LEMBAR PENGESAHAN SKRIPSI

Optimasi Metode Penanganan *Missing Values* pada *Incomplete Data Training*

Oleh:

DURRATUN NAFISAH
0610960020-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 18 Januari 2011

dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I	Pembimbing II
Bayu Rahayudi, ST, MT.	Candra Dewi, S.Kom., M.Sc
NIP. 197407122006041001	NIP. 197711142003122001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc.
NIP. 19670907 199203 1 001


UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	:	Durratun Nafisah
NIM	:	0610960020-96
Jurusan	:	Matematika
Program Studi	:	Ilmu Komputer
Penulis Skripsi berjudul	:	Optimasi Metode Penanganan *Missing Values* pada *Incomplete Data Training*

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub dalam isi dan tertulis pada daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 18 Januari 2011
Yang menyatakan,

Durratun Nafisah
NIM. 0610960020-96

UNIVERSITAS BRAWIJAYA

OPTIMASI METODE PENANGANAN *MISSING VALUES* PADA *INCOMPLETE DATA TRAINING*

ABSTRAK

Dalam pengolahan data diharapkan dapat menghasilkan informasi yang akurat. Namun dalam kenyataannya seringkali data tidak lengkap karena terdapat beberapa data yang kosong. Hal tersebut tentu mempengaruhi keakuratan informasi yang dihasilkan dalam pengolahan data. Oleh karena itu perlu dilakukan upaya untuk menjaga agar informasi yang dihasilkan dari pengolahan data yang tidak lengkap tersebut tetap memiliki keakuratan yang tinggi. Salah satu upaya yang dapat dilakukan adalah melakukan penanganan terhadap data-data yang hilang *(missing value)*.

Penanganan *missing value* dapat dilakukan dengan beberapa metode. Dalam skripsi ini digunakan metode *Mean Imputation* dan *KNN Imputation* untuk diimplementasikan pada data bertipe kuantitatif. Selain itu, metode yang digunakan adalah *Mode Imputation* dan *KNN Imputation* untuk diimplementasikan pada data bertipe kualitatif.

Nilai yang dihasilkan dari metode penanganan *missing value* selanjutnya dibandingkan dengan nilai sebenarnya dari data tersebut. Dari proses tersebut dapat diperoleh nilai keakuratan dari hasil penanganan. Hasil dari penelitian ini menunjukkan bahwa jumlah *record* dan jumlah *missing value* cukup mempengaruhi kinerja dari masing-masing metode. Pada tipe data kuantitatif, metode penanganan *Mean Imputation* sedikit lebih unggul dari *KNN Imputation* dengan rata-rata tingkat kesalahan sebesar 20.83%. Sedangkan pada tipe data kualitatif, metode penanganan *KNN Imputation* lebih unggul dari *Mode Imputation* dengan rata-rata tingkat kesalahan sebesar 2.02%.

OPTIMIZATION METHOD OF HANDLING MISSING VALUES ON INCOMPLETE TRAINING DATA

ABSTRACT

In the data processing is expected to produce accurate information. But in reality the data is often incomplete because there are some data that is missing. It affects the accuracy of the information generated in the processing of data. Therefore efforts are needed to keep the information generated from incomplete data processing still maintains high accuracy. One of the effort that can be done is handling the missing value.

Handling missing value can be done with several methods. In this paper Mean Imputation and KNN Imputation methods were implemented in quantitative data types. In addition, Mode Imputation and KNN Imputation methods were implemented in qualitative data types.

The generated value from the method of handling missing value then compared with the actual value of the data. From that process, accuracy value of handling result can be obtained. The result of this research indicate that the number of records and the number of missing value is affecting the performance of each method. In quantitative data types, methods of handling the Mean Imputation slightly superior to the KNN Imputation with an average error rate of 20,83%. While the qualitative data types, KNN Imputation treatment method is superior to Modus Imputation with an average error rate of 2,02%.

KATA PENGANTAR

Alhamdulillahi rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala Rahmat dan limpahan Hidayahnya, Tugas akhir yang berjudul "Optimasi Metode Penanganan *Missing Values* pada *Incomplete Data Training*" ini dapat terselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW, makhluk paling mulia yang senantiasa memberikan cahaya petunjuk, dan atas keluarganya dan sahabat-sahabatnya..

Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Bayu Rahayudi, ST, MT dan Candra Dewi, S.Kom, M.Sc. selaku dosen pembimbing skripsi, terima kasih atas semua saran, bantuan, kritikan, waktu, dorongan semangat dan bimbingannya
2. Nurul Hidayat, S.Pd., M.Sc. selaku dosen pembimbing akademik yang telah memberikan saran akademik, dorongan semangat dan bimbingannya
3. Drs. Marji, MT selaku Ketua Program Studi Ilmu Komputer Universitas Brawijaya Malang
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis
5. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya
6. Bapak, Ibu, kakak-kakak serta keluarga besar tercinta yang telah memberikan kasih sayang, doa yang tiada henti, serta dukungan dalam hidupku
7. Sahabat-sahabatku yang telah memberikan inspirasi, menemani hari-hariku dan berbagi banyak hal. Terima kasih telah membuatku mengetahui banyak tentang kota kelahiran tercinta ini lewat perjalanan kuliner dan wisata bersama

8. Segenap rekan-rekan Babi Inside 06 di bawah pimpinan bapak kating yang terakhir dan selamanya (Obi). Terima kasih untuk semangat kekeluargaan yang kalian berikan. Bangga bisa bertemu dan melewati moment-moment indah bersama kalian

Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan di masa mendatang.

Malang, Januari 2011

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pengolahan data dilakukan dengan tujuan untuk mendapatkan informasi dari suatu kumpulan data(*dataset*). Kelengkapan data tentu saja menjadi faktor utama dalam mendapatkan informasi yang sesuai dengan data yang ada. Namun ketika suatu *dataset* yang akan digunakan sebagai *data training* memiliki nilai yang hilang *(missing value),* maka hal tersebut dapat mempengaruhi informasi yang dihasilkan oleh suatu pengolah data (Edgar Acuna dan Caroline Rodriguez,2003). Oleh karena itu diperlukan suatu metode handal yang menghasilkan nilai pengganti yang akurat untuk menggantikan nilai yang hilang tersebut, sehingga *data training* dapat digunakan untuk melakukan pengolahan data secara tepat.

Menurut Iffat A.Gheyas dan Leslie S.Smith(2009), *missing value* merupakan hal yang biasa terdapat pada *dataset*. *Missing value* pada *dataset* didefinisikan sebagai kekosongan nilai dari variabel tertentu pada *dataset*. Banyak dari aplikasi pada dunia nyata yang mengalami kehilangan data atau tidak diketahui. Sebagi contoh dalam hasil percobaan industri, beberapa data dapat hilang karena kesalahan mekanik/elektronik selama proses perolehan data (Laksminarayan dkk, 2004; Nguyen dkk, 2003). Sebagian besar algoritma *data mining* tidak dapat bekerja secara langsung dengan *dataset* yang tidak lengkap. Pada penelitian yang dilakukan oleh Edgar Acuna dan Caroline Rodriguez (2003) dihasilkan pengaruh nilai prosentase *missing value* terhadap hasil pengolahan data. Data yang digunakan dalam penelitian tersebut adalah data klasifikasi pasien jantung yang pernah dirawat di rumah sakit. Dalam data tersebut pasien jantung diklasifikasikan menjadi 2 golongan, yaitu pasien yang mengalami hal mengerikan setelah dirawat di rumah sakit dan pasien yang tidak mengalami hal mengerikan setelah dirawat di rumah sakit. Pada tingkat *missing data* yang kurang dari 1% dapat dianggap tidak mempunyai pengaruh yang besar. Untuk *missing data* yang mencapai 1-5% masih dapat dikendalikan. Sedangkan untuk *missing data* yang mencapai 5-15% diperlukan metode yang handal untuk menanganinya. Dan untuk *missing data* yang mencapai lebih dari 15% diperkirakan dapat mempengaruhi berbagai perkiraan yang

didasarkan pada data tersebut. Oleh karena itu diperlukan suatu metode penanganan untuk diterapkan pada data dengan *missing value* seperti data pasien jantung tersebut.

Dalam skripsi ini dilakukan penelitian tentang penanganan *missing value* terhadap data dengan jumlah *missing value* antara 5-15% dengan menggunakan tiga metode penanganan *missing value*. Tiga metode tersebut adalah *Mean Imputation* (MI), *Mode Imputation* (MOI), dan *K-Nearest Neighbor Imputation* (KNNI). *Imputation* (teknik imputasi) dapat menggantikan *missing value* dengan nilai perkiraan yang didasarkan pada informasi yang terdapat pada *dataset*. Namun untuk menguji kekuatan metode ketika jumlah *missing value* melebihi 15%, maka dilakukan penelitian pula pada data dengan jumlah *missing value* sebesar 30%.

Mean Imputation diimplementasikan pada atribut bertipe data numerik. *Mean Imputation* terdiri dari tahapan menggantikan nilai/data yang hilang pada sebuah atribut dengan nilai rata-rata dari semua nilai yang diketahui dari atribut tersebut, yang berada pada kelas yang sama dengan *record* yang memiliki *missing value* tersebut. MI telah memberikan hasil uji coba yang bagus untuk melakukan klasifikasi *supervised* suatu *dataset* (P. Chan dan O.J.Dunn, 1972). Selain itu, berdasarkan pada jurnal hasil penelitian D.J.Mundfrom dan A.Whitcomb (1998) menunjukkan bahwa metode *Mean Imputation* menghasilkan tingkat akurasi klasifikasi yang lebih tinggi daripada metode *Regression* dan *Hot-Deck*. Tingkat akurasi klasifikasi yang dihasilkan oleh metode penanganan *missing value* pada penelitian tersebut adalah sebesar 74.4%, sedangkan yang dihasilkan dari metode *Regression* adalah sebesar 72%, dan dengan metode *Hot-Deck* dihasilkan tingkat akurasi sebesar 73.1%. Pada kasus *missing value* dengan atribut bertipe data *kualitatif* dapat digunakan *Mode Imputation* sebagai pengganti imputasi rata-rata.
KNNI menangani *missing value* pada suatu data dengan melakukan imputasi dengan mempertimbangkan nilai yang diberikan oleh *record* yang paling mirip. Kesamaan dari 2 *record* ditentukan dengan fungsi jarak. Menurut jurnal hasil penelitian yang dilakukan oleh Olga Troyanskaya, dkk. (2001), *KNNI* merupakan metode yang akurat untuk memperkirakan *missing value* pada data *Microarray*.

Penelitian tentang metode-metode penanganan *missing value* pernah dilakukan oleh Gustavo E.A.P.A. Batista dan Maria Carolina Monard dalam jurnal *An Analysis of Four Missing Data Treatment*

Methods for Supervised Learning, Edgar Acuna dan Caroline Rodriguez dalam jurnal *The treatment of missing values and its effect in the classifier accuracy*, serta Daniel J.Mundfrom dan Alan Whitcomb dalam jurnal *Imputing Missing Values : The Effect on the Accuracy of Classification*. Namun dalam penelitian yang telah dilakukan tersebut kurang difokuskan pada penelitian metode penanganan *missing value* yang optimal untuk tipe data kualitatif ataupun kuantitatif. Oleh karena itu, dalam skripsi ini dilakukan uji coba terhadap *missing value* yang terdapat dalam data yang bertipe kuantitatif serta kualitatif yang ditangani dengan menggunakan metode-metode yang memiliki tingkat keakuratan yang cukup tinggi seperti MI, MOI dan KNNI.

Berdasarkan latar belakang yang telah diuraikan, maka skripsi ini diberi judul " **Optimasi Metode Penanganan *Missing Values* pada *Incomplete Data Training***".

1.2 Rumusan Masalah

Rumusan masalah dalam skripsi ini adalah :

1. Bagaimana pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value Mean Imputation*
2. Bagaimana pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value Mode Imputation*
3. Bagaimana pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value KNN Imputation*
4. Metode penanganan *missing value* apa yang paling optimal untuk diimplementasikan pada tipe data kualitatif atau kuantitatif

1.3 Batasan Masalah

Batasan masalah pada skripsi ini adalah :

1. Peletakan missing value pada data training dilakukan tanpa mempertimbangkan sebaran statistika, namun berdasarkan pada keterangan atau pola missing value pada dataset
2. Metode penanganan *missing value* yang digunakan pada penelitian adalah *Mean Imputation, Mode Imputation* dan *KNN Imputation*

3. Pengujian difokuskan terhadap tipe data kualitatif dan kuantitatif
4. Algoritma penanganan *missing value Mean Imputation* diterapkan pada *dataset* dengan tipe kuantitatif, sedangkan algoritma *Mode Imputation* diterapkan dalam *dataset* bertipe kualitatif

1.4 Tujuan

Tujuan dalam skripsi ini adalah :

1 Mengetahui pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value Mean Imputation*

2 Mengetahui pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value Mode Imputation*

3 Mengetahui pengaruh jumlah *record* dan prosentase *missing value* terhadap keakuratan nilai prediksi yang dihasilkan dari metode penanganan *missing value KNN Imputation*

4 Mengetahui metode penanganan *missing value* yang paling optimal untuk diimplementasikan pada tipe data kualitatif atau kuantitatif

1.5 Manfaat

Hasil penelitian pada skripsi ini adalah untuk menangani nilai yang hilang pada suatu data dengan nilai pengganti yang paling mendekati sehingga data kembali utuh dan dapat digunakan untuk proses selanjutnya, seperti pada klasifikasi

1.6 Metode Penelitian

Metode penelitian yang dilakukan pada penelitian ini adalah :

1. Studi Literatur
   Mempelajari dan mengkaji beberapa literatur (jurnal, buku, dan artikel dari *website*) mengenai *data mining*, algoritma penanganan *missing value*, serta metode untuk mengevaluasi suatu nilai hasil perhitungan
2. Perancangan dan implementasi sistem
   Mengimplementasikan algoritma penanganan *missing value* seperti *Mean Imputation* (MI), *Mode Imputation* (MOI), dan *KNN Imputation* (KNNI). Pengimplementasian dilakukan

dengan merancang dan membangun sebuah perangkat lunak untuk menangani *dataset* yang memiliki *missing value*

3. Uji coba dan analisa hasil implementasi
   Menangani *missing value* pada *data training* dengan suatu nilai hasil dari metode penanganan *missing value*. Analisa ditentukan dengan metode evaluasi *Root Mean Squared Error* (RMSE) dan *Performance Metric Error Rate*

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut:

1. **BAB I PENDAHULUAN**
   Berisi latar belakang penelitian, perumusan masalah, batasan masalah, tujuan penelitian, manfaat, metode penelitian, dan sistematika penulisan
2. **BAB II TINJAUAN PUSTAKA**
   Bab ini berisi teori-teori dari berbagai pustaka yang menunjang penelitian dalam penulisian skripsi. Adapun teori yang tercakup dalam bab ini yaitu mengenai definisi dan konsep *data mining*, *missing value*, algoritma penanganan *missing value*, serta metode evaluasi
3. **BAB III METODOLOGI DAN PERANCANGAN SISTEM**
   Bab ini berisi mengenai perancangan perangkat lunak yang dibangun, meliputi perancangan proses, perancangan tabel dan perancangan uji coba
4. **BAB IV IMPLEMENTASI DAN PEMBAHASAN**
   Bab ini berisi hasil dari implementasi perangkat lunak yang digunakan untuk mengukur hasil penanganan *missing value*, pembahasan analisa hasil uji coba dan evaluasi hasil uji coba
5. **BAB V KESIMPULAN DAN SARAN**
   Bab ini memuat kesimpulan dari hasil penelitian dan saran-saran untuk pengembangan penelitian selanjutnya

BAB II
TINJAUAN PUSTAKA

2. 1 Optimasi

Menurut R. Jaka Arya Pradana (2008), optimasi adalah salah satu disiplin ilmu dalam matematika yang focus untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang, maupun pencarian nilai lainya dalam berbagai kasus. Optimasi sangat berguna di hampir segala bidang dalam rangka melakukan usaha secara efektif efisien untuk mencapai target hasil yang ingin dicapai. Tentunya hal ini akan sangat sesuai dengan prinsip ekonomi yang berorientasikan untuk senantiasa menekan pengeluaran untuk menghasilkan outputan yang maksimal. Optimasi ini juga penting karena persaingan saat ini sudah benar benar sangat ketat.

Seperti yang dikatakan di awal, bahwasanya optimasi sangat berguna bagi hamper seluruh bidang yang ada, maka berikut ini adalah contoh contoh bidang yang sangat terbantu dengan adanya teknik optimasi tersebut. Bidang tersebut, antara lain : Arsitektur, Data Mining, Jaringan Komputer, Signal And Immage Processing, Telekomunikasi, Ekonomi, Transportasi, Perdagangan, Pertanian, Perikanan, Perkebunan, Perhutanan, dan sebagainya.

Teknik optimasi secara umunm dapat dibagi menjadi dua bagian, yang pertama adalah Mathematical Programming, dan yang kedua adalah Combinatorial Optimatimization. Dalam bidang mathematical programming dapat dibagi menjadi dua kembali, yaitu support vector machines dan gradient descent. Dan pada bidang Combinatorial Optimization kembali difokuskan lagi ke dalam dua bidang, yaitu Graph Theory dan Genetic Algorithm. Pemfokusan pemfokusan bidang tersebut dikarenakan beberapa parameter, diantaranya, Restoration, Feature Selection, Classification, Clustering, RF assignment, Compression, dan sebagainya.

2. 2 Data

Data merupakan kumpulan kejadian nyata yang dapat dijamin kebenarannya. Data digunakan sebagai dasar pengambilan keputusan. Data dapat dikelompokkan ke dalam 2 kategori, yaitu data kualitatif dan data kuantitatif. Data kualitatif merupakan data

bertipe *string* dan bukan berupa angka/numerik. Sedangkan data kuantitatif dibagi kembali menjadi 2 macam, yaitu data diskrit dan data kontinyu. Data kuantitatif diskrit berupa bilangan bulat yang bertipe *integer*. Sedangkan data kontinyu merupakan data yang diperoleh dari hasil pengukuran yang dapat berupa bilangan pecahan dan bertipe *real*.

2. 3 *Data Mining*

2.3.1 Pengertian *data mining*

Berdasarkan kata-kata yang menyusunnya, *to mine* dalam bahasa inggris berarti mengekstrak atau menambang. Sehingga dapat dikatakan bahwa makna dari *data mining* adalah mengekstrak informasi yang berguna di antara sekumpulan data yang berjumlah sangat besar (Han dan Kamber,2000)

Menurut Hand (2001), *data mining* merupakan sebuah proses menganalisa sekumpulan data hasil penelitian, dengan tujuan untuk menemukan hubungan antar data, dan untuk meringkas data sehingga data menjadi mudah dimengerti dan berguna bagi pemiliki data.

Sedangkan menurut Turban,dkk (2005), *Data mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai *database* besar.

2.3.2 Proses *data mining*

Menurut Kantardzic (2003), dalam menemukan model dari data dalam *data mining* ada beberapa prosedur yang harus dilakukan, diantaranya :

1. Merumuskan permasalahan
   Dalam prosedur ini ditetapkan rumusan masalah dan variabel-variabel yang terlibat
2. Pengumpulan data
   Prosedur ini berkonsentrasi pada proses pembuatan data dan pengumpulan data

3. *Preprocessing data*
   Prosedur untuk menyeleksi data yang akan digunakan dalam proses
4. Mengestimasi model
   Seleksi dan implementasi terhadap metode data mining yang tepat merupakan proses utama pada prosedur ini
5. Menafsirkan model dan menarik kesimpulan

2. 4 *Preprocessing Data*

Banyak faktor yang mempengaruhi kesuksesan *Machine Learning*. Kualitas dari data yang digunakan oleh *Machine Learning* tersebut adalah yang terpenting. Jika pada data tersebut terjadi kekosongan data, redundansi informasi, *noisy* atau data tidak sesuai, maka penemuan informasi dari data tersebut selama fase training akan lebih sulit. Oleh karena itu diperlukan suatu *pre-processing data* yang dilakukan sebelum menggunakan data tersebut pada *Machine Learning*. Langkah-langkah utama dari *preprocessing data* adalah sebagai berikut :

1. Data *cleaning*
   Mengisi / mengganti nilai-nilai yang hilang, menghaluskan data yang *noisy*, mengidentifikasi dan menghilangkan data yang tidak wajar(*outliers)*, dan menyelesaikan masalah inkonsistensi data
2. Data *integration*
   Mengabungkan beberapa database dan file menjadi satu sehingga didapatkan sumber data yang besar
3. Data *transformation*
   Normalisasi dan aggregasi data
4. Data *reduction*
   Mengurangi volume data namun tetap mempertahankan arti dalam hal hasil analisis data
5. Data *discretization*
   Merupakan bagian dari data *reduction* dengan memperhitungkan data yang signifikan, khususnya pada data *kuantitatif*

Data pre-processing dapat berupa proses membersihkan, normalisasi, transformasi, dan lainnya. Hasil dari *pre-processing* adalah data training akhir yang siap digunakan.

2. 5 *Missing Value*

Menurut Iffat A.Gheyas dan Leslie S.Smith(2009), *missing value* merupakan hal yang biasa terdapat pada *dataset*. *Missing value* pada *dataset* didefinisikan sebagai kekosongan nilai dari variabel tertentu pada *dataset*. Sebagian besar algoritma *data mining* tidak dapat bekerja secara langsung dengan *dataset* yang tidak lengkap. Oleh karena itu diperlukan suatu metode penanganan untuk *missing value* pada *dataset* tersebut.

Menurut R.J.Little dan D.B.Rubin (2002), pada umumnya metode untuk menangani *missing value* dapat dibagi dalam 3 kategori, yaitu *Case/Pairwise Deletion*, Parameter *Estimation* dan Teknik Imputasi. Pada metode *Case/Pairwise Deletion*, dilakukan penghapusan terhadap *record dataset* yang variabelnya mengandung *missing value*. Pada metode Parameter *Estimation*, digunakan prosedur *Maximum Likelihood* yang menggunakan algoritma *Expectation-Maximization* untuk memperkirakan nilai dari suatu *missing value.* Pada metode Teknik Imputasi, *missing value* digantikan dengan nilai perkiraan yang didasarkan pada informasi yang terdapat pada *dataset.* Beberapa metode yang termasuk teknik imputasi diantaranya adalah *Mean Imputation*, *Mode Imputation* dan *K-Nearest Neighbor Imputation*.

2. 6 Metode Imputasi *Missing Value*

2. 6. 1 Metode *Mean Imputation*

Mean Imputation (MI) atau Imputasi Rata-rata merupakan metode imputasi yang sering digunakan. Menurut Emilio Soria Olivas, dkk. (2010), MI termasuk dalam *Statistical Solutions* dalam teknik imputasi. MI terdiri dari tahapan menggantikan nilai/data yang hilang pada sebuah atribut dengan nilai rata-rata dari semua nilai yang diketahui dari atribut tersebut yang berada pada kelas yang sama dengan *record* yang memiliki *missing value* tersebut. Diumpamakan nilai x_{ij} dari suatu kelas k, C_k, merupakan nilai yang hilang, kemudian akan dilakukan penggantian nilai hilang tersebut dengan persamaan 2.1.

$$\hat{x}_{ij} = \sum_{i:x_{ij} \epsilon C_k} \frac{x_{ij}}{n_k} \qquad (2.1)$$

Dimana n_k mewakili jumlah dari data yang tidak mengandung *missing value* pada atribut ke-j dari kelas ke-k. Berdasarkan pada Little dan Rubin (2002) hasil dari MI adalah :

a. Perkiraaan ukuran contoh yang terlalu tinggi
b. Perkiraan ragam terlalu rendah
c. Korelasi dibiaskan secara negative
d. Penyebaran dari nilai baru tidak cukup mewakili dari nilai populasi karena bentuk dari distribusi diubah dengan menambahkan nilai yang sama dengan rata-rata

Menurut (Chan dan Dunn,1972) serta (Mundfrom dan Whitcomb, 1998), mengganti semua *record* yang mengandung *missing value* dengan nilai tunggal akan menurunkan ragam dan menaikkan secara tidak alami kesignifikanan dari beberapa tes statistik yang berdasarkan padanya. Anehnya, meskipun demikian MI telah memberikan hasil uji coba yang bagus untuk melakukan klasifikasi *supervised* suatu *dataset*.

2. 6. 2 Metode *Mode Imputation*

Penanganan *missing value* menggunakan metode *Mean Imputation* hanya dapat dilakukan pada atribut dengan tipe *kuantitatif*. Namun pada permasalahan *missing value* dalam atribut bertipe *kualitatif* dapat digunakan metode *Mode Imputation*. Menurut Emilio Soria Olivas, dkk. (2010), *Mode Imputation* (MOI) termasuk dalam *Statistical Solutions* dalam teknik imputasi. Dalam *Mode Imputation* struktur hubungan antar data tidak dianggap. Menurut Edgar Acuna dan Caroline Rodriguez (2003), keberadaan relasi yang tinggi antara atribut ber-*missing value* dengan suatu atribut dapat menyebabkan teknik imputasi Mode tidak berguna atau bahkan berbahaya.

2. 6. 3 Metode *K-Nearest Neighbor Imputation*

Menurut Emilio Soria Olivas, dkk. (2010), *K-Nearest Neighbor Imputation* termasuk dalam *Machine Learning Solutions* dalam teknik imputasi. Metode ini menangani *missing value* pada suatu data dengan melakukan imputasi dengan mempertimbangkan nilai yang diberikan oleh *record* yang paling mirip.

1. Jarak *Eucledian*

Kemiripan *record* data dapat ditentukan dengan menghitung jarak terdekat dari semua *data training* terhadap *data testing*. Fungsi yang digunakan untuk menghitung jarak adalah fungsi jarak Eucledian. Fungsi Eucledian didefinisikan pada persamaan 2.2.

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

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

$$\text{dist}(x_i, x_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \quad (2.2)$$

dimana x_1 dan x_2 adalah 2 *record* dengan n atribut. Persamaan 2.2 menghitung jarak antara x_1 dan x_2, dengan tujuan untuk menentukan perbedaan nilai yang sesuai dari atribut-atribut yang berada pada *record* x_1 dan x_2.

Untuk data bertipe *kualitatif* memiliki aturan tersendiri. Dalam *software data mining* WEKA, digunakan prinsip jika data *kualitatif* yang dibandingkan berbeda, maka nilai jarak antara data tersebut adalah 1. Namun apabila data yang dibandingkan sama, maka nilai jarak antara data tersebut adalah 0.

2. Algoritma *K-Nearest Neighbor Imputation*

Algoritma Imputasi KNN adalah sebagai berikut :

1. Membagi *dataset* D kedalam 2 bagian, yaitu D_m dan D_c. D_m merupakan himpunan dari *record* yang memiliki sedikitnya 1 nilai atribut yang hilang. *Record* yang lain merupakan *record* yang tidak memiliki *missing value* (lengkap), dan dikelompokkan dalam D_c.
2. Untuk setiap vektor x dalam D_m :
   a) Membagi vektor *record* ke dalam 2 bagian, yaitu bagian yang ditemukan nilainya dalam *dataset*(x_o) dan bagian yang hilang (x_m) dengan notasi $x=[x_o;x_m]$
   b) Menghitung jarak antara x_o dan semua *vector record* dari himpunan D_c. Nilai D_c yang digunakan adalah nilai pada atribut-atribut yang ditemukan nilainya pada *vector* x.
   c) Gunakan k vektor *record* yang terdekat (K-*Nearest Neighbor)* dan lakukan *voting* terhadap perkiraan nilai yang hilang untuk atribut *kualitatif*. Untuk atribut numerik, nilai *missing value* diganti dengan nilai rata-rata dari atribut pada K-*Nearest Neighbor*.

2. 7 Evaluasi

Keakuratan metode-metode penanganan *missing value* akan dievaluasi dengan dua metode, yaitu metode evaluasi *Root Mean Squared Error* (RMSE) dan *Performance Metric*.

RMSE hanya digunakan untuk mengevaluasi nilai atribut yang bertipe numerik. RMSE dapat dihitung dengan persamaan 2.3 (Jianjun Hu,dkk. 2006).

$$RMSE = \sqrt{mean\left[\vec{y}_{imputasi} - \vec{y}_{benar}\right]^2} / \sqrt{mean\left[\vec{y}_{bener}\right]^2} \quad (2.3)$$

dimana :

$\vec{y}_{imputed}$ = nilai imputasi yang dihasilkan

$\vec{y}_{true}$ = nilai data yang sebenarnya

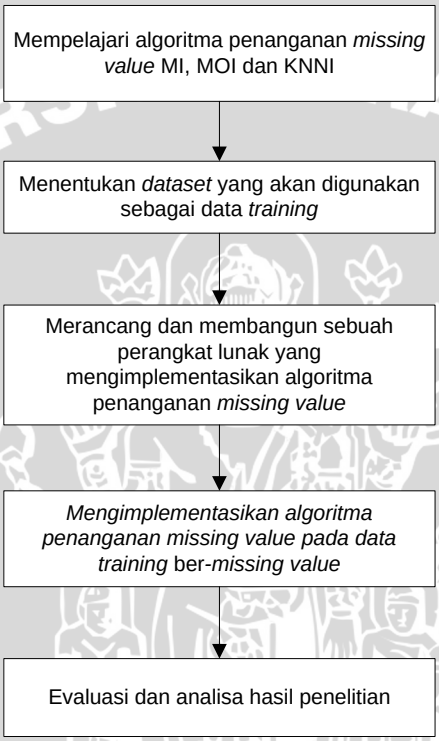
Metode evaluasi *Performance Metric Error rate* digunakan untuk mengevaluasi nilai atribut yang bertipe *kualitatif*. Menurut Tan.,dkk. (2004), *Error rate* dapat dihitung dengan persamaan 2.4.

$$Error\ rate = \frac{Jumlah\ Prediksi\ Salah}{Jumlah\ Total\ Prediksi} \quad (2.4)$$


UNIVERSITAS BRAWIJAYA

BAB III
METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang metodologi yang dilakukan dalam penelitian, serta rancangan sistem untuk melakukan uji coba dalam penelitian.



