

2020

machine **AI** **LEARNING** **& REASONING** Fuzzy Logic

Algoritma, Manual, Matlab, & Rapid Miner



Budy Santoso
Azminuddin I. S. Azis
Zohrahayaty

deepublish

Machine LEARNING & REASONING Fuzzy Logic

accuracy
sensitivity
specificity
precision
f-measure
MSE
RMSE
SEE
MAPE
shilhouette
purity
lift ratio
correlation
T-Test
anova

encoding, discretization, aggregation, normalization
missing value replacement, anomaly detection

linear regression
support vector machine
fuzzy logic
clustering
classification
regression
association
k-nearest neighbor
C4.5 fuzzy c-means
principal component analysis
singular value decomposition
mathematics statistic computing
particle swarm optimization
dimensionality reduction
feature selection / attribute weighting
feature extraction

min-max
z-score
sigmoidal
decimal scaling
softmax
naive bayes
artificial neural network
adaboost
bagging
k-means
forward selection
genetic algorithm
backward elimination

Kami persembahkan untuk:

*Orang Tua & Guru Kami
yang senantiasa mengajarkan ilmu dan adab
menuju kebijaksanaan dan kemuliaan hidup.*

*Istri & Anak Kami
yang senantiasa memberikan kekuatan
motivasi dan keikhlasan untuk menembus keterbatasan.*

Prakata

Alhamdulillah penulis panjatkan kehadiran Allah Subhanahu wa Ta'ala atas berkat rahmat dan hidayah-Nya, sehingga kami dapat menyelesaikan buku ajar berjudul: "***Machine Learning & Reasoning Fuzzy Logic***". Salam dan taslim kepada panutan kita, Nabi Muhammad Shallallahu 'Alaihi wa Sallam atas perjuangan Beliau yang telah mengantar kita dari alam kebodohan ke alam yang penuh ilmu pengetahuan.

Saat ini, metode-metode komputasi telah berkembang semakin cerdas. Pada prinsipnya, metode-metode komputasi cerdas atau biasa diistilahkan dengan *soft computing* dapat dikategorikan menjadi metode-metode *searching*, *reasoning*, dan *learning*. Metode-metode *searching* merepresentasikan masalah ke dalam *state* dan ruang masalah lalu menggunakan strategi pencarian untuk menemukan solusi. Sedangkan metode-metode *reasoning* merepresentasikan masalah ke dalam basis pengetahuan lalu menggunakan strategi penalaran untuk menemukan solusi. Pendekatan *searching* dan *reasoning* mengharuskan adanya aturan-aturan yang berlaku, namun terkadang aturan-aturan tidak selalu bisa didefinisikan secara benar dan lengkap, maka pendekatan *learning* hadir untuk mengatasi kendala tersebut yang diistilahkan dengan *machine learning*.

Jika pendekatan *searching* kesulitan dalam menentukan apakah aturan-aturan sudah benar dan lengkap karena masalah yang dihadapi cukup kompleks sehingga representasi masalah ke dalam *state* menjadi tidak efisien, maka pendekatan *reasoning* dengan representasi *logic* (bahasa formal) merupakan solusinya. Awalnya metode-metode *reasoning* digunakan pada masalah yang memiliki kepastian, bagaimana jika masalah mengandung ketidakpastian? Pendekatan seperti teori probabilitas dan *Fuzzy Logic* merupakan solusinya. Metode-metode dengan pendekatan probabilitas untuk masalah yang mengandung ketidakpastian bersifat peluang. Bagaimana jika masalah mengandung ketidakpastian yang besifat samar? *Fuzzy Logic* merupakan solusinya.

Fuzzy Logic mampu memodelkan fungsi-fungsi nonlinear, mampu mengatasi masalah yang sangat kompleks, didasarkan pada bahasa formal/alami, memiliki toleransi terhadap data yang tidak tepat, dan mampu merepresentasikan pengetahuan pakar ke dalam basis pengetahuannya sebagai aturan-aturan yang berlaku sehingga tidak memerlukan proses *learning*. Namun seperti yang telah dikatakan sebelumnya, terkadang aturan-aturan tidak selalu bisa didefinisikan secara benar dan lengkap, sehingga metode-metode *machine learning* menjadi solusi untuk itu yang juga mampu memodelkan fungsi-fungsi nonlinear, mampu mengatasi masalah yang sangat kompleks, dan memiliki toleransi terhadap data yang tidak tepat.

Kata lainnya, *Fuzzy Logic* tidak mampu melakukan pembelajaran, sementara pendekatan *machine learning* mampu melakukannya, namun tidak mampu melakukan penalaran seperti *Fuzzy Logic*. Melalui pembelajarannya, pendekatan *machine learning* mampu menggali pengetahuan, memprediksi, maupun mengenal pola dari masalah kompleks yang dihadapi, sehingga dapat memodelkan aturan-aturan. Dengan demikian, bagaimana jadinya jika pendekatan *reasoning Fuzzy*

Logic diintegrasikan dengan pendekatan *machine learning*? Bagaimana jika pendekatan *machine learning* memodelkan aturan-aturan yang sulit didefinisikan melalui pembelajarannya untuk digunakan *Fuzzy Logic*? Atau bagaimana jika pendekatan *reasoning Fuzzy Logic* mengoptimalkan proses prediksi, pengenalan pola, dan pemodelan pengetahuan yang dilakukan pendekatan *machine learning*? Buku ini merupakan salah satu jawabannya.

Bagaimana jika metode-metode *machine learning* diintegrasikan dengan metode *Fuzzy Logic*? Buku ini membahas karakteristik, algoritma, manual, dan penerapan (menggunakan *tools Matlab* dan *Rapidminer*) beberapa metode-metode *machine learning* dan pengembangannya dengan *reasoning Fuzzy Logic*, meliputi:

1. **Mengapa Machine Learning & Reasoning Fuzzy Logic?**
2. **Pengantar Machine Learning**
3. **Pra Pengolahan Data**
4. **Evaluasi Model**
5. **Fuzzy Logic**
6. **ANN, SVM, & Fuzzy**
7. **K-Means & FCM**
8. **Naïve Bayes, k-NN & Fuzzy**
9. **Bonus: C4.5, Linear Regression, & A-Priori**

Buku ini dapat pula digunakan sebagai bahan ajar oleh dosen dan mahasiswa Program Studi Teknik Informatika (Ilmu Komputer) pada Mata Kuliah:

1. **Machine Learning;**
2. **Kecerdasan Buatan;**
3. **Soft Computing;**
4. **Data Science;**
5. **Data Mining;** dan sejenisnya.

Kami menyadari sepenuhnya bahwa setiap karya tidak lepas dari bantuan dan dorongan dari berbagai pihak. Untuk itu, kami mengucapkan banyak terima kasih kepada semua pihak yang terkait. “Tak ada gading yang tak retak”, setiap karya manusia tidak ada yang sempurna, maka kami sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya semoga hasil yang telah dicapai ini dapat mendukung program pemerintah untuk mengembangkan ilmu pengetahuan dan teknologi, serta meningkatkan kesejahteraan masyarakat dan daya saing bangsa, Aamiin.

Gorontalo, Juli 2019

Penulis

Daftar Isi

Prakata.....	i
Daftar Isi.....	iii
Daftar Gambar	vii
Daftar Tabel	ix
Daftar Contoh	xi
Daftar Lampiran.....	xv
1. Mengapa Machine Learning & Reasoning Fuzzy Logic?	1
2. Pengantar Machine Learning	3
2.1 Apa itu Machine Learning?	4
2.2 Kategori Metode-Metode Machine Learning	6
2.2.1 Unsupervised Learning & Supervised Learning	7
2.2.2 Kategorikal dan Numerikal	8
2.2.3 Klasterisasi, Klasifikasi, Regresi, dan Asosiasi.....	8
2.3 Pengantar Algoritma dan Pemrograman.....	9
2.3.1 Apa itu Algoritma?	9
2.3.2 Apa itu Pemrograman Komputer?.....	10
2.3.3 Struktur Dasar Algoritma	11
2.4 Data, Informasi, & Pengetahuan	13
2.5 Struktur dan Tipe Data.....	13
2.6 Object, Variable, Parameter, Method, & Event.....	14
2.7 Soal Latihan Pengantar Machine Learning.....	16
3. Pra Pengolahan Data.....	17
3.1 Missing Value Replacement.....	18
3.2 Data Type Transformation	19
3.2.1 Encoding	19
3.2.2 Data Discretization	20
3.3 Aggregation.....	22
3.4 Smoothing Noisy Data.....	23
3.4.1 Anomaly Detection.....	23
3.4.2 Data Normalization.....	27

3.5	Feature Selection & Feature Extraction	29
3.5.1	Dimensionality Reduction	30
3.5.2	Feature Selection.....	33
3.5.3	Attribute Weighting	36
3.5.4	Feature Extraction	38
3.6	Unbalanced Class Reduction.....	39
3.7	Data Validation	43
3.8	Soal Latihan Pra Pengolahan Data	46
4.	Evaluasi Model	47
4.1	Apa itu Evaluasi Model?.....	48
4.2	Kompleksitas Algoritma.....	48
4.3	Evaluasi Model Klasifikasi	50
4.4	Interval Kepercayaan Akurasi	51
4.5	Evaluasi Model Regresi	51
4.6	Evaluasi Model Klasterisasi	53
4.6.1	Evaluasi Internal	53
4.6.2	Evaluasi Eksternal.....	54
4.6.3	Evaluasi Manual & Evaluasi Aplikasi.....	56
4.7	Evaluasi Model Asosiasi	56
4.8	Uji Korelasi Variabel.....	57
4.9	Soal Latihan Evaluasi Model	60
5.	Fuzzy Logic	61
5.1	Karakteristik Fuzzy Logic.....	62
5.2	Fuzzification	64
5.2.1	Triangular Membership Function	64
5.2.2	S & Z Shaped Membership Function.....	65
5.2.3	PI Membership Function	67
5.2.4	Trapezoidal Membership Function	68
5.2.5	Sigmoidal Membership Function.....	69
5.2.6	Gaussian Membership Function	71
5.2.7	Generalized Bell-Shaped Membership Function	72
5.2.8	Translate Parameters Between Membership Function	73

5.3	Knowledge Base.....	73
5.4	Machine Inference.....	74
5.4.1	Operasi Irisan (Intersection).....	74
5.4.2	Operasi Gabungan (Union)	75
5.4.3	Operasi Komplemen (Complement).....	75
5.5	Defuzzification.....	75
5.6	Penerapan Fuzzy Logic	76
5.6.1	Penerapan Fuzzy Logic Tsukamoto.....	76
5.6.2	Penerapan Fuzzy Logic Mamdani	78
5.6.3	Penerapan Fuzzy Logic Sugeno	84
5.7	Soal Latihan Fuzzy Logic.....	87
6.	ANN, SVM, & Fuzzy	89
6.1	ANN - Backpropagation.....	90
6.2	Adaptive Neuro Fuzzy Inference System.....	102
6.3	Support Vector Machine	106
6.4	Multi-Class SVM	115
6.5	Fuzzy SVM	119
6.6	Soal Latihan ANN, SVM, & Fuzzy.....	120
7.	K-Means & Fuzzy C-Means	121
7.1	K-Means	122
7.2	Fuzzy C-Means	129
7.3	Soal Latihan K-Means & FCM.....	132
8.	Naïve Bayes, k-NN, & Fuzzy	133
8.1	Naïve Bayes	134
8.2	Gaussian Naïve Bayes	136
8.3	Absolute Correlation Weighted Naïve Bayes	140
8.4	k-Nearest Neighbor	143
8.5	Weighted k-NN	146
8.6	Fuzzy k-NN.....	147
8.7	Fuzzy k-NN in every class.....	149
8.8	KFACWNB-NN	151
8.9	Soal Latihan Naïve Bayes, k-NN, & Fuzzy.....	152

9.	Bonus: C4.5, Linear Regression, & A-Priori.....	153
9.1	Decision Tree (C4.5).....	154
9.2	Linear Regression	166
9.2.1	Linear Regression dengan 1 Variabel Bebas.....	166
9.2.2	Linear Regression dengan 2 Variabel Bebas.....	167
9.2.3	Linear Regression dengan 3 atau Lebih Variabel Bebas	170
9.3	A-Priori	175
9.4	Soal Latihan C4.5, Linear Regression, & A-Priori.....	180
	Daftar Pustaka	181
	Glosarium.....	185
	Daftar Indeks.....	191
	Lampiran.....	197
	Biografi Penulis.....	215

Daftar Gambar

Gambar 2.1	Data Science & Machine Learning Model	5
Gambar 2.2	Metode-Metode Machine Learning.....	6
Gambar 3.1	Feature Selection & Feature Extraction Model	29
Gambar 3.2	Genetic Algorithm Model.....	35
Gambar 3.3	Bagging Model.....	39
Gambar 3.4	Boosting Model.....	39
Gambar 3.5	Stacking Model	39
Gambar 3.6	Weighted Vote Model	42
Gambar 3.7	Holdout Validation	43
Gambar 3.8	Leave-One-Out Cross Validation.....	44
Gambar 3.9	K-Fold Cross Validation	44
Gambar 5.1	Struktur Fuzzy Logic.....	63
Gambar 5.2	Grafik Kurva Trimf	64
Gambar 5.3	Grafik Kurva smf	65
Gambar 5.4	Grafik Kurva zmf.....	66
Gambar 5.5	Grafik Kurva pimf.....	67
Gambar 5.6	Grafik Kurva trapmf.....	68
Gambar 5.7	Grafik Kurva sigmf	69
Gambar 5.8	Grafik Kurva dsigmf	70
Gambar 5.9	Grafik Kurva gaussmf	71
Gambar 5.10	Grafik Kurva gbellmf.....	72
Gambar 5.11	Grafik Kurva mf2mf.....	73
Gambar 6.1	Arsitektur Jaringan ANN	90
Gambar 6.2	Struktur ANFIS	103
Gambar 6.3	Arsitektur Jaringan ANFIS	103
Gambar 6.4	Cara Kerja Hyperplane SVM.....	109
Gambar 9.1	Binary Splitting pada Atribut Binominal/Biner	154
Gambar 9.2	Binary dan Multi Splitting pada Atribut Nominal	154
Gambar 9.3	Bianry dan Multi Splitting pada Atribut Numerik	154

Daftar Tabel

Tabel 2.1 Kategori Metode-Metode Machine Learning yang Digunakan	6
Tabel 2.2 Tipe Data	14
Tabel 4.1 Confusion Matrix.....	50
Tabel 4.2 Interval Kepercayaan $Z_{\alpha/2}$ dalam Distribusi Normal	51
Tabel 4.3 Guiford Empirical Rules untuk Tafsiran Koefisien Korelasi	58

Daftar Contoh

Contoh 2.1	Data	13
Contoh 2.2	Informasi.....	13
Contoh 2.3	Function suatu Algoritma dan Obyek.....	15
Contoh 2.4	Object, Field, Property, Method, & Event.....	15
Contoh 2.5	Field/Variable.....	16
Contoh 2.6	Function dan Parameter.....	16
Contoh 3.1	Missing Value Replacement: Mean/Mode (Manual)	18
Contoh 3.2	Data Type Transformation: Encoding (Manual)	19
Contoh 3.3	Data Discretization: Binning (Manual)	21
Contoh 3.4	Data Discretization: Entropy (Manual)	22
Contoh 3.5	Agregation: Sum (Manual).....	23
Contoh 3.6	Anomaly Detection: k-NN (Manual)	24
Contoh 3.7	Anomaly Detection: k-NN (Matlab)	24
Contoh 3.8	Data Normalization (Manual).....	28
Contoh 3.9	Data Normalization: Min-Max & Z-Score (Rapidminer)	28
Contoh 3.10	Dimensionality Reduction: PCA & SVD (Rapidminer).....	32
Contoh 3.11	Feature Selection: Forward, Backward, & GA (Rapidminer)	35
Contoh 3.12	Attribute Weighting: Absolute Correlation Coefficient (Manual) ..	38
Contoh 3.13	Unbalanced Class: AdaBoost & Bagging (Rapidmider)	40
Contoh 3.14	Weighted Vote untuk Klasifikasi (Manual)	42
Contoh 3.15	K-Fold Cross Validation (Matlab).....	45
Contoh 4.1	Kompleksitas Algoritma.....	49
Contoh 4.2	Confusion Matrix (Manual).....	50
Contoh 4.3	Tingkat Kepercayaan Akurasi.....	51
Contoh 4.4	Error Estimation (Manual).....	52
Contoh 4.5	Lift Ratio (Manual)	57
Contoh 5.1	Trimf (Matlab & Manual).....	64
Contoh 5.2	Smf (Matlab & Manual)	65
Contoh 5.3	Zmf (Matlab & Manual)	66
Contoh 5.4	Pimf (Matlab & Manual)	67

