMBILAN DATASET PEGAWAI

Pengambilan Dataset Saat Praktik Kerja Lapangan



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

FAKULTAS ILMU KOMPUTER

JURUSAN TEKNIK INFORMATIKA

Jl. Veteran No.8, Malang, 65145, Indonesia

Telp. : +62-341-577911; Fax : +62-341-577911

<http://jtif.fikom.ub.ac.id>

E-mail : jtif@ub.ac.id

PKL

2C

FORMULIR PENILAIAN PRAKTIK KERJA LAPANGAN (PKL) MAHASISWA

OLEH PEMBIMBING LAPANGAN

Nama/NIM : ADAM SYARIF HIDAYATULLAH / 165150200111152
 Program Studi : TEKNIK INFORMATIKA
 Waktu Pelaksanaan : 22/12/2019 s/d 22/02/2020
 Instansi/Perusahaan : BADAN KEPEGAWAIAN DAERAH KOTA MALANG
 Unit Kerja PKL : PENILAIAN KINERJA DAN PENGHARGAAN
 Alamat Instansi/Perusahaan : JL.TUGU NO.1 KIDUL DALEM, KLOJEN, MALANG

Kriteria Penilaian	Nilai (0 – 100)	Bobot Nilai (%)	Nilai x Bobot
a. Kehadiran, Disiplin, dan Etika	90	10	9.00
b. Pemahaman Masalah	90	5	4.50
c. Kerjasama Tim	85	10	8.50
d. Pengetahuan Komprehensif (teori dan praktik)	95	10	9.50
e. Laporan	90	5	4.50
Total			36.00

Catatan : Nilai dari Pembimbing Lapangan adalah 40% dari total nilai PKL

Catatan : Terapkan metode lain dalam obyek penelitian
 Sehingga hasil akan lebih akurat

Mengetahui,

Pimpinan Instansi,

A. M. KEPOL BKPSDM



SETIJO, SE

NIP/NIK 19620611 198703 1 014

Malang, 24/02/2020

Pembimbing Lapangan,

BAOUL WIDARAS, S.Kom

NIP/NIK 19730216 2001121 004

Catatan : Penilaian dianggap sah jika dibubuhi stempel basah dari Instansi penilai

Halaman Utama:

Daftar Pegawai

Home > Informasi > Daftar Pegawai

Identitas Prajabatan Pangkat Gaji Berkala Pendidikan Jabatan Disiplin Diklat Riwayat Lain Lain

Data Pribadi

Foto :  [Delete Foto](#) [Edit](#)

Nama Dan Gelar : [Edit](#)

Nama :

Jabatan dan Unit Kerja :

Jabatan :

Unit Kerja :

Data Pengangkatan CPNS [Edit](#)

Pejabat Pengangkatan :

Nomor SK :

Tanggal SK :

Golongan :

Masa Kerja :

TMT SK :

Gaji :

Catatan :

Memo [Edit](#)

Masa Kerja dan Gaji:

No	No Sk	Tanggal SK	TMT Gaji	Golongan	Action
1	822.3/147/428.125/91	15-03-1991	01-03-1991		Detail Edit Delete
2	822.3/702/428.125/92	23-11-1992	01-03-1993		Detail Edit Delete
3	822.3/1498/428.125/9				Detail Edit Delete
4	822.3/1068/428.131/9				Detail Edit Delete
5	822.3/1575/428.131/9				Detail Edit Delete
6	822.4/108/420.406/2003				Detail Edit Delete
7	822.4/119/420.403/2005				Detail Edit Delete
8	822.4/534/35.73.403/2006				Detail Edit Delete
9	822.4/594/35.73.403/2008				Detail Edit Delete
10	822.4/27/35.73.403/2011				Detail Edit Delete

Page 1 of 2 | View 10 records | Found total 11 records

Detail Gaji Berkala Pegawai

Pejabat Pengangkatan : Walikota / Bupati

Nomor SK : 822.4/27/35.73.403/2011

Tanggal SK : 10-01-2011

Golongan : III

Masa Kerja : 22 Thn 0 Bln

TMT SK : 10-01-2011

Gaji : 2.918.900

[Close](#)



Status Kawin:

Pangkat Golongan dan Ruang	
Pangkat	:
Gol/Ruang	:
Nomor Induk	Edit
Nip Baru	:
Nip Lama	:
Data Personal	Edit
Agama	:
Jenis Kelamin	:
KPE	:
Tanggal Lahir	:
Status Kawin	Kawin
Data Alamat	Edit
Alamat	:
Kelurahan	:
Telp Rumah	:
No Handphone	:

Catatan	:
---------	---

Karpeg Edit	
Karpeg	:

Keadaan Pegawai Delete Edit	
Status Pegawai	:
Jenis Pegawai	:
Keadaan Pegawai	:

Jumlah Kenaikan Pangkat:

Detail Pangkat Pegawai									
No	No Sk	Tanggal SK	TMT Golongan	Thn	Bln	Golongan	Angka Kredit	Jenis Pangkat	Action
1	813.2/7173/032/81	31-12-1981	01-01-1982	0	0	II/a		Pengangkatan CPNS	Detail Edit Del
2	4280/PN/C.5/P	30-11-1983	01-12-1983	0	0	II/a		Pengangkatan PNS	Detail Edit Del
3	00460/KEP/851/85	16-02-1985	01-04-1985	3	3	II/b		Pangkat Reguler	Detail Edit Del
4	00016/KEP/KR.II/88/5	29-03-1988	01-10-1987	0	9	III/a		Pangkat Reguler	Detail Edit Del
5	823.3/98/112.05/94	30-09-1994	01-04-1992	5	3	III/b		Pangkat Reguler	Detail Edit Del
6	823.3/97/112.05/96	31-07-1996	01-04-1996	9	3	III/c		Pangkat Reguler	Detail Edit Del
7	43858/A2.III.2/KP/00	21-07-2000	01-04-2000	13	3	III/d		Pangkat Reguler	Detail Edit Del
8	823.4/54/428.406/01	26-05-2001	01-04-2001	14	3	IV/a		Pangkat Pilihan	Detail Edit Del
9	823.4/345/042/2005	07-02-2005	01-04-2005	18	3	IV/b	0.000	Pangkat Pilihan	Detail Edit Del
10	57/K/ TAHUN 2009	21-07-2009	01-04-2009	22	3	IV/c	0.000	Pangkat Pilihan	Detail Edit Del

Jumlah Penghargaan:

Riwayat Penghargaan				
No	Keterangan	SK	Tahun	Action
1	Karya Satya 30 Tahun	1/TK/TAHUN 2019	2019	Edit Delete



Orientasi, Integritas, Disiplin dan Kinerja:

Orientasi Pelayanan	82.00	Baik
Integritas	82.00	Baik
Komitmen	85.00	Baik
Disiplin	83.00	Baik
Kerjasama	82.00	Baik
Kepemimpinan		
Jumlah	414	
Nilai Rata Rata	82.8	Baik



Nilai Prestasi Kerja	86.32	Baik
----------------------	-------	------



DAFTAR LAMPIRAN DATASET

Keterangan:

X1 = Masa Kerja, X2 = Gaji, X3 = Jumlah Kenaikan Pangkat,

X4 = Jumlah Penghargaan, X5 = Orientasi, X6 = Integritas, X7 = Disiplin

X8 = Kerja Sama, X9 = Kawin

Kelas:

1 = Cukup, 2 = Baik, 3 = Amat Baik

Link Dataset:

[https://github.com/adamsh231/skripsi/blob/master/Dataset%20Pegawai%20\(Processed\).xlsx](https://github.com/adamsh231/skripsi/blob/master/Dataset%20Pegawai%20(Processed).xlsx)