Gambar 3. 1 Langkah-langkah Penelitian

Berdasarkan gambar 3.1 dapat diamati bahwa langkah-langkah yang dilakukan pada penelitian ini adalah :

1. Mempelajari beberapa jurnal penelitian tentang algoritma penanganan *missing value Mean Imputation, Mode Imputation,* dan *KNN Imputation*
2. Menentukan *dataset* yang akan digunakan sebagai *data training*. *Data training* yang digunakan pada penelitian adalah *data training* yang memiliki tipe data *integer* dan *string*

3. Merancang dan membangun sebuah perangkat lunak yang mengimplementasikan algoritma penanganan *missing value* seperti *Mean Imputation, Mode Imputation* dan *KNN Imputation*
4. Mengimplementasikan algoritma-algoritma penanganan *missing value* untuk menangani *missing value* pada *data training*
5. Melakukan evaluasi dan analisa terhadap hasil penghitungan tingkat kesalahan. Tingkat kesalahan diperoleh dari perbandingan nilai imputasi dengan nilai sebenarnya yang dimiliki oleh data

3.1 Deskripsi Data

Data yang digunakan sebagai *data training* dalam penelitian adalah *dataset Mushroom* dan *dataset Mammographic Mass*. *Dataset* diperoleh dari situs UCI *Machine Leraning Repository* (*http://archive.ics.uci.edu/ml/ datasets.html*). Kedua *dataset* tersebut dipilih karena tipe data atribut-atribut pada *dataset* tersebut sesuai dengan yang dibutuhkan pada penelitian yang dilakukan. Penelitian difokuskan pada *dataset* yang memiliki atribut bertipe kualitatif dan kuantitatif. Dataset *Mushroom* memiliki atribut yang bertipe kualitatif, sedangkan dataset *Mammographic Mass* memiliki atribut bertipe kuantitatif (*integer*).

Dataset Mushroom merupakan data tentang klasifikasi jamur berdasarkan karakteristik fisiknya. Jamur akan diklasifikasikan menjadi dua kelas, yaitu kelas *poisonous* dan *edible*. Dataset tersebut bersumber dari *The Audubon Society Field Guide to North American Mushrooms. Dataset* tersebut memiliki 23 atribut (termasuk atribut kelas). Jumlah data seluruhnya adalah 8124 *record* dan jumlah data yang atributnya mengandung *missing value* adalah 2480 *record* data. Pada data tersebut, pola *missing value* hanya terletak pada atribut *Stalk-root*. Atribut-atribut yang terdapat pada *dataset* tersebut adalah *Cap-shape, Cap-surface, Cap-color, Bruises, Odor, Gill-Attachment, Gill-spacing, Gill-size, Gill-color, Stalk-shape, Stalk-root, Stalk-surface-above-ring, Stalk-surface-below-ring, Stalk-color-above-ring, Stalk-color-below-ring, Veil-type, Veil-color, Ring-number, Ring-type, Spore-print-color, Population, Habitat* dan *Class*.

Dataset Mammographic Mass merupakan data tentang klasifikasi hasil *screening* kanker payudara yang dilakukan dengan menggunakan metode *Mammographic*. Kanker payudara akan diklasifikasikan menjadi dua kelas, yaitu kanker jinak dan ganas.

Dataset tersebut bersumber dari *Image Processing and Medical Engineering Department* (BMT). Dataset tersebut memiliki 6 atribut (termasuk atribut kelas). Jumlah data seluruhnya adalah 961 *record* dan jumlah data yang atributnya mengandung *missing value* adalah 162 *record* data. Pada data tersebut, pola *missing value* terdapat pada hampir semua atribut, kecuali atribut *Severity* (atribut kelas). Atribut-atribut yang terdapat pada *dataset* tersebut adalah *Age*, *Shape*, *Margin*, *Density* dan atribut kelas *Severity*.

3.2 Analisa Perangkat Lunak

3.2.1 Deskripsi umum perangkat lunak

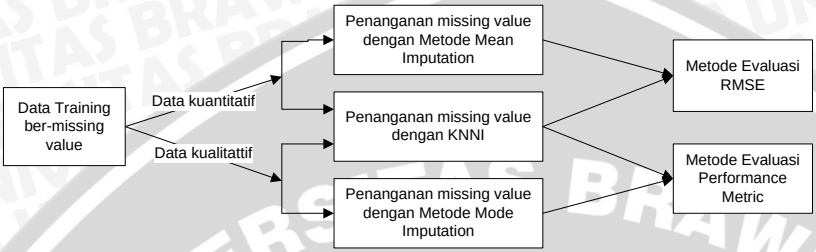
Perangkat lunak yang dibuat merupakan sebuah perangkat lunak yang mengimplementasikan proses penanganan *missing value* pada *data training* yang memiliki *missing value*. Metode-metode yang digunakan adalah *Mean & Mode Imputation*, serta *KNN Imputation*.

Perangkat lunak selanjutnya melakukan proses uji coba terhadap masing-masing metode tersebut. Parameter uji coba pada penelitian ini adalah keakuratan dari nilai imputasi yang dihasilkan dari metode-metode tersebut terhadap data yang sebenarnya. .

Proses yang terjadi ketika *user* menggunakan perangkat lunak yang dibuat adalah sebagai berikut :

1. *User* memilih *data training* ber-*missing value* yang digunakan sebagai data pengujian. *User* juga memilih d*ataset* asal yang tidak mengandung *missing value* sebagai pembanding nilai hasil penanganan. Data yang digunakan pada penelitian berupa file CSV
2. Sistem mengambil informasi dari tabel *data training* berupa nama tabel, jumlah *record*, jumlah nilai yang hilang, serta tipe atribut yang mengandung *missing value*
3. Sistem menerapkan algoritma penanganan terhadap masing-masing *missing value* pada *data training* tersebut dengan metode *Mean Imputation* dan *KNN Imputation* pada data bertipe kuantitatif, serta *Mode Imputation* dan *KNN Imputation* pada data bertipe kualitatif
4. Sistem membandingkan nilai asli data dengan nilai yang dihasilkan dari algoritma penanganan *missing value*, dan melakukan evaluasi terhadap keduanya berdasarkan persamaan yang telah didefiniskan pada subbab 2.7.

Secara global proses yang terjadi pada sistem digambarkan pada desain sistem yang ditunjukkan pada gambar 3.2.



Gambar 3. 2 Desain Sistem

3.2.2 Batasan perangkat lunak

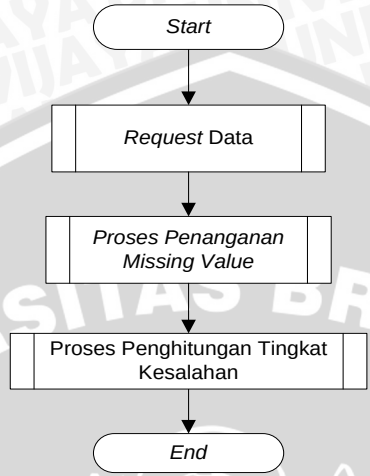
1. Jenis *file* yang digunakan untuk menyimpan *data training* berekstensi *.csv
2. Kolom yang menjelaskan kelas pada *data training* diletakkan pada urutan terakhir
3. Peletakan *missing value* pada *data training* dilakukan tanpa mempertimbangkan sebaran statistika, namun berdasarkan pada keterangan atau pola *missing value* pada *dataset*
4. Algoritma penanganan *missing value Mean Imputation* diterapkan pada *dataset* dengan tipe kuantitatif, sedangkan algoritma *Mode Imputation* diterapkan dalam *dataset* bertipe kualitatif
5. *Data training* yang digunakan pada awalnya tidak diperbolehkan mengandung *missing value*, agar sistem dapat mengetahui nilai data sebenarnya apabila dilakukan pengosongan pada nilai data tersebut

3.3 Perancangan Perangkat Lunak

3.3.1 Perancangan proses

Pada gambar 3.3 dapat diamati bahwa dalam sistem yang dibuat terdapat tiga buah proses utama, yaitu :

- proses *request data*
- proses penanganan *missing value*
- proses perhitungan tingkat kesalahan



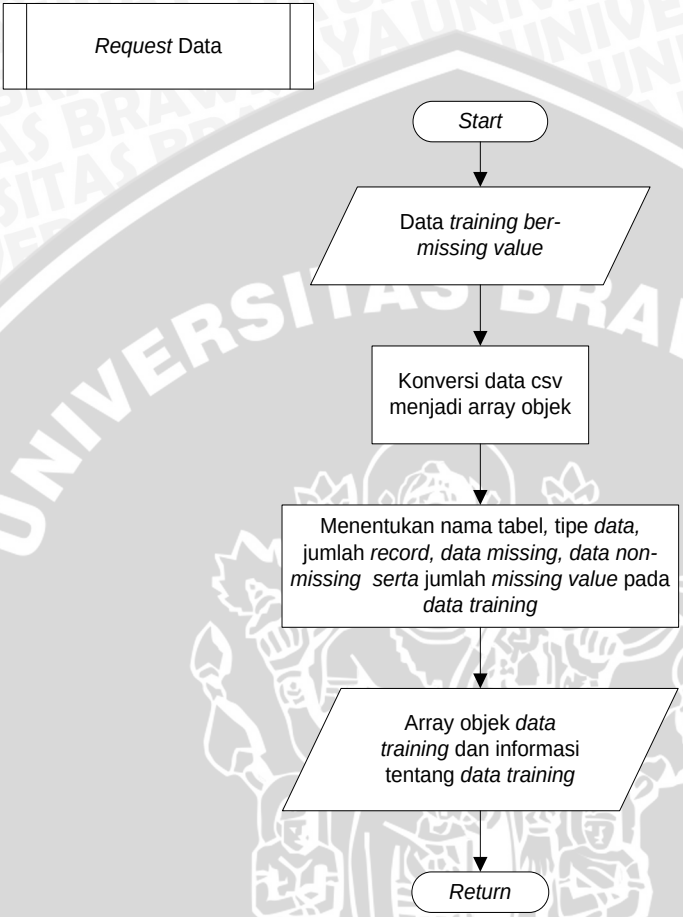
Gambar 3. 3 Alur proses sistem

1. Proses *Request Data*

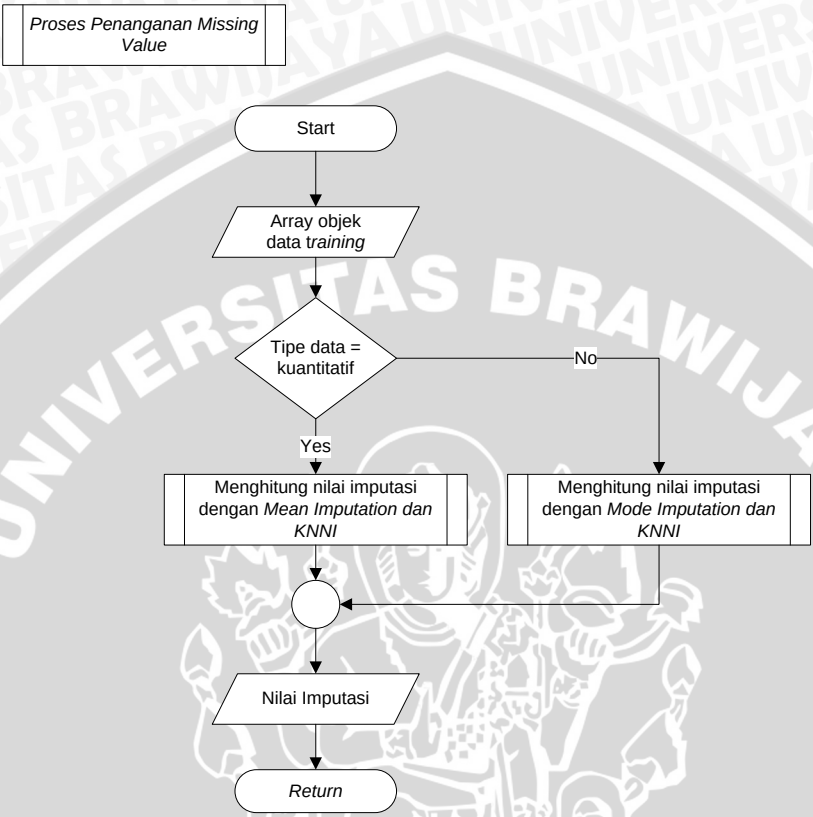
Proses *request data* terjadi ketika *user* melakukan pemilihan terhadap *data training* yang selanjutnya akan digunakan untuk penelitian. Dalam gambar 3.4 dapat diamati bahwa dari *data training* yang terpilih dapat diperoleh berbagai informasi tentang data tersebut, diantaranya nama *file*, jenis *data training*, jumlah *record*, serta jumlah *missing value* pada *data trining* tersebut. Pada akhir proses *request data*, seluruh *data training* yang telah dipilih beserta informasinya akan ditampilkan kepada *user*.

2. **Proses Penanganan *Missing Value***

Gambar 3.5 menunjukkan proses yang terjadi pada penanganan *missing value*. Pada awal proses dilakukan seleksi terhadap *data training*. Pada proses ini terdapat 2 subproses. Jika atribut data bertipe kuantitatif, maka untuk mendapatkan nilai imputasi atau nilai pengganti *missing value* digunakan metode penanganan *Mean Imputation* dan metode penanganan *KNN Imputation*. Sedangkan jika atribut bertipe kualitatif, maka nilai imputasi ditentukan dengan menggunakan metode *Mode Imputation* dan metode *KNN Imputation*. Hasil dari proses ini adalah nilai imputasi.



Gambar 3. 4 Proses *request data*



Gambar 3. 5 Proses penanganan *missing value*

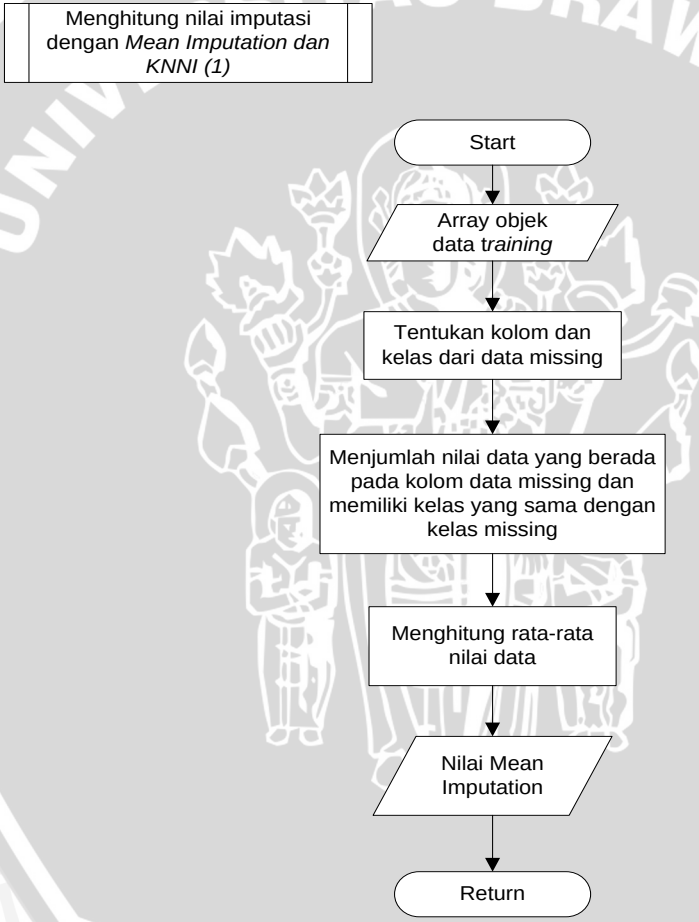
a. Menghitung nilai imputasi dengan *Mean Imputation* dan *KNNI*

Proses ini dilakukan jika *data training* yang digunakan bertipe kuantitatif.

1) Metode *Mean Imputation*

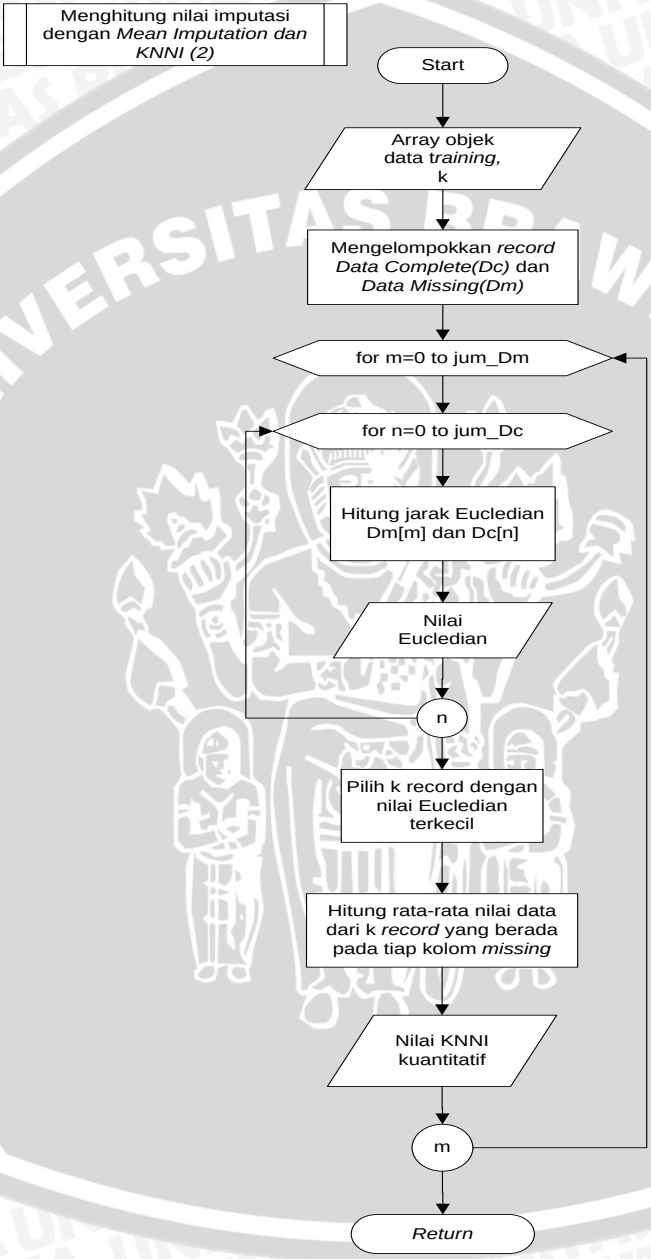
Pada gambar 3.6 ditunjukkan alur proses penanganan *missing value* dengan metode *Mean Imputation*. Pada tahap awal dilakukan pengumpulan informasi tentang kolom dan kelas dari data *missing*. Hal tersebut dilakukan terkait dengan proses *Mean Imputation* yang menggantikan nilai data yang hilang dengan rata-rata nilai dari kolom yang sama dengan kolom *missing value* serta memiliki kelas yang sama dengan kelas yang dimiliki oleh data *missing value*.

Setelah diketahui kolom dan kelas dari data *missing*, maka dilakukan penjumlahan terhadap nilai data yang berada pada kolom data *missing* serta memiliki kelas yang sama dengan data *missing*. Pada proses selanjutnya dilakukan penghitungan rata-rata terhadap sejumlah nilai data tersebut sehingga didapatkan nilai *Mean Imputation*. Apabila terdapat sejumlah *missing value* pada kolom yang sama, serta memiliki kelas yang sama, maka nilai pengganti untuk *missing value* tersebut adalah sama.



Gambar 3. 6 Proses *Mean Imputation*

2) Metode *KNN Imputation*



Gambar 3. 7 Proses *KNN Imputation* untuk data kuantitatif

Metode *KNN Imputation* yang diimplementasikan pada proses ini merupakan metode khusus untuk data bertipe kuantitatif. Pada gambar 3.7 ditunjukkan alur proses penanganan *missing value* dengan metode *KNN Imputation*. Pada tahap awal dilakukan pengelompokan *record-record* yang mengandung *missing value* (Dm) dan *record-record* yang tidak mengandung *missing value* (Dc). Pada proses selanjutnya dilakukan penghitungan nilai jarak *eucledian*. Setiap nilai data yang terdapat pada kolom Dm dihitung jarak *eucledian*-nya terhadap semua nilai data yang terdapat pada kolom Dc yang bersesuaian dengan Dm. Setelah nilai *eucledian* dihitung, maka dipilih sejumlah "k"*record* Dc yang memiliki nilai *eucledian* terkecil.

Untuk setiap Dm, dilakukan penghitungan nilai rata-rata yang berasal dari nilai data pada sejumlah "k" kolom Dc terpilih yang bersesuaian dengan Dm tersebut. Dari proses tersebut diperoleh nilai *KNN Imputation*. Proses tersebut dilakukan berulang kali untuk semua Dm.

b. Menghitung nilai imputasi dengan *Mode Imputation* dan *KNN Imputation*

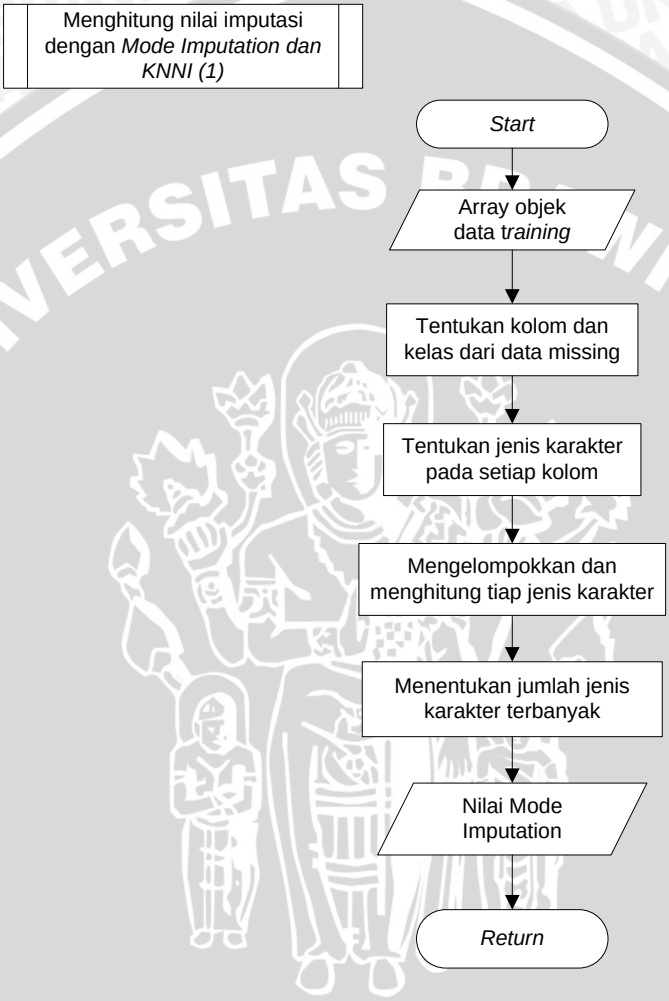
Proses ini dilakukan jika *data training* yang digunakan bertipe kualitatif.

1) Metode *Mode Imputation*

Pada gambar 3.8 ditunjukkan alur proses penanganan *missing value* dengan metode *Mode Imputation*. Pada tahap awal dilakukan pengumpulan informasi tentang kolom dan kelas dari data *missing*. Hal tersebut dilakukan terkait dengan proses *Mode Imputation* yang menggantikan nilai data yang hilang dengan Mode dari nilai-nilai data pada kolom yang sama dengan kolom *missing value* serta memiliki kelas yang sama dengan kelas yang dimiliki oleh data *missing value*. Setelah diketahui kolom dan kelas dari data *missing*, maka dilakukan pengelompokan dan penghitungan tiap jenis karakter pada setiap kolom yang mengandung *missing value* serta memiliki kelas yang sama dengan *missing value*.

Pada proses selanjutnya dilakukan penentuan karakter yang memilki jumlah terbanyak sehingga didapatkan nilai *Mode Imputation*. Apabila terdapat sejumlah *missing value* pada kolom

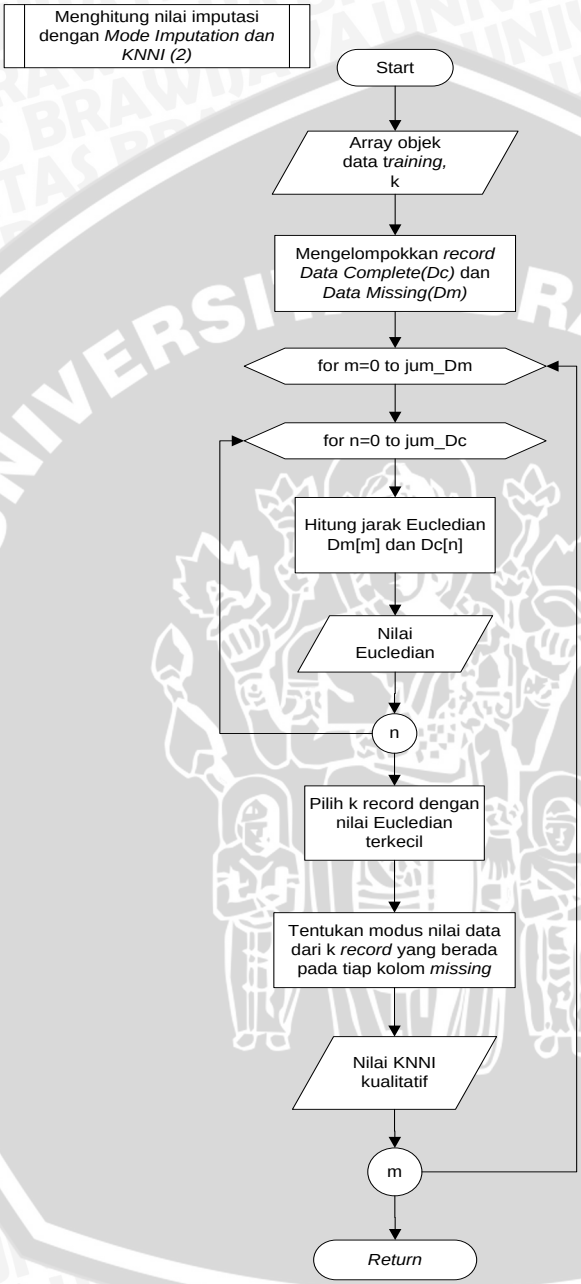
yang sama, serta memiliki kelas yang sama, maka nilai pengganti untuk *missing value* tersebut adalah sama.



Gambar 3. 8 Proses *Mode Imputation*

2) Metode *KNN Imputation*

Metode *KNN Imputation* yang diimplementasikan pada proses ini merupakan metode khusus untuk data bertipe kualitatif. Pada gambar 3.9 ditunjukkan alur proses penanganan *missing value* dengan metode *KNN Imputation*.

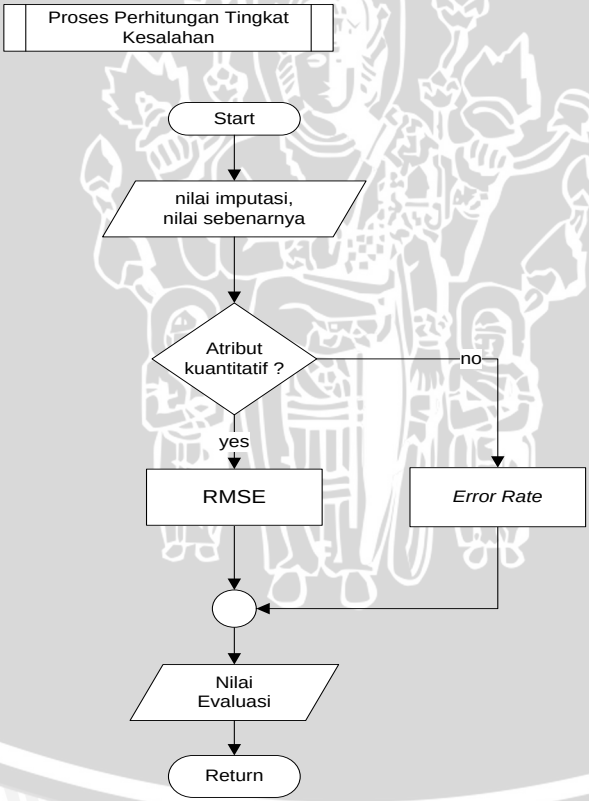


Gambar 3. 9 Proses *KNN Imputation* untuk data kualitatif

Pada tahap awal dilakukan pengelompokan *record-record* yang mengandung *missing value* (Dm) dan *record-record* yang tidak mengandung *missing value* (Dc). Pada proses selanjutnya dilakukan penghitungan nilai jarak *eucledian*. Setiap nilai data yang terdapat pada kolom Dm dihitung jarak *eucledian*-nya terhadap semua nilai data yang terdapat pada kolom Dc yang bersesuaian dengan Dm. Setelah nilai *eucledian* dihitung, maka dipilih sejumlah "k"*record* Dc yang memiliki nilai *eucledian* terkecil.

Untuk setiap Dm, dilakukan penentuan Mode data yang berasal dari nilai data pada sejumlah "k" kolom Dc terpilih yang bersesuaian dengan Dm tersebut. Dari proses tersebut diperoleh nilai *KNN Imputation*. Proses dilakukan berulang kali untuk semua Dm.