Contoh 5.5	Trapmf (Matlab & Manual).....	68
Contoh 5.6	Sigmf (Matlab & Manual).....	69
Contoh 5.7	Dsigmf (Matlab & Manual)	70
Contoh 5.8	Gaussmf (Matlab & Manual)	71
Contoh 5.9	Gbellmf (Matlab & Manual)	72
Contoh 5.10	Mf2mf (Matlab).....	73
Contoh 5.11	Machine Inference Fuzzy: Operasi MIN (Manual)	74
Contoh 5.12	Machine Inference Fuzzy: Operasi MAX (Manual).....	75
Contoh 5.13	Machine Inference Fuzzy: Operasi NOT (Manual).....	75
Contoh 5.14	Fuzzy Logic Tsukamoto (Manual).....	76
Contoh 5.15	Fuzzy Logic Mamdani (Manual & Matlab).....	78
Contoh 5.16	Fuzzy Logic Mamdani: Traffic Light Control (Matlab)	81
Contoh 5.17	Fuzzy Logic Sugeno (Manual & Matlab)	84
Contoh 6.1	ANN Backpropagation (Manual) – 1	94
Contoh 6.2	ANN Backpropagation (Manual) – 2	97
Contoh 6.3	ANN Backpropagation: Klasifikasi (Matlab)	100
Contoh 6.4	ANN Backpropagation: Regresi (Rapidminer)	102
Contoh 6.5	ANFIS (Matlab).....	103
Contoh 6.6	SVM: Klasifikasi Biner (Manual)	111
Contoh 6.7	SVM: Klasifikasi (Matlab).....	113
Contoh 6.8	SVM: Regresi (Rapidminer)	114
Contoh 6.9	1V1 SVM: Multi Classification (Matlab).....	117
Contoh 6.10	LibSVM: Multi Classification (Rapidminer)	118
Contoh 7.1	K-Means: Klasterisasi (Manual).....	124
Contoh 7.2	K-Means: Klasterisasi (Matlab)	128
Contoh 7.3	FCM: Klasterisasi (Matlab).....	130
Contoh 8.1	Naïve Bayes: Klasifikasi (Manual).....	135
Contoh 8.2	Gaussian Naïve Bayes: Klasifikasi (Manual).....	136
Contoh 8.3	Kernel Naïve Bayes: Klasifikasi (Matlab)	138
Contoh 8.4	AC W-NB: Klasifikasi (Manual).....	141
Contoh 8.5	k-NN: Klasifikasi (Manual)	144
Contoh 8.6	k-NN: Klasifikasi (Matlab)	144

Contoh 8.7	Weighted k-NN: Klasifikasi (Manual)	146
Contoh 8.8	FkNN: Klasifikasi (Manual)	148
Contoh 8.9	FkNNC: Klasifikasi (Manual)	150
Contoh 9.1	Diskretisasi Binning pada C4.5.....	156
Contoh 9.2	Diskretisasi Entropy pada C4.5.....	157
Contoh 9.3	C4.5: Klasifikasi (Manual)	158
Contoh 9.4	C4.5: Klasifikasi (Matlab)	164
Contoh 9.5	Linear Regression: Estimasi dengan 1 Variabel Bebas (Manual) .	166
Contoh 9.6	Linear Regression: Estimasi dengan 2 Variabel Bebas (Manual) .	168
Contoh 9.7	Linear Regression: Estimasi dengan 3 Variabel Bebas (Manual) .	172
Contoh 9.8	Linear Regression: Estimasi (Rapidminer).....	174
Contoh 9.9	A-Priori: Asosiasi (Manual).....	176
Contoh 9.10	FP-Growth: Asosiasi (Rapidminer).....	179

Daftar Lampiran

Lampiran 1. Dataset: Heart Disease – Cleveland	197
Lampiran 2. Dataset: Tinggi Berat Badan.....	203
Lampiran 3. Dataset: Traffic Light	206
Lampiran 4. Dataset: Pangan (Time Series).....	207
Lampiran 5. Dataset: Transaksi Penjualan Obat.....	210

1. Mengapa Machine Learning & Reasoning Fuzzy Logic?

Program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik, tidak adaptif terhadap informasi yang baru, tidak mampu melakukan pembelajaran, dan menalar dalam pengambilan keputusan. Sebaliknya, *Artificial Intelligence* (AI) memungkinkan suatu sistem komputer bersifat adaptif terhadap informasi yang baru, mampu melakukan pembelajaran, memiliki pengetahuan, dan mampu menalar. AI terus berkembang sejak tahun 1956, melambat pada periode 1966 hingga 1974, namun sejak tahun 1980 AI menjadi sebuah industri yang besar dengan perkembangan yang sangat pesat [1]. Pesatnya perkembangan Teknologi Informasi dan Komunikasi (TIK) menuntut peningkatan pengembangan metode-metode AI pula.

Saat ini, metode-metode AI telah berkembang semakin cerdas. Pada prinsipnya, metode-metode AI atau biasa diistilahkan dengan *soft computing* dapat dikategorikan menjadi metode-metode *searching*, *reasoning*, dan *learning*. Metode-metode *searching* merepresentasikan masalah ke dalam *state* dan ruang masalah lalu menggunakan strategi pencarian untuk menemukan solusi. Sedangkan metode-metode *reasoning* merepresentasikan masalah ke dalam basis pengetahuan lalu menggunakan strategi penalaran untuk menemukan solusi. Pendekatan *searching* dan *reasoning* mengharuskan adanya aturan-aturan yang berlaku, namun terkadang aturan-aturan tidak selalu bisa didefinisikan secara benar dan lengkap, maka pendekatan *learning* hadir untuk mengatasi kendala tersebut yang diistilahkan dengan *machine learning*. Dengan begitu, suatu sistem yang dibangun dengan pendekatan *machine learning*, dapat menjadi lebih cerdas dari sebelumnya.

Jika pendekatan *searching* kesulitan dalam menentukan apakah aturan-aturan sudah tepat dan lengkap karena masalah yang dihadapi cukup kompleks sehingga representasi masalah ke dalam *state* menjadi tidak efisien, maka pendekatan *reasoning* dengan representasi *logic* (bahasa formal) merupakan solusinya. Awalnya metode-metode *reasoning* digunakan pada masalah yang memiliki kepastian, bagaimana jika masalah mengandung ketidakpastian? Pendekatan seperti teori probabilitas dan *Fuzzy Logic* merupakan solusinya. Metode-metode dengan pendekatan probabilitas untuk masalah yang mengandung ketidakpastian bersifat peluang. Bagaimana jika masalah mengandung ketidakpastian yang besifat samar? *Fuzzy Logic* merupakan solusinya.

Fuzzy Logic mampu memodelkan fungsi-fungsi nonlinier, mampu mengatasi masalah yang sangat kompleks, didasarkan pada bahasa formal/alami, memiliki toleransi terhadap data yang tidak tepat, dan mampu merepresentasikan pengetahuan pakar ke dalam basis pengetahuannya sebagai aturan-aturan yang berlaku sehingga tidak memerlukan proses *learning*. Namun seperti yang telah dikatakan sebelumnya, terkadang aturan-aturan tidak selalu bisa didefinisikan secara tepat dan lengkap, sehingga metode-metode *machine learning* menjadi solusi untuk itu yang juga mampu memodelkan fungsi-fungsi nonlinier, mampu mengatasi masalah yang sangat kompleks, dan memiliki toleransi terhadap data yang tidak tepat.

Kata lainnya, *Fuzzy Logic* tidak mampu melakukan pembelajaran, sementara pendekatan *machine learning* mampu melakukannya, namun tidak mampu melakukan penalaran seperti *Fuzzy Logic*. Melalui pembelajarannya, pendekatan *machine learning* mampu menggali pengetahuan, memprediksi, maupun mengenal pola dari masalah kompleks yang dihadapi, sehingga dapat memodelkan aturan-aturan. Dengan demikian, bagaimana jadinya jika pendekatan *reasoning Fuzzy Logic* diintegrasikan dengan pendekatan *machine learning*? Bagaimana jika pendekatan *machine learning* memodelkan aturan-aturan yang sulit didefinisikan melalui pembelajarannya untuk digunakan *Fuzzy Logic*? Atau bagaimana jika pendekatan *reasoning Fuzzy Logic* mengoptimalkan proses prediksi, pengenalan pola, dan pemodelan pengetahuan yang dilakukan pendekatan *machine learning*? Buku ini merupakan salah satu jawabannya.

Bagaimana jika metode-metode *machine learning* diintegrasikan dengan metode *Fuzzy Logic*? Buku ini membahas karakteristik, algoritma, manual, dan penerapan (menggunakan *tools Matlab* dan *Rapidminer*) beberapa metode-metode *machine learning* dan pengembangannya dengan *reasoning Fuzzy Logic*, meliputi:

1. Mengapa Machine Learning & Reasoning Fuzzy Logic?
2. Pengantar Machine Learning
3. Pra Pengolahan Data
4. Evaluasi Model
5. Fuzzy Logic
6. ANN, SVM, & Fuzzy
7. K-Means & FCM
8. Naïve Bayes, k-NN & Fuzzy
9. Bonus: C4.5, Linear Regression, & A-Priori

2. Pengantar Machine Learning

No.	Materi	Tujuan Pembelajaran
1.	Apa itu <i>Machine Learning</i> ?	Anda mampu memahami dan menjelaskan pengertian <i>machine learning</i> dan hubungannya dengan <i>data science</i> .
2.	Kategori Metode-Metode <i>Machine Learning</i>	Anda mampu memahami, menjelaskan, dan membedakan metode-metode <i>machine learning</i> kategori <i>unsupervised learning</i> dan <i>supervised learning</i> ; kategorikal dan numerikal; serta klasterisasi, klasifikasi, regresi/estimasi, dan asosiasi.
3.	Pengantar Algoritma dan Pemrograman	Anda mampu memahami dan menjelaskan pengertian algoritma, pemrograman komputer, struktur dasar algoritma.
4.	Data, Informasi, dan Pengetahuan	Anda mampu memahami, menjelaskan, dan membedakan pengertian data, informasi, dan pengetahuan.
5.	Struktur dan Tipe Data	Anda mampu memahami dan menjelaskan pengertian struktur data dan tipe data.
6.	<i>Object, Variable, Method, dan Event</i>	Anda mampu memahami, menjelaskan, dan membedakan pengertian <i>type/object, variable (field, property, parameter, attribute, atau feature), method (function/procedure)</i> , dan <i>event</i> dalam pemrograman komputer dan <i>data science</i> .

2.1 Apa itu Machine Learning?

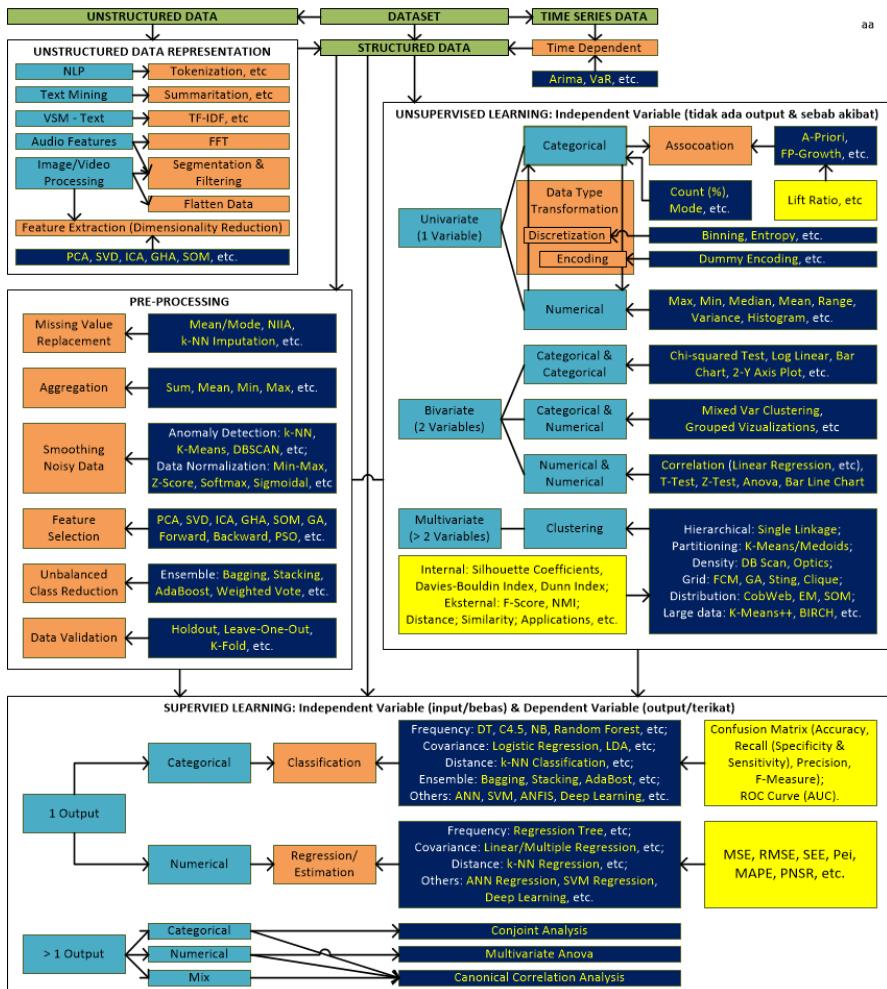
Pada awalnya, komputer hanya digunakan sebagai alat menghitung. Pemrograman komputer belum dapat melakukan pembelajaran dan penalaran seperti saat ini. Pengembangan metode-metode komputasi saat itu hanya sebatas bagaimana mengembangkan komputasi secepat mungkin atau dengan kompleksitas komputasi yang seminim mungkin. Namun sejak munculnya AI dan seiring dengan perkembangannya, membuat komputer yang bisa melakukan pembelajaran mulai terpikirkan. Sejak tahun 1960-an, konsep untuk membuat suatu sistem komputer bisa belajar mulai dikembangkan [2].

Machine learning merupakan salah satu sub disiplin ilmu dari AI. Metode-metode AI dapat diistilahkan *soft computing*, *machine learning* merupakan salah satu bagian dari itu. Tom M. Mitchell menyatakan bahwa suatu program komputer dikatakan belajar dari pengalaman E yang berhubungan dengan beberapa tugas T dan ukuran performansi P , jika performansinya pada tugas-tugas T , sebagaimana diukur menggunakan P , meningkat dengan pengalaman E [3]. Pernyataan tersebut berarti bahwa suatu program komputer secara otomatis bisa semakin cerdas melalui pembelajarannya terhadap pengalaman-pengalaman (masukan) yang diperolehnya.

Sejak tahun 1980-an, berbagai program komputer yang menggunakan pendekatan *machine learning* telah banyak diperkenalkan. Misalnya: *ALVINN* (*Autonomous Land Vehicle in Neural Network*), sebuah program untuk membuat kendaraan dapat berjalan secara otomatis (tanpa sopir) [4]; *ImageNet*, telah berhasil menghasilkan ratusan program komputer yang mampu mempelajari karakteristik jutaan citra dan mengklasifikasikannya ke dalam ribuan *class* [5]; *Dragon Speak*, program komputer untuk pengenalan ucapan manusia dengan akurasi yang tinggi [6]. Hingga saat ini, pendekatan *machine learning* dapat digunakan untuk menyelesaikan masalah-masalah kompleks di berbagai bidang yang tidak dapat diselesaikan dengan program konvensional, misalnya seperti deteksi penyakit di bidang kesehatan, program komputer untuk mengajar di bidang pendidikan, pesawat tempur tanpa awak di bidang militer, pengenalan tulisan, pencarian informasi, keamanan kendaraan, analisis pasar, dsb.

Machine learning sebagai salah satu bagian dari disiplin ilmu *soft computing* beririsan dengan berbagai disiplin ilmu lainnya, seperti *mathematics*, *statistic*, *programming*, *data science*, *big data*, *data mining*, *database*, *information retrieval*, *computer vision*, *robotic*, *game programming*, *IoT*, *expert system*, *decision support system*, *information system*, dan sebagainya. Salah satu disiplin ilmu yang populer di era *big data* saat ini, yaitu *data science*. Secara umum *data science* adalah penggalian atau mengekstrak atau analisis data untuk menemukan data yang benar sehingga menghasilkan suatu informasi atau pengetahuan yang akurat/tepat. *Data science* merupakan salah satu disiplin ilmu yang secara khusus mempelajari tentang data [7] menggunakan metode komputasi (algoritma). Oleh karena itu, peranan *machine learning* sangat penting dalam *data science*, karena berbagai jenis data baik yang terstruktur maupun tidak terstruktur dapat dianalisa dengan baik menggunakan pendekatan *machine learning*.

Profesi di bidang *data science* biasanya disebut *data analyst* atau *data scientist*. Menurut riset yang dilakukan oleh situs *LinkedIn*, *data scientist* merupakan salah satu profesi paling *hot* yang banyak dibutuhkan oleh dunia industri akhir-akhir ini. Seiring meningkatnya penggunaan teknologi secara signifikan, pengetahuan mengenai *data science* semakin dibutuhkan pula. *Data science* digunakan oleh perusahaan, instansi, dan berbagai industri untuk melakukan analisis data yang tidak bisa dilakukan dengan metode sederhana. Misalkan *marketplace* Bukalapak memerlukan suatu pengetahuan yang dapat melakukan analisis data penjualan dan pembeli di *platform* mereka, seperti *customer loyalty*, *market analysis*, dsb. Oleh karena itu, memiliki pengetahuan *machine learning*, merupakan salah satu syarat untuk menjadi seorang *data scientist*. Jason Okui, *Product Manager* Adobe, berkata: “*The biggest benefit of the program for me was that it gave me a deeper understanding of how predictive models work and how they can be applied. Just as important, it allowed me to communicate what a predictive analysis means* [8].”



Gambar 2.1 Data Science & Machine Learning Model

2.2 Kategori Metode-Metode Machine Learning

Berdasarkan outputnya, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok *unsupervised learning* dan *supervised learning*. Berdasarkan tipe data dari variabel input pada data yang diolah, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok kategorikal dan numerikal. Berdasarkan tujuannya, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok klasterisasi, klasifikasi, regresi/estimasi, dan asosiasi. Beberapa metode-metode *machine learning* ditunjukkan pada Gambar 2.2.