Id	X1	X2	X3	X4	X5	X6	X7	X8	X9	Kelas
1241	8	1717400	7	0	77	78	78	78	1	1
1242	8	1717400	7	0	82	79	72	80	1	1
1243	14	1695000	7	0	79	79	81	72	1	1
1244	14	1695000	7	0	76	77	76	75	1	1
1245	10	1686500	7	0	78	72	77	79	1	1
1246	10	1686500	7	0	77	81	77	80	3	1
1247	12	1656000	7	0	78	79	73	75	1	1
1248	12	1656000	7	0	81	72	82	75	1	1
1249	12	1656000	7	0	74	82	82	81	1	1
1250	12	1656000	7	0	78	72	82	81	3	1
1251	8	1647700	7	0	77	75	77	77	1	1
1252	10	1618000	7	0	74	81	76	78	1	1
1253	17	1588800	7	0	73	80	74	73	1	1
1254	8	1580900	7	0	76	77	77	80	1	1
1255	8	1580900	7	0	76	72	76	77	1	1
1256	8	1580900	7	0	73	73	76	77	1	1
1257	8	1580900	7	0	82	78	74	75	1	1
1258	8	1580900	7	0	81	80	74	72	1	1
1259	8	1580900	7	0	73	79	74	79	1	1



1260	19	1560200	7	0	75	77	73	72	1	1
2530	6	2810200	3	0	85	79	88	86	1	2
2531	21	2780400	3	0	82	79	74	79	1	2
2532	23	2751600	3	0	78	86	85	77	1	2
2533	4	2744500	3	0	85	80	81	84	1	2
2534	4	2744500	3	0	74	81	80	76	1	2
2535	4	2744500	3	0	78	82	77	84	2	2
2536	17	2743800	3	0	82	87	75	87	1	2
2537	17	2743800	3	0	83	88	83	88	1	2
2538	17	2743800	3	0	80	85	83	75	1	2
2539	17	2743800	3	0	78	81	89	85	1	2
2540	21	2667500	3	0	78	82	86	77	1	2
2541	21	2667500	3	0	85	87	75	76	1	2
2542	3	2660700	3	0	75	76	85	77	2	2
2543	15	2660000	3	0	87	75	75	74	1	2
2544	15	2660000	3	0	78	80	86	81	1	2
2545	15	2660000	3	0	87	79	78	78	1	2
2546	15	2660000	3	0	86	80	82	86	1	2
2547	15	2660000	3	0	81	75	78	74	1	2
2548	15	2640600	3	0	83	77	77	75	1	2
2549	15	2640600	3	0	81	86	79	83	1	2
6041	25	3237500	4	1	84	87	94	88	1	3
6042	12	3214700	4	1	90	84	87	97	3	3
6043	10	3139400	4	1	84	91	87	89	1	3
6044	10	3064400	4	1	93	84	88	88	2	3
6045	20	1147900	4	1	84	97	92	89	3	3
6046	8	2898700	4	1	92	87	83	85	1	3
6047	16	2722300	4	1	90	83	90	90	1	3
6048	6	2651100	4	1	96	88	84	85	1	3
6049	11	2586700	4	1	96	91	83	83	1	3
6050	11	2553100	4	1	92	96	97	89	1	3
6051	8	2475100	4	1	92	86	84	88	1	3
6052	21	2331400	4	1	87	85	95	93	1	3



6053	4	2326300	4	1	83	90	90	93	1	3
6054	4	2326300	4	1	90	83	89	84	1	3
6055	11	2208700	4	1	96	88	94	96	1	3
6056	21	2199400	4	1	93	92	84	90	1	3
6057	6	1177700	4	1	84	93	83	92	3	3
6058	9	2054400	4	1	91	85	87	85	1	3
6059	9	1971000	4	1	85	85	97	93	2	3
6060	9	1971000	4	1	96	88	88	95	1	3



DAFTAR LAMPIRAN HASIL AKURASI

Metode LMKHNCN (Tanpa Normalisasi)