3. Proses Penghitungan Tingkat Kesalahan



Gambar 3. 10 Proses penghitungan tingkat kesalahan

Proses penghitungan tingkat kesalahan ditunjukkan pada gambar 3.10. *Input* pada proses tersebut adalah nilai imputasi yang dihasilkan dari setiap metode penanganan *missing value* dan nilai sebenarnya dari data tersebut. Pada tahap awal dilakukan penyeleksian terhadap tipe data. Jika *data training* bertipe kuantitatif, maka tingkat kesalahan dihitung berdasarkan rumus RMSE. Namun apabila tipe data dari *data training* adalah kualitatif, maka tingkat kesalahan dihitung berdasarkan rumus *performance metric error rate*. Dengan metode-metode tersebut proses dapat menghasilkan nilai evaluasi tingkat kesalahan.

3.3.2 Perancangan tabel

Dalam penelitian ini, tabel tidak memiliki relasi. Berdasarkan pada perancangan proses, masing-masing tabel *data training* hanya berupa tabel tunggal tanpa relasi. Format tabel sama dengan *dataset* yang diperoleh dari situs UCI *Machine Learning Repository* dan telah dijelaskan pada subbab 3.1. Tabel 3.1 menunjukkan format tabel *data training* kuantitatif *Mammographic Mass* dan tabel 3.2 menunjukkan format tabel *data training* kualitatif *Mushroom*.

Tabel 3. 1 Data kuantitatif *Mammographic Mass*

BI-RADS Assessment	Age	Shape	Margin	Density	Severity
4	52	2	1	3	0
4	38	1	1	3	0
3	72	4	3	3	0
..	..	..	..	..	..
..	..	..	..	..	..

Tabel 3. 2 Data kualitatif *Mushroom*

Cap_Shape	Cap_Surface	Cap_Color	Bruises	…	…	Class
x	s	n	t			p
x	s	y	t			e
b	s	w	t			p
..	..	..	..	..	..	..

3.4 Perancangan Uji Coba

Pada penelitian ini dilakukan uji coba penanganan atau pencarian nilai pengganti *missing value* yang terdapat pada 2 macam tipe data, yaitu kuantitatif dan kualitatif. Untuk *missing value* yang terdapat pada data bertipe kuantitatif digunakan metode penanganan *Mean Imputation* dan *KNN Imputation*. Sedangkan untuk *missing value* pada data bertipe kualitatif digunakan metode penanganan *Mode Imputation* dan *KNN Imputation*. Nilai pengganti *missing value* yang dihasilkan dari metode penanganan tersebut kemudian dibandingkan dengan nilai yang sebenarnya dari *missing value* tersebut sehingga diketahui tingkat kesalahan dari nilai hasil penanganan tersebut. Dengan demikian dapat diketahui kinerja dari metode *Mean Impuatation* dan *KNN Imputation* dalam menangani *missing value* pada data bertipe kuantitatif, serta kinerja dari *Mode Imputation* dan *KNN Imputation* dalam menangani *missing value* pada data bertipe kualitatif.

Pada mulanya *data training* yang digunakan pada penelitian ini telah memiliki *missing value*. Namun *record - record* yang mengandung *missing value* pada data tersebut kemudian dihilangkan karena jika data *training* dengan *missing value* tersebut langsung digunakan untuk penelitian, maka nilai yang sebenarnya dari *missing value* tersebut tidak diketahui, sehingga tidak bisa dilakukan evaluasi tingkat kesalahan dari hasil penanganan *missing value*. Dengan *data training* tanpa *missing value* tersebut, jika kemudian dilakukan pengosongan beberapa nilai data pada posisi manapun selain keterangan kelas, maka hal tersebut tidak menjadi masalah. Data sebenarnya dari data yang telah dikosongi dapat diketahui berdasarkan data yang lengkap tersebut, sehingga pembandingan nilai hasil penanganan yang dihasilkan dari penanganan *missing value* dengan nilai yang sebenarnya tetap dapat dilakukan. Pada uji coba, pengosongan beberapa nilai data pada *data training* dilakukan secara random, dengan memperhatikan pola *missing value* pada *data training* sebelum dimodifikasi. Pola *missing value* sebelumnya perlu diperhatikan karena tidak semua atribut data diperbolehkan memiliki *missing value*.

Pada uji coba digunakan 57 *data training*, yang terdiri dari 3 tabel yang tidak mengandung *missing value*, dan 54 *data training* mengandung *missing value* yang terdiri dari 27 *data training* bertipe data kuantitatif dan 27 *data training* bertipe data kualitatif. Pengujian

dilakukan pada *data training* ber-*missing value* tersebut yang masing-masing memiliki perbedaan pada jumlah *record*, jumlah persentase *missing value*, serta peletakan dari *missing*. Sedangkan 3 tabel *data training* yang lengkap atau tidak mengandung *missing value* masing-masing dibedakan berdasarkan jumlah *record*, yaitu 300 *record*, 150 *record* dan 75 *record* . *Data training* yang lengkap tersebut digunakan sebagai pembanding dari nilai hasil penghitungan dengan menggunakan metode penanganan *missing value*. Jadi, setiap nilai hasil penanganan *missing value* dibandingkan dengan nilai pada posisi yang sama pada *data training* yang lengkap. Perbandingan hanya dapat dilakukan antara *data training* lengkap dan *data training* ber-*missing value* yang memiliki jumlah *record* yang sama.

Pengujian berdasarkan jumlah *data training* dilakukan dengan 3 variasi jumlah *record* data, yaitu 300 *record*, 150 *record*, dan 75 *record*. Pada masing-masing jumlah *record* tersebut ditentukan selisih sebesar 50% agar pengujian tidak dilakukan dengan jumlah data yang hampir sama atau selisihnya terlalu dekat. Untuk pengujian berdasarkan *missing value* digunakan 4 macam nilai persentase *missing value*, yaitu 5%, 10%, 15% dan 30%. Nilai persentase tersebut dipilih karena menurut Edgar Acuna dan Caroline Rodriguez (2003), *missing value* dengan persentase diantara 5% hingga 15% dapat ditangani dengan metode penanganan yang handal. Hal tersebut sangat sesuai dengan tujuan dari pengujian yang bertujuan untuk mengetahui kinerja dari metode penanganan *missing value*, apakah dapat menjadi metode penanganan yang handal untuk menangani *missing value*. Namun untuk menguji kehandalan metode penanganan pada persentase yang melebihi nilai pada teori tersebut, maka pada penelitian ini metode penanganan juga diujikan pada *data training* dengan persentase *missing value* 30%. Jumlah *missing value* pada pengujian ditentukan dari prosentase *missing value* yang telah ditentukan terhadap jumlah data keseluruhan pada kolom yang mengandung *missing value*, bukan jumlah *record*.

Pada setiap pengujian, nilai evaluasi hasil perbandingan nilai penanganan disimpan dalam tabel pengujian yang dapat dilihat pada tabel 3.3 dan tabel 3.4. Tabel 3.3 merupakan tabel untuk hasil uji coba terhadap data bertipe kuantitatif, sedangkan tabel 3.4 merupakan tabel untuk hasil uji coba terhadap data bertipe kualitatif. Setelah dilakukan tiga kali uji coba pada *data training* dengan posisi *missing value* yang berbeda, maka dilakukan perhitungan nilai rata-

rata terhadap ketiga nilai evaluasi tersebut. Hal ini ditujukan untuk menemukan nilai evaluasi yang akurat.

Pada pengujian menggunakan metode KNNI digunakan 3 parameter nilai tetangga/*record* terdekat (k). Nilai k yang digunakan pada pengujian adalah 3, 5, 10 dan nilai k maksimal yang diperbolehkan. Nilai k=3 dipilih sebagai nilai minimum untuk jumlah tetangga/*record* terdekat. Nilai k=5 dan k=10 dipilih agar tidak terlalu jauh dengan nilai minimum k sehingga dapat terlihat pola perubahan nilai *Error rate* pada pengujian. Sedangkan nilai k = maksimal dipilih untuk mengetahui *Error rate* jika semua *record* tidak ber-*missing value* dipilih sebagai tetangga/*record* terdekat.

Tabel 3. 3 Hasil pengujian *data training* kuantitatif

Jumlah *record*	Jumlah Missing Value	Metode	Uji coba	Nama Tabel	RMSE	Rata-rata RMSE
75, 150, 300 *record*	5%, 10%, 15%, 30%	Mean Imputation	uji 1			
			uji 2			
			uji 3			
		KNNI (k=3,5,10,maks)	uji 1			
			uji 2			
			uji 3			

Tabel 3. 4 Hasil pengujian *data training* kualitatif

Jumlah *record*	Jumlah Missing Value	Metode	Uji coba	Nama Tabel	Error Rate	Rata-rata Error Rate
75, 150, 300 *record*	5%, 10%, 15%, 30%	Mode Imputation	uji 1			
			uji 2			
			uji 3			
		KNNI (k=3,5,10,maks)	uji 1			
			uji 2			
			uji 3			

3.5 Contoh Perhitungan

3.5.1 Perhitungan nilai imputasi tipe data kuantitatif

Data pada contoh perhitungan diambil dari *dataset Mammographic Mass*. Tabel 3.5 menunjukkan data-data yang akan digunakan pada perhitungan.

Tabel 3. 5 Contoh data kuantitatif

BI-RADS Assessment	Age	Shape	Margin	Density	Severity
4	52	2	1	3	0
4	38	1	1	3	0
3	72	4	3	3	0
5	80	4	3	3	1
5	76	4	3	3	1
4	62	3	1	3	0
5	64	4	5	3	1
5	42	4	5	3	0
4	64	4	5	3	0
4	63	4	4	3	1

Tabel 3. 6 Data kuantitatif ber-*missing value*

Record ke-	BI-RADS Assessment	Age	Shape	Margin	Density	Severity
1	4	52	2	1	3	0
2	4	38	1	1	3	0
3	3	72		3	3	0
4		80	4	3		1
5	5	76	4	3	3	1
6	4	62	3	1	3	0
7	5	64	4		3	1
8		42	4	5	3	0
9	4		4	5	3	0
10	4	63	4	4	3	1

Pada *dataset* tersebut dilakukan penghilangan beberapa nilai sebagai *missing value*. Tidak semua atribut dapat dihilangkan nilainya. Oleh karena itu pengosongan dilakukan berdasarkan pada ketentuan yang dimiliki oleh *dataset.* Pada *dataset Mammographic Mass* , atribut yang boleh dihilangkan nilainya adalah semua atribut kecuali atribut *Severity* yang merupakan atribut kelas. Hasil penghilangan beberapa data dapat dilihat pada tabel 3.6.

Nilai-nilai yang dihilangkan pada *data training* tersebut dapat dilihat pada Tabel 3.7.

Tabel 3. 7 Posisi dan nilai data yang dihilangkan

Record ke-	Atribut missing (kolom ke-)	Nilai sebenarnya
3	3	4
4	1	5
	5	3
7	4	5
8	1	5
9	2	64

Perhitungan dilakukan dengan tujuan untuk mencari nilai prediksi dari nilai yang dihilangkan tersebut. Seperti yang telah dijelaskan pada subbab Perancangan Uji Coba, untuk data yang memiliki tipe data numerik, metode penanganan data hilang (*missing value*) yang digunakan adalah *Mean Imputation* dan *KNN Imputation*.

1. *Mean Imputation*

Nilai *Mean Imputation* dapat diperoleh dengan menggunakan persamaan 2.1. Penanganan *missing value* dengan *Mean Imputation* dilakukan dengan tahapan sebagai berikut :

Tabel 3. 8 Kolom dan kelas *data missing*

Kolom	Kelas ber-*missing value*
1	0
	1
2	0
3	0
4	1
5	1

- Pada setiap kolom atribut *dataset* yang mengandung *missing value*, tentukan kelas-kelas yang memiliki *missing value*. Tabel 3.8 menunjukkan kelas-kelas yang mengandung *missing value* pada tiap kolom
- Pada setiap kolom ber-*missing value*, kelompokkan nilai-nilai atribut yang memiliki kelas yang sama dengan kelas ber-*missing value*. Kemudian hitung rata-ratanya. Tabel 3.9 menunjukkan nilai rata-rata yang dihasilkan dari penghitungan kelompok nilai data. Nilai *mean* tersebut merupakan hasil dari metode *Mean Imputation.*

2. *KNN Imputation*

Dalam menangani *missing value* pada data *kuantitatif* dapat digunakan metode *K-Nearest Neighbor Imputation* (KNNI). Langkah-langkah penanganan *missing value* dengan menggunakan KNNI adalah sebagai berikut :

1) Membagi *dataset* menjadi *dataset complete* (Dc) dan *dataset missing* (Dm)

Dataset complete merupakan *record-record* dari *dataset* yang tidak mengandung *missing value,* sedangkan *dataset missing* merupakan *record-record* dari *dataset* yang tidak mengandung *missing value*.

D_m = {*record 3,record 4,record 7,record 8, record 9, record 10*}

D_c = { *record 1, record 2, record 5, record 6, record 1*}

Pembagian *record dataset* dapat dilihat lebih jelas pada tabel 3.10 dan tabel 3.11.

Tabel 3. 9 Hasil *Mean Imputation*

Kolom	Baris	Kelas ber-*missing value*	Nilai data	Mean nilai data
1	8	0	4	4
			4	
			3	
			4	
			4	
	4	1	5	5
			5	
			4	
2	9	0	52	53
			38	
			72	
			62	
			42	
3	3	0	2	3
			1	
			3	
			4	
			4	
4	7	1	3	3
			3	
			4	
5	4	1	3	3
			3	
			3	

Tabel 3. 10 Data *missing* (Dm) pada data kuantitatif

Record ke-	BI-RADS Assessment	Age	Shape	Margin	Density	Severity
3	3	72		3	3	0
4		80	4	3		1
7	5	64	4		3	1
8		42	4	5	3	0
9	4		4	5	3	0

Tabel 3. 11 Data *complete* (Dc) pada data kuantitatif

Record ke-	BI-RADS Assessment	Age	Shape	Margin	Density	Severity
1	4	52	2	1	3	0
2	4	38	1	1	3	0
5	5	76	4	3	3	1
6	4	62	3	1	3	0
10	4	63	4	4	3	1

2) Pada setiap *record* Dm
   a. Lakukan pembagian atribut-atribut dalam Dm, antara bagian yang hilang (x_m) dan bagian yang ditemukan nilainya dalam *record* tersebut (x_o)
      Misalkan untuk *record* ke-3 yang termasuk dalam Dm :
      x_o={BI-RADS Assessment=3;Age=72;Margin=3;Density =3}
      x_m={Shape=?}
   b. Hitung jarak *Eucledian* antara x_o dengan semua D_c
      Jarak *Eucledian* dihitung dengan menggunakan persamaan 2.2.
      Misalkan dihitung jarak antara x_o pada *record* ke-3 dengan semua D_c :

x_o *record* ke-3 = { BI-RADS Assessment=3; Age=72; Margin=3; Density =3}

Pada x_o *record* ke-3, nilai atribut *Shape* adalah *missing value*, maka pada D_c nilai dari atribut *Shape* juga dihilangkan karena atribut yang nilainya digunakan pada perhitungan jarak adalah atribut yang termasuk dalam x_o *record* ke-3.

- Untuk x_o pada D_c *record* ke-1 = { BI-RADS Assessment=4; Age=52; Margin=2; Density =3}
  Jarak = $\sqrt{(3-4)^2 + (72-52)^2 + (3-2)^2 + (3-3)^2}$ = 20.05
- Untuk x_o pada D_c *record* ke-2 = { BI-RADS Assessment=4; Age=38; Margin=1; Density =3}
  Jarak = $\sqrt{(3-4)^2 + (72-38)^2 + (3-1)^2 + (3-3)^2}$ = 34.07
- Untuk x_o pada D_c *record* ke-5= { BI-RADS Assessment=5; Age=76; Margin=3; Density =3}
  Jarak = $\sqrt{(3-5)^2 + (72-76)^2 + (3-3)^2 + (3-3)^2}$ = 4.47
- Untuk x_o pada D_c *record* ke-6= { BI-RADS Assessment=4; Age=62; Margin=1; Density =3}
  Jarak = $\sqrt{(3-4)^2 + (72-62)^2 + (3-1)^2 + (3-3)^2}$ = 10.25
- Untuk x_o pada D_c *record* ke-10= { BI-RADS Assessment=4; Age=63; Margin=4; Density =3}
  Jarak = $\sqrt{(3-4)^2 + (72-63)^2 + (3-4)^2 + (3-3)^2}$ = 9.11

c. Tentukan "k" nilai *Eucledian* terkecil

Misalkan k=3, maka akan dipilih 3 *record* D_c yang memiliki nilai Eucledian terkecil. Karena *missing value* berada pada atribut *Shape*, maka dari masing-masing ketiga *record* tersebut diambil nilai dari atribut *Shape*. Tabel 3.12 menampilkan nilai-nilai atribut *missing* dari "k" Dc dengan *eucledian* terkecil.

Tabel 3. 12 Nilai atribut eucledian terkecil

Data Complete	Nilai atribut missing *Shape*
Dc *record* ke-5	4
Dc *record* ke-10	4
Dc *record* ke-6	3

d. Hitung rata-rata nilai atribut dari sejumlah Dc dengan *Eucledian* terkecil

Nilai-nilai pada tabel 3.12 tersebut kemudian dihitung rata-ratanya.

rata-rata = $\frac{4+4+3}{3} = \frac{11}{3} = 3.67 \sim 4$

Maka nilai imputasi dari *missing value* atribut *Shape* pada *record* ke-3 adalah 4.

Lakukan proses perulangan hingga dapat ditemukan nilai imputasi untuk semua *missing value*. Nilai imputasi yang dihasilkan dengan metode penanganan *missing value* KNNI dapat dilihat pada tabel 3.13.

Tabel 3. 13 Hasil *KNN Imputation*

Record ke-	Atribut missing (kolom ke-)	Nilai Imputasi
3	3	3.6
4	1	4.3
	5	3
7	4	2.6
8	1	4
9	2	67

3.5.2 Perhitungan nilai imputasi tipe data kualitatif

Data pada contoh perhitungan diambil dari *dataset Mushroom*. Tabel 3.14 menunjukkan data-data yang akan digunakan pada perhitungan.

Pada *data training* tersebut dilakukan penghilangan beberapa nilai sebagai *missing value*. Hasil penghilangan beberapa nilai data dapat dilihat pada tabel 3.15. Sedangkan nilai-nilai data yang dihilangkan pada *data training* tersebut dapat dilihat pada Tabel 3.16.

Tabel 3. 14 Contoh data kualitatif

Gill Size	Gill Color	Stalk Shape	Stalk Root	Stalk Surface Above Ring	Stalk Surface Below Ring	Class
b	k	t	e	s	f	e
b	k	e	c	s	s	e
b	k	t	e	f	f	e
b	n	e	r	s	y	e
n	k	e	e	s	s	p
n	k	e	e	s	s	p
b	k	t	e	f	s	e
b	n	e	r	s	y	e
b	n	t	e	f	f	e
b	n	t	e	s	s	e

Tabel 3.15 Data kualitatif ber-*missing value*

Gill Size	Gill Color	Stalk Shape	Stalk Root	Stalk Surface Above Ring	Stalk Surface Below Ring	Class
b	k	t			f	e
b	k		c	s	s	e
b	k	t	e	f	f	e
b	n	e	r	s	y	e
	k	e	e	s	s	p
n	k	e	e	s	s	p
b	k	t		f		e
b	n	e	r	s	y	e
b	n	t	e	f	f	e
b	n	t	e	s	s	e

Tabel 3.16 Posisi dan nilai data yang dihilangkan

Record ke-	Atribut missing (kolom ke-)	Nilai sebenarnya
1	4	e
	5	s
2	3	e
5	1	n
7	4	e
	6	s

Perhitungan dilakukan dengan tujuan untuk mencari nilai prediksi dari nilai yang dihilangkan tersebut. Seperti yang telah dijelaskan pada subbab Perancangan Uji Coba, untuk data yang memiliki tipe data kualitatif, metode penanganan data hilang (*missing value*) yang digunakan adalah *Mode Imputation* dan *KNN Imputation*.

1. *Mode Imputation*

Nilai *Mode Imputation* dapat diperoleh dengan mencari nilai yang paling sering muncul dengan ketentuan tertentu. Penanganan *missing value* dengan *Mode Imputation* dilakukan dengan tahapan sebagai berikut :

1) Pada setiap kolom atribut *dataset* yang mengandung *missing value*, tentukan kelas-kelas yang memiliki *missing value*. Tabel 3.17 menunjukkan kelas-kelas yang mengandung *missing value* pada tiap kolom

Tabel 3. 17 Kolom dan kelas data *missing*

Kolom	Kelas ber-*missing value*
1	p
3	e
4	e
5	e
6	e

2) Pada setiap kolom ber-*missing value*, kelompokkan nilai-nilai atribut yang memiliki kelas yang sama dengan kelas data *missing*

Tabel 3.18 Hasil *Mode Imputation*

Kolom	Kelas ber-missing value	Nilai data	Kelompok kategori	Mode
1	p	n	n = 1	n
3	e	t	t = 5	t
		t	e = 2	
		e		
		t		
		e		
		t		
		t		
4	e	c	c = 1	e
		e	e = 3	
		r	r = 2	
		r		
		e		
		e		
5	e	s	s = 4	s
		f	f = 3	
		s		
		f		
		s		
		f		
		s		
6	e	f	f = 3	f
		s	s = 2	
		f	y = 2	
		y		
		y		
		f		
		s		

Misalnya untuk kolom 3, *missing value* termasuk dalam kelas e. Maka selanjutnya dilakukan seleksi pada kolom tersebut untuk menemukan nilai data yang memiliki kelas e.

Data-data yang terseleksi : t,t,e,t,e,t dan t

Jenis kategori data yang sama kemudian dikelompokkan dan dihitung jumlahnya. Pada contoh tersebut didapatkan 3 kelompok kategori yang terseleksi:

- Kategori t sejumlah 5
- Kategori e sejumlah 2

Jenis kategori data yang memiliki jumlah terbesar merupakan Mode dari kumpulan nilai data tersebut. Pada contoh, kategori t merupakan Mode dari kumpulan nilai data tersebut, sehingga *missing value* pada kolom 3 yang memiliki kelas e dapat diganti dengan nilai Mode, yaitu t.

Lakukan proses perulangan hingga dapat ditemukan nilai imputasi untuk semua *missing value*. Nilai imputasi yang dihasilkan dengan metode penanganan *missing value Mode Imputation* dapat dilihat pada tabel 3.18.

2. *KNN Imputation*

Dalam menangani *missing value* pada data *kualitatif* dapat digunakan metode *k-Nearest Neighbor Imputation* (KNNI). Langkah-langkah penanganan *missing value* dengan menggunakan KNNI adalah sebagai berikut :

1) Membagi *dataset* menjadi *dataset complete* (Dc) dan *dataset missing* (Dm)

   Dataset complete merupakan *record-record* dari *dataset* yang tidak mengandung *missing value,* sedangkan *dataset missing* merupakan *record-record* dari *dataset* yang tidak mengandung *missing value*.

   D_m = {*record 3,record 4,record 7,record 8, record 9, record 10*}

   D_c = { *record 1, record 2, record 5, record 6, record 1*}

   Pembagian *record dataset* dapat dilihat lebih jelas pada tabel 3.19 dan tabel 3.20.

Tabel 3. 19 Data *missing* (Dm) pada data kualitatif

Record ke-	Gill Size	Gill Color	Stalk Shape	Stalk Root	Stalk Surface Above Ring	Stalk Surface Below Ring	Class
1	b	k	t			f	e
2	b	k		c	s	s	e
5		k	e	e	s	s	p
7	b	k	t		f		e

Tabel 3. 20 Data *complete* (Dc) pada data kualitatif

Record ke-	Gill Size	Gill Color	Stalk Shape	Stalk Root	Stalk Surface Above Ring	Stalk Surface Below Ring	Class
3	b	k	t	e	f	f	e
4	b	n	e	r	s	y	e
6	n	k	e	e	s	s	p
8	b	n	e	r	s	y	e
9	b	n	t	e	f	f	e
10	b	n	t	e	s	s	e

2) Pada setiap *record* Dm

a. Lakukan pembagian atribut-atribut dalam Dm, antara bagian yang hilang (x_m) dan bagian yang ditemukan nilainya dalam *record* tersebut (x_o)

Misalkan untuk *record* ke-1 yang termasuk dalam Dm :

x_o={Gill Size=b; Gill Color=k; Stalk Shape=t; Stalk Surface Below Ring=f}

x_m={Stalk Root=?; Stalk Surface Above Ring=?}

b. Tentukan jarak *Eucledian* antara x_o dengan semua D_c

Pada subbab 2.6 telah dijelaskan prinsip penentuan jarak *Eucledian* untuk data kualitatif. Jarak ditentukan dengan prinsip jika data kualitatif yang dibandingkan berbeda, maka nilai jarak antara data tersebut adalah 1. Namun apabila data

yang dibandingkan sama, maka nilai jarak antara data tersebut adalah 0.

Misalkan ditentukan jarak untuk x_o pada *record* ke-1 dengan semua D_c :

x_o *record* ke-1 = { Size=b; Gill Color=k; Stalk Shape=t; Stalk Surface Below Ring=f}

Pada x_o *record* ke-1, nilai atribut Stalk Root dan Stalk Surface Above Ring adalah *missing value*, maka pada D_c nilai dari atribut Stalk Root dan Stalk Surface Above Ring juga dihilangkan karena atribut yang nilainya digunakan pada perhitungan jarak adalah atribut yang termasuk dalam x_o *record* ke-1.

- Untuk x_o pada D_c *record* ke-3 = { Size=b; Gill Color=k; Stalk Shape=t; Stalk Surface Below Ring=f }
  Jarak = $\sqrt{(0)^2 + (0)^2 + (0)^2 + (0)^2}$ = 0
- Untuk x_o pada D_c *record* ke-4 = { Size=b; Gill Color=n; Stalk Shape=e; Stalk Surface Below Ring=y}
  Jarak = $\sqrt{(0)^2 + (1)^2 + (1)^2 + (1)^2}$ = 1.73
- Untuk x_o pada D_c *record* ke-6 = { Size=n; Gill Color=k; Stalk Shape=e; Stalk Surface Below Ring=s}
  Jarak = $\sqrt{(1)^2 + (0)^2 + (1)^2 + (1)^2}$ = 1.73
- Untuk x_o pada D_c *record* ke-8 = { Size=b; Gill Color=n; Stalk Shape=e; Stalk Surface Below Ring=y}
  Jarak = $\sqrt{(0)^2 + (1)^2 + (1)^2 + (1)^2}$ = 1.73
- Untuk x_o pada D_c *record* ke-9 = { Size=b; Gill Color=n; Stalk Shape=t; Stalk Surface Below Ring=f}
  Jarak = $\sqrt{(0)^2 + (1)^2 + (0)^2 + (0)^2}$ = 1
- Untuk x_o pada D_c *record* ke-10 = { Size=b; Gill Color=n; Stalk Shape=t; Stalk Surface Below Ring=s}
  Jarak = $\sqrt{(0)^2 + (1)^2 + (0)^2 + (1)^2}$ = 1.4

c. Tentukan "k" nilai *Eucledian* terkecil

Misalkan k=3, maka akan dipilih 3 *record* D_c yang memiliki nilai Eucledian terkecil. Karena *missing value* berada pada atribut Stalk Root dan Stalk Surface Above Ring, maka dari

ketiga *record* tersebut diambil nilai dari atribut Stalk Root dan Stalk Surface Above Ring. Tabel 3.21 menampilkan nilai-nilai atribut *missing* dari "k" Dc dengan *eucledian* terkecil.

Tabel 3. 21 Nilai atribut dengan *eucledian* terkecil

Data Complete	Nilai atribut missing *Stalk Root*	Nilai atribut missing *Stalk Surface Above Ring*
Dc record ke-3	e	f
Dc record ke-9	e	f
Dc record ke-10	e	s

d. Tentukan kategori / karakter yang paling sering muncul (Mode)

Kategori/karakter yang paling sering muncul pada tabel 3.21 digunakan sebagai nilai *Mode Imputation* atribut-atribut *missing value* pada *record* tersebut. Jadi, dari tabel 3.21 tersebut *Mode Imputation* untuk atribut-atribut yang merupakan *missing value* pada *record* ke-1 adalah :

- Atribut Stalk Root = e
- Atribut Stalk Surface Above Ring = f

Lakukan proses perulangan hingga dapat ditemukan nilai imputasi untuk semua *missing value*. Nilai imputasi yang dihasilkan dengan metode penanganan *missing value* KNNI dapat dilihat pada tabel 3.22.

Tabel 3. 22 Hasil *KNN Imputation*

Record ke-	Atribut missing (kolom ke-)	Nilai Imputasi
1	4	e
	5	f
2	3	t
5	1	b
7	4	e
	6	f

3.5.3 Perhitungan evaluasi nilai imputasi

Evaluasi dilakukan dengan cara membandingkan nilai hasil imputasi dengan nilai yang sebenarnya dari *missing value* tersebut. Metode untuk menghitung nilai evaluasi pada hasil imputasi terdiri dari 2 macam, yaitu RMSE (*Root Mean Squared Error*) dan *Performance Metric Error Rate*.

a. RMSE

RMSE digunakan untuk mengevaluasi data numerik. RMSE dapat diperoleh dengan persamaan 2.3. Karena RMSE digunakan untuk mengevaluasi data kuantitatif, maka metode evaluasi RMSE ditujukan untuk mengevaluasi hasil imputasi dari metode *Mean Imputation* dan *KNN Imputation*.

- Evaluasi hasil metode *Mean Imputation*

  Pada proses evaluasi ini, nilai hasil imputasi *Mean Imputation* dari contoh perhitungan manual yang ditampilkan pada tabel 3.9 dibandingkan dengan nilai yang sebenarnya dari *data training* yang ditampilkan pada tabel 3.7. Perbandingan tersebut ditunjukkan pada tabel 3.23.