Artificial Neural Network	Support Vector Machine	Decision Tree	Function/Regression
Perceptron Backpropagation Probabilistic Neural Network Long-Short Term Memory Radial Basis Function Network Hopfield Network Recurrent Neural Network ... <i>Others ANN</i>	Support Vector Machine Support Vector Regression Support Vector Clustering LibSVM PSO-SVM Evolutionary SVM Fuzzy SVM ... <i>Others SVM</i>	ID3 C4.5 Random Forest Gradient Boosted Trees CHAID Random Tree CHART ... <i>Others DT</i>	Linear Regression Logistic Regression Stepwise Regression Ordinary Least Squares Reg. MARS LOESS ... <i>Others Regression</i>
Function	Instance-Based Learning	Natural Inspired	Association
Linear Regression Polynomial Regression Vector Linear Regression ... <i>Others Function</i>	K-Nearest Neighbor Fuzzy k-NN Learning Vector Quantization ... <i>Others IBL</i>	Evolutionary Algorithm Swarm Intelligence Bio-Inspired Computation ... <i>Others NIC</i>	A-Priory FP-Growth ... <i>Others Association</i>
Bayesian	Clustering	Ensemble	Deep Learning
Naïve Bayes Gaussian Naïve Bayes Kernel Naïve Bayes Bayesian Belief Network Augmented Naïve Bayes NB-Tree Selective Neighborhood NB Hidden Naïve Bayes ... <i>Others NB</i>	K-Means X-Means K-Medoids Fuzzy C-Means DBSCAN Self Organizing Map Agglomerative Clustering Hierarchical Clustering ... <i>Others Clustering</i>	Bagging Stacking Adaboost Bucket and Models Gradient Boosting Machine Likelihood-based Boosting Deep Forest Weighted Vote ... <i>Others Ensemble</i>	Convolutional Neural Network Deep Boltzman Machine Deep Belief Network Stacked Auto-Encoders ... <i>Others Deep Learning</i>

Gambar 2.2 Metode-Metode Machine Learning

Tentu saja tidak semua metode-metode yang tertuang pada Gambar 2.2 dibahas dalam buku ini. Hanya beberapa metode-metode populer saja (Tabel 2.1) yang dibahas dalam buku ini, termasuk pengembangannya dengan *Fuzzy Logic*, dan beberapa pengembangan yang dilakukan oleh penulis.

Tabel 2.1 Kategori Metode-Metode Machine Learning yang Digunakan

Algoritma	ANN	SVM	C4.5	k-NN	NB	LR	k-Means	FCM	A-Priori
Unsupervised							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Supervised	<input checked="" type="checkbox"/>								
Kategorikal			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Numerikal	<input checked="" type="checkbox"/>								
Klasterisasi							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Klasifikasi	<input checked="" type="checkbox"/>								
Regresi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>			
Asosiasi								<input checked="" type="checkbox"/>	

2.2.1 Unsupervised Learning & Supervised Learning

Berdasarkan outputnya, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok *unsupervised learning* dan *supervised learning*. Metode-metode *machine learning* yang masuk dalam kategori *unsupervised learning* melakukan pembelajaran tanpa pengawasan, atau biasa diistilahkan dengan pembelajaran tanpa guru. Metode ini berusaha mememukan hubungan antar variabel bebas [9], atau hubungan antar data pada suatu variabel, atau menghitung nilai statistik tertentu dari satu atau lebih variabel bebas. Dengan demikian, jenis metode pembelajaran ini tidak memiliki variabel output. Hubungan antar variabel bebas yang ditemukan tersebut diwakili dalam suatu struktur yang disebut sebagai model/pola/pengetahuan [9]. Pengetahuan tersebutlah merupakan hasil pembelajaran yang memberi tahu pola dari data yang ada yang kemudian dapat digunakan untuk pekerjaan laiinya/selanjutnya, misalnya untuk pekerjaan klasifikasi atau estimasi/regresi. Namun dalam hal ini, hubungan antar variabel bebas (misalnya korelasi) bukan menunjukkan hubungan sebab akibat. klaterisasi dan asosiasi masuk dalam kategori ini.

Apabila terdapat 1 variabel bebas (*univariate*) bertipe kategorikal, maka itu merupakan pekerjaan asosiasi atau dapat pula untuk mencari nilai *mode* maupun *count (%)* dari variabel tersebut. Namun apabila variabel tersebut bertipe numerik, maka biasanya itu untuk mencari nilai minimal, maksimal, *median*, *mean*, *variance*, dll. Apabila terdapat 2 variabel bebas (*bivariate*) bertipe kategorikal, maka biasanya itu untuk memperoleh nilai *Chi-squared Test*, *Log Linear*, atau menggambarkan *bar chart*, *2-Y axis plot*, dsb dari kedua variabel tersebut. Namun apabila kedua variabel tersebut bertipe kategorikal dan numerik, maka biasanya itu untuk *grouped visualization*, dsb. Sedangkan apabila kedua variabel tersebut bertipe numerik dan numerik, maka biasanya itu merupakan pekerjaan korelasi, atau untuk memperoleh nilai *T-Test*, *Z-Test*, *Anova*, atau untuk menggambarkan *bar line chart* dari kedua variabel tersebut. Apabila terdapat lebih dari 2 variabel bebas (*multivariate*), maka itu merupakan pekerjaan klasterisasi.

Sementara itu, metode-metode *machine learning* yang masuk dalam kategori *unsupervised learning* melakukan pembelajaran dengan pengawasan, atau biasa diistilahkan pembelajaran dengan guru. Metode ini berusaha mememukan hubungan antara inputan dengan output [9], yang mana output inilah yang disebut sebagai guru atau yang melakukan pengawasan terhadap pembelajaran yang dilakukan. Dengan demikian, jenis metode pembelajaran ini memiliki variabel output (bebas) dan variabel input (terikat), sehingga ada hubungan sebab akibat. Hubungan antara inputan dengan output yang ditemukan tersebut diwakili dalam suatu struktur yang disebut sebagai model/pola/pengetahuan [9]. Pengetahuan tersebutlah merupakan hasil pembelajaran yang kemudian digunakan untuk memprediksi output dari inputan yang belum diketahui outputnya (kasus baru). Klasifikasi dan regresi masuk dalam kategori ini.

Apabila hanya terdapat 1 variabel output (terikat) bertipe kategorikal, maka itu merupakan pekerjaan klasifikasi. Namun apabila variabel output (terikat) tersebut bertipe numerik, maka itu merupakan pekerjaan regresi/estimasi. Sedangkan apabila terdapat lebih dari 1 variabel output (terikat) bertipe kategorikal, maka dapat dilakukan analisis menggunakan pendekatan *Conjoint Analysis*.

Apabila 2 atau lebih variabel output (terikat) tersebut bertipe numerikal, maka dapat dilakukan analisis menggunakan pendekatan *Multivariate Anova*. Apabila 2 atau lebih variabel output (terikat) tersebut bertipe campuran (kategorikal dan numerikal), maka dapat dilakukan analisis menggunakan pendekatan *Canonical Correlation Analysis*.

2.2.2 Kategorikal dan Numerikal

Berdasarkan tipe data dari atribut-atribut input yang digunakan, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok kategorikal dan numerikal. Namun pada pengembangannya, metode-metode *machine learning* dapat mengolah inputan yang bertipe kategorikal maupun numerikal. Misalnya seperti *Naïve Bayes* (NB) yang bekerja pada inputan bertipe kategorikal, namun untuk mengolah inputan bertipe numerikal, pendekatan distribusi *Gaussian* dapat diterapkan pada *Naïve Bayes*. Sebaliknya begitupun C4.5 yang dapat melakukan diskretisasi pada inputan bertipe numerikal. Selain itu, pra pengolahan data dapat dilakukan agar metode *machine learning* yang digunakan dapat mengolah data tersebut, misalnya dengan melakukan diskretisasi untuk mentransformasi tipe data numerik ke kategori, atau sebaliknya melakukan *encoding* untuk mentransformasikan tipe data kategori ke numerik. Transformasi tipe data akan dibahas dalam sub pokok bahasan lain.

2.2.3 Klasterisasi, Klasifikasi, Regresi, dan Asosiasi

Berdasarkan tujuannya, metode-metode *machine learning* dapat dikategorikan ke dalam kelompok klasterisasi, klasifikasi, regresi/estimasi, dan asosiasi. Klasterisasi mengacu pada pengelompokan atau pengamatan pada data yang tidak memiliki variabel output [10]. Suatu klaster adalah kumpulan dari catatan/data yang mirip satu sama lainnya, dan berbeda dengan catatan dalam kelompok lainnya. Tugas klasterisasi tidak mencoba untuk mengklasifikasikan, memperkirakan, atau memprediksi output, namun mencari segmen seluruh data yang ditetapkan menjadi sub kelompok yang relatif homogen/klaster, yang mana kesamaan data dalam klaster dimaksimalkan dan kesamaan klaster di luar data diminimalkan [10].

Klasifikasi mirip dengan klasterisasi, keduanya merupakan pengelompokan, namun klasifikasi tidak berusaha mememukan hubungan antar inputan, melainkan hubungan antara inputan dengan output [9]. Output inilah yang disebut sebagai guru atau yang melakukan pengawasan terhadap pembelajaran yang dilakukan. Dengan demikian, jenis metode pembelajaran ini memiliki variabel input (bebas) dan variabel output (terikat), sehingga ada hubungan sebab akibat. Hasil pembelajaran klasifikasi merupakan suatu pengetahuan/pola yang digunakan untuk memprediksi label-label output yang bertipe kategorikal dari inputan yang belum diketahui label outputnya (memprediksi kasus baru) [9]. Dengan demikian klasifikasi mengolah data dengan variabel output yang bertipe kategorikal (diskrit) [10]. Contohnya apabila variabel output adalah pendapatan, maka bisa digunakan label output pendapatan besar, menengah, atau kecil.

Regresi atau estimasi mirip dengan klasifikasi. Bedanya, atribut output pada regresi bertipe kontinyu [10], itulah mengapa disebut estimasi. Sedangkan asosiasi

berusaha menemukan inputan yang muncul bersamaan dalam suatu transaksi. Asosiasi hanya memiliki 1 variabel input. Dalam dunia bisnis, sering disebut dengan *affinity analysis* atau *market basket analysis* [10]. Asosiasi berusaha mencari aturan hubungan antar data pada 1 variabel input/bebas. Berangkat dari pola “If *antecedent*, then *consequent*,” bersamaan dengan pengukuran *Support (Coverage)* dan *Confidence* yang terasosiasi dalam aturan [10].

2.3 Pengantar Algoritma dan Pemrograman

Suatu program komputer disusun dari serangkaian metode komputasi (algoritma). Sedangkan kegiatan mengembangkan program komputer dengan menggunakan bahasa pemrograman tertentu disebut pemrograman. Dengan demikian, program komputer atau aplikasi merupakan produk dari kegiatan pemrograman menggunakan serangkaian algoritma dengan alat bantu bahasa pemrograman, atau dengan notasi bahasa pemrograman tertentu.

2.3.1 Apa itu Algoritma?

Metode komputasi atau algoritma merupakan jantung dari ilmu komputer. Ditinjau dari asal usul katanya, awalnya orang hanya menemukan kata “*Algorism*” yang berarti proses menghitung dengan angka arab [11]. Para ahli sejarah matematika akhirnya menemukan asal kata tersebut berasal dari nama penulis buku arab yang terkenal, yaitu Abu Ja’far Muhammad Ibnu Musa Al-Khuwarizmi.

Al-Khuwarizmi (oleh orang barat menjadi *Algorism*) menulis buku yang berjudul *Kitab Al Jabar Wal-Muqabala* yang artinya “Buku pemugaran dan pengurangan”. Karena *Algorism* sering dihubungkan dengan *arithmetic*, sehingga akhirnya *-sm* berubah menjadi *-thm* [12]. Akhirnya lambat laun kata *algorithm* berangsur-angsur digunakan sebagai metode perhitungan (komputasi). Dalam Bahasa Indonesia, kata *algorithm* diserap menjadi algoritma.

“Algoritma adalah serangkaian instruksi yang telah didefinisikan dengan baik untuk menghitung, yang dimulai dari suatu kondisi awal dan inputan awal, yang mana instruksi-instruksi tersebut menjelaskan suatu komputasi yang bila dieksekusi dan diproses melewati sejumlah urutan kondisi terbatas yang terdefenisi dengan baik, yang pada akhirnya menghasilkan suatu keluaran dan akan berhenti di kondisi akhir [11].” Pendapat lainnya menyatakan bahwa, “Algoritma adalah prosedur komputasi yang mengambil suatu atau beberapa nilai sebagai inputan, kemudian memprosesnya, hingga menghasilkan suatu atau beberapa nilai sebagai output [13].” Dengan demikian, dalam bidang matematika dan informatika, algoritma adalah serangkaian prosedur yang saling berinteraksi untuk mencapai suatu tujuan dalam memecahkan masalah tertentu. Sehingga algoritma dapat digunakan untuk penghitungan, pemrosesan data, pencarian, penalaran, optimasi, pembelajaran, dan sejenisnya untuk menyelesaikan suatu masalah.

Agar dapat dilaksanakan oleh komputer, maka algoritma harus ditulis dalam notasi bahasa pemrograman untuk menjadi program komputer. Jadi program komputer adalah perwujudan atau implementasi teknis algoritma yang ditulis dalam

bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer. Umumnya, algoritma dapat ditulis dengan notasi [14]:

1. Bahasa alamiah;
2. *Pseudocode*;
3. *Flowchart*;
4. Bagan drakon; atau pun
5. Bahasa pemrograman.

Ekspresi bahasa alamiah terhadap algoritma cenderung sangat banyak dan rancu, sehingga jarang digunakan untuk algoritma yang kompleks. *Pseudocode* dan *flowchart* merupakan cara terstruktur, mudah, dan umum untuk menggambarkan algoritma. Sedangkan bahasa pemrograman ditujukan untuk mengekspresikan algoritma dalam sebuah bentuk yang dapat dieksekusi oleh komputer.

Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman untuk diproses oleh komputer. Meski pun setiap komputer berbeda teknologinya, tetapi secara umum semua komputer dapat melakukan operasi-operasi dasar dalam pemrograman, misalnya operasi pembacaan data, perbandingan, aritmatika, dsb. Sampai saat ini, perkembangan teknologi komputer tidak mengubah operasi-operasi dasarnya, yang berubah hanyalah kecepatan, fasilitas, biaya, atau pun tingkat ketelitianya. Pada sisi lain setiap program dalam bahasa tingkat tinggi selalu diterjemahkan kedalam bahasa mesin sebelum akhirnya dikerjakan oleh komputer. Setiap instruksi dalam bahasa mesin menyajikan operasi dasar yang sesuai, dan menghasilkan efek netto yang sama pada setiap komputer. Analoginya sama dengan resep membuat kue. Sebuah resep kue dapat ditulis dalam bahasa apapun. Apapun bahasanya, kue yang dihasilkan tetap sama asalkan semua aturan pada resep diikuti oleh juru masak (sebagai pemroses) yang dapat melakukan operasi dasar yang sama, seperti mengocok telur, menimbang berat gula, dsb.

2.3.2 Apa itu Pemrograman Komputer?

Pemrograman komputer merupakan kegiatan mengembangkan program komputer. Sedangkan program komputer merupakan kumpulan instruksi atau perintah atau pernyataan (algoritma) menggunakan suatu bahasa pemrograman komputer yang disusun sedemikian rupa sehingga tepat, benar, akurat dalam menyelesaikan suatu permasalahan. Sementara itu, bahasa pemrograman adalah suatu jenis aplikasi komputer yang digunakan sebagai alat bantu dalam kegiatan membuat program komputer. Jadi, bahasa pemrograman merupakan alat bantu dalam pemrograman.

Profesi dalam pemrograman komputer disebut sebagai *programmer*. Jadi, seorang *programmer* memerlukan keterampilan dalam algoritma, logika, pemrograman, bahasa pemrograman, dan pengetahuan-pengetahuan lain seperti matematika atau pun pengetahuan yang berhubungan dengan masalah yang dihadapi. Jika dihubungkan dengan profesi *data scientist*, maka seorang *data scientist* yang handal memiliki kemampuan sebagai *programmer* pula, begitupun sebaliknya.

Pemrograman merupakan sebuah seni dalam menggunakan satu atau lebih algoritma yang saling berhubungan dengan menggunakan sebuah bahasa pemrograman tertentu sehingga menjadi sebuah program komputer. Bahasa pemrograman yang berbeda mendukung gaya pemrograman yang berbeda pula. Gaya pemrograman ini biasa disebut paradigma pemrograman.

Yang perlu diperhatikan adalah belajar pemrograman tidaklah sama dengan belajar bahasa pemrograman atau *tools* Bahasa pemrograman. Belajar pemrograman berarti mempelajari metodologi pemecahan masalah, kemudian menuliskan algoritma pemecahan masalah dalam notasi tertentu, dan belajar prinsip-prinsip instruksi-instruksi pemrograman. Sedangkan belajar bahasa pemrograman berarti belajar menggunakan suatu bahasa komputer, aturan tata bahasanya, instruksi-instruksinya, pengoperasiannya, *compiler*-nya, dan memanfaatkan instruksi-instruksi tersebut untuk membuat program yang ditulis hanya dalam bahasa pemrograman itu saja.

2.3.3 Struktur Dasar Algoritma

Suatu algoritma dapat dibangun dari tiga instruksi dasar, antara lain [14]:

1. *Sequence* (runtunan), setiap *statement* dikerjakan secara beruntun sesuai dengan urutannya.
2. *Selection* (seleksi), *statement* yang dikerjakan adalah *statement* yang memenuhi kondisi tertentu (yang terpilih).
3. *Loop* (pengulangan), *statement* yang dikerjakan secara berulang-ulang hingga batas tertentu.

Instruksi *sequence* merupakan runtunan pernyataan yang dikerjakan secara beruntun sesuai dengan urutannya.