K	Akurasi (10-Fold Cross Validation)										Deviasi Standar
	1	2	3	4	5	6	7	8	9	10	
1	0.916	0.914	0.910	0.920	0.927	0.910	0.917	0.898	0.910	0.908	0.0074
2	0.908	0.904	0.911	0.916	0.920	0.920	0.907	0.916	0.910	0.913	0.0053
3	0.917	0.908	0.904	0.899	0.913	0.916	0.925	0.905	0.920	0.910	0.0076
4	0.907	0.914	0.908	0.901	0.904	0.911	0.911	0.920	0.901	0.924	0.0073
5	0.917	0.907	0.914	0.910	0.902	0.902	0.913	0.913	0.919	0.904	0.0058
6	0.908	0.914	0.904	0.908	0.910	0.901	0.905	0.911	0.910	0.924	0.0059
7	0.924	0.905	0.913	0.907	0.911	0.910	0.890	0.905	0.907	0.908	0.0063
8	0.907	0.919	0.901	0.911	0.902	0.911	0.913	0.89	0.904	0.908	0.0061
9	0.905	0.907	0.919	0.89	0.911	0.902	0.910	0.913	0.89	0.902	0.0075
10	0.901	0.911	0.907	0.919	0.899	0.913	0.904	0.908	0.910	0.892	0.0075
11	0.895	0.898	0.911	0.907	0.917	0.898	0.910	0.907	0.908	0.911	0.0075
12	0.911	0.896	0.898	0.910	0.908	0.919	0.899	0.910	0.907	0.908	0.0066
13	0.907	0.914	0.896	0.898	0.911	0.908	0.916	0.898	0.908	0.907	0.0066

14	0.907	0.908	0.914	0.898	0.896	0.910	0.904	0.916	0.899	0.910	0.0064
15	0.911	0.907	0.908	0.916	0.896	0.896	0.910	0.902	0.916	0.899	0.0070

Metode LMKHNCN (Normalisasi)

K	Akurasi (10-Fold Cross Validation)										Deviasi Standar
	1	2	3	4	5	6	7	8	9	10	
1	0.954	0.943	0.948	0.957	0.949	0.957	0.958	0.954	0.943	0.936	0.0070
2	0.963	0.962	0.963	0.968	0.968	0.968	0.965	0.969	0.968	0.957	0.0039
3	0.960	0.963	0.968	0.955	0.972	0.972	0.966	0.972	0.969	0.971	0.0054
4	0.969	0.963	0.966	0.966	0.957	0.971	0.975	0.968	0.969	0.966	0.0045
5	0.965	0.969	0.965	0.968	0.968	0.957	0.969	0.974	0.971	0.968	0.0042
6	0.968	0.968	0.969	0.966	0.963	0.968	0.957	0.969	0.971	0.969	0.0037
7	0.969	0.966	0.966	0.971	0.965	0.965	0.968	0.955	0.965	0.971	0.0041
8	0.969	0.971	0.965	0.966	0.975	0.965	0.966	0.965	0.955	0.962	0.0050
9	0.958	0.963	0.969	0.965	0.966	0.974	0.966	0.966	0.962	0.957	0.0046
10	0.957	0.957	0.963	0.968	0.965	0.966	0.974	0.966	0.965	0.963	0.0046
11	0.962	0.954	0.957	0.962	0.969	0.963	0.966	0.971	0.965	0.963	0.0048
12	0.958	0.962	0.951	0.957	0.962	0.971	0.962	0.965	0.969	0.965	0.0054

13	0.965	0.958	0.955	0.949	0.957	0.963	0.969	0.962	0.965	0.969	0.0059
14	0.966	0.966	0.958	0.955	0.949	0.957	0.955	0.969	0.960	0.965	0.0058
15	0.965	0.968	0.966	0.955	0.955	0.949	0.955	0.957	0.968	0.957	0.0060

Metode KNN

REPOSITORY.UB.AC.ID

REPOSITORY.UB.AC.ID

K	Akurasi (10-Fold Cross Validation)										Deviasi Standar
	1	2	3	4	5	6	7	8	9	10	
1	0.949	0.949	0.954	0.943	0.960	0.957	0.958	0.946	0.952	0.951	0.0050
2	0.945	0.943	0.943	0.955	0.937	0.960	0.960	0.951	0.946	0.943	0.0073
3	0.960	0.951	0.955	0.962	0.963	0.942	0.963	0.962	0.955	0.954	0.0063
4	0.948	0.949	0.940	0.949	0.951	0.962	0.939	0.958	0.968	0.951	0.0084
5	0.955	0.963	0.962	0.949	0.957	0.960	0.963	0.943	0.954	0.972	0.0076
6	0.972	0.952	0.954	0.946	0.940	0.948	0.952	0.966	0.940	0.954	0.0096
7	0.962	0.972	0.958	0.955	0.957	0.942	0.952	0.958	0.963	0.951	0.0076
8	0.945	0.958	0.971	0.957	0.952	0.952	0.942	0.946	0.952	0.962	0.0081
9	0.962	0.942	0.960	0.971	0.957	0.952	0.958	0.943	0.946	0.954	0.0084
10	0.946	0.963	0.945	0.958	0.971	0.944	0.945	0.949	0.940	0.942	0.0094
11	0.945	0.955	0.965	0.943	0.957	0.974	0.957	0.948	0.955	0.945	0.0091