  Tabel 3. 23 Perbandingan nilai *Mean Imputation* dan nilai sebenarnya

Baris	Kolom	Nilai Imputasi (y_{im})	Nilai sebenarnya (y_{true})	$(y_{im}-y_{true})^2$	$(y_{true})^2$
3	3	3	4	1	16
4	1	5	5	0	25
4	5	3	3	0	9
7	4	3	5	4	25
8	1	4	5	1	25
9	2	53	64	121	4096
Jumlah				127	4196

Berdasarkan tabel tersebut, maka nilai evaluasi RMSE adalah:

$$\text{RMSE} = \sqrt{\left(\frac{\sum(\text{yim} - \text{ytrue})2}{6}\right)} \Big/ \sqrt{\frac{(\text{ytrue})2}{6}}$$

$$\text{RMSE} = \sqrt{\left(\frac{127}{6}\right)} \Big/ \sqrt{\frac{4196}{6}}$$
$$= \frac{4.601}{26.445} = 0.174$$

- Evaluasi hasil metode *KNN Imputation*

  Pada proses evaluasi ini, nilai hasil imputasi *KNN Imputation* dari contoh perhitungan manual yang ditampilkan pada tabel 3.13 dibandingkan dengan nilai yang sebenarnya dari *data training* yang ditampilkan pada tabel 3.7. Perbandingan tersebut ditunjukkan pada tabel 3.24.

Tabel 3.24 Perbandingan nilai *KNN Imputation* dengan nilai sebenarnya

Baris	Kolom	Nilai Imputasi (y_{im})	Nilai sebenarnya (y_{true})	$(y_{im}-y_{true})^2$	$(y_{true})^2$
3	3	4	4	0	16
4	1	4	5	1	25
4	5	3	3	0	9
7	4	3	5	4	25
8	1	4	5	1	25
9	2	67	64	9	4096
Jumlah				15	4196

Berdasarkan tabel tersebut, maka nilai evaluasi RMSE adalah:

$$\text{RMSE} = \sqrt{\left(\frac{\sum(\text{yim} - \text{ytrue})2}{6}\right)} \Big/ \sqrt{\frac{(\text{ytrue})2}{6}}$$

$$\text{RMSE} = \sqrt{\left(\frac{15}{6}\right)} \Big/ \sqrt{\frac{4196}{6}}$$
$$= \frac{1.581}{26.445} = 0.06$$

b. *Performance Metric Error Rate*

Performance Metric Error Rate digunakan untuk mengevaluasi data *kualitatif*. *Error Rate* dapat diperoleh dengan persamaan 2.4. Karena tingkat kesalahan digunakan untuk mengevaluasi data kualitatif, maka metode evaluasi tingkat kesalahan ditujukan untuk

mengevaluasi hasil imputasi dari metode *Mode Imputation* dan *KNN Imputation*.

- Evaluasi hasil metode *Mode Imputation*

  Pada proses evaluasi ini, nilai hasil imputasi *Mode Imputation* dari contoh perhitungan manual yang ditampilkan pada tabel 3.18 dibandingkan dengan nilai yang sebenarnya dari *data training* yang ditampilkan pada tabel 3.16. Perbandingan tersebut ditunjukkan pada tabel 3.25.

Tabel 3.25 Perbandingan nilai *Mode Imputation* dengan nilai sebenarnya

Baris	Kolom	Nilai Imputasi	Nilai sebenarnya	Nilai prediksi salah
1	4	e	e	0
1	5	s	s	0
2	3	t	e	1
5	1	n	n	0
7	4	e	e	0
7	6	f	s	1
Jumlah prediksi salah				2

Berdasarkan tabel tersebut, maka nilai evaluasi tingkat kesalahan adalah:

$$Error\ Rate = \frac{2}{6} = 0.33$$

- Evaluasi hasil metode *KNN Imputation*

  Pada proses evaluasi ini, nilai hasil imputasi *KNN Imputation* dari contoh perhitungan manual yang ditampilkan pada tabel 3.22 dibandingkan dengan nilai yang sebenarnya dari *data training* yang ditampilkan pada tabel 3.16. Perbandingan tersebut ditunjukkan pada tabel 3.26.

  Berdasarkan tabel tersebut, maka nilai evaluasi tingkat kesalahan adalah:

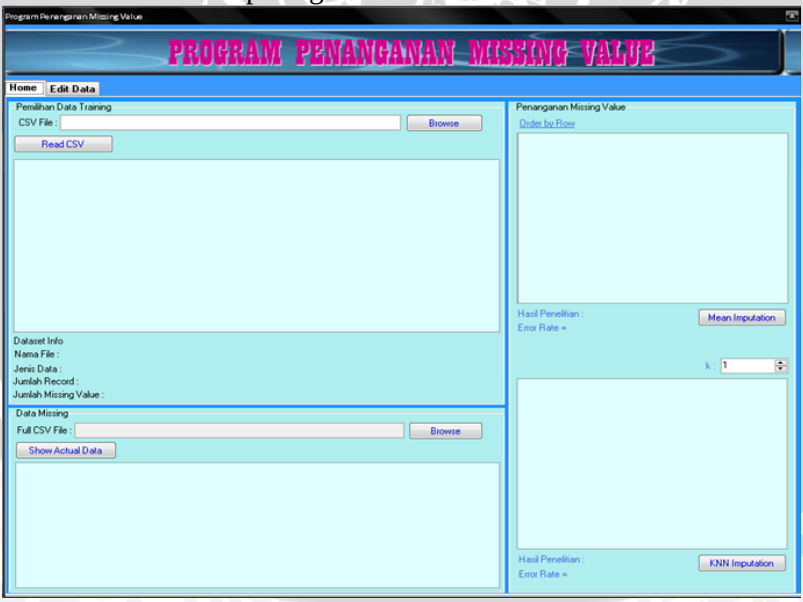
$$Error\ Rate = \frac{4}{6} = 0.67$$

Tabel 3. 26 Perbandingan nilai *KNN Imputation* dengan nilai sebenarnya

Baris	Kolom	Nilai Imputasi	Nilai sebenarnya	Nilai prediksi salah
1	4	e	e	0
1	5	f	s	1
2	3	t	e	1
5	1	b	n	1
7	4	e	e	0
7	6	f	s	1
Jumlah				4

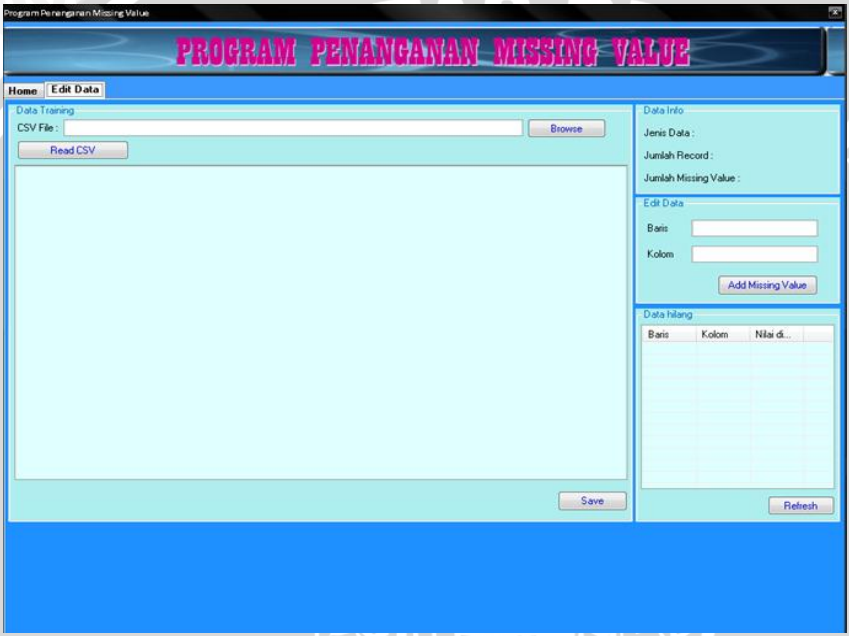
3.6 Perancangan Antarmuka (*interface*)

Antarmuka pada pengujian ini terdiri dari 2 *form*, yaitu *form Home* dan *form Edit*. *Form Home* merupakan *form* utama yang menampilkan menu bagi *user* untuk memilih *data training* ber-*missing value* yang akan diuji, *data training* lengkap sebagai data pembanding, serta menampilkan hasil dari pengujian sistem. *Form Home* diilustrasikan pada gambar 3.11.



Gambar 3. 11 *Form Home*

Sedangkan *form Edit* merupakan *form* yang menampilkan menu bagi *user* untuk memilih *data training* lengkap yang akan dilakukan pengosongan beberapa nilai. Selain itu pada *form* tersebut ditampilkan berbagai informasi mengenai *data training*, menu untuk melakukan pengosongan terhadap beberapa nilai pada *data training*, serta posisi-posisi dan nilai-nilai data yang telah dikosongi. *Form Edit* diilustrasikan pada gambar 3.12.



Gambar 3. 12 *Form Edit*

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

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

4.1.1 Lingkungan implementasi perangkat keras

Perangkat keras yang digunakan dalam penanganan *missing value* dengan metode *Mean Imputation, Mode Imputation* dan *k-Nearest Neighbor* ini adalah:

1. Prosesor Intel Core 2 Duo
2. Memori 2 GB
3. Harddisk dengan kapasitas 250 GB
4. Monitor 12”
5. *Keyboard*
6. *Mouse*

4.1.2 Lingkungan implementasi perangkat lunak

Perangkat lunak yang digunakan dalam penanganan *missing value* dengan metode *Mean Imputation, Mode Imputation* dan *k-nearest Neighbors* ini adalah :

1. Sistem Operasi *Microsoft Windows XP Professional*
2. *Microsoft Visual Studio Professional* 2008 dengan bahasa pemrograman C#
3. *Microsoft Office Excel* 2007

4.2 Implementasi Program

Pada subbab implementasi program ini akan dijelaskan mengenai implementasi dari rancangan perangkat lunak yang dijelaskan sebelumnya pada subbab 3.3.

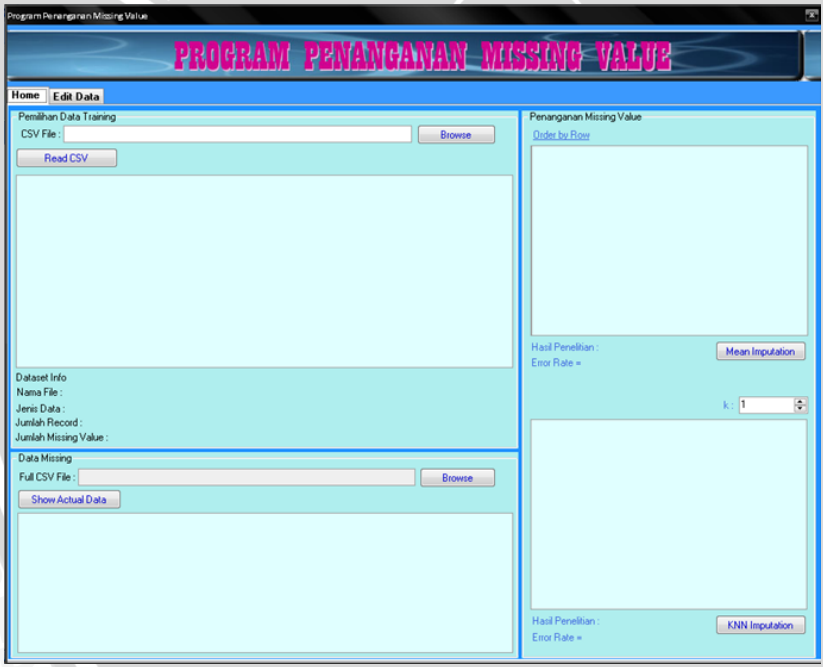
4.2.1 Implementasi antarmuka

Implementasi antarmuka dari sistem terdiri dari dua buah *form*. *Form* pertama yaitu *form Home* yang merupakan implementasi antarmuka untuk proses *request* data, proses penanganan *missing*

value, serta proses penghitungan tingkat kesalahan. *Form* kedua merupakan *form Edit Data*. Pada *form* tersebut *user* dapat melakukan pengosongan beberapa nilai secara langsung terhadap *data training* terpilih.

1. *Form Home*

Form Home merupakan *form* yang pertama kali muncul. Pada *form* tersebut *user* dapat memilih *data training* ber-*missing value* untuk selanjutnya dilakukan penanganan terhadapnya. Pada *form* tersebut pula dapat dilihat hasil penanganan serta hasil proses penghitungan tingkat kesalahan. *Form Home* diilustrasikan pada gambar 4.1.



Gambar 4. 1 *Form Home*

Pada *form Home* terdapat tombol *Browse* untuk memilih *file data training* yang ber-*missing value* dan kemudian dilakukan pembacaan pada *file* tersebut dengan menekan tombol *Read CSV* sehingga data dapat muncul pada *listview*. Nilai-nilai yang sebenarnya dari *missing value* tersebut dapat diketahui dengan memilih *file data training*

yang tidak mengandung *missing value* dan kemudian menekan tombol *Show Actual Data* untuk menampilkan nilai yang sebenarnya dari *missing value* pada *listview*. Pada *form* tersebut juga ditampilkan info tentang *data training* ber-*missing value* yang dipilih. Info yang ditampilkan diantaranya adalah nama *file*, jenis data, jumlah *record* dan jumlah *missing value*. Ketika dilakukan pemilihan kedua *data training* tersebut, maka *data training* akan ditampilkan pada *listview* seperti pada gambar 4.2.

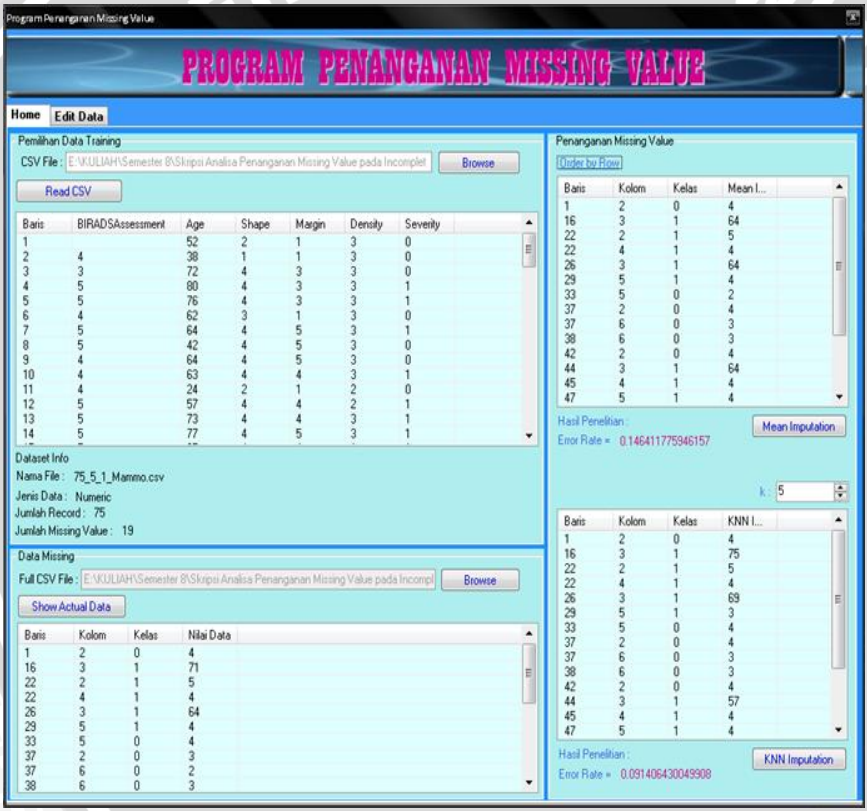
Gambar 4. 2 Tampilan *data training* pada *form Home*

Untuk melakukan penanganan *missing value*, *user* dapat menekan tombol *Mode Imputation* (untuk data kuantitatif) atau *Mean Imputation* (untuk data kualitatif) dan *KNN Imputation*. Sebelum menekan tombol *KNN Imputation*, *user* terlebih dahulu menentukan nilai k yang akan digunakan. Hasil penanganan *missing value* ditampilkan pada *listview* seperti pada gambar 4.3. Pada *form*

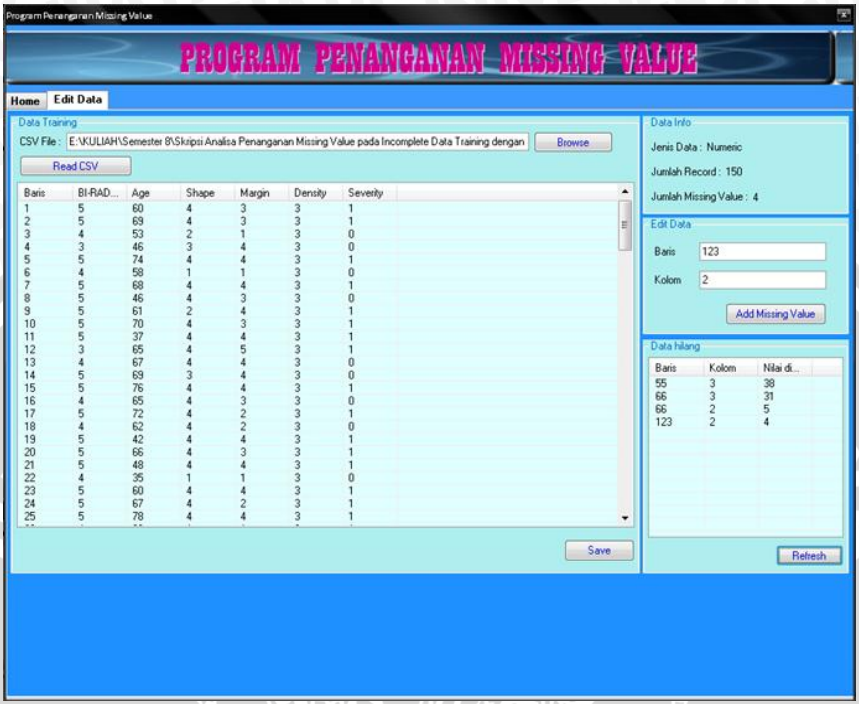
tersebut juga ditampilkan nilai kesalahan dari hasil penanganan setelah dicocokkan dengan nilai sebenarnya dari *missing value*.

2. *Form Edit Data*

Pada *form Edit Data user* dapat memilih *data training* tanpa *missing value*. Selanjutnya *user* dapat melakukan pengosongan nilai pada beberapa posisi yang dipilih dan menyimpan hasil pengubahan pada *data training* tersebut. *Form Edit Data* diilustrasikan pada gambar 4.4.



Gambar 4. 3 Tampilan hasil penanganan *missing value* pada *form Home*



Gambar 4. 4 *Form Edit Data*

4.2.2 Implementasi kelas

Pada implementasi program, daftar program dibuat secara modular yang diimplementasikan dalam bentuk kelas-kelas. Pada pembangunan sistem ini terdapat lima buah kelas, yaitu : `class Data`, `class Position`, `class MeanImputation`, `class ModeImputation` dan `class KNNImputation`.

1. Kelas `Data`

Kelas Data berisi struktur data dari *data training*, prosedur, serta fungsi-fungsi yang berkaitan dengan operasi pada *data training*. Kelas `Data` disimpan pada *file* Data.cs. Struktur kelas **`Data`** dapat dilihat pada gambar 4.5

Struktur data yang digunakan pada kelas tersebut adalah struktur data *list*, *array*, dan tipe data primitif seperti *integer* dan *string*. **`PosisiData`** merupakan array 2 dimensi untuk menampung objek-

objek yang menyimpan informasi tentang posisi, nilai, serta kelas dari semua data pada *data training*, termasuk data yang hilang. **PosisiMV** merupakan *list* objek **Position** yang dapat menyimpan infomasi tentang posisi serta kelas yang dimiliki oleh data yang hilang tersebut. **PosisiVal** juga merupakan *List* berisi objek **Position** yang dapat menyimpan infomasi tentang posisi, nilai, serta kelas dari suatu data pada *data training*, tidak termasuk *missing value*. **DistinctMV** merupakan list objek **Position** yang menyimpan infomasi tentang kolom dan kelas dari missing value. Atribut **DataComp** merupakan *list integer* untuk menyimpan data posisi baris data yang tidak mengandung *missing value* atau komplit. Variabel **jum_row** dan **jum_col** masing-masing menyimpan informasi tentang baris dan kolom data. Variabel **csvline** merupakan *array string* yang menyimpan informasi tentang data-data hasil pembacaan *file* csv.

```
1   public class Data
2   {
3       public Position[,] PosisiData;
4       public List<Position> PosisiMV ;
5       public List<Position> PosisiVal ;
6       public List<Position> DistinctMV ;
7       public List<int> DataComp ;
8       public int jum_row;
9       public int jum_col;
10
11      public void ConvertNilai(string[] csvline)
12      public void GetData()
13      public void GetMV()
14      public int getJumlahMV()
15      public void GetDistinctMV()
16      public int NilaiToInt(int baris,int kolom)
17  }
```

Gambar 4. 5 *Listing* struktur data, prosedur dan fungsi class Data

Untuk penjelasan mengenai prosedur dan fungsi-fungsi pada kelas **Data**, disajikan pada tabel 4.1.

Tabel 4. 1 Deskripsi prosedur dan fungsi pada kelas `Data`

Nama prosedur/fungsi	Deskripsi
`public void ConvertNilai (string[] csvline)`	Fungsi untuk merubah bentuk data csv menjadi bentuk *array* 2 dimensi
`public void GetData()`	Fungsi untuk menyimpan informasi tentang data-data, kemudian menyimpannya dalam *list* objek
`public void GetMV()`	Fungsi untuk menyimpan posisi dari data yang nilainya hilang (*missing*), dan menyimpannya dalam *list* objek
`public int getJumlahMV()`	Fungsi untuk menghitung jumlah *missing value*
`public void GetDistinctMV()`	Fungsi untuk mengetahui kolom dan kelas dari data hilang (*missing value*)
`public int NilaiToInt(int baris,int kolom)`	Fungsi untuk mengubah bentuk nilai data yang tersimpan dalam bentuk *aray* 2 dimensi yang bertipe *string* menjadi nilai *integer*

a. Fungsi `ConvertNilai`

Input *data training* pada program adalah berupa *file* CSV. Namun pada proses selanjutnya tidak memungkinkan untuk menggunakan secara langsung data dari *file* tersebut. Oleh karena itu diimplementasikan fungsi yang dapat mengubah bentuk data dari *file* CSV menjadi bentuk *array* objek 2 dimensi. Fungsi tersebut adalah fungsi **`ConvertNilai`** yang ditunjukkan pada gambar 4.6.

Pada mulanya data-data pada setiap *record file* csv dipisahkan oleh tanda koma. Untuk menghilangkan tanda koma tersebut digunakan *method split*. Setelah tanda koma dihilangkan, data kemudian dipisahkan dan disimpan dalam *array* objek 2 dimensi **`PosisiData`**. Pada *array* objek 2 dimensi tersebut disimpan informasi mengenai baris, kolom, nilai dan kelas seluruh data. Pada fungsi tersebut juga dilakukan penyeleksian *record-record* yang

tidak mengandung *missing value*. Hasil penyeleksian tersebut disimpan pada list *integer*.

```
1   public void ConvertNilai(string[] csvline)
2   {
3     string[] dt = csvline[0].Split(',');
4     jum_row = csvline.Count();
5     jum_col = dt.Count();
6     DataComp.Clear();
7     PosisiData = new Position[jum_row,jum_col];
8     for (int x = 0; x < jum_row; x++)
9     {
10      dt = csvline[x].Split(',');
11      int jum_miss = 0;
12      for (int y = 0; y < jum_col; y++)
13      {
14        PosisiData[x,y] = new Position(x, y, dt[y],
15  dt[jum_col - 1]);
16        if (dt[y].Equals(""))
17        {
18           jum_miss++;
19           PosisiData[x,y].empty_data = true;
20        }
21      }
22      if (jum_miss == 0)
23      { DataComp.Add(x);}
24  }
```

Gambar 4. 6 Fungsi ConvertNilai

b. Fungsi `GetData`

Untuk mengelompokkan data yang tidak hilang maka dilakukan proses penyeleksian terhadap *data training*. Untuk melakukan proses tersebut maka diimplementasikan fungsi **`GetData`** yang ditunjukkan pada gambar 4.7.

Pada fungsi tersebut dilakukan penyeleksian terhadap setiap data. Jika variabel `empty_data` pada *array* **`PosisiData`** bernilai *false*, maka data pada posisi tersebut tidak hilang dan memiliki nilai. Data tersebut kemudian dikelompokkan ke dalam *list* **`PosisiVal`**. Dalam **`PosisiVal`** informasi-informasi yang disimpan diantaranya adalah baris, kolom, nilai dan kelas dari data yang terseleksi tersebut.

```
1   public void GetData()
2   {
3     for (int h = 0; h < PosisiData.GetLength(0); h++)
4     {
5       for(int l = 0; l < PosisiData.GetLength(1); l++)
6       {
7         if(PosisiData[h, l].empty_data == false)
8         {PosisiVal.Add(PosisiData[h, l]);}
9       }
10    }
11  }
```

Gambar 4. 7 Fungsi GetData

c. Fungsi `GetMV`

Untuk mengelompokkan data yang hilang (*missing*) maka dilakukan proses penyeleksian terhadap *data training*. Untuk melakukan proses tersebut maka diimplementasikan fungsi **`GetMV`** yang ditunjukkan pada gambar 4.8.

Pada fungsi tersebut dilakukan penyeleksian terhadap setiap data. Jika variabel `empty_data` pada *array* **`PosisiData`** bernilai *true*, maka data pada posisi tersebut adalah *missing* data. Data tersebut kemudian dikelompokkan ke dalam *list* **`PosisiMV`**. Dalam **`PosisiMV`** informasi-informasi yang disimpan diantaranya adalah baris, kolom, nilai dan kelas dari data yang terseleksi tersebut.

```
1   public void GetMV()
2   {
3      PosisiMV.Clear();
4      for (int h = 0; h < PosisiData.GetLength(0); h++)
5      {
6         for(int l = 0;l< PosisiData.GetLength(1);l++)
7         {
8           if (PosisiData[h, l].empty_data == true)
9           {
10             PosisiMV.Add(PosisiData[h, l]);
11          }
12        }
13     }
14  }
```

Gambar 4. 8 Fungsi GetMV

d. Fungsi `getJumlahMV`

Untuk mempermudah mengetahui jumlah data yang *missing*, maka diimplementasikan fungsi **`getJumlahMV`**. Pada fungsi tersebut digunakan *method count.* Fungsi tersebut ditunjukkan pada gambar 4.9.

```
1   public int getJumlahMV()
2   {
3       return PosisiMV.Count;
4   }
```

Gambar 4. 9 Fungsi getJumlahMV

e. Fungsi `GetDistinctMV`

Untuk menentukan kolom dan kelas dari data *missing* diperlukan hasil dari proses pengelompokan data hilang yang telah dijelaskan pada fungsi **`GetMV`**. Dalam proses pengelompokan data hilang tersebut hasil dari proses disimpan dalam *list* **`PosisiMV`**. Dalam penentuan kolom dan kelas dari data *missing* diimplementasikan fungsi **`GetDistinctMV`** yang dapat diamati pada gambar 4.10.

```
1   public void GetDistinctMV()
2   {
3     var distinc =(from k in PosisiMV
4                   orderby k.Y, k.co
5                   select new { k.Y, k.co}).Distinct();
6
7     foreach (var t in distinc)
8     {
9        DistinctMV.Add(new Position(t.Y, t.co));
10    }
11  }
```

Gambar 4. 10 Fungsi GetDistinctMV

f. Fungsi `NilaiToInt`

Seluruh nilai data pada program penelitian pada awalnya diubah menjadi data bertipe *string*. Hal tersebut dilakukan untuk mempermudah proses-proses yang dilakukan pada fungsi **`ConvertNilai`**. Pengubahan nilai data bertipe *string* menjadi *integer* hanya dilakukan pada *data training* yang memiliki tipe data kuantitatif. Hal tersebut dilakukan terkait dengan proses penanganan *missing value* yang menggunakan nilai *integer* tersebut. Proses

penanganan *missing value* yang memerlukan nilai bertipe *integer* adalah *Mean Imputation* dan *KNN Imputation*. Untuk melakukan proses pengubahan tipe data tersebut diimplementasikan fungsi **`NilaiToInt`** yang dapat diamati pada gambar 4.11.

```
1   public int NilaiToInt(int baris,int kolom)
2   {
3      int nil;
4      if (PosisiData[baris, kolom].nilai.Equals(""))
5      {
6          nil = 0;
7      }
8      else
9      {
10      nil=Convert.ToInt32(PosisiData[baris,kolom].nilai);
11     }
12     return nil;
13  }
```

Gambar 4. 11 Fungsi NilaiToInt

2. Kelas `Position`

```
1   public class Position
2   {
3      public int X {get; set;}
4      public int Y { get; set; }
5      public string nilai { get; set; }
6      public string co { get; set; }
7      public bool empty_data { get; set; }
8      public double euc {get; set; }
9
10     public Position(int X, int Y, string nilai, string co)
11     public Position(int X, double euc)
12     public Position(int Y, string co)
13     public Position(int X, int Y, string nilai)
14  }
15
```

Gambar 4. 12 *Listing* struktur data dan konstruktor kelas Position

Kelas **`Position`** berisi konstruktor-konstruktor kelas dengan parameter berbeda yang berfungsi untuk menyimpan berbagai informasi tentang *data training*. Hal ini disebut dengan *overloading* konstruktor. Setiap kali objek diinisialisasi, maka konstruktor kelas dijalankan. Konstruktor yang dijalankan tergantung pada instansiasi objek yang dilakukan sesuai dengan parameter yang dimiliki

konstruktor kelas. Struktur kelas **`Position`** dapat diamati pada gambar 4.12. Untuk penjelasan tentang konstruktor kelas dan parameternya dapat dilihat pada tabel 4.2.