Syntax Sequence:

Statement1;	1
Statement2;	1
StatementN;	1

Instruksi *selection* merupakan logika keputusan dalam suatu algoritma, dengan maksud melaksanakan satu atau lebih *statement* berdasarkan keputusan yang benar (memenuhi). Instruksi ini menguji beberapa kondisi dan melaksanakan satu atau lebih pernyataan tergantung dari hasil pengujian, yaitu bernilai benar atau salah. Terdapat aksi yang berbeda antara kondisi yang bernilai benar dan kondisi yang bernilai salah. Perlu diperhatikan bahwa konstruksi ini tidak akan menguji kondisi yang lain jika sebuah kondisi sudah memenuhi, sebab untuk apa membuat kondisi dengan maksud yang sama.

Syntax If – Then:

If Condition A	1
Statements	2
End	2

Syntax If – Then – Else:

If Condition A	1
Statements	2
Else	2
Statements	3
End	3

Syntax If – Then – Else If – Then – Else:

If Condition A	1
Statements	2
ElseIf Condition B	3
Statements	4
ElseIf Condition C	5
Statements	6
Else	6
Statements	7
End	7

Syntax Switch Case:

Switch Condition	1
Case Condition A	2
Statements	3
Case Condition B	3
Statements	4
Case Condition C	4
Statements	5
Case Else	5
Statements	6
End	6

Instruksi perulangan atau *loop* merupakan logika perulangan dalam suatu algoritma, dengan maksud melaksanakan satu atau lebih *statement* secara berulang-ulang hingga batas tertentu. Struktur ini melaksanakan satu atau lebih pernyataan yang berada pada sebuah blok *loop* program dan mengulanginya hingga suatu kondisi (*expression*) bernilai benar, atau suatu kondisi bernilai salah, atau hingga waktu tertentu yang ditentukan, atau untuk setiap sekelompok data.

Syntax Repeat – Until:

Repeat	1
Statements	2
Until Condition	2

Syntax While:

While Condition	1
Statements	2
End	2

Syntax For:

For Counter = Lower To Upper	1
Statements	2
End	2

2.4 Data, Informasi, & Pengetahuan

Data merupakan entitas yang tidak memiliki arti, meskipun kemungkinan memiliki nilai di dalamnya. Tentunya data perlu disimpan, namun yang lebih penting dari itu adalah bagaimana menggali atau menemukan pengetahuan dari data yang disimpan. Data merupakan kumpulan/rekaman/catatan dari fakta, transaksi, atau objek tentang suatu kejadian yang tidak membawa arti. Dapat pula merupakan suatu catatan terstruktur dari suatu transaksi. Sedangkan informasi merupakan data yang telah diolah sehingga memiliki nilai. Dengan demikian, data merupakan materi penting dalam membentuk informasi.

Sedangkan pengetahuan merupakan gabungan dari informasi yang bertujuan untuk memberikan suatu informasi yang baru. Pengetahuan dapat berupa solusi pemecahan suatu masalah atau petunjuk suatu pekerjaan, yang mana nilainya bisa ditingkatkan, dipelajari, dan diajarkan kepada orang lain. Thomas H. Davenport & Laurence Prusak menyatakan bahwa pengetahuan merupakan gabungan suatu pengalaman, nilai, informasi dan pandangan pakar yang memberikan suatu *framework* untuk mengevaluasi dan menciptakan pengalaman dan informasi baru.

Contoh 2.1 Data

NIDN	Tgl	Datang	Pulang
0914107901	2/12/2004	7:20	15:40
0908048403	2/12/2004	7:45	15:33
...
0924048601	2/12/2004	8:00	16:15

Contoh 2.2 Informasi

NIDN	Masuk	Alpa	Ijin	Telat	Sakit
0914107901	30	0	0	7	0
0908048403	27	1	1	1	1
...
0924048601	20	3	7	2	0

Dari data dan informasi yang ditunjukkan pada tabel-tabel tersebut, maka dapat ditarik suatu pengetahuan menggunakan metode tertentu, tentang kebiasaan jam datang dan pulang dosen. Pengetahuan tersebut tentunya dapat mendukung pengambilan keputusan atau kebijakan tentang bagaimana teknik/metode agar dapat meningkatkan kehadiran dosen.

Suatu data dapat dikatakan suatu objek. Sementara suatu objek dapat memiliki satu atau lebih *variable* (*parameter, attribute, feature, field*, atau *property*), *record*, dan *method* (*function/procedure*). Pembahasan lebih dalam mengenai materi ini akan dibahas pada sub pokok yang lain.

2.5 Struktur dan Tipe Data

Algoritma dan struktur data tidak bisa terpisahkan. Algoritma untuk memanipulasi data, sementara wadah dari data tersebut merupakan struktur data untuk mengatur/mengolah data tersebut di dalam wadahnya. Dalam istilah informatika, struktur data adalah cara penyimpanan, penyusunan dan pengaturan

data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien [13], [15]. Dengan sifatnya tersebut, maka suatu struktur data dapat diterapkan untuk pengolahan data, pengolahan *database*, pengolahan teks, *spreadsheet*, pengolahan citra, pengolahan suara, atau pun juga pemanfaatan berkas dengan teknik tertentu yang memanfaatkan struktur data, dsb.

Data *type* (tipe data) dapat dikatakan jenis dari suatu data. Jadi umumnya, suatu data dapat ditetapkan atau didefinisikan tipenya [14]. Tipe data dapat diklasifikasikan dalam dua kategori besar, yaitu:

1. Tipe data terstruktur, dapat mendefinisikan suatu data dengan jelas.
 2. Tipe data tidak terstruktur, belum dapat mendefinisikan suatu data dengan jelas.
- Contoh: data *image*, *video*, *tex*, *voice*, *spasial*, dan sejenisnya. Data yang bertipe tidak terstruktur harus diolah lebih dahulu menjadi terstruktur agar data tersebut dapat diolah dalam *data science*, pemrograman, *database*, dan sejenisnya. Pembahasan lebih dalam mengenai materi ini, akan di bahas pada pokok bahasan pra pengolahan data.

Walaupun berbagai tipe data terstruktur antara *tools* pemrograman (seperti Visual Studio dan Java Netbeans), *data science* (seperti Phyton, Matlab, dan Rapidminer), maupun *database* (seperti SQL Server dan My SQL) berbeda-beda, namun pada prinsipnya sama. Secara umum, tipe data standar ditunjukkan pada Tabel 2.2.

Tabel 2.2 Tipe Data

Tipe		Contoh	
Kualitatif (Discrete/Categorical)	Nominal	Char, String, Varchar, Boolean,	[pria, wanita]
	Ordinal	Date, Time, dsb.	[dingin, normal, panas]
Kuantitatif (Continue/Numerical)	Interval	Integer, Real, Double, Float,	suhu (dalam celcius)
	Ratio	Numeric, dsb.	umur

2.6 Object, Variable, Parameter, Method, & Event

Dalam pemrograman komputer dan *data science*, *type* merupakan gambaran dari *object* (obyek) yang dapat menampung/memiliki berbagai *variable* (*field*, *property*, *parameter*, *attribute*, atau *feature*), *method* (*function/procedure*), dan *event* [14]. Umumnya, *type* terdiri dari *class* dan *structure*. Dengan demikian, *class* dan *structure* merupakan obyek yang dapat menampung *variable*, *method*, dan *event*. Analoginya, *class* ibarat sebuah gambar bangunan di atas kertas, sedangkan obyek merupakan bangunan sudah jadi yang dibuat berdasarkan gambar tersebut.

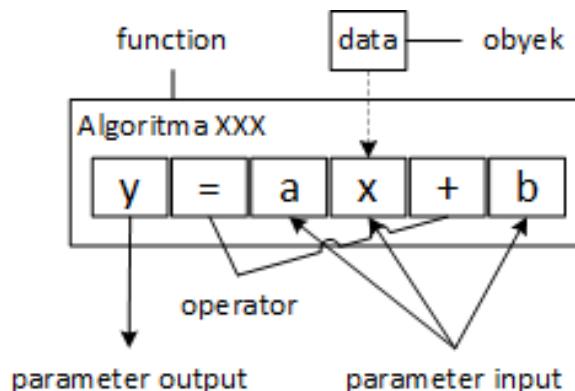
Variable atau *field* merupakan tempat untuk menyimpan atau menampung nilai/data, sehingga dapat mendefinisikan dirinya sebagai suatu *type/object* (*class* atau *structure*), dapat menjadi anggota dari suatu obyek, dapat bernilai suatu obyek, namun tidak dapat memiliki *parameter* dan *method* di dalamnya.

Property atau *parameter* mirip dengan *field/variable*. Bedanya, visibilitas *field/variable* tidak dapat diakses langsung dari luar obyeknya, sedangkan *property/parameter* dapat diakses langsung dari luar obyeknya. *Variable/field* dan *property/parameter* dapat membentuk suatu vektor.

Method (function/procedure) merupakan suatu aksi atau tindakan yang dimiliki suatu obyek untuk menyelesaikan masalah tertentu yang djalankan/dilaksanakan oleh *event*. Suatu *function* dapat mendefinisikan dirinya sebagai suatu obyek, sehingga memiliki nilai/output. Suatu *function* dapat pula memiliki parameter inputan. Suatu *function* dari suatu obyek dapat mengimplementasikan *variable*, *property*, dan obyek lain di dalamnya. Dengan demikian, *method/function* dapat dikatakan sebagai suatu atau serangkaian algoritma. Perlu diketahui bahwa dalam pemrograman prosedural tidak ada istilah *event* karena program djalankan dalam *main method* secara sekuensial/beruntun.

Perhatikan gambaran variable atau parameter dari suatu *function* algoritma pada Contoh 2.3. Perhatikan contoh *variable/field*, *property/parameter*, *method/function*, dan *event* pada suatu obyek tabel/matrix XXX yang ditunjukkan pada Contoh 2.4. Begitupun contoh-contohnya dalam pemrograman *data science*, pemrograman berorientasi objek, dan pemrograman prosedural yang ditunjukkan pada Contoh 2.5 dan Contoh 2.6.

Contoh 2.3 Function suatu Algoritma dan Obyek



Contoh 2.4 Object, Field, Property, Method, & Event

Type/Obyek/Class/Matriks/Tabel XXX								
	Record ke-	Property						
	Field1	Field2	Field3	Field4	Field5	Field6	Field7	
Vektor1	1	9	9,3	B	Abc	1500	True	0
Vektor2	2	1	1,5	C	Def	1200	False	1
Vektor3	...	7	7,15	A	Ghi	750	False	1
Vektor4	n	3	3,2	E	Jkl	2000	True	0
	VektorF1	VektorF2	VektorF3	VektorF4	VektorF5	VektorF6	VektorF7	

↑ Method beraksi (data di obyek/tabel XXX bertambah)

Method ← Input Data	Param1	Param2	Param3	Param4	Param5	Param6	Param7
	?	?	?	?	?	?	?

↑ Eksekusi method

Event ← Ada Data Baru	2	7.3	D	Mno	9000	True	1
-----------------------	---	-----	---	-----	------	------	---

Contoh 2.5 Field/Variable

Contoh *field/variable* dalam *data science programming* (Matlab):

```
a = [1, 2, 3; 4, 5, 6]; b = a(2,3); %6
```

Contoh *field/variable* dalam *object oriented programming* (VB .NET):

```
Dim a(,) As integer = New Integer(1,2) {{1, 2, 3}, {4, 5, 6}}
b = a(1,2) '6
```

Contoh *field/variable* dalam *procedural programming* (Pascal):

```
Var a : array[1..6] of integer;
```

Contoh 2.6 Function dan Parameter

Contoh *function* dalam *data science programming* (Matlab):

```
function [output1, output2] = Algoritma1(input1, input2, input3)
    ...
end
```

Contoh *function* dalam *object oriented programming* (VB .NET):

```
Function Algoritma1(input1, input2, input3) As Double
    ...
End Function
```

Contoh *function* dalam *procedural programming* (Pascal):

```
function Algoritma1(var input1 : integer; var input2 : integer) : real;
begin
    ...
End
```

2.7 Soal Latihan Pengantar Machine Learning

1. Sebutkan beberapa metode machine learning yang belum disebutkan dalam bab ini?
2. Kumpulkan suatu *dataset* publik, kemudian jelaskan karakteristik *dataset* tersebut!
3. Berdasarkan soal No. 2:
 - Sebutkan analisis *data science* apa yang dapat diterapkan pada *dataset* tersebut dan mengapa memilih analisis tersebut?
 - Sebutkan metode *machine learning* apa yang dapat digunakan untuk menganalisis *dataset* tersebut dan mengapa memilih metode tersebut?
 - Sebutkan satu atau lebih pra pengolahan apa yang dapat diterapkan pada *dataset* tersebut lengkap dengan metodenya dan mengapa memilih metode serta pra pengolahan tersebut?
 - Sebutkan metode evaluasi apa yang dapat diterapkan terhadap metode *machine learning* yang digunakan untuk menganalisis *dataset* tersebut yang telah disebutkan sebelumnya?

3. Pra Pengolahan Data

No.	Materi	Tujuan Pembelajaran
1.	<i>Missing Value Replacement</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>missing value replacement</i> . Selanjutnya anda mampu menerapkan pendekatan imputasi <i>mean/mode</i> untuk <i>missing value replacement</i> .
2.	<i>Data Type Transformation</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>data type transformation</i> . Selanjutnya anda mampu menerapkan pendekatan <i>Encoding</i> untuk transformasi tipe data kategorikal ke numerikal dan menerapkan pendekatan <i>data discretization</i> menggunakan teknik <i>Binning</i> dan <i>Entropy</i> untuk transformasi tipe data numerikal ke kategorikal.
3.	<i>Aggregation</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>aggregation</i> . Selanjutnya anda mampu menerapkan pendekatan <i>sum, min, max, mean, median</i> , dll untuk <i>aggregation</i> .
4.	<i>Smoothing Noisy Data</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>smoothing noisy data</i> . Selanjutnya anda mampu menerapkan algoritma k-NN untuk <i>outlier detection</i> dan menerapkan metode <i>Min-Max Normalization, Z-Score, Decimal Scaling, Sigmoidal</i> , dan <i>Softmax</i> untuk <i>data normalization</i> .
5.	<i>Feature Selection & Feature Extraction</i>	Anda mampu memahami, menjelaskan, dan membedakan prosedur dan tujuan melakukan <i>dimensionality reduction, feature selection, attribute weighting</i> , dan <i>feature extraction</i> . Selanjutnya anda mampu menerapkan algoritma PCA dan SVD untuk <i>dimensionality reduction</i> , <i>Genetic Algorithm</i> dan <i>Forward/Backward</i> untuk <i>feature selection</i> , serta ACC untuk <i>attribute weighting</i> .
6.	<i>Unbalanced Class Reduction</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>unbalanced class reduction</i> . Selanjutnya anda mampu menerapkan pendekatan <i>ensemble</i> menggunakan algoritma <i>AdaBoost, Bagging, Stacking</i> , dan <i>Weighted Vote</i> untuk <i>unbalanced class reduction</i> .
7.	<i>Data Validation</i>	Anda mampu memahami dan menjelaskan prosedur dan tujuan melakukan <i>data validation</i> . Selanjutnya anda mampu menerapkan metode <i>Holdout, Leave-One-Out Cross Validation</i> , dan <i>K-Fold Cross Validation</i> untuk <i>data validation</i> .

3.1 Missing Value Replacement

Missing value yang jumlahnya sedikit dapat diatasi dengan cara membuangnya. Namun jika *missing value* jumlahnya banyak, membuangnya dapat menghilangkan informasi yang mungkin penting, mengakibatkan bias [16]. Oleh karena itu, *missing value replacement* penting dilakukan. Terdapat beberapa pendekatan yang dapat dilakukan untuk *missing value replacement*, antara lain dengan mengabaikannya, melalui estimasi parameter, dan melalui imputasi [16]. Mengabaikan *missing value*, tidak jauh berbeda dengan membuangnya, dapat mengakibatkan bias. Pendekatan yang paling umum dilakukan adalah imputasi. Pendekatan imputasi terdiri dari *case substitution*, *mean/mode (average)*, *hot deck* & *cold deck*, dan dapat pula dengan menggunakan pendekatan model prediksi [16].

Salah satu metode *missing value replacement* dengan pendekatan imputasi yaitu *Nonparametric Iterative Imputation Algorithm* (NIIA) [17]. Sedangkan salah satu metode *missing value replacement* dengan pendekatan imputasi menggunakan model prediksi yaitu *k-Nearest Neighbor* (*k*-NN) *Imputation* [16]. Namun metode *missing value replacement* dengan pendekatan imputasi yang lebih sederhana dan umum digunakan yaitu *mean/mode (average)*, yang mana *missing value* diganti dengan nilai *mean* pada variabel kontinyu atau nilai *mode* pada variabel diskrit [16], [17], [18], [19]. Perlu diketahui bahwa menurut kamus Oxford, makna *average* berbeda dengan *mean*. *Average* merupakan ukuran pusat data, dapat berupa *mean*, *mode*, atau *median* tergantung pada konteksnya.

Terdapat 2 cara yang dapat dilakukan untuk *missing value replacement* menggunakan imputasi *mean/mode*, yaitu dengan melibatkan label *class* atau tidak. Tentu saja jika persoalannya bukan klasifikasi, maka label *class* tidak perlu dilibatkan, karena memang data tidak memiliki label *class*. Jika variabel *missing value* bertipe *integer*, maka sebaiknya bulatkan nilai *mean* yang diperoleh, karena bisa jadi tipe data asli atribut tersebut adalah *ordinal* yang ditransformasi ke *integer* dengan *encoding* numerik. Namun jika variabel *missing value* bertipe *real*, maka gunakan nilai *mean* yang diperoleh. Perhatikan contoh *missing value replacement* menggunakan imputasi *mean/mode* berikut ini.