12	0.942	0.942	0.936	0.960	0.946	0.955	0.963	0.960	0.949	0.965	0.0086
13	0.955	0.945	0.949	0.949	0.963	0.948	0.957	0.965	0.958	0.951	0.0063
14	0.945	0.945	0.943	0.946	0.936	0.960	0.945	0.954	0.960	0.958	0.0079
15	0.955	0.949	0.949	0.946	0.948	0.943	0.962	0.943	0.957	0.960	0.0063



DAFTAR LAMPIRAN HASIL PENGUJIAN TERBAIK

Akurasi = 97.416%, F1-Score = 97.72%

Id	Kelas	Prediksi
1072	1	1
6023	3	3
2384	1	1
4868	2	2
4363	2	2
949	1	1
2777	2	1
861	1	2
4575	2	2
1092	1	1
5778	3	3
4537	2	2
4817	2	2
1899	1	1
4008	2	2
4820	2	2
4968	2	2
5233	2	2
6547	3	3
1041	1	1
2969	2	2
1076	1	1
5191	2	2
1710	1	1
3048	2	2
4385	2	2
3812	2	2
1078	1	1
4750	2	2



1922	1	1
6066	3	3
4393	2	2
1533	1	1
2002	1	1
5834	3	3
1767	1	1
928	1	1
3377	2	2
4277	2	2
1030	1	1
6391	3	3
433	1	1
1527	1	1
484	1	1
6454	3	3
2533	2	2
179	1	1
2069	1	1
520	1	1
4131	2	2
5079	2	2
4733	2	2
2624	2	2
1294	1	1
472	1	1
4299	2	2
2900	2	2
4589	2	2
4333	2	2
5304	2	2
2670	2	2
4794	2	2



6555	3	3
5137	2	2
1612	1	1
13	1	1
6516	3	3
1278	1	1
5565	2	2
5203	2	2
824	1	1
6183	3	3
2092	1	1
4801	2	2
1088	1	1
3471	2	2
5228	2	2
4855	2	2
293	1	1
4100	2	2
4961	2	2
5122	2	2
4985	2	2
4938	2	2
5585	2	2
6075	3	2
4380	2	2
2672	2	2
2840	2	2
3863	2	2
1309	1	1
883	1	1
5375	2	2
816	1	1
4498	2	2



1188	1	2
2127	1	1
1020	1	1
2187	1	1
2578	2	2
56	1	1
2714	2	2
5310	2	2
262	1	1
1848	1	1
4731	2	2
4095	2	2
1987	1	1
4418	2	2
5923	3	3
6301	3	3
6279	3	3
5642	2	2
314	1	1
1096	1	1
5796	3	3
476	1	1
2207	1	1
4170	2	2
3821	2	2
4407	2	2
2763	2	2
2229	1	1
2587	2	2
6278	3	3
5766	3	3
6429	3	3
4977	2	2



1508	1	1
714	1	1
5501	2	2
1214	1	1
77	1	1
4682	2	2
6509	3	3
5215	2	2
1930	1	1
892	1	1
3178	2	2
5067	2	2
4967	2	2
4069	2	2
2863	2	2
2739	2	2
152	1	2
3946	2	2
5594	2	2
440	1	1
397	1	2
5135	2	2
3429	2	2
4102	2	1
204	1	1
626	1	1
4912	2	2
2455	1	1
3209	2	2
510	1	1
3918	2	2
403	1	2
4314	2	2