Tabel 4. 2 Deskripsi konstruktor pada kelas `Position`

Nama konstruktor	Deskripsi
`public Position(int X, int Y, string nilai, string co)`	Konstruktor untuk menyimpan informasi tentang *data training*, yaitu baris data, kolom data, nilai data dan kelas data
`public Position(int X, double euc)`	Konstruktor untuk menyimpan informasi tentang *data training*, yaitu baris data dan nilai *eucledian* yang digunakan pada metode *KNN Imputation*
`public Position(int Y, string co)`	Konstruktor untuk menyimpan informasi tentang *data training*, yaitu kolom data dan kelas data

3. Kelas `MeanImputation`

Kelas MeanImputation berisi struktur data serta fungsi yang berkaitan dengan proses yang terjadi pada metode penanganan *missing value Mean Imputation* khusus untuk data bertipe numerik. Struktur data dan fungsi kelas **`MeanImputation`** dapat dilihat pada gambar 4.13.

```
1  public class MeanImputation : Data
2  {
3     public List<int> BarisMV = new List<int>();
4     public List<Position> PosisiAvg = new List
5     <Position>();
6
7     public void GetAverage(string[]
8     csvdata)
   }
```

Gambar 4. 13 *Listing* struktur data dan fungsi kelas MeanImputation

Kelas **`MeanImputation`** merupakan kelas turunan dari kelas Data. Proses pada metode penanganan *Mean Imputation* menggunakan data-data hasil pengolahan dari kelas **`Data`**. Oleh

karena itu, kelas **MeanImputation** merupakan kelas turunan dari kelas **Data**.

Struktur data yang digunakan pada kelas tersebut adalah *list*. **BarisMV** merupakan *list integer* yang dapat menyimpan baris data yang menyimpan *missing value*. Sedangkan **PosisiAvg** merupakan *list* objek **Position** yang dapat menyimpan informasi tentang baris, kolom dan kelas yang dimiliki oleh data yang hilang, serta nilai pengganti dari data yang hilang tersebut. Nilai pengganti tersebut merupakan hasil dari penanganan *missing value* dengan *Mean Imputation*.

```
1  public void GetAverage(string[] csvdata)
2  {
3      foreach (var r in DistinctMV)
4      {
5          total = 0; pembagi = 0;
6          for (c = 0; c < jum_row; c++)
7          {
8              if (PosisiData[c,r.Y].nilai != "" &&
9              PosisiData[c, jum_col-1].nilai == r.co)
10             {
11                 total = total + NilaiToInt(c,r.Y);
12                 pembagi++;
13             }
14             else if
15             (PosisiData[c,r.Y].empty_data==true &&
16             PosisiData[c,jum_col-1].nilai==r.co)
17             {
18                  BarisMV.Add(c);
19             }
20         }
21         ave = total / pembagi;
22
23         foreach (int brs in BarisMV)
23         {
25           PosisiAvg.Add(new Position(brs,r.Y,
26           ave.ToString(), r.co));
27         }
28     }
29 }
```

Gambar 4. 14 Fungsi GetAverage

Fungsi yang terdapat pada kelas `MeanImputation` hanya fungsi **GetAverage**. Parameter dari fungsi tersebut adalah `csvdata`. Fungsi tersebut digunakan untuk melakukan proses untuk

menemukan nilai pengganti dari *missing value* sesuai dengan metode *Mean Imputation*. Data yang diolah adalah *data training* yang telah tersimpan pada variabel **csvdata**. Fungsi **GetAverage** dapat dilihat pada gambar 4.14. Fungsi tersebut mengimplementasikan persamaan 2.6 pada subbab 2.7.1. Hasil dari *Mean Imputation* disimpan dalam *list* **PosisiAvg** beserta informasi baris, kolom, dan kelas dari data yang hilang.

4. Kelas ModeImputation

Kelas **ModeImputation** berisi struktur data serta fungsi yang berkaitan dengan proses yang terjadi pada metode penanganan *missing value Mode Imputation* khusus pada data bertipe kualitatif. Struktur data dan fungsi kelas **ModeImputation** dapat dilihat pada gambar 4.15.

```
1  public class ModeImputation : Data
2  {
3     public List<Position> PosisiMod = new List
4     <Position>();
5
6     public void GetMode (string[] csvdata)
7  }
```

Gambar 4.15 *Listing* struktur data dan fungsi kelas ModeImputation

Kelas **ModeImputation** merupakan kelas turunan dari kelas **Data**. Proses pada metode penanganan *Mode Imputation* menggunakan data-data hasil pengolahan dari kelas **Data**. Oleh karena itu, kelas **ModeImputation** merupakan kelas turunan dari kelas **Data.**

Struktur data yang digunakan pada kelas tersebut adalah list. **PosisiMod** merupakan *list* objek **Position** yang dapat menyimpan informasi tentang baris,kolom dan kelas yang dimiliki oleh data yang hilang, serta nilai pengganti dari data yang hilang tersebut. Nilai pengganti tersebut merupakan hasil dari penanganan *missing value* dengan *Mode Imputation*.

Fungsi yang terdapat pada kelas **ModeImputation** hanya fungsi **GetMode**. Parameter dari fungsi tersebut adalah **csvdata**. Fungsi tersebut digunakan untuk melakukan proses untuk menemukan nilai pengganti dari *missing value* sesuai dengan metode *Mode Imputation*. Data yang diolah adalah *data training* yang telah

tersimpan pada variabel **csvdata**. Fungsi **GetMode** dapat diamati pada gambar 4.16. Hasil dari *Mode Imputation* disimpan dalam *list* PosisiMod beserta informasi baris, kolom, dan kelas dari data yang hilang.

```
public void GetMode (string[] csvdata)
{
    foreach (var r in DistinctMV)
    {
      var group =(from f in PosisiVal
                  where f.Y == r.Y
                  select f.nilai).Distinct();

      //atribut-atribut untuk menyimpan info Mode
      int maks = 0; string kar_maks = "";
      foreach (var p in group)
      {
         jum_kar = 0;
         for (int q = 0; q < jum_row; q++)
         {
           if (PosisiData[q, r.Y].nilai == p &&
           PosisiData[q, jum_col - 1].nilai ==
           r.co)
           {jum_kar++;}
         }
         if (jum_kar>maks)
         {
            kar_maks=p;
            maks=jum_kar;
         }
      }
     //linq untuk menyimpan baris yang mengandung MV
      var baris = from br in PosisiMV
                  where br.Y == r.Y && br.co==r.co
                  select br.X;
      foreach (var brs in baris)
      {
         PosisiMod.Add(new Position(brs,r.Y,
         kar_maks, r.co));
      }}}
```

Gambar 4. 16 Fungsi GetMode

5. Kelas KNNImputation

Kelas **KNNImputation** berisi struktur data serta fungsi yang berkaitan dengan proses yang terjadi pada penanganan *missing value K-Nearest Neighbor Imputation*. Metode tersebut dapat diterapakan

pada dua jenis tipe data yaitu kuantitatif (*integer*) dan kualitatif. Struktur data dan fungsi kelas **KNNImputation** dapat dilihat pada gambar 4.17.

```
1  public class KNNImputation : Data
2  {
3      public List<Position> PosisiEuc = new List
4      <Position>();
5      public List<Position> PosisiKNNI_num = new List
6      <Position>();
7      public List<Position> PosisiKNNI_kar = new List
8      <Position>();
9      public   List<string>   karakter   =   new   List
10     <string>();
11
12     public int ratarata(int jum, int pembagi)
13     public int kuadrat(int jum)
14     public double eucledian(double nilai_kuadrat)
15     public  void  GetKNN_Num(string[]  csvdata,  int
16     jum_k)
17     public  void  GetKNN_Cat(string[]  csvdata,  int
18     jum_k)
19 }
```

Gambar 4. 17 *Listing* struktur data dan fungsi kelas KNNImputation

Kelas **KNNImputation** merupakan kelas turunan dari kelas **Data**. Proses pada metode penanganan *KNN Imputation* menggunakan data-data hasil pengolahan dari kelas **Data**. Oleh karena itu, kelas **KNNImputation** merupakan kelas turunan dari kelas **Data**.

Struktur data yang digunakan pada kelas tersebut adalah *list*. **PosisiEuc** merupakan list objek **Position** yang dapat menyimpan baris data yang mengandung *missing value* serta nilai *eucledian* dari baris tersebut. **PosisiKNNI_num** dan **PosisiKNNI_kar** merupakan *list* objek **Position** yang dapat menyimpan baris, kolom dan kelas dari *missing value*, serta nilai hasil penanganannya menggunakan metode penanganan *KNN Imputation*. **PosisiKNNI_num** digunakan untuk data yang bertipe kuantitatif, sedangkan **PosisiKNNI_kar** digunakan untuk data bertipe kualitatif. Karakter merupakan *list integer* yang menyimpan data yang terletak pada baris yang termasuk dalam jarak *eucledian* terkecil, serta pada kolom yang mengandung *missing value*.

Untuk penjelasan mengenai prosedur dan fungsi-fungsi pada class **`Data`**, disajikan pada tabel 4.3.

Tabel 4. 3 Deskripsi fungsi kelas `KNNImputation`

Nama Prosedur/Fungsi	Deskripsi
`public int ratarata(int jum, int pembagi)`	Fungsi untuk menghitung nilai rata-rata pada data bertipe kuantitatif
`public int kuadrat(int jum)`	Fungsi untuk menghitung nilai kuadrat yang digunakan pada penentuan nilai jarak *eucledian*
`public double eucledian (double nilai_kuadrat)`	Fungsi untuk menghitung nilai jarak *eucledian* antara data *missing* (Dm) dengan data *complete* (Dc)
`public void GetKNN_Num (string[] csvdata, int jum_k)`	Fungsi untuk menentukan nilai penanganan *missing value* pada data bertipe kuantitatif
`public void GetKNN_Cat (string[] csvdata, int jum_k)`	Fungsi untuk menentukan nilai penanganan *missing value* pada data bertipe kualitatif

KNN Imputation yang diimplementasikan pada program telah disesuaikan dengan penjelasan algoritma *KNN Imputation* pada subbab 2.7.3. Dalam *KNN Imputation* dilakukan imputasi dengan mempertimbangkan nilai yang diberikan oleh sejumlah *record* yang paling mirip. Kemiripan dari sejumlah *record* ditentukan dengan fungsi jarak *Eucledian* seperti pada persamaan 2.2. Berbeda dengan *Mean Imputation* dan *Mode Imputation*, *KNN Imputation* dapat diterapkan pada data bertipe kuantitatif maupun kualitatif. Dalam proses penanganan *missing value* dengan *KNN Imputation* digunakan data yang telah tersimpan dalam *array* **`PosisiData`**.

1) *KNN Imputation* pada data kuantitatif

Hasil pengolahan data yang telah diperoleh pada proses-proses sebelumnya digunakan untuk menentukan nilai *KNN Imputation*. Perhitungan nilai *KNN Imputation* untuk tipe data kuantitatif diimplemetasikan pada fungsi **`GetKNN_Num`** yang dapat diamati pada gambar 4.18. Hal yang membedakan pada penghitungan *KNN Imputation* untuk data kuantitatif dilakukan penghitungan terhadap nilai rata-rata dari sejumlah *neighbor* terdekat. Hasil *KNN Imputation* pada data kuantitatif selanjutnya disimpan pada *list* **`PosisiKNNI_num`** beserta informasi baris, kolom dan kelas dari data yang hilang.

```
1   public void GetKNN_Num(string[] csvdata, int jum_k)
2   {
3      int sum, nil_kuadrat;
4      double mean, euc;
5      ConvertNilai(csvdata);
6
7      //Fungsi untuk menyeleksi posisi data missing
8      GetMV();
9
10     //linq untuk mencari baris record yg mengandung
11     missing value
12     var miss = (from b in PosisiMV
13                 select new { b.X }).Distinct();
14
15     foreach (var m in miss)
16     //pada setiap baris yang mengandung MV
17     {
18       //linq untuk memilih kolom yg mengandung MV
19       pada baris tertentu
20       var kolom_MV = from colMV in PosisiMV
21                      where colMV.X == m.X
22                      select colMV.Y;
23       PosisiEuc.Clear();
24       foreach (int u in DataComp)
25       //pada setiap baris yang complete(tidak
26       mengandung MV)
27       {
28          nil_kuadrat = 0;
29          for (int kol = 0; kol < jum_col-1; kol++)
30          {
31            //jika data yg ditemui merupakan MV
32            if (NilaiToInt(m.X, kol) == 0)
33            { sum = 0; }
```

```
34              else
35              {
36                  sum = NilaiToInt(m.X, kol)-
37                  NilaiToInt(u,kol);
38              }
39              nil_kuadrat = nil_kuadrat + kuadrat(sum);
40          }
41
42          //Hitung Eucledian
43          euc = eucledian(nil_kuadrat);
44          PosisiEuc.Add(new Position(u, euc));
45      }
46
47      //Mengurutkan data pada PosisiEuc dari jml euc
48      terkecil serta memilih barisnya saja
49      var euc_order = from h in PosisiEuc
50                      orderby h.euc ascending
51                      select h.X;
52
53      //convert hsl linq euc_order mjd array int
54      int[] arr_euc=euc_order.ToArray();
55
56      foreach (var h in kolom_MV)
57      //untuk setiap kolom yg mengandung MV
58      {
59        int jum = 0;
60        for (int f = 0; f < jum_k; f++)
61        //pilih sejumlah record Dc min sebanyak k
62        {
63            jum = jum + NilaiToInt(arr_euc[f], h);
64        }
65        //menghitung rata-rata dari k neighbour
66        terdekat(euc min)
67         mean = Math.Round(average(jum, jum_k));
68        PosisiKNNI_num.Add(new Position(m.X, h,
69        mean.ToString(), PosisiData[m.X, jum_col -
70        1].nilai));
71       }
    }}
```

Gambar 4.18 Fungsi GetKNN_Num

2) *KNN Imputation* pada data kualitatif

Hasil pengolahan data yang telah diperoleh pada proses-proses sebelumnya digunakan untuk menentukan nilai *KNN Imputation*. Perhitungan nilai *KNN Imputation* untuk tipe data kualitatif diimplemetasikan pada fungsi **`GetKNN_Num`** yang dapat diamati pada

gambar 4.19. Hal yang membedakan pada penghitungan *KNN Imputation* untuk data kualitatif dilakukan penentuan Mode dari sejumlah *neighbor* terdekat.

Pada fungsi **`GetKNN_Cat`** juga dilakukan pemanggilan terhadap fungsi *eucledian* untuk menghitung nilai rata-rata dari sejumlah *neighbor* terdekat. Hasil *KNN Imputation* pada data kualitatif selanjutnya disimpan pada *list* **`PosisiKNNI_kar`** beserta informasi baris, kolom dan kelas dari data yang hilang.

```
1  public void GetKNN_Cat (string[] csvdata, int jum_k)
2  {
3    //linq untuk mencari baris record yg mengandung
4    missing value(Dm)
5    var miss = (from b in PosisiMV
6                select new { b.X }).Distinct();
7
8    foreach (var m in miss)
9    {
10     //linq untuk memilih kolom yg mengandung MV pada
11     baris tertentu (yg mengandung MV)
12     var kolom_MV = from colMV in PosisiMV
13                    where colMV.X == m.X
14                    select colMV.Y;
15
16     PosisiEuc.Clear();
17     foreach (int u in DataComp)
18     {
19       jml_jarak = 0;
20       for (int kol = 0; kol < jum_col - 1; kol++)
21       {
22         //jika data yg ditemui merupakan MV atau
23         milai Dc dam Dm sama
24         if (PosisiData[m.X, kol].nilai.Equals("")||
25         PosisiData[m.X, kol].nilai == PosisiData
26         [u, kol].nilai)
27         {
28           jml_jarak = jml_jarak + 0;
29         }
30
31         else if (PosisiData[m.X, kol].nilai !=
32         PosisiData[u, kol].nilai)
33         {
34            jml_jarak = jml_jarak + 1;
35         }
36       }
37       //Hitung Eucledian
38       euc = eucledian(jml_jarak);
```

```
        PosisiEuc.Add(new Position(u, euc));
      }
      //Mengurutkan data pada PosisiEuc dari jml euc
      terkecil serta memilih barisnya saja

      var euc_order = from h in PosisiEuc
                      orderby h.euc ascending
                      select h.X;

      //convert hsl linq euc_order mjd array int
      int[] arr_euc = euc_order.ToArray();

      Karakter.Clear();
      foreach (var h in kolom_MV)
      {
        //proses deteksi karakter2 yg ada pada baris
        euc_min & kolom MV
        for (int d = 0; d < jum_k; d++)
        {
          Karakter.Add(PosisiData[arr_euc[d],
          h].nilai);
        }
        //seleksi karakter2 yang terdeteksi
        var group = Karakter.Distinct();
        int jum_maks_kar = 0; string kar_maks = "";
        foreach (var gr_krktr in group)
        {
         int jum_kar = 0;
         for (int y = 0; y < jum_k; y++)
         {
          if(PosisiData[arr_euc[y],h].nilai==gr_krktr)
            {
               jum_kar++;
            }
         }
         if (jum_kar > jum_maks_kar)
         {
           kar_maks = gr_krktr;
           jum_maks_kar = jum_kar;
         }
       }
       PosisiKNNI_kar.Add(new
       position(m.X,h,kar_maks,
       PosisiData[m.X,jum_col- 1].nilai));
      }
    }
}
```

Gambar 4. 19 Fungsi GetKNN_Cat

Selain kedua fungsi utama tersebut, pada kelas **KNNImputation** juga terdapat 3 fungsi pendukung yang dipanggil dalam fungsi **GetKNN_Num** dan fungsi **GetKNN_Cat**.

a. Fungsi kuadrat

Pada fungsi **kuadrat** dilakukan penghitungan kuadrat dari suatu nilai. Fungsi **kuadrat** dapat diamati pada gambar 4.20.

```
1  public int kuadrat(int jum)
2  {  return jum*jum;}
```

Gambar 4. 20 Fungsi kuadrat

b. Fungsi eucledian

Dalam fungsi **eucledian** dilakukan penghitungan jarak *eucledian* dari nilai kuadrat yang telah dihitung pada fungsi **kuadrat**. Penghitungan jarak *eucledian* disesuaikan dengan persamaan 2.2. Fungsi **eucledian** dapat diamati pada gambar 4.21.

```
1  public double eucledian(int nilai_kuadrat)
2  { return Math.Sqrt(nilai_kuadrat);}
```

Gambar 4. 21 Fungsi eucledian

c. Fungsi average

Fungsi **average** digunakan untuk melakukan penghitungan rata-rata dari suatu nilai. Fungsi tersebut dipanggil pada fungsi **GetKNN_Num** untuk menghitung nilai rata-rata dari sejumlah nilai data yang memiliki jarak *eucledian* terkecil. Fungsi **average** dapat diamati pada gambar 4.22.

```
1  public double average(int pembilang, int penyebut)
2  {
3     double a, b, hasil, selisih;
4     int bagi;
5     a = Convert.ToDouble(pembilang);
6     b = Convert.ToDouble(penyebut);
7     hasil = a / b;
8     bagi = pembilang / penyebut;
9     selisih = hasil - bagi;
10    if (selisih >= 0.5)
11    { hasil = hasil + 0.5;}
12    return hasil;}
```

Gambar 4. 22 Fungsi average

6. Kelas `Form1`

`Form 1` merupakan kelas utama yang mengandung fungsi yang mengatur tampilan *data training*, hasil penanganan *missing value*, berbagai informasi yang terkait dengan *data training* serta fungsi untuk penghitungan tingkat kesalahan.

Penghitungan tingkat kesalahan dilakukakn berdasarkan perbandingan antara nilai data yang ditangani dengan metode penanganan *missing value* dan nilai data sebenarnya yang dimiliki oleh data pada posisi tersebut. Metode penghitungan tingkat kesalahan dibagi menjadi 2 berdasarkan tipe dari *data training*, yaitu *Root Mean Squared Error*(RMSE) untuk tipe data kuantitatif serta *Performance Metric Error Rate* untuk tipe data kualitatif.

a. *Root Mean Squared Error* (RMSE)

```
1  public double Rmse(double jum_selisih, double
2  jum_actual, double jum_MV)
3
4  {
5     double err_rate_num = Math.Sqrt(jum_selisih /
6     jum_MV) / Math.Sqrt(jum_actual / jum_MV);
7     return err_rate_num;
8  }
```

Gambar 4. 23 Fungsi Rmse

Untuk tipe data kuantitatif, tingkat kesalahan dihitung dengan metode RMSE seperti pada persamaan 2.8. RMSE diimplementasikan pada fungsi `Rmse` yang dapat diamati pada gambar 4.23.

Fungsi **`Rmse`** memiliki parameter berupa nilai **`jum_selisih`**, **`jum_actual`** dan **`jum_MV`**. Nilai **`jum_selisih`** merupakan jumlah nilai selisih antara nilai imputasi dengan nilai sebenarnya dari suatu data, kemudian dikuadratkan. Nilai **`jum_actual`** adalah jumlah dari nilai sebenarnya dari data yang dikuadratkan.

b. *Performance Metric Error Rate*

Untuk tipe data kualitatif, tingkat kesalahan dihitung dengan metode *Performance Metric Error Rate* seperti pada persamaan 2.9. *Error Rate* diimplementasikan pada fungsi `perform_metric` yang dapat diamati pada gambar 4.24.

Fungsi **perform_metric** memiliki parameter berupa nilai **salah** dan **jum_mv**. Nilai **salah** merupakan jumlah dari nilai yang tidak sama. Jika nilai imputasi berbeda dengan nilai sebenarnya, maka jumlah nilai salah bertambah 1, namun bila nilai imputasi dan nilai sebenarnya sama, maka jumlah nilai salah tidak bertambah. Nilai **jum_mv** merupakan jumlah *missing value* pada *data training* tersebut.

```
1  public double Perform_metric(double salah,double
2  jum_MV)
3
4  {
5     double err_rate = salah / jum_MV;
6     return err_rate;
7  }
```

Gambar 4. 24 Fungsi Perform_metric

4.3 Implementasi Pengujian

4. 3. 1 Hasil uji

Hasil pengujian yang dilakukan terhadap *data training* ber-*missing value* diperlihatkan pada tabel 4.4, 4.5, tabel 4.6, dan tabel 4.7. Tabel 4.4 dan tabel 4.5 menunjukkan hasil pengujian metode *Mean Imputation* dan *KNN Imputation* untuk *data training Mammographic Mass* yang bertipe kuantitatif. Sedangkan tabel 4.6 dan tabel 4.7 menunjukkan hasil pengujian metode *Mode Imputation* dan *KNN Imputation* untuk *data training Mushroom* yang bertipe kualitatif. Masing-masing nilai hasil uji yang ditampilkan pada tabel-tabel tersebut merupakan hasil rata-rata dari penghitungan tingkat kesalahan 3 macam data uji coba.

Dari tabel-tabel hasil uji coba tersebut dapat diperoleh informasi tentang pengaruh jumlah *record*, prosentase *missing value* serta peletakan *missing value* pada data uji terhadap tingkat kesalahan dari nilai imputasi yang dihasilkan dari masing-masing metode.

Tabel 4. 4 Hasil pengujian data kuantitatif dengan *Mean Imputation*

Jumlah *record*	Jumlah MV	Rata-rata RMSE
75	5%	0.148
	10%	0.195
	15%	0.203
	30%	0.211
150	5%	0.202
	10%	0.233
	15%	0.216
	30%	0.24
300	5%	0.196
	10%	0.217
	15%	0.202
	30%	0.237

Tabel 4. 5 Hasil pengujian data kuantitatif dengan *KNN Imputation*

Jumlah *record*	Jumlah MV	Rata-rata RMSE			
		k=3	k=5	k=10	k maks
75	5%	0.162	0.143	0.125	0.175
	10%	0.225	0.199	0.206	0.25
	15%	0.231	0.231	0.22	0.238
	30%	0.243	0.237	0.222	0.221
150	5%	0.208	0.216	0.211	0.197
	10%	0.241	0.251	0.228	0.262
	15%	0.256	0.237	0.228	0.242
	30%	0.265	0.246	0.239	0.249
300	5%	0.242	0.224	0.205	0.206
	10%	0.246	0.227	0.216	0.231
	15%	0.229	0.212	0.207	0.225
	30%	0.26	0.254	0.248	0.266

Tabel 4.6 Hasil pengujian data kualitatif dengan *Mode Imputation*

Jumlah *record*	Jumlah MV	Rata-rata *Error Rate*
75	5%	0.583
	10%	0.458
	15%	0.361
	30%	0.348
150	5%	0.292
	10%	0.289
	15%	0.217
	30%	0.193
300	5%	0.4
	10%	0.378
	15%	0.326
	30%	0.396

Tabel 4. 7 Hasil pengujian data kualitatif dengan *KNN Imputation*

Jumlah *record*	Jumlah MV	Rata-rata *Error Rate*			
		k=3	k=5	k=10	k maks
75	5%	0	0	0	0.583
	10%	0	0	0.042	0.458
	15%	0	0	0.028	0.361
	30%	0	0	0.029	0.348
150	5%	0	0.042	0.042	0.292
	10%	0.067	0.089	0.089	0.289
	15%	0.029	0.044	0.058	0.217
	30%	0.037	0.029	0.067	0.193
300	5%	0	0.022	0	0.6
	10%	0	0	0	0.556
	15%	0.007	0.007	0	0.467
	30%	0	0	0	0.515

1. Tingkat kesalahan hasil uji *data training* bertipe kuantitatif

Pada *data training* bertipe kuantitatif diimplementasikan 2 macam metode penanganan *missing value*, yaitu *Mean Imputation* dan *KNN Imputation*. Pada implementasi metode *KNN Imputation* dilakukan pengujian dengan 4 macam parameter nilai k. Hasil pengujian telah ditampilkan pada tabel 4.4 dan tabel 4.5. Untuk lebih jelasnya, hasil pengujian disusun kembali berdasarkan jumlah *record* dari *data training*.

a. *Data training* terdiri dari 75 *record*

Hasil pengujian untuk 75 *record* data ditampilkan pada tabel 4.8. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.25.

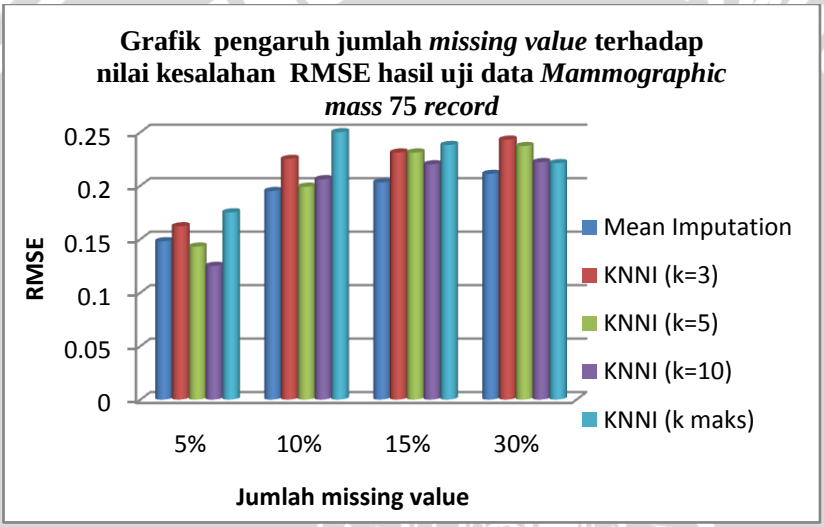
Tabel 4. 8 Tingkat kesalahan pada pengujian data *Mammographic mass* 75 *record*

	Jml MV=5%	Jumlah MV=10%	Jumlah MV=15%	Jumlah MV=30%
Mean Imputation	0.148	0.195	0.203	0.211
KNNI (k=3)	0.162	0.225	0.231	0.243
KNNI (k=5)	0.143	0.199	0.231	0.237
KNNI (k=10)	0.125	0.206	0.22	0.222
KNNI (k maks)	0.175	0.25	0.238	0.221

Pada tabel 4.8, atribut kolom tabel merupakan prosentase jumlah *missing value* dalam *data training* yang diuji. Dari gambar 4.20 dan tabel 4.8 dapat diamati secara keseluruhan bahwa rata-rata tingkat kesalahan RMSE yang diperoleh dari pengujian hasil metode *Mean Imputation* yang ditunjukkan dengan warna biru tua mengalami kenaikan seiring dengan bertambahnya jumlah *missing value*. Rata-rata nilai kesalahan RMSE hasil pengujian dari metode *Mean Imputation* meningkat dimulai dari 0.148 (14.8%) pada prosentase *missing value* 5% hingga 0.211(21.1%) pada prosentase *missing value* 30%.

Untuk rata-rata nilai kesalahan RMSE dari metode *KNN Imputation* pada grafik ditunjukkan oleh warna merah (untuk k=3), hijau (untuk k=5), ungu (untuk k=10) dan biru muda (untuk k

maksimum). Untuk rata-rata nilai kesalahan RMSE dengan k = 3, 5, dan 10 mengalami kenaikan seiring dengan pertambahan jumlah *missing value*. Namun untuk nilai k maksimum, rata-rata nilai RMSE mengalami pola yang tidak beraturan. Ketika prosentase jumlah *missing value* sebanyak 5% dan 10%, rata-rata RMSE mengalami kenaikan dari 0.175(17.5%) menjadi 0.25 (25%). Namun ketika prosentase jumlah *missing value* bertambah menjadi 15% dan 30%, rata-rata RMSE mengalami penurunan menjadi 0.238(23.8%) dan 0.221(22.1%).