Contoh 3.1 Missing Value Replacement: Mean/Mode (Manual)

No	Data Asli		Missing Value Replacement				Class
			Cara 1		Cara 2		
	A	B	A	B	A	B	
1.	2	Kuning	2	Kuning	2	Kuning	1
2.	3	Merah	3	Merah	3	Merah	1
3.	2	Kuning	2	Kuning	2	Kuning	1
4.	?	?	2	Hijau	2 (2,33)	Kuning	1
5.	?	?	2	Hijau	2 (2,33)	Kuning	1
6.	1	Hijau	1	Hijau	1	Hijau	0
7.	?	?	2	Hijau	1 (1,5)	Hijau	0
8.	1	Hijau	1	Hijau	1	Hijau	0
9.	3	Merah	3	Merah	3	Merah	0
10.	1	Hijau	1	Hijau	1	Hijau	0

3.2 Data Type Transformation

Transformasi data (*data transformation*) adalah perubahan bentuk, rupa, sifat, fungsi, dll data dengan cara menambah, mengurangi, atau menata kembali unsur-unsur dari data tersebut. Dengan demikian, *data type transformation*, *feature selection*, *feature extraction*, *aggregation*, dan bahkan *Kernel Trick* merupakan *data transformation*. *Data type transformation* bertujuan untuk merubah satu atau lebih tipe variabel pada suatu *dataset* yang biasanya agar data tersebut dapat diolah oleh algoritma yang digunakan. *Feature selection* bertujuan untuk mengurangi variabel-variabel pada suatu *dataset*. *Feature extraction* bertujuan untuk merubah data yang belum terstruktur menjadi data terstruktur. *Aggregation* bertujuan untuk menggabungkan beberapa variabel pada suatu *dataset*. *Kernel trick* dalam *machine learning* merupakan suatu fungsi yang digunakan untuk merubah data ke dimensi baru (*feature space*) yang lebih tinggi agar data dapat dipisahkan secara linier (dapat mengakibatkan *curse of dimensionality*). Dalam sub pokok bahasan ini hanya membahas tentang *data type transformation*, sedangkan *data transformation* lainnya dibahas pada sub pokok bahasan lainnya dalam bab ini.

Umumnya ada dua jenis *data type transformation*, yaitu kategorikal ke numerikal atau sebaliknya numerikal ke kategorikal [20]. Terdapat pula transformasi tipe data dari numerikal atau kategorikal ke *date* dan sebaliknya. Transformasi tipe data dari kategorikal ke numerikal diistilahkan dengan *encoding*, sedangkan numerikal ke kategorikan diistilahkan dengan *data discretization*. Perlu diperhatikan bahwa perubahan tipe data suatu variabel tidak boleh merubah sifat dari variabel tersebut.

3.2.1 Encoding

Terdapat 2 cara yang mudah dan umum digunakan untuk melakukan *data type transformation* dari kategorikal ke numerikal (*encoding*), yaitu dengan *numerical encoding* dan *dummy encoding*. Perhatikan contoh berikut, yang mana terdapat 2 variabel/parameter/atribut kategorikal yang ditransformasikan ke numerikal.

Contoh 3.2 Data Type Transformation: Encoding (Manual)

Asli			Hasil Transformasi				
Jinkel	Warna	Tinggi	Jinkel	Warna1	Warna2	Warna3	Tinggi
Binomial	Nominal	Ordinal	Integer	Integer	Integer	Integer	Integer
			Encoding	Dummy Coding			Encoding
Pria	Merah	Pendek	0	1	0	0	1
Wanita	Kuning	Ideal	1	0	1	0	2
Wanita	Hijau	Tinggi	1	0	0	1	3
Pria	Hijau	Ideal	0	0	0	1	2
Wanita	Merah	Pendek	1	1	0	1	1

3.2.2 Data Discretization

Diskretisasi data (*data discretization*) bertujuan untuk mentransformasi data numerikal ke kategorikal atau dari kuantitatif ke kualitatif [21]. Algoritma seperti standar *Decision Tree* (DT) dan *Naïve Bayes* (NB) membutuhkan diskretisasi pada data numerikal [22]. Namun pengembangan dari algoritma-algoritma tersebut sudah dapat menangani data numerikal, misalnya C4.5 yang merupakan pengembangan dari DT, atau *Gaussian Naïve Bayes* yang merupakan pengembangan dari *Naïve Bayes*. Perlu diketahui bahwa diskretisasi data dapat menghilangkan informasi yang mungkin penting.

Ada banyak pendekatan yang dapat digunakan untuk melakukan diskretisasi data, dua di antaranya yang umum dan sering digunakan, yaitu *Binning* dan *Entropy*. Pendekatan *Binning* menggunakan persamaan *Entropy* (1) dan (2), *Information Gain* (3), dan *Gain* (4). Pendekatan *Binning* mendefinisikan kumpulan *class* nominal untuk setiap atribut (variabel input), kemudian menetapkan setiap nilai atribut ke dalam salah satu *class*. Misalnya, jika domain atribut numerik mempunyai nilai dari 0 sampai dengan 100, domain tersebut dapat dibagi menjadi empat bin {0..24, 25..49, 50..74, 75..100}. Setiap nilai atribut akan dikonversi menjadi atribut nominal/kategorikal yang berkorespondensi dengan salah satu *bin*. Oleh karena itu pendekatan *Binning* disebut *unsupervised discretization method*. Namun pendekatan ini memiliki kelemahan, pendekatan ini dapat menyebabkan banyak informasi yang bisa hilang.

$$E(S) = - \sum_{i=1}^m P(\omega_i|S) \log_2 P(\omega_i|S) \quad (1)$$

$$E(s_j) = - \sum_{i=1}^m P(\omega_i|s_j) \log_2 P(\omega_i|s_j) \quad (2)$$

$$IG(s) = \sum_{j=1}^n P(s_j|s) E(s_j) \quad (3)$$

$$G(s) = E(S) - IG(s) \quad (4)$$

- $E(S)$: menyatakan *Entropy* variabel output (*class*).
- $E(s_j)$: menyatakan *Entropy* label ke- j dari *bin s*.
- $IG(s)$: menyatakan *Information Gain* *bin s*.
- $G(s)$: menyatakan *Gain* *bin s*.
- m : menyatakan banyaknya label *class*.
- n : menyatakan banyaknya pemecahan/*split* (label) dari *bin s*.
- $P(\omega_i|S)$: menyatakan proporsi label *class* ke- i dari seluruh data, yaitu jumlah data label *class* ke- i dibagi dengan jumlah seluruh data.
- $P(\omega_i|s_j)$: menyatakan proporsi label *class* ke- i dari data yang diproses dalam label ke- j *bin s*, yaitu jumlah data label *class* ke- i dalam label ke- j *bin s* dibagi dengan jumlah seluruh data dalam label ke- j *bin s*.
- $P(s_j|s)$: menyatakan proporsi label ke- j dari *bin s* dari data yang diproses dalam *bin s* (jumlah seluruh data).

Contoh 3.3 Data Discretization: Binning (Manual)

No	Cuaca	Suhu	Kelembaban Angin	Bermain (Class)
1	Cerah	85	85 Biasa	Tidak
2	Cerah	80	90 Kencang	Tidak
3	Mendung	83	78 Biasa	Ya
4	Hujan	70	96 Biasa	Ya
5	Hujan	68	80 Biasa	Ya
6	Hujan	65	70 Kencang	Tidak
7	Mendung	64	65 Kencang	Ya
8	Cerah	72	95 Biasa	Tidak
9	Cerah	69	70 Biasa	Ya
10	Hujan	75	80 Biasa	Ya
11	Cerah	75	70 Kencang	Ya
12	Mendung	72	90 Kencang	Ya
13	Mendung	81	75 Biasa	Ya
14	Hujan	71	80 Kencang	Tidak

Label 'Ya' : 9
 Label 'Tidak' : 5
 Jumlah : 14
 Entropy : **0.9403**

Suhu	v = 70		v = 75		v = 80	
	<=70	>70	<=75	>75	<=80	>80
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Jumlah	5	9	10	4	11	3
	14		14		14	
Entropy	0,7219	0,9911	0,8813	1,0000	0,9457	0,9183
Gain	-0,8950		-0,9152		-0,9398	

Maka variabel 'Suhu' akan dibagi menjadi ≤ 70 dan >70 .

Sedangkan pendekatan *Entropy* berusaha menemukan pemisah terbaik dengan menghitung *Split Info*, *Entropy*, *Information Gain*, dan *Gain* antara dua sampel [23]. Algoritmanya adalah: urutkan data secara *ascending*, hitung *Split Info* (5), hitung *Entropy* setiap label *Split Info* (6), hitung *Information Gain* setiap *Split Info* (7), hitung *Gain* setiap *Split Info* (8), dan gunakan *Split Info* yang memiliki *Gain* tertinggi.

$$s = \frac{a_j + a_{j+1}}{2} \quad (5)$$

$$E(s_j) = - \sum_{i=1}^m P(\omega_i | s_j) \log_2 P(\omega_i | s_j) \quad (6)$$

$$IG(s) = \sum_{j=1}^n P(s_j | s) E(s_j) \quad (7)$$

$$G(s) = E(S) - IG(s) \quad (8)$$

Contoh 3.4 Data Discretization: Entropy (Manual)

Umur	Jinkel	Bacaan	Tujuan	Lingkungan	Minat		
41	Wanita	Non Fiksi	Wawasan	Terbuka	Sedang		
47	Wanita	Fiksi	Wawasan	Tertutup	Sedang		
38	Wanita	Fiksi	Wawasan	Tertutup	Sedang		
42	Wanita	Fiksi	Wawasan	Terbuka	Tinggi		
37	Wanita	Non Fiksi	Wawasan	Tertutup	Tinggi		
35	Wanita	Non Fiksi	Wawasan	Tertutup	Tinggi		
35	Pria	Fiksi	Motivasi	Terbuka	Sedang		
Urutkan secara ascending atribut kontinyu ‘Umur’							
35					Tinggi		
35					Sedang		
37					Tinggi		
38					Sedang		
41					Sedang		
42					Tinggi		
47					Sedang		
Split Info	Rendah	Sedang	Tinggi	Jumlah	Entropy	Info	Gain
35	<=36	-	1	1	2	1	
	>36	-	3	2	5	0,970951	0,97925
37	<=37,5	-	1	2	3	0,918296	0,857143
	>37,5	-	3	1	4	0,811278	0,128085
38	<=39,5	-	2	2	4	1	
	>39,5	-	2	1	3	0,918296	0,964984
41	<=41,5	-	3	2	5	0,970951	0,020244
	>41,5	-	1	1	2	1	
42	<=44,5	-	3	3	6	1	
	>44,5	-	1	0	1	0	0,005978
							0,4
	Jumlah	0	4	3	7	0,985228	

Maka variabel ‘Umur’ akan dibagi menjadi $\leq 37,5$ dan $> 37,5$.

3.3 Aggregation

Agregasi (*aggregation*) merupakan pengkombinasian beberapa nilai pada data menjadi suatu nilai tunggal [20]. Agregasi dapat digunakan apabila terdapat sejumlah nilai dalam suatu variabel pada data yang bisa dijadikan satu kelompok. Beberapa pendekatan agregasi yang umum digunakan, antara lain *sum*, *mean*, *min*, dan *max* [20].

Mengapa agregasi dilakukan? Pertama, data yang lebih kecil tentu saja membutuhkan kompleksitas komputasi yang lebih sedikit dan waktu proses yang lebih cepat; Kedua, merubah cara pandang terhadap data dari level rendah ke level tinggi; Ketiga, agregasi sering kali lebih stabil dari pada menggunakan data aslinya/awalnya [20].

Contoh 3.5 Agregation: Sum (Manual)

Atribut A	Atribut B	Atribut C	Atribut D
Data Asli			
Makassar	01001	19 Nov 1992	700.000
Makassar	01002	19 Nov 1992	250.000
Jakarta	02001	01 Des 2017	100.000
Jakarta	02002	01 Des 2017	200.000
Jakarta	02003	14 Okt 1979	900.000
Data Setelah Agregasi			
Atribut A	Atribut C	Atribut D	
Makassar	19 Nov 1992	950.000	
Jakarta	01 Des 2017	300.000	
Jakarta	14 Okt 1979	900.000	

3.4 Smoothing Noisy Data

Data yang karakteristiknya secara signifikan berbeda atau menyimpang dengan karakteristik data pada umumnya disebut *outlier* [24]. Misalnya data umur manusia, umumnya umur manusia dari 1 hingga 90 tahun, umur manusia di atas 100 tahun dapat disebut *outlier/anomali*. Adanya *outliers* dapat disebabkan beberapa hal, antara lain [20]:

1. Data berasal dari *class* atau mekanisme atau pola yang berbeda.
2. Variasi natural. Data dapat dimodelkan dengan distribusi statistik, misalnya dengan distribusi normal (*Gaussian*), yang mana probabilitas kemunculan data akan menurun secara drastis sebagai peningkatan fungsi jarak dari pusat distribusi. Misalnya, data jam bangun pagi didistribusikan secara normal, yang mana pusat distribusinya yaitu pada jam 5 kerena merupakan jumlah tertinggi pada data jam bangun pagi dari jangkaun nilai jam 2 hingga jam 8. Dengan memangdang distribusi tersebut, maka data jam bangun pagi dengan nilai ≤ 1 atau ≥ 9 terdeteksi sebagai *outlier*.
3. Kesalahan pengumpulan dan pengukuran data.

Adanya *outlier* atau anomali pada data dapat menyebabkan *noisy data*, berdampak buruk terhadap kinerja model suatu algoritma *machine learning* [18], [19], [25]. Oleh karena itu, dalam pekerjaan klasifikasi, estimasi, klasterisasi, ataupun asosiasi, perlu dilakukan *smoothing noisy data* yang dapat diatasi dengan pendekatan *anomaly detection* atau *data normalization*.

3.4.1 Anomaly Detection

Anomaly detection melakukan *smoothing noisy data* dengan cara mendeteksi *outlier* kemudian membuangnya (*outlier removal*). Sedangkan *data normalization* tidak mendeteksi *outlier* kemudian membuangnya, namun mentransformasi data untuk *smoothing noisy data*. Selain itu, *anomaly detection* dapat pula dimanfaatkan bukan untuk *smoothing noisy data*, melainkan memang untuk memprediksi *outlier*, seperti masalah deteksi kecurangan, deteksi penyusupan, dll. *Anomaly detection* sebenarnya merupakan salah satu pendekatan klasterisasi.

Metode-metode *machine learning* dapat pula digunakan untuk *anomaly detection*, seperti *k-Nearest Neighbor*, *K-Means*, *DBSCAN* dll. Perlu diketahui bahwa menggunakan k-NN untuk *anomaly detection* kurang mampu menangani data dengan tingkat kepadatan yang berbeda-beda, karena menggunakan ambang global yang sebenarnya tidak dapat dipertanggung jawabkan pada data dengan tangkat kepadatan yang berbeda-beda [20]. Berikut ini contoh penyelesaian manual *anomaly detection* menggunakan *k-Nearest Neighbor* (k-NN).

Contoh 3.6 Anomaly Detection: k-NN (Manual)

Data Ke-	Atribut A	Atribut B	Atribut C	Atribut D
1	1	2	4	5
2	9	3	6	7
3	7	8	9	8

Jarak Euclidean Antar Data:

Data Ke-	Atribut A	Atribut B	Atribut C	Atribut D	Jarak
1	0	0	0	0	0
2	$(1 - 9)^2 = 64$	$(2 - 3)^2 = 1$	$(4 - 6)^2 = 4$	$(5 - 7)^2 = 4$	8,5440
3	$(1 - 7)^2 = 36$	$(2 - 8)^2 = 36$	$(4 - 9)^2 = 25$	$(5 - 8)^2 = 9$	10,2956
2	0	0	0	0	0
1	$(9 - 1)^2 = 64$	$(3 - 2)^2 = 1$	$(6 - 4)^2 = 4$	$(7 - 5)^2 = 4$	8,5440
3	$(9 - 7)^2 = 4$	$(3 - 8)^2 = 25$	$(6 - 9)^2 = 9$	$(7 - 8)^2 = 1$	6,2450
3	0	0	0	0	0
1	$(7 - 1)^2 = 36$	$(8 - 2)^2 = 36$	$(9 - 4)^2 = 25$	$(8 - 5)^2 = 9$	10,2956
2	$(7 - 9)^2 = 4$	$(8 - 3)^2 = 25$	$(9 - 6)^2 = 9$	$(8 - 7)^2 = 1$	6,2450

Jarak NxN data (beri nilai maksimal untuk jarak = 0 = 999):

Data Ke	1	2	3
	999	8,5440	10,2956
8,5440	999	6,2450	
10,2956	6,2450	999	

k tetangga terdekat, $k = 2$

Data Ke	1	2	3
k=1	8,5440	6,2450	6,2450
k=2	10,2956	8,5440	10,2956
k=3	999,0000	999,0000	999,0000
Mean k=2	9,4198	7,3945	8,2703

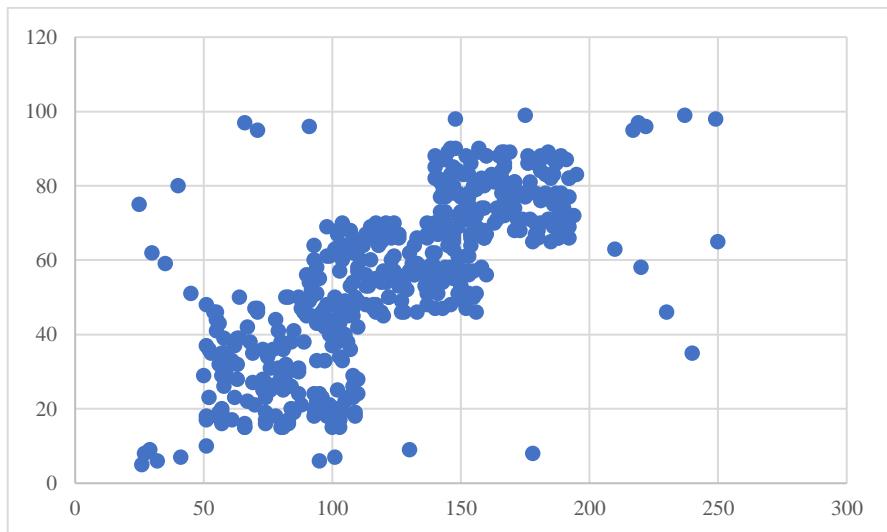
Jika $T \geq 8$, maka data ke-1 dan data ke-3 yang dianggap sebagai *outlier*.