4053	2	2
2535	2	2
4972	2	2
4724	2	2
5979	3	3
527	1	1
5080	2	2
3378	2	1
4006	2	2
477	1	1
2853	2	2
5139	2	2
4931	2	2
5131	2	2
131	1	1
3710	2	2
3404	2	2
6253	3	3
5454	2	2
1549	1	1
2561	2	2
4637	2	2
2534	2	2
96	1	1
6317	3	3
2633	2	2
5468	2	2
3379	2	2
223	1	1
5571	2	2
1834	1	1
3875	2	2
4015	1	1



5691	2	2
5762	3	3
2813	2	2
1365	1	1
1054	1	1
723	1	1
1391	1	1
3420	2	2
773	1	1
3307	2	2
4134	2	2
5718	2	2
289	1	1
2674	2	2
4369	2	2
6363	3	3
3890	2	2
3314	2	2
2112	1	1
6129	3	3
2902	2	2
3199	2	2
3551	2	2
2819	2	2
2431	1	1
3922	2	2
5589	2	2
3095	2	2
4120	2	2
266	1	1
5416	2	2
2410	1	1
2783	2	2



544	1	1
3562	2	2
3523	2	2
5247	2	2
381	1	2
5235	2	2
3832	2	2
1023	1	1
1141	1	1
3984	2	2
5549	2	2
5023	2	2
3688	2	2
6254	3	3
3207	2	2
611	1	1
2279	1	1
3311	2	2
6470	3	3
1310	1	1
4375	2	2
4599	2	2
1354	1	1
1151	1	1
83	1	1
1147	1	1
2791	2	2
2747	2	2
3660	2	2
2369	1	1
6179	3	3
8	1	1
1399	1	1



2151	1	1
5965	3	3
243	1	1
4270	2	2
4384	2	2
3225	2	2
4267	2	2
4545	2	2
6145	3	3
1485	1	1
4812	2	2
5390	2	2
1347	1	1
5640	2	2
5746	2	2
1116	1	1
1705	1	1
5433	2	2
5292	2	2
6522	3	3
4932	2	2
399	1	2
2226	1	1
5788	3	3
6570	3	3
5395	2	2
5802	3	3
1011	1	1
1377	1	1
4495	2	2
2335	1	1
5231	2	2
1320	1	1



5712	2	2
4413	2	2
4866	2	2
189	1	1
3189	2	2
170	1	1
5503	2	2
6037	3	3
3040	2	2
3502	2	1
2833	2	2
3891	2	2
2078	1	1
1959	1	1
4161	2	2
760	1	1
1593	1	1
2377	1	1
3780	2	2
4631	2	2
2741	2	2
663	1	1
761	1	1
2459	1	1
5180	2	2
345	1	1
5333	2	2
3743	2	2
1700	1	1
4032	2	2
2835	2	2
2584	2	2
5739	2	2



1224	1	1
5202	2	2
6124	3	3
2984	2	2
3195	2	2
1067	1	1
3608	2	2
4597	2	2
1948	1	1
3295	2	2
4026	2	2
3477	2	2
2044	1	1
3592	2	2
1517	1	1
317	1	1
3556	2	2
76	1	1
3435	2	2
814	1	1
921	1	1
4903	2	2
4644	2	2
5837	3	3
1779	1	1
1539	1	1
2420	1	1
5060	2	2
4048	2	2
3769	2	2
2728	2	2
1471	1	1
2074	1	1



1604	1	1
5986	3	3
531	1	1
6202	3	3
2933	2	2
3902	2	2
1019	1	1
6485	3	3
6428	3	3
6315	3	3
5452	2	2
5406	2	2
1384	1	1
1480	1	1
5740	2	2
915	1	1
4990	2	2
4347	2	2
4024	2	2
2140	1	1
5265	2	2
1297	1	1
365	1	1
2663	2	2
4878	2	2
235	1	1
3284	2	2
6474	3	3
1274	1	1
2048	1	1
499	1	1
6403	3	3
5758	3	3



2291	1	1
5619	2	2
4614	2	2
3590	2	2
1279	1	1
2194	1	1
5855	3	3
4265	2	2
4399	2	2
3599	2	2
5584	2	2
5179	2	2
2079	1	1
5383	2	2
6451	3	3
3835	2	2
4580	2	2
6031	3	3
453	1	1
5243	2	2
3270	2	2
340	1	1
4439	2	2
5705	2	2
3465	2	2
5093	2	2
3594	2	2
5367	2	2
4132	2	1
1567	1	1
4298	2	2
2826	2	2
2931	2	2