Gambar 4. 25 Grafik tingkat kesalahan pada data *Mammographic mass* 75 *record*

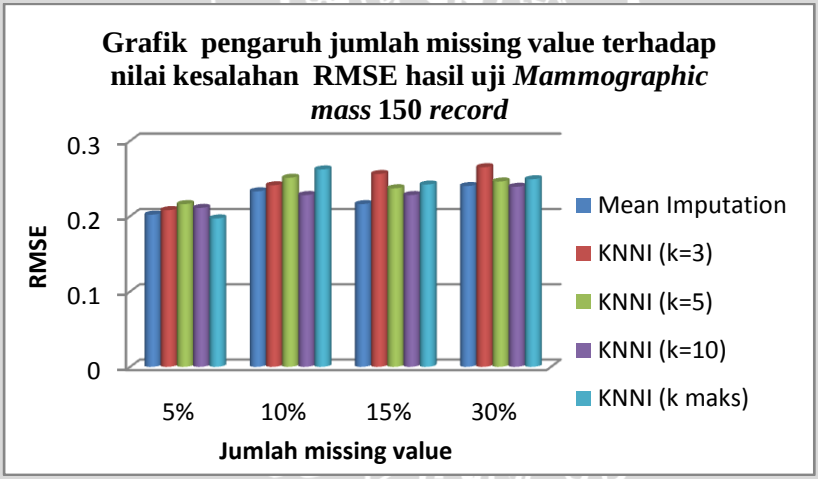
Jika diamati pada setiap jenis prosentase jumlah *missing value*, rata-rata kesalahan RMSE yang dihasilkan dari metode *Mean Imputation* selalu lebih rendah dari rata-rata kesalahan RMSE dari *KNN Imputation*. Untuk *KNN Imputation*, nilai rata-rata kesalahan RMSE mengalami penurunan untuk setiap peningkatan nilai k dari 3 hingga 10 pada setiap jenis prosentase jumlah *missing value*. Namun untuk nilai rata-rata kesalahan RMSE dari *KNN Imputation* dengan k maksimum cenderung lebih tinggi daripada nilai rata-rata kesalahan RMSE dari *KNN Imputation* dengan k = 3, 5, dan 10.

b. *Data training* terdiri dari 150 *record*

Hasil pengujian untuk 150 *record* data ditampilkan pada tabel 4.9. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.26.

Tabel 4. 9 Tingkat kesalahan pada pengujian data *Mammographic mass* 150 *record*

	Jml MV=5%	Jml MV=10%	Jml MV=15%	Jml MV=30%
Mean Imputation	0.202	0.233	0.216	0.24
KNNI (k=3)	0.208	0.241	0.256	0.265
KNNI (k=5)	0.216	0.251	0.237	0.246
KNNI (k=10)	0.211	0.228	0.228	0.239
KNNI (k maks)	0.197	0.262	0.242	0.249



Gambar 4. 26 Grafik tingkat kesalahan pada data *Mammographic mass* 150 *record*

Dari gambar 4.26 dan tabel 4.9 dapat diamati secara keseluruhan bahwa rata-rata tingkat kesalahan RMSE yang diperoleh dari hasil pengujian metode *Mean Imputation* dan *KNN Imputation* tidak menunjukkan perbedaan yang terlalu besar.

Dari pengujian hasil metode *Mean Imputation* yang ditunjukkan dengan warna biru tua mengalami pola yang kurang beraturan seiring

dengan bertambahnya jumlah *missing value*. Rata-rata nilai kesalahan RMSE hasil pengujian dari metode *Mean Imputation* pada awalnya meningkat dimulai dari 0.202 (20.2%) pada prosentase *missing value* 5% hingga 0.233 (23.3%) pada prosentase *missing value* 10%. Namun pada prosentase *missing value* 15% terjadi penurunan nilai rata-rata kesalahan RMSE menjadi 0.216 (21.6%) dan kembali mengalami kenaikan pada prosentase *missing value* 30% yaitu dengan nilai RMSE 0.24 (24%).

Untuk rata-rata nilai kesalahan RMSE dari metode *KNN Imputation* pada grafik ditunjukkan oleh warna merah (untuk k=3), hijau (untuk k=5), ungu (untuk k=10) dan biru muda (untuk k maksimum). Untuk rata-rata nilai kesalahan RMSE dengan k = 3 dan 10 cenderung mengalami kenaikan seiring dengan pertambahan jumlah *missing value*. Namun untuk nilai k = 5 rata-rata nilai RMSE mengalami pola yang tidak beraturan. Pada prosentase *missing value* 5% hingga 10% nilai rata-rata RMSE mengalami kenaikan dari 0.216 (21.6%) menjadi 0.251 (25.1%). Tetapi pada prosentase *missing value* 15% menurun menjadi 0.237 (23.7%) dan kembali naik menjadi 0.246 (24.6%) pada prosentase 30%. Demikian pula dengan nilai k maksimum, rata-rata nilai RMSE mengalami pola yang tidak beraturan. Ketika prosentase jumlah *missing value* sebanyak 5% dan 10%, rata-rata RMSE mengalami kenaikan dari 0.197(19.7%) menjadi 0.262 (26.2%). Namun ketika prosentase jumlah *missing value* bertambah menjadi 15%, rata-rata RMSE mengalami penurunan menjadi 0.242(24.2%) dan kembali mengalami kenaikan pada prosentase *missing value* 30% menjadi 0.249 (24.9%). Pada setiap peningkatan nilai k dari 3 hingga 10 pada setiap jenis prosentase jumlah *missing value*, nilai RMSE mengalami penurunan dan meningkat kembali pada saat k maksimum.

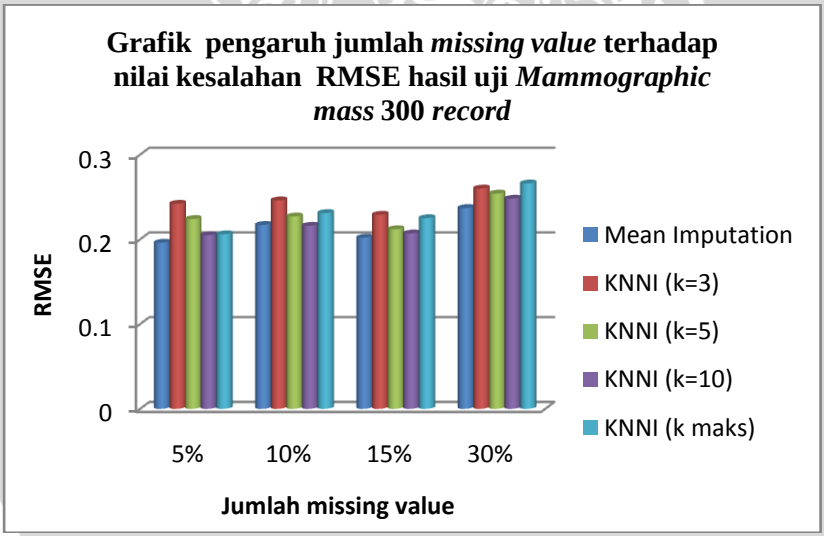
Jika diamati pada setiap jenis prosentase jumlah *missing value*, rata-rata kesalahan RMSE yang dihasilkan dari metode *Mean Imputation* cenderung lebih rendah dari rata-rata kesalahan RMSE *KNN Imputation*. Namun pada prosentase *missing value* 5% dan 10% nilai rata-rata kesalahan RMSE terendah diperoleh dari *KNN Imputation* dengan k maksimum (untuk prosentase *missing value* 5%) dan k=10 (untuk prosentase *missing value* 10%).

c. *Data training* terdiri dari 300 *record*

Hasil pengujian untuk 300 *record* data ditampilkan pada tabel 4.10. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.27.

Tabel 4. 10 Tingkat kesalahan pada pengujian data *Mammographic mass* 300 *record*

	Jml MV=5%	Jml MV=10%	Jml MV=15%	Jml MV=30%
Mean Imputation	0.196	0.217	0.202	0.237
KNNI (k=3)	0.242	0.246	0.229	0.26
KNNI (k=5)	0.224	0.227	0.212	0.254
KNNI (k=10)	0.205	0.216	0.207	0.248
KNNI (k maks)	0.206	0.231	0.225	0.266



Gambar 4. 27 Grafik tingkat kesalahan pada data *Mammographic mass* 300 *record*

Dari gambar 4.27 dan tabel 4.10 dapat diamati secara keseluruhan bahwa rata-rata tingkat kesalahan RMSE yang diperoleh dari hasil

pengujian metode *Mean Imputation* dan *KNN Imputation* tidak menunjukkan perbedaan yang terlalu besar.

Untuk hasil pengujian metode *Mean Imputation*, nilai RMSE membentuk pola tidak beraturan. Pada prosentase *missing value* 5% hingga 10%, nilai RMSE bertambah dari 0.196 (19.6%) menjadi 0.217 (21.7%). Namun ketika prosentase *missing value* 15%, nilai RMSE menurun menjadi 0.202 (20.2%) dan bertambah kembali menjadi 0.237 (23.7%) pada prosentase *missing value* 30%.

Untuk rata-rata nilai kesalahan RMSE dari metode *KNN Imputation* pada grafik ditunjukkan oleh warna merah (untuk k=3), hijau (untuk k=5), ungu (untuk k=10) dan biru muda (untuk k maksimum). Pada hasil pengujian *KNN Imputation*, secara umum tidak ada perubahan yang terlalu besar dari nilai RMSE seiring dengan peningkatan jumlah *missing value*. Pada setiap jenis prosentase *missing value*, nilai rata-rata RMSE semakin menurun seiring dengan penambahan nilai k dari 3 hingga 10, dan kembali meningkat pada k maksimum.

Jika diamati pada setiap jenis prosentase jumlah *missing value*, rata-rata kesalahan RMSE yang dihasilkan dari metode *Mean Imputation* cenderung lebih rendah dari rata-rata kesalahan RMSE *KNN Imputation.*

2. Tingkat kesalahan hasil uji *data training* bertipe kualitatif

Pada *data training* bertipe kualitatif diimplementasikan 2 macam metode penanganan *missing value*, yaitu *Mode Imputation* dan *KNN Imputation*. Hasil pengujian telah ditampilkan pada tabel 4.6 dan tabel 4.7. Untuk lebih jelasnya, hasil pengujian disusun kembali berdasarkan jumlah *record* dari *data training*.

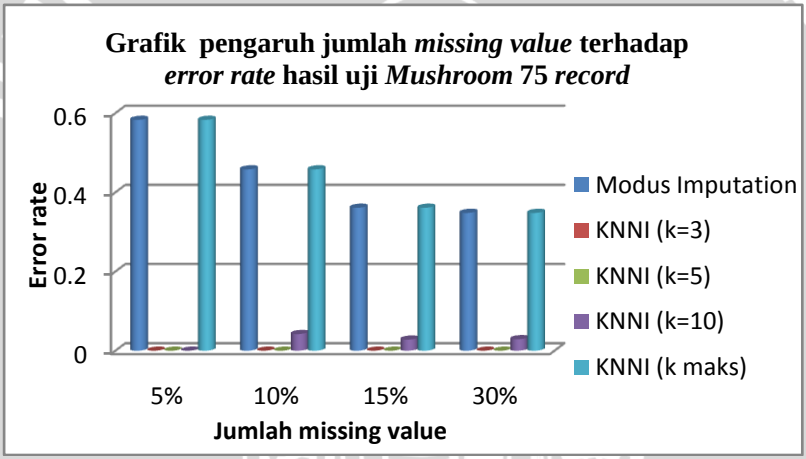
a. *Data training* terdiri dari 75 *record*

Hasil pengujian untuk 75 *record* data ditampilkan pada tabel 4.11. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.28.

Dari tabel 4.11 dan gambar 4.28 terlihat perbedaan yang mencolok pada rata-rata nilai *error rate*. Untuk hasil pengujian dengan *Mode Imputation*, rata-rata nilai *error rate* menurun seiring dengan penambahan prosentase jumlah *missing value*, yaitu dari 0.583 (58.3%), 0.458 (45.8%), 0.361 (36.1%) dan 0.348 (34.8%).

Tabel 4. 11 Tingkat kesalahan pada pengujian data *Mushroom* 75 *record*

	Jml MV=5%	Jml MV=10%	Jml MV=15%	Jml MV=30%
Mode Imputation	0.583	0.458	0.361	0.348
KNNI (k=3)	0	0	0	0
KNNI (k=5)	0	0	0	0
KNNI (k=10)	0	0.042	0.028	0.029
KNNI (k maks)	0.583	0.458	0.361	0.348



Gambar 4. 28 Grafik tingkat kesalahan pada data *Mushroom* 75 *record*

Untuk hasil pengujian dengan *KNN Imputation*, pada setiap penambahan nilai k pada setiap jenis prosentase *missing value*, rata-rata *error rate* cenderung bertambah. Pada prosentase *missing value* 5%, *error rate* untuk k = 3, 5 dan 10 tetap dengan nilai 0, namun untuk k maksimum *error rate* meningkat tajam menjadi 0.583 (58.3%). Pada prosentase *missing value* 10%, *error rate* untuk k = 3 dan 5 tetap dengan nilai 0. Sedangkan untuk k = 10 *error rate* bernilai 0.042 (4.2%) dan untuk k maksimum *error rate* meningkat tajam menjadi 0.583 (58.3%). Pada prosentase *missing value* 15%,

error rate untuk k = 3 dan 5 tetap dengan nilai 0. Sedangkan untuk k = 10 *error rate* bernilai 0.028 (2.8%) dan untuk k maksimum *error rate* meningkat tajam menjadi 0.361 (36.1%). Pada prosentase *missing value* 30%, *error rate* untuk k = 3 dan 5 tetap dengan nilai 0. Sedangkan untuk k = 10 *error rate* bernilai 0.029 (2.9%) dan untuk k maksimum *error rate* meningkat tajam menjadi 0.348 (34.8%). Jika diperhatikan kembali, nilai *error rate* pada setiap k maks menyamai *error rate* dari *Mode Imputation* pada jumlah prosentase yang sama.

b. *Data training* terdiri dari 150 *record*

Hasil pengujian untuk 150 *record* data ditampilkan pada tabel 4.12. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.29.

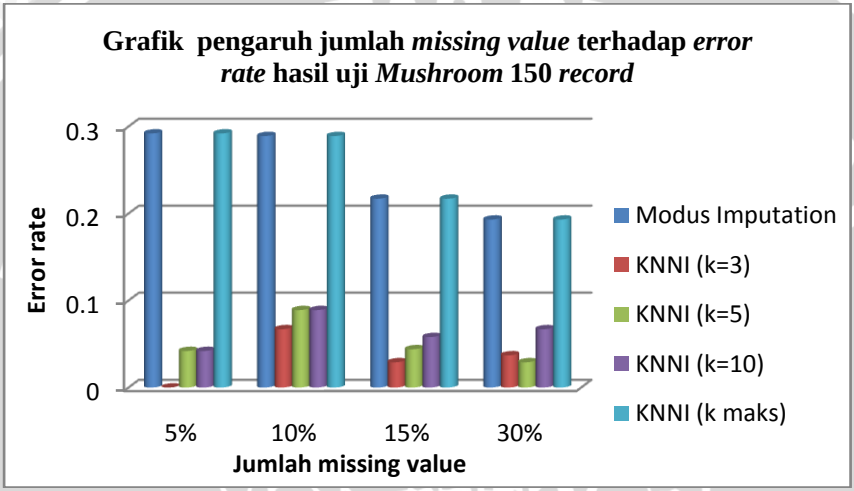
Tabel 4. 12 Tingkat kesalahan pada pengujian data *Mushroom* 150 *record*

	Jml MV=5%	Jml MV=10%	Jml MV=15%	Jml MV=30%
Mode Imputation	0.292	0.289	0.217	0.193
KNNI (k=3)	0	0.067	0.029	0.037
KNNI (k=5)	0.042	0.089	0.044	0.029
KNNI (k=10)	0.042	0.089	0.058	0.067
KNNI (k maks)	0.292	0.289	0.217	0.193

Dari tabel 4.12 dan gambar 4.29 terlihat perbedaan yang mencolok pada rata-rata nilai *error rate*. Untuk hasil pengujian dengan *Mode Imputation*, rata-rata nilai *error rate* cenderung menurun seiring dengan penambahan prosentase jumlah *missing value*, yaitu dari 0.292 (29.2%), 0.289 (28.9%), 0.217 (21.7%) dan 0.193 (19.3%).

Untuk hasil pengujian dengan *KNN Imputation*, *error rate* pada prosentase *missing value* 5% cenderung bertambah seiring dengan bertambahnya nilai k, yaitu 0 (untuk k = 3), 0.042 (untuk k =5 dan 10), dan 0.292 (untuk k maksimum). Pada prosentase 10% nilai *error rate* adalah 0.067 (untuk k=3), 0.089 (untuk k = 5 dan 10), dan 0.289 (untuk k maksismum). Pada prosentase 15% nilai *error rate* adalah 0.029 (untuk k=3), 0.044 (untuk k=5), 0.058 (untuk k=10) dan 0.217 (untuk k maksismum). Pada prosentase 30% terjadi

penyimpangan pada nilai *error rate*. Untuk k=3, nilai *error rate* adalah 0.037. Jika pada pola sebelumnya seharusnya nilai *error rate* bertambah jika nilai k bertambah, namun pada nilai k=5 nilai *error rate* turun menjadi 0.029. Pada nilai k=10 *error rate* kembali bertambah menjadi 0.067 dan 0.193 pada k maksimum. Jika diperhatikan kembali, nilai *error rate* pada setiap k maks menyamai *error rate* dari *Mode Imputation* pada jumlah prosentase yang sama.



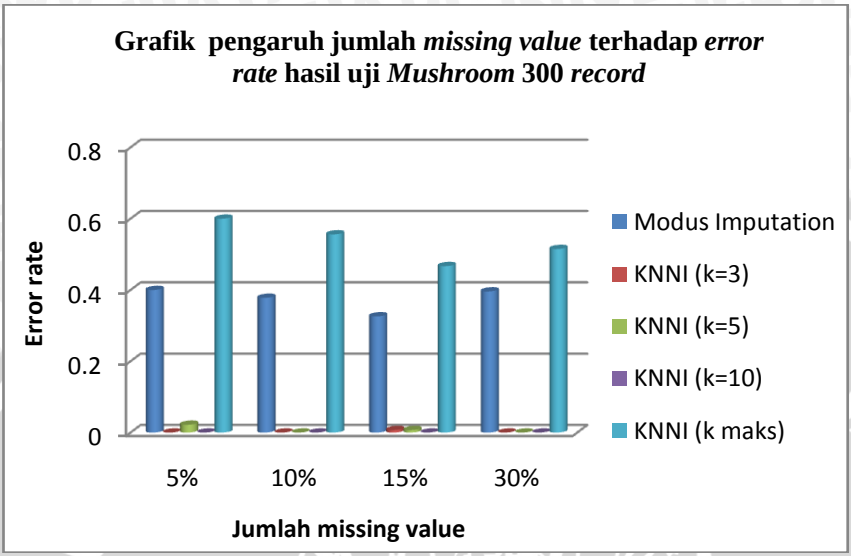
Gambar 4. 29 Grafik tingkat kesalahan pada data *Mushroom* 150 *record*

c. *Data training* terdiri dari 300 *record*

Hasil pengujian untuk 300 *record* data ditampilkan pada tabel 4.13. Data pada tabel tersebut diimplementasikan pula dalam bentuk grafik pada gambar 4.30.

Tabel 4. 13 Tingkat kesalahan pada pengujian data *Mushroom* 300 *record*

	Jml MV=5%	Jml MV=10%	Jml MV=15%	Jml MV=30%
Mode Imputation	0.4	0.378	0.326	0.396
KNNI (k=3)	0	0	0.007	0
KNNI (k=5)	0.022	0	0.007	0
KNNI (k=10)	0	0	0	0
KNNI (k maks)	0.6	0.556	0.467	0.515



Gambar 4. 30 Grafik tingkat kesalahan pada data *Mushroom* 300 *record*

Dari tabel 4.13 dan gambar 4.30 terlihat perbedaan yang mencolok pada nilai rata-rata *error rate*. Untuk hasil pengujian dengan *Mode Imputation*, rata-rata nilai *error rate* cenderung menurun seiring dengan penambahan prosentase jumlah *missing value*, yaitu dari 0.4 (40%), 0.378 (37.8%), dan 0.326 (32.6%). Namun pada prosentase *missing value* 30% nilai *error rate* kembali bertambah menjadi 0.396 (39.6%).

Untuk hasil pengujian dengan *KNN Imputation* dengan nilai k = 3, 5, dan 10, nilai *error rate* pada setiap perubahan prosentase *missing value* menunjukkan nilai yang tidak terlalu besar, yaitu 0.02 (untuk k=5 pada prosentase *missing value* 5%) dan 0.007 (untuk k=3 dan 5 pada prosentase *missing value* 15%). Selebihnya, hasil pengujian menunjukkan angka 0. Namun hal tersebut tidak berlaku pada k maksimum yang memiliki nilai *error rate* yang cukup jauh berbeda. Pada prosentase *missing value* 5%, nilai *error rate* adalah 0.6 (60%). Pada prosentase *missing value* 10%, nilai *error rate* adalah 0.556 (55.6%). Pada prosentase *missing value* 15%, nilai *error rate* adalah 0.467 (46.7%). Sedangkan pada prosentase *missing value* 30%, nilai *error rate* adalah 0.515 (51.5%).

4. 3. 2 Analisa hasil

1. Analisa penyimpangan hasil pengujian

Berdasarkan hasil uji coba yang telah dibahas pada subbab 4.3.1 diketahui bahwa terjadi beberapa kasus penyimpangan nilai hasil pengujian. Hal tersebut diamati dari penyimpangan pola grafik yang terjadi.

a. Analisa penyimpangan hasil uji *data training* kuantitatif

Salah satu kasus penyimpangan pola grafik terdapat pada gambar 4.26. Pada grafik tersebut terjadi penyimpangan ketika dilakukan pengujian dengan prosentase *missing value* 5% pada 150 *record Mammographic mass*. Pada umumnya grafik menunjukkan nilai RMSE dari metode *Mean Imputation* lebih rendah dari nilai RMSE yang dihasilkan dari *KNN Imputation*. Namun pada grafik tersebut, nilai rata-rata RMSE terendah diperoleh dari *KNN Imputation* dengan k maksimum. Seharusnya nilai rata-rata RMSE dari k maksimum lebih besar. Selain itu, pada grafik umumnya nilai RMSE dari *KNN Imputation* menunjukan penurunan pada setiap peningkatan nilai k dari 3, 5 hingga 10. Namun nilai rata-rata RMSE untuk k=3 lebih rendah dari nilai rata-rata RMSE yang dihasilkan dengan k = 5 dan 10.

Hipotesa sementara untuk kasus tersebut adalah penyimpangan disebabkan oleh peletakan *missing value* yang berbeda pada *data training* dengan jumlah *record* dan jumlah *missing value* yang sama. Peletakan *missing value* yang berbeda tersebut terkadang menghasilkan nilai pengujian tingkat kesalahan yang berbeda. Untuk membuktikan kebenaran hipotesa tersebut, maka dilakukan pengamatan kembali pada hasil uji coba yang telah dilampirkan dan disederhanakan kembali pada tabel 4.15 untuk mempermudah pengamatan.

Dari tabel 4.14, perbedaan dari uji 1, 2 dan 3 terletak pada posisi *missing value* yang berbeda. Sedangkan persamaan dari uji 1, 2 dan 3 tersebut adalah jumlah *record data training* terdiri dari 150 *record* dan prosentase jumlah *missing value* 5%.

Dari tabel 4.11 dan gambar 4.28 terlihat perbedaan yang mencolok pada rata-rata nilai *error rate*. Untuk hasil pengujian dengan *Mode Imputation*, rata-rata nilai *error rate* menurun seiring dengan penambahan prosentase jumlah *missing value*, yaitu dari 0.583 (58.3%), 0.458 (45.8%), 0.361 (36.1%) dan 0.348 (34.8%).

Tabel 4. 14 Hasil pengujian *data Mammographic mass* 150 *record* dan prosentase *missing value* 5%

Jumlah *record*	Jumlah *missing value*	Metode penanganan		Uji coba	RMSE	Rata-rata RMSE
150	5%	Mean Imputation		uji 1	0.208	0.202333
				uji 2	0.285	
				uji 3	0.114	
		KNNI	k=3	uji 1	0.262	0.207667
				uji 2	0.207	
				uji 3	0.154	
			k=5	uji 1	0.263	0.215667
				uji 2	0.249	
				uji 3	0.135	
			k=10	uji 1	0.262	0.211
				uji 2	0.201	
				uji 3	0.17	
			k maks	uji 1	0.204	0.197333
				uji 2	0.243	
				uji 3	0.145	

b. Analisa penyimpangan hasil uji *data training* kualitatif

Salah satu kasus penyimpangan pola grafik terdapat pada gambar 4.29. Pada grafik tersebut terjadi penyimpangan ketika dilakukan pengujian dengan prosentase *missing value* 30% pada 150 *record Mushroom*. Pada umumnya grafik nilai rata-rata *Error rate* dari metode *KNN Imputation* menunjukkan pola meningkat untuk setiap penambahan nilai k dari 3, 5 hingga 10. Namun pada prosentase *missing value* 30%, nilai rata-rata *error rate* dari *KNN Imputation* membentuk pola yang tidak beraturan. Pada *KNN Imputation* dengan k=5, nilai rata-rata *error rate* seharusnya labih tinggi dibandingkan dengan nilai sebelumnya, namun pada grafik nilai *error rate* k=5 lebih rendah dari nilai *error rate* k=3.

Hipotesa sementara untuk kasus tersebut adalah penyimpangan disebabkan oleh peletakan *missing value* yang berbeda pada *data*

training dengan jumlah *record* dan jumlah *missing value* yang sama. Peletakan *missing value* yang berbeda tersebut terkadang menghasilkan nilai pengujian tingkat kesalahan yang berbeda. Untuk membuktikan kebenaran hipotesa tersebut, maka dilakukan pengamatan kembali pada hasil uji coba yang telah dilampirkan dan disederhanakan kembali pada tabel 4.15 untuk mempermudah pengamatan.

Tabel 4. 15 Hasil pengujian data *Mushroom* 150 record dan prosentase missing value 30%

Jumlah *record*	Jumlah *missing value*	Metode penanganan		Uji coba	*Error rate*	Rata-rata *Error rate*
300	30%	Mode Imputation		uji 1	0.244	0.192666667
				uji 2	0.178	
				uji 3	0.156	
		KNNI	k=3	uji 1	0.067	0.037
				uji 2	0.044	
				uji 3	0	
			k=5	uji 1	0.044	0.029333333
				uji 2	0.044	
				uji 3	0	
			k=10	uji 1	0.111	0.066666667
				uji 2	0.089	
				uji 3	0	
			k maks	uji 1	0.244	0.192666667
				uji 2	0.178	
				uji 3	0.156	

Dari tabel 4.15, perbedaan dari uji 1, 2 dan 3 terletak pada posisi *missing value* yang berbeda. Sedangkan persamaan dari uji 1, 2 dan 3 tersebut adalah jumlah *record data training* terdiri dari 150 *record* dan prosentase jumlah *missing value* 30%.

Pada hasil uji coba dengan metode *KNN Imputation*, nilai *Error rate* dari uji 1, 2 dan 3 menunjukkan perbedaan yang cukup besar.

Sehingga hal tersebut mempengaruhi nilai rata-rata dari *Error rate*. Dengan demikian posisi *missing value* yang berbeda pada pengujian sangat mempengaruhi nilai tingkat kesalahan yang dihasilkan.

Tabel 4. 16 Uji kebenaran pengaruh posisi *missing value*

Jenis pengosongan	Jumlah MV	MOI	KNNI	
			Jumlah k	*Error rate*
Selain dominasi	1	1	k=3	0
			k=5	0
			k=10	0
			k=74 (maks)	1
	5	1	k=3	0
			k=5	0
			k=10	0.2
			k=70 (maks)	1
	20	1	k=3	0.35
			k=5	0.3
			k=10	0.9
			k=65 (maks)	1
Dominasi (karakter 'e')	1	0	k=3	0
			k=5	0
			k=10	0
			k=74 (maks)	0
	5	0	k=3	0
			k=5	0
			k=10	0
			k=74 (maks)	0
	20	0	k=3	0
			k=5	0
			k=10	0
			k=55 (maks)	0

Namun untuk lebih memperkuat kembali hipotesa dari kasus pada *data training* kualitatif tersebut, maka dilakukan pengujian terhadap pengaruh jenis karakter yang dihilangkan. Jika karakter-karakter yang dihapus dari *data training* merupakan karakter-karakter yang mendominasi pada kolom ber-*missing value* tersebut, maka hal tersebut dapat mempengaruhi hasil dari pengujian.

Untuk membuktikan kebenaran hipotesa tersebut maka dilakukan 2 macam uji kebenaran dengan pengaturan terhadap karakter yang dipilih untuk dihilangkan. Pada uji kebenaran 1 dilakukan pengosongan terhadap karakter lain yang bukan merupakan karakter dominasi pada kolom 11 (kolom yang mengandung *missing value* pada data *Mushroom*). Sedangkan pada uji kebenaran 2 dilakukan pengosongan terhadap karakter yang mendominasi kolom tersebut. Pengosongan dilakukan hanya pada kolom 11 sesuai dengan kondisi *data training Mushroom* yang hanya mengandung *missing value* pada kolom tersebut.

Hasil uji kebenaran ditampilkan pada tabel 4.16. Dalam kolom jenis pengosongan terdapat 2 jenis pengosongan, yaitu pengosongan karakter selain dominasi dan pengosongan hanya pada karakter dominasi. Karakter dominasi pada kasus tersebut adalah karakter ‘e’. Kolom jumlah MV menunjukkan jumlah *missing value* yang diimplementasikan pada pengujian, yaitu sebanyak 1, 5 dan 20 karakter yang dihilangkan. Kolom MOI menampilkan nilai *error rate* dari metode penanganan *Mode Imputation*. Sedangakan kolom KNNI menampilkan *error rate* dari hasil pengujian dengan menggunakan 4 macam nilai k, yaitu 3, 5, 10 dan maksimum.