Contoh 3.7 Anomaly Detection: k-NN (Matlab)

Dataset : *dsTinggiBeratBadan* (terlampir).

k-NN : $k = 7$; *Treshold (T) = 0,2*; *Distance measure = Cityblock*.

	Range			Total
	1	2	3	
Tinggi Badan	50 - 110	90 - 160	140 - 195	50 - 195
Berat Badan	15 - 50	45 - 70	65 - 90	15 - 90
Total	172	164	164	500
	34.40	32.80	32.80	100
Outlier 1-10	Tinggi Badan: No 1 - 5 > 195 & No 6 - 10 < 50			
Outlier 11-20	Berat Badan: No 11 - 15 > 90 & No 16 - 20 < 15			
Ootlier 21-30	No 21 - 25: Tinggi Badan < 50 & Berat Badan < 15 No 26 - 30: Tinggi Badan > 195 & Berat Badan > 90			
Class	Outlier (1)	Not (0)		Total
	30	470		500
	6%	94%		100



Data yang jauh dari pusat data merupakan *outlier*. Berikut ini kode programnya menggunakan Matlab.

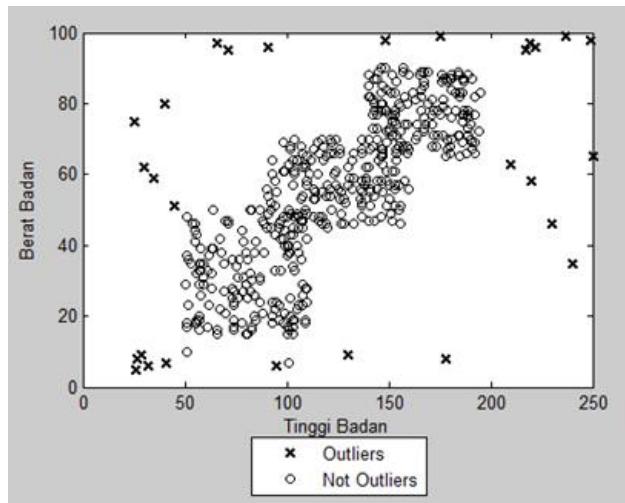
Script "AnomalyDetectionKNN.m":

```
clc; clear all; close all; warning off all;
% baca data
dsTinggiBeratBadan = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsTinggiBeratBadan');
% kolom: 1 ID, 2 Tinggi Badan, 3 Berat Badan, 4 Class
ds = dsTinggiBeratBadan(:,1:4);
K=7; % K
T=0.2; % rata2 max tetangga terdekat sebesar T persen
distance='cityblock'; % fungsi jarak (euclidean, cityblock, dll)
dataWithIDnClass = ds; % data lengkap, ID, input, output/class
data = ds(:,2:3); % data input (Tinggi & Berat) yang akan diolah
% Fungsi KNN Anomaly Detection
[outliers, avgJarak, tetangga, akurasi] = AnomalyDetectionKNNaa(data,
dataWithIDnClass(:,4), K, T, distance);
% Tampilan grafik hasilnya
figure('Position',[300 100 420 350]);
plot(data(outliers,1), data(outliers,2),
'kx','MarkerSize',8,'LineWidth',2);
hold on
plot(data(xor(1,outliers),1),data(xor(1,outliers),2),'ko','MarkerSize
',5,'LineWidth',1);
hold off
legend('Outliers','Not Outliers','Location','SouthOutside');
xlabel('Tinggi Badan');
ylabel('Berat Badan');
% data yang diprediksi sebagai outlier
dataOutliers = dataWithIDnClass(outliers,:)
```

Function "AnomalyDetectionKNNaa.m"

```
function [outliers, avgJarak, tetangga, akurasi] =
AnomalyDetectionKNNaa(data, class, K, T, distance)
[n,m] = size(data); % n baris, m kolom
jarak = pdist(data, distance); % jarak antar data
jarak = squareform(jarak); % jarak antar data n x n
```

```
% jarak data dengan dirinya sendiri = 0 diubah ke = 999, karena K
tetangga akan diambil dari jarak minimal dst hingga K data
jarak(logical(eye(n))) = 999;
for i=1:n
    ttg(i,:) = sort(jarak(i,:)); % urutkan asc
    tetangga(i,:) = ttg(i,1:K); % ambil data n x n yang masuk K
end
avgJarak = mean(tetangga)'; % transform agar sesuai dng dimensi data
T = T * max(avgJarak); % batas data yang diprediksi sebagai outlier
% indeks = 1 jika data K tetangga avg jaraknya >= T, ambil indeks
% data K tetangga tersebut
indeks = find(avgJarak >= T);
outliers = zeros(n,1); % beri nilai 0 untuk setiap indeks data
outliers(indeks) = 1; % 1 untuk data yang sesuai indeks (ke-indeks)
% Evaluasi dengan Confusion Matrix
conMat = confusionmat(class, outliers);
jmlData = sum(conMat(:));
hasilBenar = sum(diag(conMat));
akurasi = 100 * (hasilBenar / jmlData);
outliers = logical(outliers); % ubah ke logical agar bisa ambil data
end
```



Setelah berbagai percobaan pada nilai k , T , dan fungsi jarak, akurasi terbaik = 99.60% diperoleh pada $k = 7$, $T = 0.2$ (20% dari rata-rata data k tetangga terdekat yang maksimum), dan fungsi jarak = *Cityblock*. Fungsi jarak *Cityblock* biasanya lebih tangguh untuk *anomaly detection* berbasis jarak, karena *Cityblock/Manhattan* menggunakan jumlah selisih absolut sehingga mampu memberikan jarak terjauh antara 2 data. Dari 30 outlier yang ada pada data, k-NN salah mendekripsi 2 data, yaitu data ke-16 dan data ke-19 sebagai berikut.

Data ke-	Tinggi	Berat	Actual Class	k-NN Detection
16	51	10	1	0
19	101	7	1	0

3.4.2 Data Normalization

Selain dengan pendekatan *anomaly detection*, *smoothing noisy data* dapat pula dilakukan dengan pendekatan *data normalization*. Beberapa metode data *normalization*, yaitu: *Min-Max Normalization*, *Z-Score*, *Decimal Scaling*, *Sigmoidal*, dan *Softmax*.

Min-Max Normalization (9) merupakan pendekatan normalisasi data yang melakukan transformasi linier terhadap data.

$$x'_i = \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} ((new_{\max} - new_{\min}) + new_{\min}) \quad (9)$$

Z-Score (10) merupakan pendekatan normalisasi data yang berdasarkan *mean* (μ) dan *standard deviation* (σ) dari data.

$$x'_i = \frac{(x_i - \mu)}{\sigma} \quad (10)$$

Mean (μ) dan *standard deviation* (σ) diperoleh melalui persamaan berikut ini.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (11)$$

$$\sigma = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2 \right)^{1/2} \quad (12)$$

Decimal Scaling (13) merupakan pendekatan normalisasi data yang menggerakkan nilai data ke arah yang diinginkan. Jika y merupakan nilai *scaling* yang diinginkan, maka persamaannya dapat didefinisikan sebagai berikut.

$$x'_i = \frac{x_i}{10^y} \quad (13)$$

Sigmoidal (14) merupakan pendekatan normalisasi data secara nonlinier ke dalam *range* [-1, 1] dengan menggunakan fungsi *sigmoid*. Pendekatan ini sangat berguna pada data yang memiliki banyak *outlier*. Jika y merupakan *Z-Score* (10) dan e merupakan eksponensial = 2,718281828, maka persamaannya dapat didefinisikan sebagai berikut.

$$x'_i = \frac{(1 - e^{-y})}{(1 + e^{-y})} \quad (14)$$

Softmax (15) merupakan pendekatan normalisasi data pengembangan transformasi secara linier, yang mana outputnya adalah [0, 1]. Pendekatan ini sangat berguna pada data yang memiliki banyak *outlier*. Jika y merupakan respon linier dari *standard deviation* (ditentukan oleh *user*), maka persamaannya dapat didefinisikan sebagai berikut.

$$x'_i = \frac{1}{(1 + e^t)} \quad (15)$$

Nilai t pada Persamaan (15) diperoleh melalui persamaan berikut ini.

$$t = \frac{(x_i - \mu)}{(y(\sigma/(2\pi)))} \quad (16)$$

Contoh 3.8 Data Normalization (Manual)

$x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];$	Hasil
Min-Max [0, 1] $x(5) = ((5 - 1) / (10 - 1)) * (1 - 0) + 0$	0,44
Z-Score $x(5) = (5 - 5,5) / 3,03$	-0,17
DS [0,1] $x(5) = 5 / 10^{0,1}$	3,97
Sigmoid $x(5) = (1 - 2,72^{0,17}) / (1 + 2,72^{0,17})$	0,08

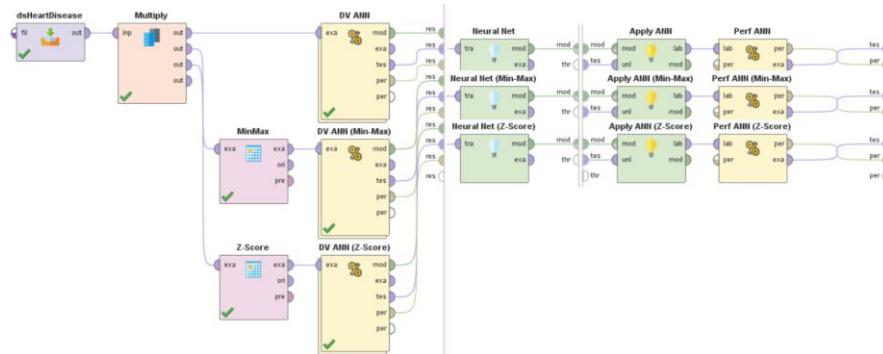
Contoh 3.9 Data Normalization: Min-Max & Z-Score (Rapidminer)

Dataset : dsHeartDiseaseCleveland – Class: {0,1} (terlampir)

Data normalization : Min-Max Normalization (range: 0 – 1)
Z-Score

Classifier : ANN (1 hidden layer dengan 10 neurons)

Data validation : 10-Fold Cross Validation



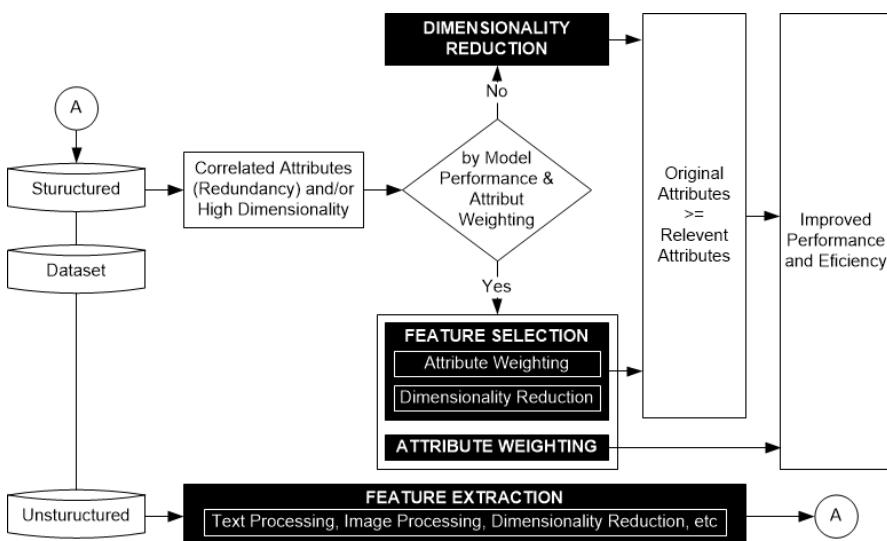
Berikut ini adalah hasil evaluasinya menggunakan *Confusion Matrix*. Dalam kasus ini menunjukkan bahwa penerapan *smoothing noisy data* dengan pendekatan *data normalization* (Min-Max dan Z-Score) dapat meningkatkan kinerja model.

ANN			
accuracy: 78,88% +/- 5,19% (micro average: 78,88%)			
	true 0	true 1	precision
prediction 0	134	34	79,76%
prediction 1	30	105	77,78%
recall	81,71%	75,54%	
ANN + Min - Max Normalization			
accuracy: 80,89% +/- 7,63% (micro average: 80,86%)			
	true 0	true 1	precision
prediction 0	138	32	81,18%
prediction 1	26	107	80,45%
recall	84,15%	76,98%	
ANN + Z-Score			
accuracy: 80,89% +/- 9,33% (micro average: 80,86%)			
	true 0	true 1	precision
prediction 0	137	31	81,55%
prediction 1	27	108	80,00%
recall	83,54%	77,70%	

3.5 Feature Selection & Feature Extraction

Ketika terjadi beberapa redudansi pada atribut-atribut (variabel input/bebas), maksudnya ketika terdapat atribut-atribut yang saling berkorelasi, maka tentu saja lebih efisien untuk mereduksi, menggabungkan, atau membuang atribut-atribut yang saling berkorelasi tersebut. Dapat pula dengan memberikan bobot (*weight*) pada atribut-atribut untuk memperoleh tingkat relevansi setiap atribut. Pada prinsipnya, atribut-atribut (variabel input/bebas) diasumsikan bersifat independen/bebas. Dengan kata lain, atribut-atribut tidak saling berkorelasi/terikat, atribut-atribut input hanya terikat dengan atribut output (variabel-variabel input/bebas hanya mempengaruhi variabel output/terikat). Padahal dalam banyak kasus, biasanya terdapat atribut-atribut yang saling berkorelasi, sehingga asumsi ini tidak selalu tepat. Pada keadaan lain, ketika atribut-atribut berjumlah sangat banyak, atau dengan kata lainnya *dataset* berdimensi sangat tinggi, maka tentu saja lebih efisien untuk mereduksi dimensi data (membuang atribut yang tidak atau bahkan kurang relevan), sehingga kompleksitas komputasi terhadap waktu maupun ruang yang begitu besar dapat pula direduksi. Keadaan-keadaan data seperti ini memicu dilakukannya *dimensionality reduction*, *feature selection*, atau *attribute weighting* untuk mengoptimalkan kinerja model. Biasanya ketiga pendekatan tersebut dianggap *feature selection*, karena memiliki sebab, input, proses, dan output yang mirip.

Jika *feature selection* (*dimensionality reduction*, *feature selection*, *attribute weighting*) dilakukan pada data yang terstruktur, maka pada data yang tidak terstruktur, dilakukan *feature extraction* dengan tujuan untuk memperoleh data yang terstruktur. Perlu diketahui bahwa metode-metode *dimensionality reduction* dapat pula digunakan untuk *feature extraction*. Untuk lebih jelasnya, perhatikan perbedaan *feature selection* dan *feature extraction* pada gambar berikut ini.



Gambar 3.1 Feature Selection & Feature Extraction Model

3.5.1 Dimensionality Reduction

Ketika dimensi data terlalu tinggi sehingga membutuhkan kompleksitas komputasi terhadap waktu dan ruang yang besar pula, atau ketika terdapat atribut-atribut yang saling berkorelasi (atribut yang tidak perlu untuk ikut dianalisis karena berkorelasi dengan atribut lainnya), maka pendekatan *dimensionality reduction* dapat diterapkan untuk meningkatkan efisiensi dan kinerja model. *Dimensionality reduction* yang dimaksudkan dalam hal ini adalah pada data yang terstruktur, karena metode-metode *dimensionality reduction* dapat pula digunakan untuk *feature extraction*. *Dimensionality reduction* bekerja dengan cara menemukan karakteristik data melalui pemetaan data (transformasi data) dari dimensi semula ke dimensi yang lebih rendah [20]. Dengan demikian, *dimensionality reduction* bertujuan untuk membuang komponen/atribut yang kurang berpengaruh atau sama sekali tidak berpengaruh. Dengan melakukan pemetaan data ke dimensi yang lebih rendah dalam memilih atribut-atribut yang berpengaruh untuk digunakan oleh model, maka metode-metode *dimensionality reduction* dapat pula digunakan untuk *feature extraction*.

Dimensionality reduction memang mirip dengan *feature selection*. Seringkali kedua pendekatan ini memang cukup membingungkan, karena keduanya mereduksi dimensi data dengan alasan yang sama. Bedanya, *feature selection* memilih atribut-atribut yang relevan terhadap model berdasarkan kinerja dari model atas atribut-atribut dan melakukan pembobotan terhadap setiap atribut [26] (perhatikan Gambar 3.1). Atribut yang tidak meningkatkan kinerja model yang dibuang. Sementara *dimensionality reduction* tidak mempertimbangkan kinerja model (dilakukan sebelum model dievaluasi), dan tidak melakukan pembobotan terhadap setiap atribut [26]. *Feature selection* biasanya digunakan untuk pekerjaan *supervised learning* (klasifikasi dan regresi/estimasi) [26], sedangkan *dimensionality reduction* dapat digunakan pada pekerjaan *supervised learning* maupun *unsupervised learning*.