3774	2	2
3220	2	2
832	1	1
132	1	1
5132	2	2
2186	1	1
116	1	1
2163	1	1
2327	1	1
57	1	1
1868	1	1
2454	1	1
6047	3	3
3654	2	1
5558	2	2
4671	2	2
6158	3	3
719	1	1
2588	2	2
5528	2	2
591	1	1
4412	2	2
2746	2	2
6015	3	3
5189	2	2
6514	3	3
2021	1	1
1532	1	1
1086	1	1
296	1	1
3741	2	2
4993	2	2
3942	2	2



3631	2	2
4738	2	2
1049	1	1
1335	1	1
3444	2	2
1337	1	1
6537	3	3
3838	2	2
3204	2	2
2233	1	1
2596	2	2
3762	2	2
4917	2	2
6143	3	3
669	1	1
375	1	1
5955	3	3
66	1	1
3864	2	2
127	1	1
2765	2	2
1029	1	1
6336	3	3
94	1	1
1140	1	1
2983	2	2
1770	1	1
2461	1	1
1040	1	1
4900	2	2
3569	2	2
6508	3	3
4215	2	2



423	1	1
794	1	1
1053	1	1
2873	2	2
1257	1	1
2146	1	1
2278	1	1
1866	1	1
4220	2	2
4452	2	2
5541	2	2
2643	2	2
1596	1	1
2752	2	2
1474	1	1
6092	3	3
1316	1	1
1885	1	1
4807	2	2
2326	1	1
142	1	1
5883	3	3
2530	2	2
5747	3	3
434	1	1
4035	2	2
847	1	1
6054	3	2
3879	2	2
1392	1	1
310	1	1
2830	2	2
385	1	1



4660	2	2
830	1	1
6493	3	3
2556	2	2
4309	2	2
461	1	1
3852	2	2
4526	2	2
5990	3	3
5764	3	3
5288	2	2
2766	2	2
2161	1	1
1045	1	1
1603	1	1
2483	1	1
4305	2	2
4587	2	2
2991	2	2
4525	2	2
415	1	1
388	1	2
27	1	1
63	1	1
6065	3	3
4023	2	2
6297	3	3
1173	1	1
5017	2	2
5469	2	2
5035	2	2
5489	2	2
4669	2	2



4165	2	2
3797	2	2
4663	2	2
820	1	1
5505	2	2
6311	3	3
3294	2	2
2757	2	2
1629	1	1
3601	2	2
1520	1	1
2258	1	1
1681	1	1
6357	3	3
6180	3	3
1557	1	1
1124	1	1
2075	1	1
4424	2	2
6478	3	3
5995	3	3
3793	2	2
5976	3	3
2648	2	2
1907	1	1
5047	2	2
1949	1	1
4510	2	2
2116	1	1
6173	3	3
1143	1	1
4755	2	2
1	1	1



1071	1	1
5575	2	2
1207	1	1
2228	1	1
2344	1	1
4674	2	2
5905	3	3
2560	2	2
2162	1	1
3154	2	2
647	1	1
2009	1	1
274	1	1
5431	2	2
2938	2	2
1016	1	1
2290	1	1
5478	2	2
4494	2	2
704	1	1
2358	1	1
938	1	1
5472	2	2
144	1	1
2305	1	1
3730	2	2
3911	2	2
4136	2	2
4383	2	2
2859	2	2
1938	1	1
5617	2	2
2805	2	2



215	1	1
3527	2	2
6271	3	3
1236	1	1
6559	3	3
2113	1	1
4535	2	2
1829	1	1
2235	1	1
2234	1	1
90	1	1
4677	2	2
4116	2	2
786	1	1
2525	2	2
4411	2	2
5579	2	2
1524	1	1
630	1	1
5573	2	2
3468	2	2
5936	3	3
6472	3	3
1780	1	1
1281	1	1
3034	2	2
4774	2	2
2241	1	1
1103	1	1
181	1	2
2490	1	1
5174	2	2