Dari nilai *error rate* yang ditampilkan pada tabel 4.16 terlihat perbedaan yang mencolok antara hasil uji kebenaran 1 dengan uji kebenaran 2.

- **Pengujian metode *Mode Imputation***

Pada hasil pengujian dengan metode *Mode Imputation*, nilai *error rate* ketika dilakukan pemilihan karakter selain dominasi untuk dihilangkan, maka dihasilkan nilai 1 untuk semua jenis jumlah *missing value* yang diujikan. Hal tersebut menunjukkan bahwa semua karakter yang dihasilkan dengan *Mode Imputation* adalah salah secara keseluruhan. Karakter-karakter yang dihasilkan dari pengujian tersebut adalah karakter dominasi.

Sedangkan ketika dilakukan pemilihan karakter ‘e’ yang merupakan karakter dominasi untuk dihilangkan, maka dihasilkan

nilai 0 untuk semua jenis jumlah *missing value* yang diujikan. Hal tersebut menunjukkan bahwa semua karakter yang dihasilkan adalah benar secara keseluruhan. Hal tersebut disebabkan karena pada metode *Mode Imputation* diterapkan konsep pencarian karakter yang paling banyak ditemukan (Mode). Hasil dari pengujian tersebut seluruhnya adalah karakter 'e'.

- **Pengujian metode *KNN Imputation***

Pada hasil pengujian dengan metode *KNN Imputation*, ketika dilakukan pemilihan karakter selain karakter yang mendominasi untuk dihilangkan, maka masih terdapat beberapa nilai *error rate* yang lebih dari 0. Pada k maksimum selalu didapatkan nilai 1 untuk *error rate*. Hal tersebut disebabkan karena ketika nilai k yang dipilih sejumlah nilai k yang diperbolehkan, yaitu sejumlah *record data training* yang tidak mengandung *missing value*, maka hal tersebut sangat mempengaruhi karakter yang dihasilkan, karena pada *KNN Imputation* digunakan konsep pencarian Mode karakter. Sehingga jika kolom ber-*missing value* pada sejumlah *record* yang tidak mengandung *missing value* tersebut lebih banyak mengandung karakter dominasi, maka hasil prediksi *missing value* yang dihasilkan adalah karakter dominasi tersebut. Karena karakter yang dihasilkan sama dengan karakter dominasi, maka pada uji kebenaran 1 tersebut nilai *error rate* adalah 1 atau salah semua.

Sedangkan ketika dilakukan pemilihan karakter 'e' yang merupakan karakter dominasi untuk dihilangkan, maka dihasilkan nilai 0 untuk semua jenis jumlah *missing value* yang diujikan. Hal tersebut menunjukkan bahwa semua karakter yang dihasilkan adalah benar secara keseluruhan.

2. Analisa pengaruh jumlah *record* dan jumlah *missing value* terhadap tingkat kesalahan

Berdasarkan hasil pengujian yang telah ditampilkan pada tabel 4.4 sampai tabel 4.7, maka dapat dilakukan analisa sebagai berikut.

a. **Analisa pengaruh jumlah *record* dan jumlah *missing value* pada data kuantitatif**

Nilai rata-rata tingkat kesalahan yang dihasilkan dari hasil pengujian data kuantitatif dengan metode *Mean Imputation* dapat dilihat pada tabel 4.4. Sedangkan untuk hasil pengujian data

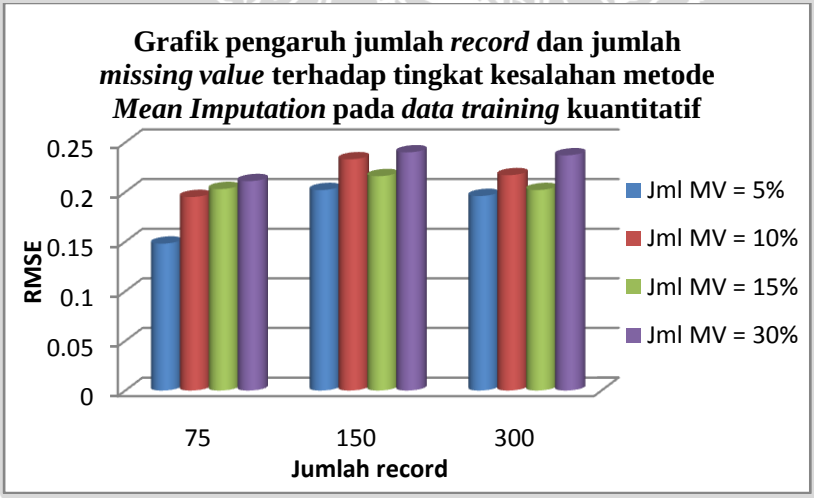
kuantitatif dengan metode *KNN Imputation* dapat dilihat pada tabel 4.5.

- **Analisa pada hasil uji metode *Mean Imputation***

Hasil pengujian yang dilakukan terhadap data kuantitatif dengan menerapkan metode *Mean Imputation* disederhanakan kembali ke dalam grafik yang terdapat pada gambar 4.31.

Berdasarkan gambar 4.31 terlihat nilai RMSE yang hampir sama pada setiap peningkatan jumlah *record*. Hal tersebut menunjukkan bahwa peningkatan jumlah *record* tidak berpengaruh besar terhadap perubahan nilai RMSE pada hasil penanganan *missing value* dengan *Mean Imputation*.

Sedangkan untuk pengaruh jumlah *missing value* dapat diamati pula pada gambar 4.31. Pada gambar tersebut juga terlihat nilai kesalahan RMSE yang hampir sama. Hal tersebut juga menunjukkan bahwa peningkatan jumlah *missing value* tidak berpengaruh besar terhadap perubahan nilai kesalahan RMSE pada hasil penanganan *missing value* dengan *Mean Imputation*.



Gambar 4.31 Grafik pengaruh jumlah *record* dan jumlah *missing value* pada hasil *Mean Imputation*

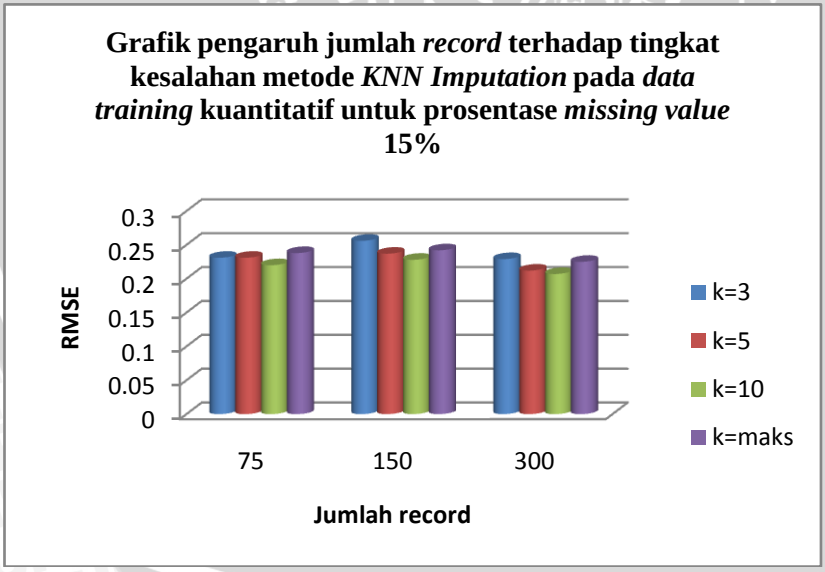
Beberapa pola penyimpangan yang terjadi pada hasil pengujian tersebut disebabkan oleh peletakan *missing value* yang berbeda pada *data training* dengan jumlah *record* dan *missing value* yang sama, sehingga mempengaruhi nilai rata-rata dari nilai kesalahan tersebut.

Hal tersebut telah dijabarkan pada analisa penyimpangan hasil pengujian.

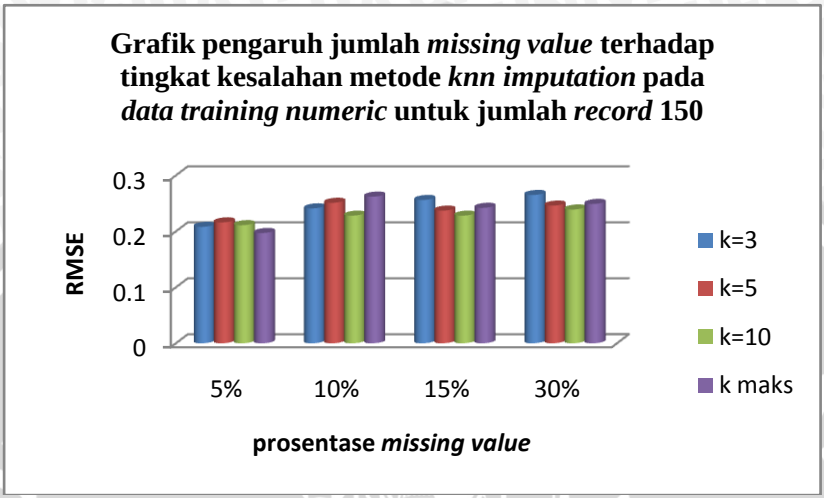
- **Analisa pada hasil uji metode *KNN Imputation***

Hasil pengujian yang dilakukan terhadap data kuantitatif dengan menerapkan metode *KNN Imputation* pada tabel 4.5 disederhanakan kembali ke dalam grafik yang terdapat pada gambar 4.32 dan gambar 4.33.

Gambar 4. 32 merupakan grafik pengaruh jumlah *record* terhadap tingkat kesalahan pada *data training* kuantitatif *Mammographic mass* untuk prosentase *missing value* 15%. Pada gambar tersebut dapat diamati bahwa perubahan jumlah *record* tidak terlalu berpengaruh terhadap nilai kesalahan dari hasil penanganan *missing value*, karena nilai kesalahan berada pada *range* 0.207 sampai 0.256. Demikian pula untuk prosentase *missing value* 5%, nilai kesalahan berada pada *range* 0.125 sampai 0.242. Untuk prosentase *missing value* 10%, nilai kesalahan berada pada *range* 0.199 hingga 0.262. Sedangkan untuk prosentase *missing value* 30%, nilai kesalahan berada pada *range* 0.221 sampai 0.266.



Gambar 4. 32 Grafik pengaruh jumlah *record* pada hasil *KNN Imputation* untuk *data training* kuantitatif



Gambar 4. 33 Grafik pengaruh jumlah *missing value* pada hasil *KNN Imputation* untuk *data training* kuantitatif

Gambar 4.33 merupakan grafik pengaruh jumlah *missing value* terhadap tingkat kesalahan pada *data training* kuantitatif *Mammographic mass* untuk jumlah *record* 150. Pada gambar tersebut dapat diamati bahwa pada setiap nilai k, penambahan jumlah *missing value* tidak terlalu berpengaruh pada nilai kesalahan dari hasil penanganan *missing value* dengan *KNN Imputation*, karena nilai kesalahan berada pada *range* yang tidak terlalu besar, yaitu 0.197 sampai 0.265. Demikian pula untuk jumlah *record* 75 nilai kesalahan berada pada *range* 0.125 sampai 0.25. Sedangkan untuk jumlah *record* 300 nilai kesalahan berada pada *range* 0.205 sampai 0.266.

b. **Analisa pengaruh jumlah *record* dan jumlah *missing value* pada data kualitatif**

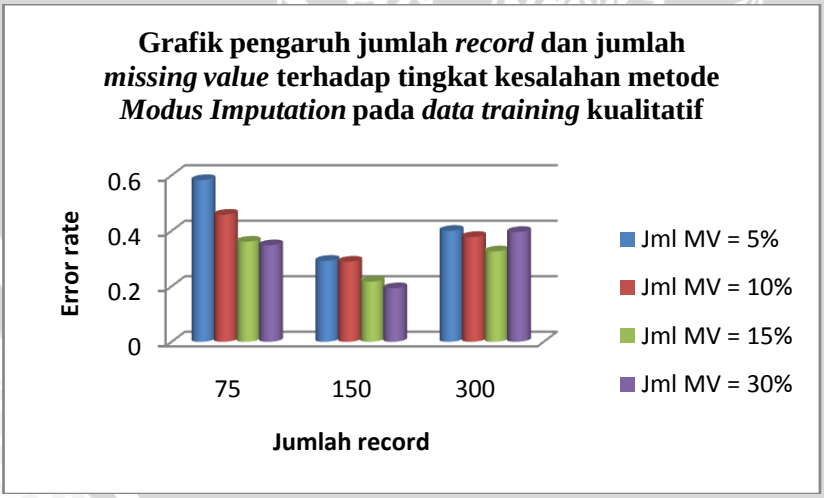
Nilai rata-rata tingkat kesalahan yang dihasilkan dari hasil pengujian data kualitatif dengan metode *Mode Imputation* dapat dilihat pada tabel 4.6. Sedangkan untuk hasil pengujian data kualitatif dengan metode *KNN Imputation* dapat dilihat pada tabel 4.7.

- **Analisa pada hasil uji metode *Mode Imputation***

Hasil pengujian yang dilakukan terhadap data kualitatif dengan menerapkan metode *Mode Imputation* disederhanakan kembali ke dalam grafik yang terdapat pada gambar 4.34.

Gambar 4.34 merupakan grafik pengaruh jumlah *record* dan jumlah *missing value* terhadap nilai kesalahan pada *data training* kualitatif *Mushroom*. Pada gambar tersebut dapat diamati bahwa ketika dilakukan penambahan jumlah *record*, maka terbentuk pola yang tidak beraturan dari nilai kesalahan. Pada penambahan jumlah *record* 75 menjadi 150, terjadi pola menurun pada grafik. Namun pada penambahan jumlah *record* dari 150 menjadi 300, pola kembali naik. Berdasarkan analisa penyimpangan hasil pengujian yang telah dibahas pada subbab sebelumnya, penyimpangan pola yang terjadi pada data kualitatif tersebut dapat disebabkan karena peletakan *missing value* dilakukan secara random, sehingga meskipun data yang digunakan memiliki jumlah *record* yang sama, nilai kesalahan yang dihasilkan dari evaluasi hasil penanganan dapat berbeda.

Pada gambar tersebut dapat pula diamati pengaruh penambahan jumlah *missing value* pada nilai kesalahan evaluasi. Nilai kesalahan semakin menurun seiring dengan penambahan prosentase *missing value*.

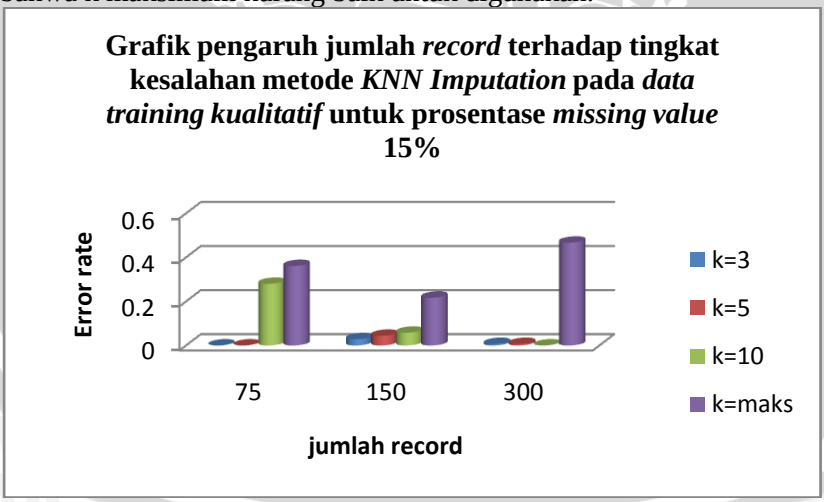


Gambar 4. 34 Grafik pengaruh jumlah *record* dan jumlah *missing value* pada hasil *Mode Imputation* untuk *data training* kualitatif

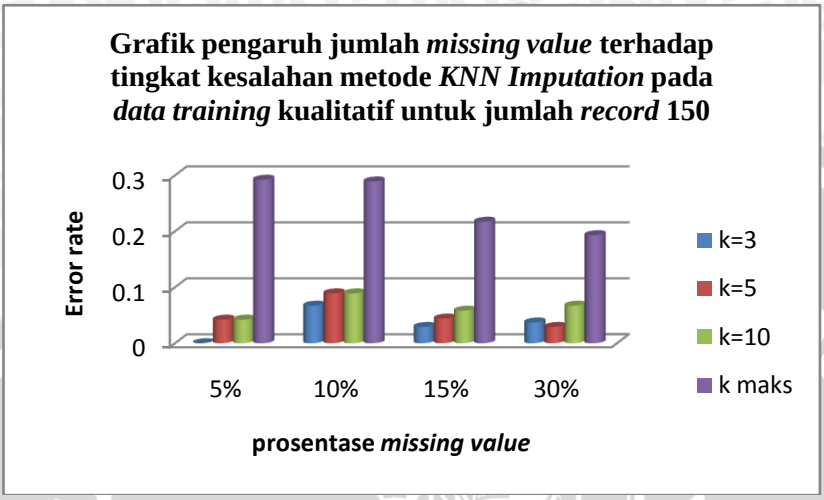
- **Analisa pada hasil uji metode *KNN Imputation***

Hasil pengujian yang dilakukan terhadap data kualitatif dengan menerapkan metode *Mode Imputation* disederhanakan kembali ke dalam grafik yang terdapat pada gambar 4.35.

Gambar 4. 35 merupakan grafik pengaruh jumlah *record* terhadap tingkat kesalahan pada *data training* kualitatif *Mushroom* untuk prosentase *missing value* 15%. Pada gambar tersebut dapat diamati bahwa dengan nilai k maksimum, maka nilai kesalahan selalu tinggi, yaitu dengan *range* nilai kesalahan 0.217 hingga 0.467. Namun jika nilai k = 3, 5 dan 10, maka nilai kesalahan mengalami kenaikan pada setiap penambahan nilai k, namun tidak terlalu besar, yaitu pada *range* 0 sampai 0.058. Demikian pula untuk prosentase *missing value* 5%, nilai kesalahan yang dihasilkan dari k maksimum jauh lebih tinggi daripada nilai kesalahan yang dihasilkan dengan nilai k lainnya yang hanya berada pada *range* 0 hingga 0.042. Untuk *missing value* dengan prosentase 10% didapatkan pula nilai kesalahan hasil k maksimum yang berada pada *range* 0.289 hingga 0.556, sedangkan untuk k lainnya nilai kesalahan hanya berada pada *range* 0 hingga 0.089. Untuk *missing value* dengan prosentase 30% didapatkan pula nilai kesalahan hasil k maksimum yang berada pada *range* 0.193 hingga 0.515, sedangkan untuk k lainnya nilai kesalahan hanya berada pada *range* 0 hingga 0.067. Hal tersebut menunjukkan bahwa k maksimum kurang baik untuk digunakan.



Gambar 4. 35 Grafik pengaruh jumlah *record* pada hasil *KNN Imputation* untuk *data training* kualitatif



Gambar 4. 36 Grafik pengaruh jumlah *missing value* pada hasil *KNN Imputation* untuk *data training* kualitatif

Gambar 4.36 merupakan grafik pengaruh jumlah *missing value* terhadap tingkat kesalahan pada *data training* kualitatif *Mushroom* untuk jumlah *record* 150. Pada gambar tersebut dapat diamati bahwa dengan nilai k maksimum, maka nilai kesalahan akan sangat tinggi, yaitu dengan *range* nilai kesalahan 0.193 hingga 0.289. Namun jika nilai k = 3, 5 dan 10, maka nilai kesalahan mengalami kenaikan pada setiap penambahan nilai k, namun tidak terlalu besar, yaitu pada *range* 0 hingga 0.089. Untuk jumlah *record* 75 didapatkan pula nilai kesalahan hasil k maksimum yang tinggi berada pada *range* 0.348 hingga 0.583, sedangkan untuk k lainnya nilai kesalahan hanya berada pada *range* 0 hingga 0.042. Untuk jumlah *record* 300 didapatkan pula nilai kesalahan hasil k maksimum yang tinggi berada pada *range* 0.467 hingga 0.6, sedangkan untuk k lainnya nilai kesalahan hanya berada pada *range* 0 hingga 0.022. Hal tersebut menunjukkan bahwa k maksimum kurang baik untuk digunakan.

3. Analisa kinerja metode penanganan *missing value*

a. Metode penanganan *missing value* untuk *data training* kuantitatif

Berdasarkan hasil pengujian data kuantitatif *Mammographic mass* yang ditampilkan pada tabel 4.17, dapat diamati kinerja dari masing-

masing metode penanganan, yaitu *Mean Imputation* dan *KNN Imputation*. Jika dilihat berdasarkan *range* rata-rata nilai kesalahan, metode *Mean Imputation* tampak sedikit lebih baik dengan rata-rata nilai kesalahan yang lebih rendah daripada rata-rata nilai kesalahan dari metode *KNN Imputation*. Berdasarkan tabel tersebut, pada jumlah data sebanyak 75 *record*, 150 *record* dan 300 *record*, prosentase nilai kesalahan yang lebih rendah ditunjukkan oleh hasil dari penanganan *missing value* menggunakan *Mean Imputation*, Berdasarkan tabel 4.4 dan 4.5, rata-rata nilai kesalahan RMSE dari hasil evaluasi metode *Mean Imputation* mencapai 0.208 (20.8%), sedangkan untuk metode *KNN Imputation* sebesar 0.225 (22.5%).

Tabel 4.17 *Range* nilai kesalahan RMSE data kuantitatif *Mammographic Mass*

Jumlah Record	*Range Nilai Kesalahan RMSE*	
	MI	KNNI
75 record	14.8%-21.1%	12.5%-25%
150 record	20.2%-24%	19.7%-26.5%
300 record	19.6%-23.7%	20.5%-26.6%

b. **Metode penanganan *missing value* untuk *data training* kualitatif**

Berdasarkan hasil pengujian data kuantitatif *Mammographic mass* yang ditampilkan pada tabel 4.18, dapat diamati kinerja dari masing-masing metode penanganan, yaitu *Mode Imputation* dan *KNN Imputation*. Jika dilihat berdasarkan rata-rata nilai kesalahan, metode *KNN Imputation* jauh lebih baik dibandingkan metode *Mode Imputation* dengan nilai kesalahan yang lebih rendah, bahkan menyentuh angka 0. Namun hal tersebut tidak berlaku untuk nilai k maksimum, sebab pada hasil pengujian, rata-rata nilai kesalahan dengan k maksimum terpaut sangat jauh dengan rata-rata nilai kesalahan dengan nilai k lainnya. Oleh karena itu hasil dari k maksimum tidak diperhitungkan atau dihilangkan. Berdasarkan tabel tersebut nilai kesalahan yang dihasilkan oleh metode *Mode Imputation* mencapai puluhan persen, namun nilai kesalahan yang dihasilkan oleh metode *KNN Imputation* kurang dari 10%. Jika diamati berdasarkan tabel 4.6 dan 4.7, rata-rata nilai *Error Rate* dari

hasil evaluasi metode *Mode Imputation* mencapai 0.353 (35.3%), sedangkan untuk metode *KNN Imputation* sebesar 0.02 (2%).

Tabel 4. 18 *Range Error Rate* data kualitatif *Mushroom*

Jumlah Record	*Range Error Rate*	
	MOI	KNNI
75 record	34.8%-58.3%	0%-4.2%
150 record	19.3%-29.2%	0%-8.9%
300 record	32.6%-40%	0%-2.2%

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari skripsi ini adalah :

1. Penanganan *missing value* dengan metode *Mean Imputation* menunjukkan bahwa jumlah *record* dan jumlah *missing value* tidak terlalu berpengaruh terhadap nilai kesalahan RMSE yang berada pada *range* nilai 14.8% hingga 24%

2. Penanganan *missing value* dengan metode *Mode Imputation* menunjukkan bahwa jumlah *record* cukup mempengaruhi nilai *Error Rate*. Pada hasil percobaan, *range* nilai *Error Rate* sebesar 34.8%-58.3% (untuk 75 *record*), 19.3%-29.2% (untuk 150 *record*), dan 32.6%-40% (untuk 300 *record*). Selain itu, pada metode *Mode Imputation* nilai *Error Rate* juga dipengaruhi oleh jumlah *missing value*. Nilai kesalahan semakin menurun seiring dengan penambahan prosentase *missing value*

3. Penanganan *missing value* dengan metode *KNN Imputation* dilakukan pada kedua jenis data, yaitu kuantitatif dan kualitatif. Pada penanganan *missing value* dalam data kuantitatif, jumlah *record* dan jumlah *missing value* tidak terlalu berpengaruh terhadap nilai kesalahan RMSE yang berada pada *range* nilai 12.5% hingga 26.6%. Sedangakan pada implementasi *KNN Imputation* dalam data kualitatif, hasil penanganan dengan jumlah *Neighbors* (k) maksimum dihilangkan karena hasilnya terpaut jauh dibandingkan dengan hasil penanganan dengan jumlah *Neighbors* (k) yang tidak terlalu besar. Sehingga berdasarkan hasil pengujian pada k=3, 5 dan 10, jumlah *record* dan jumlah *missing value* cukup mempengaruhi nilai kesalahan. Seiring dengan penambahan nilai k, nilai kesalahan semakin meningkat namun tidak terlalu besar

4. Metode penanganan *missing value Mean Imputation* sedikit lebih baik untuk digunakan pada data bertipe kuantitatif jika dibandingkan dengan *KNN Imputation*. Hal tersebut ditunjukkan dengan nilai rata-rata kesalahan dari *Mean Imputation* sebesar 20.8%, sedangkan untuk *KNN Imputation*

sebesar 22.5%. Selain itu, pada data bertipe kualitatif disimpulkan bahwa metode penanganan *KNN Imputation* lebih baik untuk diterapkan pada penanganan *missing value* jika dibandingkan dengan metode *Mode Imputation*. Hal tersebut ditunjukkan dengan nilai rata-rata kesalahan dari *KNN Imputation* hanya sebesar 2%, sedangkan untuk *Mode Imputation* sebesar 35.3%. Untuk *KNN Imputation*, sebaiknya jumlah *neighbor* (k) yang dipilih tidak terlalu besar. Hal tersebut ditunjukkan oleh nilai *Error Rate* pada data kualitatif yang jauh lebih tinggi jika dibandingkan dengan nilai *Error Rate* yang dihasilkan dari jumlah *neighbor* (k) yang kecil

5.2 Saran

Beberapa saran yang mungkin menjadi pertimbangan adalah sebagai berikut :

1. Tipe data yang ditangani bukan hanya *integer* dan *string*, tetapi mencakup tipe data *double, float,* dan sebagainya
2. Sebaran statistika pada data perlu diperhatikan

DAFTAR PUSTAKA

Acuna,Edgar dan Rodriguez,Caroline. 2003. *The Treatment of Missing Values and its Effect in the Classifier Accuracy*. University of Puerto Rico at Mayaguez, Mayaguez.

Anonymous. Pengertian dan Karakteristik Data. http://meilanynonsi.upy.ac.id/files/stat/modul2.pdf. diakses tanggal 15 April 2010

Anonymous. Pengertian Data dan Informasi. http://erma_sova.staff.gunadarma.ac.id/.../PENGERTIAN+DATA+&+INFORMASI.ppt. Diakses tanggal 15 April 2010.

Batista, G.E.A.P.A., Monard,M.C. 2003. *An Analysis of Four Missing Data Treatment Methods for Supervised*. University of SaoPaulo,USP

Chan, P. and Dunn, O.J. 1972. *The Treatment of Missing Values in Discriminant Analysis*. Journal of the American Statistical Association, 6, 473-477.

Gheyas, Iffat A. dan Smith,Leslie S.2009. *A Novel Nonparametric Multiple Imputation Algorithm for Estimating Missing Data*. Proceedings of the World Congress on Engineering 2009 Vol II WCE 2009, July 1 - 3, 2009, London, U.K.

Guo, Yike dan Grossman, Robert. 1999. *High Performance Data Mining*. Kluwer Academic Publishers, USA.

Han, Jiawei and Khamber, Micheline. 2001. *Data mining : Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco,USA.

Kantardzic, Mehmed. 2003. *Data mining : Concepts, Models, Methods and Algorithm*. John Wiley & Sons, New York.

Kusrini, Luthfi, dan Taufiq ,Emha. 2009. *Algoritma Data mining*. Andi Offset, Yogyakarta.

Lakshminarayan, K., Harp, S. A., & Samad, T. 2004. *Imputation of Missing Data in Industrial Databases*. Applied Intelligence, hal. 259–275.

Larose, Daniel T. 2005. *Discovering Knowledge in Data : An Introduction to Data mining.* John Willey&Sons,Inc, New York.

Little, R. J. and Rubin, D.B. 2002. *Statistical Analysis with Missing Data Second Edition*. John Wiley and Sons, New York.

Mannila, Smyth, and Hand, David.2001. *Principle of Data mining*. MIT Press, Cambridge.

Moertini,Veronika S. 2002. *Data mining sebagai Solusi Bisnis*. Integral,vol.7 no.1

Moradian,Mehdi dan Baraani,Ahmad. 2009. *KNNBA : K-Nearest-Neighbor-Based-Association Algorithm*. University of Isfahan, Isfahan, Iran. http://jatit.org . Diakses tanggal 2 Maret 2010.

Mundfrom, Daniel J., dan Whitcommb, Alan. 1998. *Imputation Missing values : The Effect on the Accuracy of Classification*. Paper presented at the Annual Meeting of the American Educational Research Association (San Diego,CA)

Nguyen, L. N., & Scherer, W. T. 2003. *Imputation Techniques to Account for Missing Data in Support of Intelligent Transportation Systems Applications* University of Virginia, USA.