Beberapa metode yang dapat digunakan untuk *dimensionality reduction*, antara lain *Principal Component Analysis* (PCA), *Singular Value Decomposition* (SVD), *Independent Component Analysis* (ICA), *Generalized Hebbian Algorithm* (GHA), dan *Self Organizing Map* (SOM) [26]. PCA dan SVD merupakan metode yang klasik dan umum digunakan untuk *dimensionality reduction*, sehingga menjadi metode yang akan dibahas lebih dalam pada sub pokok bahasan ini daripada metode ICA, GHA, dan SOM.

PCA mereduksi dimensi data menggunakan matriks *covariance*. PCA merupakan prosedur matematika yang menggunakan transformasi ortogonal untuk mengubah seperangkat pengamatan dari atribut yang mungkin berkorelasi menjadi satu set nilai atribut tidak berkorelasi yang disebut komponen utama, yang mana jumlah komponen utama kurang dari atau sama dengan jumlah atribut asli [26]. Harap diperhatikan bahwa PCA sensitif terhadap *relative scaling* dari atribut asli [26]. Selain itu, PCA merupakan metode yang agak *arbitrary*, misalnya hasil berbeda akan diperoleh antara penggunaan atribut *Fahrenheit* dan *Celsius* [26].

PCA memerlukan masukan data yang mempunyai sifat *new zero-mean* pada setiap atributnya. Sifat *zero-mean* setiap atribut bisa didapatkan dengan mengurangkan semua nilai dengan rata-ratanya (17).

$$x_{ij} = x_{ij} - \bar{x}_j \quad (17)$$

Selanjutnya dilakukan perhitungan matriks kovarian (18), yang mana X^T adalah matriks transpos dari data X . Formula yang digunakan adalah *dot-product* pada setiap atribut.

$$C_x = \frac{1}{M} X^T X \quad (18)$$

PCA bertujuan untuk meminimalkan redundansi yang diukur dengan nilai jarak dari kovarian dan memaksimalkan nilai keluaran pemetaan yang diukur dengan varian [20]. Jika Y adalah matriks data hasil pemetaan, C_Y adalah matriks kovarian dari Y , maka cara yang umum digunakan untuk mendapatkan C_Y adalah dengan *eigenvalue* dan *eigenvector*. Nilai *eigenvalue* dan *eigenvector* dari matriks data X berturut-turut adalah nilai skala λ dan vektor u yang memenuhi Persamaan (19).

$$Xu = \lambda u \quad (19)$$

Dengan mencari matriks ortonormal P , yang mana $Y = PX$ dan $C_Y = \frac{1}{M} YY^T$ adalah matriks diagonal, dan kolom dari P adalah komponen utama (*principal component*) dari X , Persamaan C_Y bisa dijabarkan sebagai berikut:

$$C_Y = \frac{1}{M} YY^T = \frac{1}{M} (PX)(PX)^T = \frac{1}{M} PXX^TP^T = P \left(\frac{1}{M} XX^T \right) P^T$$

Dengan mensubstitusikan Persamaan (18), maka diperoleh matriks C_Y (20), yang mana setiap baris matriks P adalah eigenvector C_X .

$$C_Y = PC_X P^T \quad (20)$$

Jika PCA menggunakan *eigenvalue* dan *eigenvector* untuk memperoleh solusi, SVD menggunakan dekomposisi nilai tunggal untuk memperoleh solusi. Matriks data X dapat dibentuk dengan Persamaan (21).

$$A = \sum_{i=1}^{\text{rank}(A)} \sigma_i U_i V_i^T \quad (21)$$

σ_i adalah nilai *singular* ke- i dari A (nilai ke- i pada diagonal Σ), U_i adalah vektor *singular* kiri dari A (kolom ke- i dari U), dan V_i adalah vektor singular kanan ke- i dari A (kolom ke- i dari V).

SVD mereduksi dimensi data dengan menghilangkan atribut yang tidak perlu yang secara linier tergantung pada sudut pandang Aljabar Linier [26]. Misalnya terdapat atribut suhu air dan atribut sifat air (padat, cair, atau gas), maka SVD dengan mudah menganalisis bahwa atribut sifat air bergantung pada atribut suhu air, sehingga atribut kedua tidak penting untuk digunakan.

Salah satu metode *dimensionality reduction* yang dapat menangkap struktur penting pada data, termasuk jika untuk *feature extraction*, yaitu ICA. Metode ini

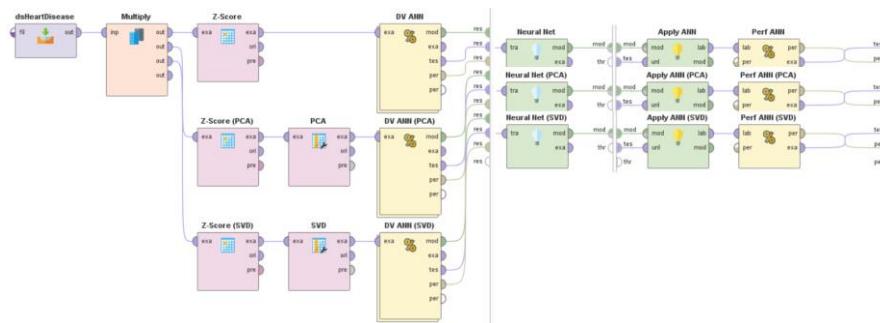
menganalisis atribut-atribut secara linear dan merubahnya menjadi komponen yang secara maksimal independen satu dengan lainnya. ICA digunakan untuk mengungkapkan faktor-faktor tersembunyi yang mendasari data. ICA mirip dengan PCA. Namun ICA merupakan teknik yang jauh lebih kuat [26]. ICA mengimplementasikan algoritma FastICA dari A. Hyvärinen dan E. Oja [26]. Algoritma FastICA lebih baik daripada ANN, karena bersifat paralel, terdistribusi, komputasi lebih sederhana, dan membutuhkan lebih sedikit ruang memori [26].

GHA merupakan algoritma ANN (*Feedforward Network*) linier untuk *unsupervised learning* [26]. Dari sudut pandang komputasi, dapat digunakan untuk menyelesaikan masalah nilai *eigen* dengan metode iteratif yang tidak perlu menghitung matriks kovarian secara langsung. Metode ini berguna ketika data berisi banyak atribut (ratusan atau bahkan ribuan) [26].

Selain GHA, SOM juga merupakan algoritma ANN untuk *unsupervised learning* [26]. Metode ini menghasilkan representasi dimensi rendah (biasanya dua dimensi). SOM berguna untuk memvisualisasikan data berdimensi rendah dari data berdimensi tinggi, mirip dengan penskalaan multidimensi. Algoritma ini pertama kali digambarkan sebagai ANN oleh Teuvo Kohonen, dan kadang-kadang disebut *Kohonen map* [26]. Seperti kebanyakan ANN, SOM beroperasi dalam dua mode: pelatihan dan pemetaan. Pelatihan membuat peta menggunakan contoh input, sedangkan pemetaan secara otomatis mengklasifikasikan vektor input baru [26].

Contoh 3.10 Dimensionality Reduction: PCA & SVD (Rapidminer)

Dataset : dsHeartDiseaseCleveland – Class: {0,1} (terlampir)
Dimensionality reduction : PCA (variance threshold = 0,95)
 SVD (percentage threshold = 0,95)
Classifier : ANN (1 hidden layer dengan 10 neurons)
Data normalization : Z-Score
Data validation : 10-Fold Cross Validation



Berikut ini adalah hasil evaluasinya menggunakan *Confusion Matrix*. Hasil ini menunjukkan bahwa penerapan pendekatan *dimensionality reduction* menggunakan metode PCA dan SVD dapat meningkatkan kinerja model. Dalam kasus ini, metode SVD yang memberikan kinerja terbaik, dengan akurasi 82,84%, mampu meningkatkan kinerja akurasi ANN sebesar 1,98%.

ANN + Z-Score			
accuracy: 80,89% +/- 9,33% (micro average: 80,86%)			
	true 0	true 1	precision
prediction 0	137	31	81,55%
prediction 1	27	108	80,00%
recall	83,54%	77,70%	
ANN + Z-Score + PCA			
accuracy: 81,86% +/- 7,27% (micro average: 81,85%)			
	true 0	true 1	precision
prediction 0	143	34	80,79%
prediction 1	21	105	83,33%
recall	87,20%	75,54%	
ANN + Z-Score + SVD			
accuracy: 82,86% +/- 5,47% (micro average: 82,84%)			
	true 0	true 1	precision
prediction 0	144	32	81,82%
prediction 1	20	107	84,25%
recall	87,80%	76,98%	

Berikut ini merupakan data hasil prediksi yang menunjukkan bahwa jumlah variabel input sebelumnya adalah 13, berubah menjadi 12, berarti SVD membuang 1 variabel yang dianggap tidak berpengaruh.

ID	Class	Prediction	SVD1	...	SVD12
6	0	0	-0,072		-0,071
...
297	1	1	0,087	...	0,169

3.5.2 Feature Selection

Seperti penjelasan sebelumnya bahwa *feature selection* memilih atribut-atribut yang relevan terhadap model berdasarkan kinerja dari model atas atribut-atribut dan melakukan pembobotan terhadap setiap atribut [26]. Atribut yang tidak meningkatkan kinerja model yang dibuang. Hal ini yang membedakannya dengan *dimensionality reduction*. *Feature selection* biasanya digunakan untuk pekerjaan *supervised learning* (klasifikasi dan regresi/estimasi) [26].

Beberapa metode yang dapat digunakan untuk *feature selection*, antara lain *Forward Selection*, *Backward Elimination*, *Bruto Force*, *Weight-Guided*, dan *Evolutionary (Genetic Algorithm)* [26]. *Forward Selection*, *Backward Elimination*, dan *Genetic Algorithm* (GA) merupakan metode yang cukup populer digunakan untuk *feature selection*. Ketiga metode tersebut memiliki sub proses karena bekerja dengan cara memperhitungkan kinerja model dalam melakukan transformasi data dan memberikan atribut-atribut yang dapat meningkatkan kinerja model. Dengan demikian, metode-metode tersebut membutuhkan memori komputasi yang besar. Dalam sub pokok bahasan ini, metode-metode tersebut akan dibahas.

Dalam setiap sub proses *Forward Selection*, vektor kinerja dari model akan selalu dikembalikan. Dimulai dengan pemilihan atribut yang kosong dan di setiap iterasi sub proses, atribut ditambahkan. Hanya atribut yang memberikan peningkatan kinerja tertinggi ditambahkan ke seleksi (dipilih). Ada beberapa kriteria/pilihan untuk menghentikan sub proses *Forward Selection*, yaitu iterasi terus berjalan selama ada peningkatan kinerja, iterasi terus berjalan selama

peningkatan kinerja memenuhi nilai yang ditentukan, dan iterasi akan berhenti jika peningkatan kinerja tidak signifikan [26]. Terdapat pula parameter untuk iterasi spekulatif yang dapat menentukan berapa banyak iterasi dilakukan setelah kriteria berhenti terpenuhi [26]. Jika kinerja meningkat lagi selama iterasi spekulatif, pemilihan akan dilanjutkan. Namun jika tidak, maka semua atribut tambahan yang dipilih akan dihapus, seolah-olah tidak ada iterasi spekulatif yang telah dieksekusi. Hal ini agar tidak terjebak dalam *local optima* [26].

Berikut ini merupakan algoritma *Forward Selection* untuk *feature selection*, yang mana fungsi f pada Persamaan (22) dimaksimalkan untuk menemukan fitur-fitur (atribut-atribut) k dalam set fitur F [27].

$$x_j = \underset{x_j \in F}{\operatorname{argmax}} f(x_j, y, F \setminus \{x_j\}) \quad (22)$$

Input:

feature set F , an objective function f , k features to select, initialize an empty set $f \emptyset$

1. Maximize the objective function (22);
2. Update relevant feature set such that $F \emptyset \leftarrow F \emptyset \cup x_j$;
3. Remove relevant feature from the original set $F \leftarrow F \setminus x_j$;
4. Repeat until $|F \emptyset| = k$;

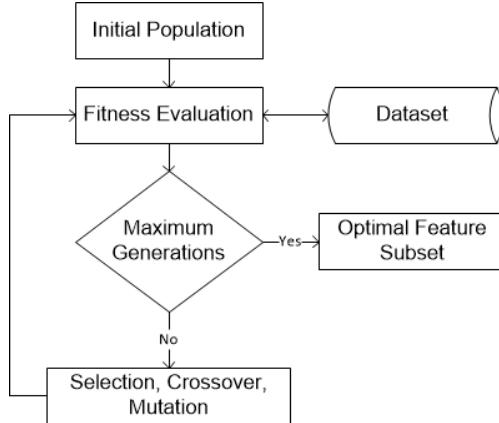
Jika *Forward Selection* dimulai dengan atribut yang kosong, sebaliknya *Bakward Selection* dimulai dengan atribut yang lengkap, yang mana disetiap iterasi sub prosesnya akan menghapus atribut yang memberikan paling sedikit penurunan kinerja [26]. Sebenarnya kedua metode ini (*Forward Selection* dan *Bakward Elimination*) dapat diintegrasikan untuk melakukan *feature selection* yang lebih optimal. Hanya saja akan membutuhkan kompleksitas komputasi yang tentunya sangat besar. Dalam Rapidminer, pengembangan ini diberi nama *Optimize Selection*.

GA merupakan algoritma pencarian heuristik yang meniru proses evolusi alami (*inheritance, mutation, selection, & crossover*). GA sangat populer digunakan untuk menghasilkan solusi dari masalah-masalah optimasi dan pencarian, sehingga dapat pula digunakan untuk *feature selection*. Proses *mutation* dalam GA untuk *feature selection* berarti menghidupkan dan mematikan atribut. *Crossover* berarti menukar atribut yang digunakan. Sedangkan *selection* dilakukan berdasarkan pendekatan *selection*.

Berikut ini prosedur kerja GA untuk *feature selection* (Rapidminer) [26]:

1. Inisialisasi populasi awal yang terdiri dari p individu. Setiap atribut diaktifkan dengan probabilitas p_i . Nilai p dan p_i dapat disesuaikan dengan ukuran populasi dan p menginisialisasi parameter masing-masing.
2. Lakukan *mutation* untuk semua individu dalam populasi. Misalnya, atur atribut yang digunakan untuk tidak digunakan dengan probabilitas p_m dan sebaliknya. Probabilitas p_m dapat disesuaikan dengan parameter *mutation p*.
3. Pilih dua individu dari populasi dan lakukan *crossover* dengan probabilitas p_c . Probabilitas p_c dapat disesuaikan dengan parameter *crossover p*. Pendekatan *crossover* dapat dipilih melalui parameter *crossover type*.

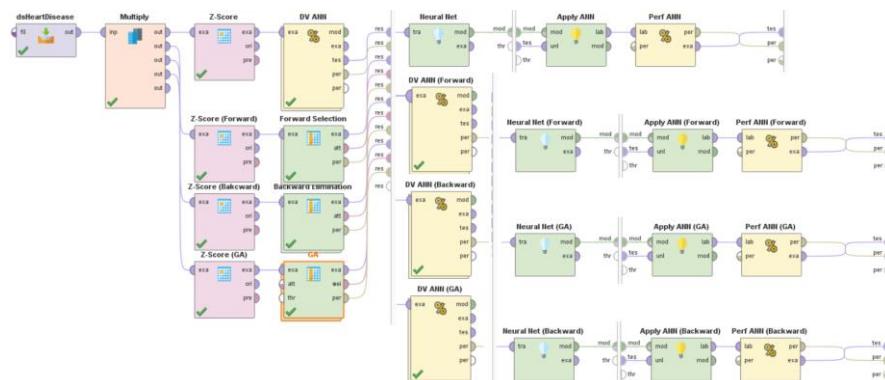
4. Lakukan *selection*, petakan semua individu sesuai dengan nilai *fitness* individu tersebut dan ambil individu p secara acak sesuai dengan probabilitasnya, yang mana p adalah ukuran populasi. Selama nilai *fitness* membaik, lanjutkan ke langkah nomor 2.



Gambar 3.2 Genetic Algorithm Model

Contoh 3.11 Feature Selection: Forward, Backward, & GA (Rapidminer)

<i>Dataset</i>	: <i>dsHeartDiseaseCleveland – Class: {0, 1}</i> (terlampir)
<i>Feature Selection</i>	: <i>Forward Selection (max attributes = 10)</i> <i>Backward Elimination (max elimination = 10)</i> <i>Genetic Algorithm (min attributes = 1, population = 5)</i>
<i>Classifier</i>	: ANN (1 hidden layer dengan 10 neurons)
<i>Data normalization</i>	: Z-Score
<i>Data validation</i>	: 10-Fold Cross Validation



Berikut ini adalah hasil evaluasinya menggunakan *Confusion Matrix*. Hasil ini menunjukkan bahwa penerapan pendekatan *feature selection* menggunakan metode *Forward*, *Backward*, dan *Evolutionary* dapat meningkatkan kinerja model. Dalam kasus ini, metode *Backward Elimination* yang memberikan kinerja terbaik, dengan akurasi 84,49%, mampu meningkatkan kinerja akurasi ANN sebesar 3,63%.