Olivas, Emilio Soria,dkk. 2010. *Handbook of Research on Machine Learning Application and Trends : Algorithms, Methods, and Techniques*. IGI Global, United States of America

Pradana, R Jaka Arya. 2008. Teknik Optimasi. http://aajaka. multiply.com/ journal/item/6 / TEKNIK_OPTIMASI_. Diakses tanggal 19 Januari 2011

Tan, Pang-Ning, Michael Steinbach dan Vipin Kumar. 2004. *Introducing to data mining*. New York. http://www-

users.cs.umn.edu/~kumar/dmbook/index.php. Diakses tanggal 14 Februari 2010.

The Scribner. 1979. *Bantam English Dictionary*. Amerika Serikat dan Canada. http://ronawajah.wordpress.com/2007/05/29/kinerja-apa-itu/. Diakses tanggal 20 November 2010.

Turban,dkk. 2005. *Decision Support Systems and Intelligent Systems*. Andi Offset, Yogyakarta

Troyanskaya, Olga, dkk. 2001. *Missing value estimation methods for DNA microarrays*. Oxford University Press, Inggris. http://sci2s.ugr.es/MVDM/pdf/troyanskaya_cantor_sherlock01.pdf . Diakses tanggal 15 Februari 2010.

Ramadhan,Riza. 2006. *Penerapan Pohon untuk Klasifikasi Dokumen Teks Berbahasa Inggris*. Sekolah Teknik Elektro dan Informatika,Institut Teknologi Bandung

LAMPIRAN

Lampiran 1. Deskripsi data *Mammographic Mass*

1. Title: Mammographic Mass Data
2. Sources:
   (a) Original owners of database:
   Prof. Dr. R diger Schulz-Wendtland Institute of Radiology, Gynaecological Radiology, University Erlangen-Nuremberg Universit tsstra e 21-23 91054 Erlangen, Germany

   (b) Donor of database:
   Matthias Elter Fraunhofer Institute for Integrated Circuits (IIS) Image Processing and Medical Engineering Department (BMT) Am Wolfsmantel 3391058 Erlangen, Germany
   matthias.elter@iis.fraunhofer.de
   (49) 9131-7767327

   (c) Date received: October 2007

3. Past Usage:
   M. Elter, R. Schulz-Wendtland and T. Wittenberg (2007) :
   The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process.
   Medical Physics 34(11), pp. 4164-4172
4. Relevant Information:
   Mammography is the most effective method for breast cancer screening available today. However, the low positive predictive value of breast biopsy resulting from mammogram interpretation leads to approximately 70% unnecessary biopsies with benign outcomes. To reduce the high number of unnecessary breast biopsies, several computer-aided diagnosis (CAD) systems have been proposed in the last years.These systems help physicians in their decision to perform a breast biopsy on a suspicious lesion seen in a mammogram or to perform a short term follow-up examination instead. This data set can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes

together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006. Each instance has an associated BI-RADS assessment ranging from 1 (definitely benign) to 5 (highly suggestive of malignancy) assigned in a double-review process by physicians. Assuming that all cases with BI-RADS assessments greater or equal a given value (varying from 1 to 5), are malignant and the other cases benign, sensitivities and associated specificities can be calculated. These can be an indication of how well a CAD system performs compared to the radiologists.

5. Number of Instances: 961
6. Number of Attributes: 6 (1 goal field, 1 non-predictive, 4 predictive attributes)
7. Attribute Information:
   1. BI-RADS assessment: 1 to 5 (ordinal)
   2. Age: patient's age in years (integer)
   3. Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
   4. Margin: mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)
   5. Density: mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)
   6. Severity: benign=0 or malignant=1 (binominal)
8. Missing Attribute Values: Yes
   - BI-RADS assessment: 2
   - Age: 5
   - Shape: 31
   - Margin: 48
   - Density: 76
   - Severity: 0
9. Class Distribution: benign: 516; malignant: 445

Lampiran 2. Deskripsi data *Mushroom*.

1. Title: Mushroom Database
2. Sources:
   (a) Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf
   (b) Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)
   (c) Date: 27 April 1987
3. Relevant Information:
   This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like ``leaflets three, let it be'' for Poisonous Oak and Ivy.
4. Number of Instances: 8124
5. Number of Attributes: 22 (all nominally valued)
6. Attribute Information: (classes: edible=e, poisonous=p)
   1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
   2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
   3. cap-color: brown=n,buff=b,cinnamon=c, gray=g, green=r, pink=p,purple=u,red=e,white=w,yellow=y
   4. bruises?: bruises=t,no=f
   5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
   6. gill-attachment: attached=a,descending=d,free=f, notched=n
   7. gill-spacing: close=c,crowded=w,distant=d
   8. gill-size: broad=b,narrow=n
   9. gill-color: black=k,brown=n,buff=b, chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w, yellow=y
   10. stalk-shape: enlarging=e,tapering=t
   11. stalk-root: bulbous=b,club=c,cup=u, equal=e, rhizomorphs=z, rooted=r,missing=?
   12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
   13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s

14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g, orange=o, pink=p,red=e, white=w, yellow=y
15. stalk-color-below-ring: brown= n,buff=b,cinnamon=c, gray=g, orange=o, pink=p, red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19.ring-type: cobwebby=c,evanescent=e,flaring=f, large=l, none=n,pendant=p,sheathing=s,zone=z
20.spore-print-color: black=k,brown=n,buff=b,chocolate=h, green=r, orange=o,purple=u, white=w, yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

8. Missing Attribute Values: 2480 of them (denoted by "?"), all for attribute #11.

9. Class Distribution:
   -- edible: 4208 (51.8%)
   -- poisonous: 3916 (48.2%)
   -- total: 8124 instances

Lampiran 3. Hasil Pengujian data *Mammographic Mass.*

Jumlah Record	Jumlah Missing Value	Metode		Uji coba	Nama Tabel	RMSE	Rata-rata RMSE
75 record	5%	Mean Imputation		uji 1	75_5_1_Mammo.csv	0.146	0.148
				uji 2	75_5_2_Mammo.csv	0.112	
				uji 3	75_5_3_Mammo.csv	0.186	
		KNNI	k=3	uji 1	75_5_1_Mammo.csv	0.107	0.162
				uji 2	75_5_2_Mammo.csv	0.17	
				uji 3	75_5_3_Mammo.csv	0.209	
			k=5	uji 1	75_5_1_Mammo.csv	0.091	0.143333333
				uji 2	75_5_2_Mammo.csv	0.151	
				uji 3	75_5_3_Mammo.csv	0.188	
			k=10	uji 1	75_5_1_Mammo.csv	0.105	0.125
				uji 2	75_5_2_Mammo.csv	0.119	
				uji 3	75_5_3_Mammo.csv	0.151	
			k maks	uji 1	75_5_1_Mammo.csv	0.161	0.175
				uji 2	75_5_2_Mammo.csv	0.179	
				uji 3	75_5_3_Mammo.csv	0.185	

	10%	Mean Imputation		uji 1	75_10_1_Mammo.csv	0.305	0.194666667
				uji 2	75_10_2_Mammo.csv	0.159	
				uji 3	75_10_3_Mammo.csv	0.12	
		KNNI	k=3	uji 1	75_10_1_Mammo.csv	0.316	0.224666667
				uji 2	75_10_2_Mammo.csv	0.182	
				uji 3	75_10_3_Mammo.csv	0.176	
			k=5	uji 1	75_10_1_Mammo.csv	0.29	0.199
				uji 2	75_10_2_Mammo.csv	0.146	
				uji 3	75_10_3_Mammo.csv	0.161	
			k=10	uji 1	75_10_1_Mammo.csv	0.291	0.206333333
				uji 2	75_10_2_Mammo.csv	0.158	
				uji 3	75_10_3_Mammo.csv	0.17	
			k maks	uji 1	75_10_1_Mammo.csv	0.339	0.25
				uji 2	75_10_2_Mammo.csv	0.22	
				uji 3	75_10_3_Mammo.csv	0.191	
	15%	Mean Imputation		uji 1	75_15_1_Mammo.csv	0.145	0.202666667
				uji 2	75_15_2_Mammo.csv	0.241	
				uji 3	75_15_3_Mammo.csv	0.222	

		KNNI	k=3	uji 1	75_15_1_Mammo.csv	0.211	0.231
				uji 2	75_15_2_Mammo.csv	0.25	
				uji 3	75_15_3_Mammo.csv	0.232	
			k=5	uji 1	75_15_1_Mammo.csv	0.213	0.231333333
				uji 2	75_15_2_Mammo.csv	0.255	
				uji 3	75_15_3_Mammo.csv	0.226	
			k=10	uji 1	75_15_1_Mammo.csv	0.2	0.22
				uji 2	75_15_2_Mammo.csv	0.246	
				uji 3	75_15_3_Mammo.csv	0.214	
			k maks	uji 1	75_15_1_Mammo.csv	0.186	0.238333333
				uji 2	75_15_2_Mammo.csv	0.258	
				uji 3	75_15_3_Mammo.csv	0.271	
	30%	Mean Imputation		uji 1	75_30_1_Mammo.csv	0.209	0.210666667
				uji 2	75_30_2_Mammo.csv	0.205	
				uji 3	75_30_3_Mammo.csv	0.218	
		KNNI	k=3	uji 1	75_30_1_Mammo.csv	0.272	0.243
				uji 2	75_30_2_Mammo.csv	0.235	
				uji 3	75_30_3_Mammo.csv	0.222	

			k=5	uji 1	75_30_1_Mammo.csv	0.237	0.236666667
				uji 2	75_30_2_Mammo.csv	0.246	
				uji 3	75_30_3_Mammo.csv	0.227	
			k=10	uji 1	75_30_1_Mammo.csv	0.221	0.222
				uji 2	75_30_2_Mammo.csv	0.235	
				uji 3	75_30_3_Mammo.csv	0.21	
			k maks	uji 1	75_30_1_Mammo.csv	0.222	0.220666667
				uji 2	75_30_2_Mammo.csv	0.221	
				uji 3	75_30_3_Mammo.csv	0.219	
150 record	5%	Mean Imputation		uji 1	150_5_1_Mammo.csv	0.208	0.202333333
				uji 2	150_5_2_Mammo.csv	0.285	
				uji 3	150_5_3_Mammo.csv	0.114	
		KNNI	k=3	uji 1	150_5_1_Mammo.csv	0.262	0.207666667
				uji 2	150_5_2_Mammo.csv	0.207	
				uji 3	150_5_3_Mammo.csv	0.154	
			k=5	uji 1	150_5_1_Mammo.csv	0.263	0.215666667
				uji 2	150_5_2_Mammo.csv	0.249	
				uji 3	150_5_3_Mammo.csv	0.135	

			k=10	uji 1	150_5_1_Mammo.csv	0.262	0.211
				uji 2	150_5_2_Mammo.csv	0.201	
				uji 3	150_5_3_Mammo.csv	0.17	
			k maks	uji 1	150_5_1_Mammo.csv	0.204	0.197333333
				uji 2	150_5_2_Mammo.csv	0.243	
				uji 3	150_5_3_Mammo.csv	0.145	
	10%	Mean Imputation		uji 1	150_10_1_Mammo.csv	0.175	0.233
				uji 2	150_10_2_Mammo.csv	0.168	
				uji 3	150_10_3_Mammo.csv	0.356	
		KNNI	k=3	uji 1	150_10_1_Mammo.csv	0.156	0.241
				uji 2	150_10_2_Mammo.csv	0.209	
				uji 3	150_10_3_Mammo.csv	0.358	
			k=5	uji 1	150_10_1_Mammo.csv	0.189	0.250666667
				uji 2	150_10_2_Mammo.csv	0.204	
				uji 3	150_10_3_Mammo.csv	0.359	
			k=10	uji 1	150_10_1_Mammo.csv	0.164	0.228
				uji 2	150_10_2_Mammo.csv	0.185	
				uji 3	150_10_3_Mammo.csv	0.335	

			k maks	uji 1	150_10_1_Mammo.csv	0.209	0.262333333
				uji 2	150_10_2_Mammo.csv	0.219	
				uji 3	150_10_3_Mammo.csv	0.359	
	15%	Mean Imputation		uji 1	150_15_1_Mammo.csv	0.233	0.216
				uji 2	150_15_2_Mammo.csv	0.196	
				uji 3	150_15_3_Mammo.csv	0.219	
		KNNI	k=3	uji 1	150_15_1_Mammo.csv	0.265	0.256333333
				uji 2	150_15_2_Mammo.csv	0.246	
				uji 3	150_15_3_Mammo.csv	0.258	
			k=5	uji 1	150_15_1_Mammo.csv	0.245	0.236666667
				uji 2	150_15_2_Mammo.csv	0.224	
				uji 3	150_15_3_Mammo.csv	0.241	
			k=10	uji 1	150_15_1_Mammo.csv	0.231	0.228333333
				uji 2	150_15_2_Mammo.csv	0.228	
				uji 3	150_15_3_Mammo.csv	0.226	
			k maks	uji 1	150_15_1_Mammo.csv	0.239	0.241666667
				uji 2	150_15_2_Mammo.csv	0.228	
				uji 3	150_15_3_Mammo.csv	0.258	

	30%	Mean Imputation		uji 1	150_30_1_Mammo.csv	0.242	0.24
				uji 2	150_30_2_Mammo.csv	0.212	
				uji 3	150_30_3_Mammo.csv	0.266	
		KNNI	k=3	uji 1	150_30_1_Mammo.csv	0.252	0.265333333
				uji 2	150_30_2_Mammo.csv	0.233	
				uji 3	150_30_3_Mammo.csv	0.311	
			k=5	uji 1	150_30_1_Mammo.csv	0.247	0.245666667
				uji 2	150_30_2_Mammo.csv	0.209	
				uji 3	150_30_3_Mammo.csv	0.281	
			k=10	uji 1	150_30_1_Mammo.csv	0.243	0.239
				uji 2	150_30_2_Mammo.csv	0.189	
				uji 3	150_30_3_Mammo.csv	0.285	
			k maks	uji 1	150_30_1_Mammo.csv	0.246	0.249333333
				uji 2	150_30_2_Mammo.csv	0.237	
				uji 3	150_30_3_Mammo.csv	0.265	
300 record	5%	Mean Imputation		uji 1	300_5_1_Mammo.csv	0.185	0.196333333
				uji 2	300_5_2_Mammo.csv	0.213	
				uji 3	300_5_3_Mammo.csv	0.191	

		KNNI	k=3	uji 1	300_5_1_Mammo.csv	0.275	0.242
				uji 2	300_5_2_Mammo.csv	0.218	
				uji 3	300_5_3_Mammo.csv	0.233	
			k=5	uji 1	300_5_1_Mammo.csv	0.239	0.223666667
				uji 2	300_5_2_Mammo.csv	0.205	
				uji 3	300_5_3_Mammo.csv	0.227	
			k=10	uji 1	300_5_1_Mammo.csv	0.214	0.205333333
				uji 2	300_5_2_Mammo.csv	0.205	
				uji 3	300_5_3_Mammo.csv	0.197	
			k maks	uji 1	300_5_1_Mammo.csv	0.189	0.205666667
				uji 2	300_5_2_Mammo.csv	0.214	
				uji 3	300_5_3_Mammo.csv	0.214	
	10%	Mean Imputation		uji 1	300_10_1_Mammo.csv	0.176	0.217
				uji 2	300_10_2_Mammo.csv	0.265	
				uji 3	300_10_3_Mammo.csv	0.21	
		KNNI	k=3	uji 1	300_10_1_Mammo.csv	0.224	0.246
				uji 2	300_10_2_Mammo.csv	0.244	
				uji 3	300_10_3_Mammo.csv	0.27	

			k=5	uji 1	300_10_1_Mammo.csv	0.182	0.227
				uji 2	300_10_2_Mammo.csv	0.244	
				uji 3	300_10_3_Mammo.csv	0.255	
			k=10	uji 1	300_10_1_Mammo.csv	0.166	0.216333333
				uji 2	300_10_2_Mammo.csv	0.241	
				uji 3	300_10_3_Mammo.csv	0.242	
			k maks	uji 1	300_10_1_Mammo.csv	0.188	0.231
				uji 2	300_10_2_Mammo.csv	0.258	
				uji 3	300_10_3_Mammo.csv	0.247	
	15%	Mean Imputation		uji 1	300_15_1_Mammo.csv	0.208	0.202333333
				uji 2	300_15_2_Mammo.csv	0.192	
				uji 3	300_15_3_Mammo.csv	0.207	
		KNNI	k=3	uji 1	300_15_1_Mammo.csv	0.246	0.229
				uji 2	300_15_2_Mammo.csv	0.227	
				uji 3	300_15_3_Mammo.csv	0.214	
			k=5	uji 1	300_15_1_Mammo.csv	0.233	0.212333333
				uji 2	300_15_2_Mammo.csv	0.215	
				uji 3	300_15_3_Mammo.csv	0.189	

			k=10	uji 1	300_15_1_Mammo.csv	0.222	0.207
				uji 2	300_15_2_Mammo.csv	0.218	
				uji 3	300_15_3_Mammo.csv	0.181	
			k maks	uji 1	300_15_1_Mammo.csv	0.236	0.225333333
				uji 2	300_15_2_Mammo.csv	0.222	
				uji 3	300_15_3_Mammo.csv	0.218	
	30%	Mean Imputation		uji 1	300_30_1_Mammo.csv	0.235	0.237
				uji 2	300_30_2_Mammo.csv	0.255	
				uji 3	300_30_3_Mammo.csv	0.221	
		KNNI	k=3	uji 1	300_30_1_Mammo.csv	0.26	0.260333333
				uji 2	300_30_2_Mammo.csv	0.252	
				uji 3	300_30_3_Mammo.csv	0.269	
			k=5	uji 1	300_30_1_Mammo.csv	0.24	0.253666667
				uji 2	300_30_2_Mammo.csv	0.261	
				uji 3	300_30_3_Mammo.csv	0.26	
			k=10	uji 1	300_30_1_Mammo.csv	0.239	0.247666667
				uji 2	300_30_2_Mammo.csv	0.262	
				uji 3	300_30_3_Mammo.csv	0.242	

			k maks	uji 1	300_30_1_Mammo.csv	0.251	0.265666667
				uji 2	300_30_2_Mammo.csv	0.292	
				uji 3	300_30_3_Mammo.csv	0.254	

Lampiran 4. Hasil Pengujian data *Mushroom*.

Jumlah Record	Jumlah Missing Value	Metode		Uji coba	Nama Tabel	Error Rate	Rata-rata Error Rate
75 record	5%	Mode Imputation		uji 1	75_5_1_Mush.csv	0.5	0.583333333
				uji 2	75_5_2_Mush.csv	0.75	
				uji 3	75_5_3_Mush.csv	0.5	
		KNNI	k=3	uji 1	75_5_1_Mush.csv	0	0
				uji 2	75_5_2_Mush.csv	0	
				uji 3	75_5_3_Mush.csv	0	
			k=5	uji 1	75_5_1_Mush.csv	0	0
				uji 2	75_5_2_Mush.csv	0	
				uji 3	75_5_3_Mush.csv	0	
			k=10	uji 1	75_5_1_Mush.csv	0	0
				uji 2	75_5_2_Mush.csv	0	
				uji 3	75_5_3_Mush.csv	0	
			k maks	uji 1	75_5_1_Mush.csv	0.5	0.583333333
				uji 2	75_5_2_Mush.csv	0.75	
				uji 3	75_5_3_Mush.csv	0.5	

	10%	Mode Imputation		uji 1	75_10_1_Mush.csv	0.625	0.458333333
				uji 2	75_10_2_Mush.csv	0.375	
				uji 3	75_10_3_Mush.csv	0.375	
		KNNI	k=3	uji 1	75_10_1_Mush.csv	0	0
				uji 2	75_10_2_Mush.csv	0	
				uji 3	75_10_3_Mush.csv	0	
			k=5	uji 1	75_10_1_Mush.csv	0	0
				uji 2	75_10_2_Mush.csv	0	
				uji 3	75_10_3_Mush.csv	0	
			k=10	uji 1	75_10_1_Mush.csv	0.125	0.041666667
				uji 2	75_10_2_Mush.csv	0	
				uji 3	75_10_3_Mush.csv	0	
			k maks	uji 1	75_10_1_Mush.csv	0.625	0.458333333
				uji 2	75_10_2_Mush.csv	0.375	
				uji 3	75_10_3_Mush.csv	0.375	
	15%	Mode Imputation		uji 1	75_15_1_Mush.csv	0.333	0.361
				uji 2	75_15_2_Mush.csv	0.333	
				uji 3	75_15_3_Mush.csv	0.417	

		KNNI	k=3	uji 1	75_15_1_Mush.csv	0	0
				uji 2	75_15_2_Mush.csv	0	
				uji 3	75_15_3_Mush.csv	0	
			k=5	uji 1	75_15_1_Mush.csv	0	0
				uji 2	75_15_2_Mush.csv	0	
				uji 3	75_15_3_Mush.csv	0	
			k=10	uji 1	75_15_1_Mush.csv	0	0.027666667
				uji 2	75_15_2_Mush.csv	0	
				uji 3	75_15_3_Mush.csv	0.083	
			k maks	uji 1	75_15_1_Mush.csv	0.333	0.361
				uji 2	75_15_2_Mush.csv	0.333	
				uji 3	75_15_3_Mush.csv	0.417	
	30%	Mode Imputation		uji 1	75_30_1_Mush.csv	0.348	0.348
				uji 2	75_30_2_Mush.csv	0.348	
				uji 3	75_30_3_Mush.csv	0.348	
		KNNI	k=3	uji 1	75_30_1_Mush.csv	0	0
				uji 2	75_30_2_Mush.csv	0	
				uji 3	75_30_3_Mush.csv	0	

			k=5	uji 1	75_30_1_Mush.csv	0	0
				uji 2	75_30_2_Mush.csv	0	
				uji 3	75_30_3_Mush.csv	0	
			k=10	uji 1	75_30_1_Mush.csv	0.044	0.029333333
				uji 2	75_30_2_Mush.csv	0	
				uji 3	75_30_3_Mush.csv	0.044	
			k maks	uji 1	75_30_1_Mush.csv	0.348	0.348
				uji 2	75_30_2_Mush.csv	0.348	
				uji 3	75_30_3_Mush.csv	0.348	
150 record	5%	Mode Imputation		uji 1	150_5_1_Mush.csv	0.25	0.291666667
				uji 2	150_5_2_Mush.csv	0.25	
				uji 3	150_5_3_Mush.csv	0.375	
		KNNI	k=3	uji 1	150_5_1_Mush.csv	0	0
				uji 2	150_5_2_Mush.csv	0	
				uji 3	150_5_3_Mush.csv	0	
			k=5	uji 1	150_5_1_Mush.csv	0.125	0.041666667
				uji 2	150_5_2_Mush.csv	0	
				uji 3	150_5_3_Mush.csv	0	

			k=10	uji 1	150_5_1_Mush.csv	0.125	0.041666667
				uji 2	150_5_2_Mush.csv	0	
				uji 3	150_5_3_Mush.csv	0	
			k maks	uji 1	150_5_1_Mush.csv	0.25	0.291666667
				uji 2	150_5_2_Mush.csv	0.25	
				uji 3	150_5_3_Mush.csv	0.375	
	10%	Mode Imputation		uji 1	150_10_1_Mush.csv	0.4	0.289
				uji 2	150_10_2_Mush.csv	0.2	
				uji 3	150_10_3_Mush.csv	0.267	
		KNNI	k=3	uji 1	150_10_1_Mush.csv	0.133	0.066666667
				uji 2	150_10_2_Mush.csv	0	
				uji 3	150_10_3_Mush.csv	0.067	
			k=5	uji 1	150_10_1_Mush.csv	0.2	0.089
				uji 2	150_10_2_Mush.csv	0	
				uji 3	150_10_3_Mush.csv	0.067	
			k=10	uji 1	150_10_1_Mush.csv	0.2	0.089
				uji 2	150_10_2_Mush.csv	0	
				uji 3	150_10_3_Mush.csv	0.067	

			k maks	uji 1	150_10_1_Mush.csv	0.4	0.289
				uji 2	150_10_2_Mush.csv	0.2	
				uji 3	150_10_3_Mush.csv	0.267	
	15%	Mode Imputation		uji 1	150_15_1_Mush.csv	0.261	0.217333333
				uji 2	150_15_2_Mush.csv	0.261	
				uji 3	150_15_3_Mush.csv	0.13	
		KNNI	k=3	uji 1	150_15_1_Mush.csv	0.087	0.029
				uji 2	150_15_2_Mush.csv	0	
				uji 3	150_15_3_Mush.csv	0	
			k=5	uji 1	150_15_1_Mush.csv	0.087	0.043666667
				uji 2	150_15_2_Mush.csv	0.044	
				uji 3	150_15_3_Mush.csv	0	
			k=10	uji 1	150_15_1_Mush.csv	0.087	0.058333333
				uji 2	150_15_2_Mush.csv	0.044	
				uji 3	150_15_3_Mush.csv	0.044	
			k maks	uji 1	150_15_1_Mush.csv	0.261	0.217333333
				uji 2	150_15_2_Mush.csv	0.261	
				uji 3	150_15_3_Mush.csv	0.13	

	30%	Mode Imputation		uji 1	150_30_1_Mush.csv	0.244	0.192666667
				uji 2	150_30_2_Mush.csv	0.178	
				uji 3	150_30_3_Mush.csv	0.156	
		KNNI	k=3	uji 1	150_30_1_Mush.csv	0.067	0.037
				uji 2	150_30_2_Mush.csv	0.044	
				uji 3	150_30_3_Mush.csv	0	
			k=5	uji 1	150_30_1_Mush.csv	0.044	0.029333333
				uji 2	150_30_2_Mush.csv	0.044	
				uji 3	150_30_3_Mush.csv	0	
			k=10	uji 1	150_30_1_Mush.csv	0.111	0.066666667
				uji 2	150_30_2_Mush.csv	0.089	
				uji 3	150_30_3_Mush.csv	0	
			k maks	uji 1	150_30_1_Mush.csv	0.244	0.192666667
				uji 2	150_30_2_Mush.csv	0.178	
				uji 3	150_30_3_Mush.csv	0.156	
300 record	5%	Mode Imputation		uji 1	300_5_1_Mush.csv	0.4	0.4
				uji 2	300_5_2_Mush.csv	0.4	
				uji 3	300_5_3_Mush.csv	0.4	

		KNNI	k=3	uji 1	300_5_1_Mush.csv	0	0
				uji 2	300_5_2_Mush.csv	0	
				uji 3	300_5_3_Mush.csv	0	
			k=5	uji 1	300_5_1_Mush.csv	0	0.022333333
				uji 2	300_5_2_Mush.csv	0.067	
				uji 3	300_5_3_Mush.csv	0	
			k=10	uji 1	300_5_1_Mush.csv	0	0
				uji 2	300_5_2_Mush.csv	0	
				uji 3	300_5_3_Mush.csv	0	
			k maks	uji 1	300_5_1_Mush.csv	0.533	0.6
				uji 2	300_5_2_Mush.csv	0.667	
				uji 3	300_5_3_Mush.csv	0.6	
	10%	Mode Imputation		uji 1	300_10_1_Mush.csv	0.333	0.377666667
				uji 2	300_10_2_Mush.csv	0.4	
				uji 3	300_10_3_Mush.csv	0.4	
		KNNI	k=3	uji 1	300_10_1_Mush.csv	0	0
				uji 2	300_10_2_Mush.csv	0	
				uji 3	300_10_3_Mush.csv	0	

			k=5	uji 1	300_10_1_Mush.csv	0	0
				uji 2	300_10_2_Mush.csv	0	
				uji 3	300_10_3_Mush.csv	0	
			k=10	uji 1	300_10_1_Mush.csv	0	0
				uji 2	300_10_2_Mush.csv	0	
				uji 3	300_10_3_Mush.csv	0	
			k maks	uji 1	300_10_1_Mush.csv	0.467	0.555666667
				uji 2	300_10_2_Mush.csv	0.6	
				uji 3	300_10_3_Mush.csv	0.6	
	15%	Mode Imputation		uji 1	300_15_1_Mush.csv	0.311	0.325666667
				uji 2	300_15_2_Mush.csv	0.333	
				uji 3	300_15_3_Mush.csv	0.333	
		KNNI	k=3	uji 1	300_15_1_Mush.csv	0	0.007333333
				uji 2	300_15_2_Mush.csv	0.022	
				uji 3	300_15_3_Mush.csv	0	
			k=5	uji 1	300_15_1_Mush.csv	0	0.007333333
				uji 2	300_15_2_Mush.csv	0.022	
				uji 3	300_15_3_Mush.csv	0	

			k=10	uji 1	300_15_1_Mush.csv	0	0
				uji 2	300_15_2_Mush.csv	0	
				uji 3	300_15_3_Mush.csv	0	
			k maks	uji 1	300_15_1_Mush.csv	0.467	0.466666667
				uji 2	300_15_2_Mush.csv	0.444	
				uji 3	300_15_3_Mush.csv	0.489	
	30%	Mode Imputation		uji 1	300_30_1_Mush.csv	0.333	0.396
				uji 2	300_30_2_Mush.csv	0.444	
				uji 3	300_30_3_Mush.csv	0.411	
		KNNI	k=3	uji 1	300_30_1_Mush.csv	0	0
				uji 2	300_30_2_Mush.csv	0	
				uji 3	300_30_3_Mush.csv	0	
			k=5	uji 1	300_30_1_Mush.csv	0	0
				uji 2	300_30_2_Mush.csv	0	
				uji 3	300_30_3_Mush.csv	0	
			k=10	uji 1	300_30_1_Mush.csv	0	0
				uji 2	300_30_2_Mush.csv	0	
				uji 3	300_30_3_Mush.csv	0	

			k maks	uji 1	300_30_1_Mush.csv	0.511	0.514666667
				uji 2	300_30_2_Mush.csv	0.522	
				uji 3	300_30_3_Mush.csv	0.511	


UNIVERSITAS BRAWIJAYA