ANN + Z-Score			
accuracy: 80,90% +/- 6,80% (micro average: 80,86%)			
	true 0	true 1	precision
prediction 0	135	29	82,32%
prediction 1	29	110	79,14%
recall	82,32%	79,14%	
ANN + Z-Score + Forward Selection			
accuracy: 83,90% +/- 9,32% (micro average: 83,83%)			
	true 0	true 1	precision
prediction 0	148	33	81,77%
prediction 1	16	106	86,89%
recall	90,24%	76,26%	
ANN + Z-Score + Backward Elimination			
accuracy: 84,53% +/- 6,86% (micro average: 84,49%)			
	true 0	true 1	precision
prediction 0	143	26	84,62%
prediction 1	21	113	84,33%
recall	87,20%	81,29%	
ANN + Z-Score + Genetic Algorithm			
accuracy: 83,84% +/- 6,27% (micro average: 83,83%)			
	true 0	true 1	precision
prediction 0	145	30	82,86%
prediction 1	19	109	85,16%
recall	88,41%	78,42%	

Hasil *feature selection* dalam kasus ini menunjukkan bahwa *Forward Selection* membuang variabel ke-1 hingga ke-8 dan ke-10. *Backward Elimination* membuang variabel ke-4 dan ke-10. *Genetic Algorithm* membuang variabel ke ke-2, ke-4, dan ke-10. Ketiga metode tersebut membuang variabel ke-10.

3.5.3 Attribute Weighting

Attribute weighting sebenarnya merupakan salah satu bagian dari *feature selection*, karena dalam proses *feature selection*, dilakukan pula *attribute weighting*. Ketiga istilah ini, *dimensionality reduction*, *feature selection*, dan *attribute weighting* memang cukup membingungkan. Sebenarnya boleh disatukan dalam satu istilah saja, yaitu *feature selection*, karena alasan dan tujuannya sama walaupun prosesnya sedikit berbeda (perhatikan Gambar 3.1).

Seperti penjelasan sebelumnya, *dimensionality reduction* tidak melakukan *attribute weighting* berdasarkan kinerja dari model. Sedangkan *feature selection* melakukannya. Jika begitu, apa bedanya *attribute weighting* dengan *feature selection*? Terkadang optimalisasi terhadap model berdasarkan atribut-atribut data dapat pula dilakukan tanpa mereduksi dimensi data, inilah yang dilakukan *attribute weighting* dalam hal ini. Dengan demikian, dalam hal ini *attribute weighting* tidak mereduksi dimensi data dan inilah yang membedakannya dengan *feature selection* dan *dimensionality reduction*.

Metode-metode yang digunakan untuk *feature selection* dapat pula digunakan untuk *attribute weighting*, antara lain *Forward Selection*, *Backward Elimination*, *Bruto Force*, *Weight-Guided*, *Evolutionary (Genetic Algorithm)*, dan *Particle Swarm Optimization (PSO)* [26]. Salah satu metode *attribute weighting* yang tidak umum namun telah terbukti dapat meningkatkan kinerja dari model *machine*

learning yang digunakan yaitu *Absolute Correlation Coefficient* (ACC) [28]. Oleh karena itu, metode ACC untuk *attribute weighting* yang dibahas dalam sub pokok bahasan ini dan akan dilanjutkan pada pokok bahasan pengembangan algoritma *Naïve Bayes*.

ACC (25) dapat menentukan kekuatan antar atribut dan bekerja pada atribut bertipe numerik [28]. Metode ini menggunakan pendekatan *mean* (11) dan *standard deviation* (12) [28]. Dasarnya adalah *correlation coefficient* (22) yang merupakan penentu kekuatan hubungan antara dua variabel/atribut numerik [29].

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2} \sqrt{\sum(y - \bar{y})^2}} \quad (22)$$

\bar{x} adalah nilai *mean* (11) dari x , dan \bar{y} adalah nilai *mean* (11) dari y .

Beberapa penelitian telah menggunakan/mengembangkan *correlation coefficient*. Guyon *et al.*, mengusulkan metode *weighting* [30] dengan menggunakan koefisien (23) dari penelitian yang dilakukan Golub [31], didefinisikan sebagai berikut.

$$w_i = \frac{(\mu_i(+) - \mu_i(-))}{(\sigma_i(+) + \sigma_i(-))} \quad (23)$$

μ_i adalah *mean* (11) dan σ_i adalah *standard deviation* (12) dari atribut ke- i untuk *class* (+) dan *class* (-) masing-masing. w_i dengan nilai positif yang besar menunjukkan kekuatan hubungan yang kuat dengan *class* (+), sebaliknya w_i dengan nilai negatif yang besar menunjukkan kekuatan hubungan yang kuat dengan *class* (-).

Zhang, meningkatkan kinerja *Weighted Naïve Bayes* menggunakan *correlation coefficient* pula [32]. Sementara itu, Pavlidis, *et al.*, mengusulkan *associated coefficients* (24) yang didefinisikan sebagai berikut [33].

$$w_i = \frac{(\mu_i(+) - \mu_i(-))^2}{(\sigma_i(+)^2 + \sigma_i(-)^2)} \quad (24)$$

Furey, *et al.*, menggunakan nilai *absolute* dari w_i (23) sebagai metode untuk *feature selection* pada SVM dalam klasifikasi kanker [34]. Begitupun Asmono, Wahono, & Syukur, menggunakan nilai *absolute* dari w_i (23) sebagai metode *attribute weighting* pada NB (*Absolute Correlation – Weighted Naïve Bayes*) dalam prediksi cacat *software* [28]. Metode *weighting* tersebut kemudian dinamakan *Absolute Correlation Coefficient* (25), didefinisikan sebagai berikut.

$$w_i = \frac{|(\mu_{ij} - \mu_{ij})|}{|(\sigma_{ij} - \sigma_{ij})|} \quad (25)$$

Keterangan:

w_i : *weight* dari atribut ke- i .

μ_{ij} : nilai *mean* (11) dari atribut ke- i pada *class* j .

μ_{ij} : nilai *mean* (11) dari atribut ke- i pada *class* non j .

σ_{ij} : nilai *standard deviation* (12) dari atribut ke- i pada *class* j .

σ_{ij} : nilai *standard deviation* (12) dari atribut ke- i pada *class* non j .

Contoh 3.12 Attribute Weighting: Absolute Correlation Coefficient (Manual)

	X1	X2	Class
Data	11	9,9	1
	10	5,3	1
	11	1,4	1
	12	2,9	2
	10	7,3	2
	10	6,1	2
	10	9,4	2
	11	9,9	3
	10	1,3	3
	10	4,2	3
$\mu(1)$	10,67	5,53	1
$\mu(2)$	10,50	6,43	2
$\mu(3)$	10,33	5,13	3
$\sigma(1)$	0,58	4,25	1
$\sigma(2)$	1	2,72	2
$\sigma(3)$	0,58	4,38	3
w	1,6874	0,1899	

Keterangan:

$$\mu(X1, \text{Class } 1) = (11 + 10 + 11) / 3 = 10,67$$

$$\mu(X2, \text{Class } 1) = (9,9 + 5,3 + 1,4) / 3 = 5,53$$

$$\mu(X1, \text{Class } 2) = (12 + 10 + 10 + 10) / 4 = 10,50$$

$$\mu(X2, \text{Class } 2) = (2,9 + 7,3 + 6,1 + 9,4) / 4 = 6,43$$

$$\mu(X1, \text{Class } 3) = (11 + 10 + 10) / 3 = 10,50$$

$$\mu(X2, \text{Class } 3) = (9,9 + 1,3 + 4,2) / 3 = 5,13$$

$$\sigma(X1, \text{Class } 1) = ((11-10,67)^2 + ((10-10,67)^2) + ((11-10,67)^2) / 3 = 0,67 \\ (0,67 / (3-1)) ^ 0,5 = 0,58$$

$$\sigma(X2, \text{Class } 1) = ((9,9-5,53)^2 + ((5,3-5,53)^2) + ((1,4-5,53)^2) / 3 = 36,21 \\ (36,21 / (3-1)) ^ 0,5 = 4,25$$

$$\sigma(X1, \text{Class } 2) = ((12-10,50)^2 + ((10-10,50)^2) + ((10-10,50)^2) + ((10-10,50)^2) / 3 = 3 \\ (3 / (4-1)) ^ 0,5 = 1$$

$$\sigma(X2, \text{Class } 2) = ((2,9-6,43)^2 + ((7,3-6,43)^2) + ((6,1-6,43)^2) + ((9,4-6,43)^2) / 3 = 22,15 \\ (22,15 / (4-1)) ^ 0,5 = 2,72$$

$$\sigma(X1, \text{Class } 3) = ((11-10,33)^2 + ((10-10,33)^2) + ((10-10,33)^2) / 3 = 0,67 \\ (0,67 / (3-1)) ^ 0,5 = 0,58$$

$$\sigma(X2, \text{Class } 3) = ((9,9-5,13)^2 + ((1,3-5,13)^2) + ((4,2-5,13)^2) / 3 = 38,29 \\ (38,29 / (3-1)) ^ 0,5 = 4,38$$

$$w(X1) = (10,67 - 10,50 - 10,33) / (0,58 + 1 + 0,58) = 1,6874$$

$$w(X1) = (5,53 - 6,43 - 5,13) / (4,25 + 2,72 + 4,38) = 0,1899$$

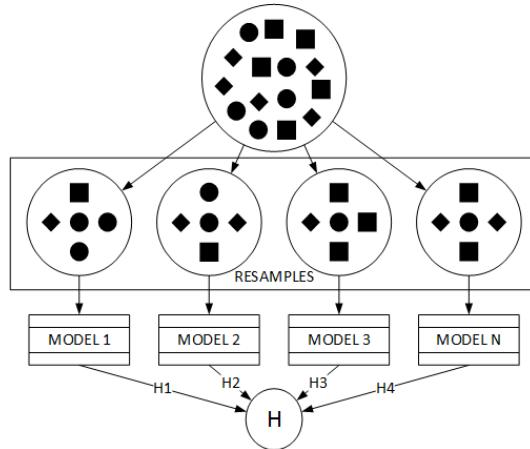
Dengan demikian nilai bobot atribut X1 = 1,6874 dan X2 = 0,1899.

3.5.4 Feature Extraction

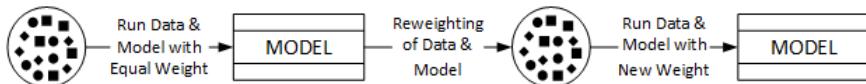
Seperti penjelasan sebelumnya, *feature selection (dimensionality reduction, feature selection, attribute weighting)* dilakukan pada data yang terstruktur, sedangkan pada data yang tidak terstruktur, dilakukan *feature extraction* dengan tujuan untuk memperoleh data yang terstruktur (perhatikan Gambar 3.1). Pendekatan yang dapat digunakan untuk *feature extraction* yaitu *text processing, image processing*, dll. Perlu diketahui bahwa metode-metode *dimensionality reduction* dapat pula digunakan untuk *feature extraction*.

3.6 Unbalanced Class Reduction

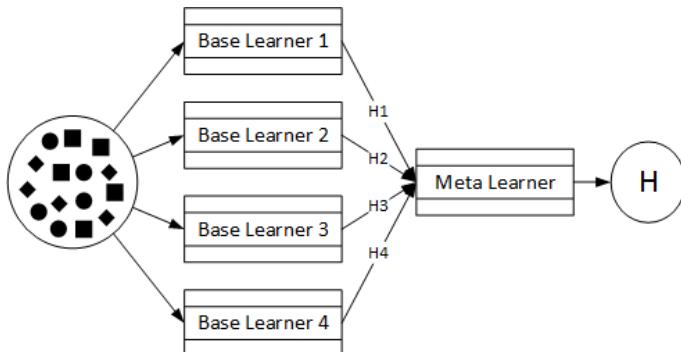
Unbalanced class merupakan suatu keadaan ketika data mengalami ketidakseimbangan antar *class* yang terlalu besar. Misalnya suatu data yang berjumlah 1000 *record* terdiri dari 2 label *class*, yaitu *class 0* dan *class 1*, yang mana jumlah data pada *class 0* = 150 (15%), sementara *class 1* = 850 (85%). Masalah ini dapat menurunkan kinerja suatu model metode *machine learning* [35]. Pendekatan *ensemble* merupakan salah satu cara yang populer untuk menangani *unbalanced class* [36], [37], [38] dan terbukti mampu meningkatkan kinerja suatu model *machine learning* [19], [38]. Beberapa metode *ensemble* yang populer, antara lain *Bagging*, *Stacking*, *Adaboost*, dan *Weighted Vote*.



Gambar 3.3 Bagging Model



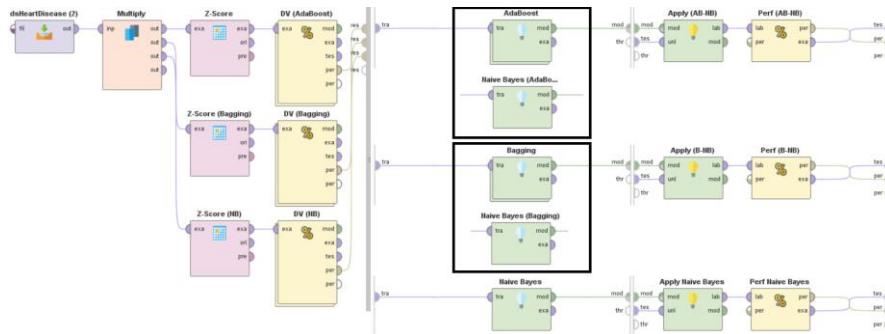
Gambar 3.4 Boosting Model



Gambar 3.5 Stacking Model

Contoh 3.13 Unbalanced Class: AdaBoost & Bagging (Rapidmider)

- Dataset* : *dsHeartDiseaseCleveland* – *Class: {0,1,2,3,4}* (terlampir)
Ensemble : *AdaBoost (iteration=10)*
Bagging (iteration=10, sample ratio=0,1)
Classifier : *Naïve Bayes (Gaussian untuk probabilitas numerik)*
Data normalization : *Z-Score*
Data validation : *10-Fold Cross Validation*



Berikut ini adalah hasil evaluasinya menggunakan *Confusion Matrix*. Hasil ini menunjukkan bahwa penerapan metode *ensemble* (*Bagging & AdaBoost*) untuk *unbalanced class reduction* dapat meningkatkan kinerja model. Dalam kasus ini, metode *Bagging* yang memberikan kinerja terbaik, dengan akurasi 58,75%, mampu meningkatkan kinerja akurasi *Naïve Bayes* sebesar 2,97%.

Naïve Bayes + Z-Score						
accuracy: 55.85% +/- 9.04% (micro average: 55.78%)						
	true 0	true 2	true 1	true 3	true 4	precision
pred. 0	140	2	26	2	0	82,35
pred. 2	5	8	11	13	3	20,00
pred. 1	15	14	13	9	5	23,21
pred. 3	1	11	4	8	5	27,59
pred. 4	3	1	1	3	0	0,00
recall	85,37	22,22	23,64	22,86	0,00	
Naïve Bayes + Z-Score + Bagging						
accuracy: 58.81% +/- 6.15% (micro average: 58.75%)						
	true 0	true 2	true 1	true 3	true 4	precision
pred. 0	154	11	33	4	1	75,86
pred. 2	4	9	7	12	3	25,71
pred. 1	5	12	11	15	4	23,40
pred. 3	1	4	4	4	5	22,22
pred. 4	0	0	0	0	0	0,00
recall	93,90	25,00	20,00	11,43	0,00	
Naïve Bayes + Z-Score + AdaBoost						
accuracy: 56.78% +/- 8.64% (micro average: 56.77%)						
	true 0	true 2	true 1	true 3	true 4	precision
pred. 0	141	2	24	2	0	83,43
pred. 2	5	8	11	15	4	18,60
pred. 1	15	11	14	7	5	26,92
pred. 3	1	11	4	9	4	31,03
pred. 4	2	4	2	2	0	0,00
recall	85,98	22,22	25,45	25,71	0,00	

Salah satu contoh penerapan metode *Weighted Vote* pada metode-metode *machine learning* *Decision Tree*, *Support Vectore Machine*, *Naïve Bayes*, dan *Memory Based Learner*, yaitu pada prediksi Kanker Payudara, menunjukkan kinerja akurasi *Weighted Vote* secara signifikan lebih unggul dari pada metode-metode *machine learning* tersebut [19]. Oleh karena itu, sub pokok bahasan ini akan membahas pendekatan *ensemble* menggunakan metode *Weighted Vote* secara lebih mendalam. Algoritmanya *Weighted Vote* adalah sebagai berikut:

1. Hitung acc_i , yaitu akurasi rata-rata menggunakan Persamaan (11) berdasarkan akurasi dari tiap-tiap K dalam *K-Fold Cros Validation* untuk metode *machine learning* ke- $i = (1, 2, \dots, n)$, yang mana n adalah banyaknya metode *machine learning* yang digunakan.
2. Hitung $accmax_i$, yaitu akurasi maksimum ($\max(accuracy) = \{accuracy_1, accuracy_2, \dots, accuracy_k\}$) berdasarkan akurasi dari tiap-tiap K dalam *K-Fold Cros Validation* untuk metode *machine learning* ke- i .
3. Hitung $accmin_i$, yaitu akurasi minimum ($\min(accuracy) = \{accuracy_1, accuracy_2, \dots, accuracy_k\}$) berdasarkan akurasi dari tiap-tiap K dalam *K-Fold Cros Validation* untuk metode *machine learning* ke- i .
4. Hitung M_i , yaitu hasil normalisasi acc_i , $accmax_i$, dan $accmin_i$ menggunakan metode *Min-Max Normalization* (9) dalam interval $[0,1; 1]$ untuk metode *machine learning* ke- i .
5. Hitung m_i , yaitu selisih antara 1 dengan M_i ($1 - M_i$) untuk metode *machine learning* ke- i .
6. Hitung w_i menggunakan Persamaan (26) berikut ini untuk metode *machine learning* ke- i .

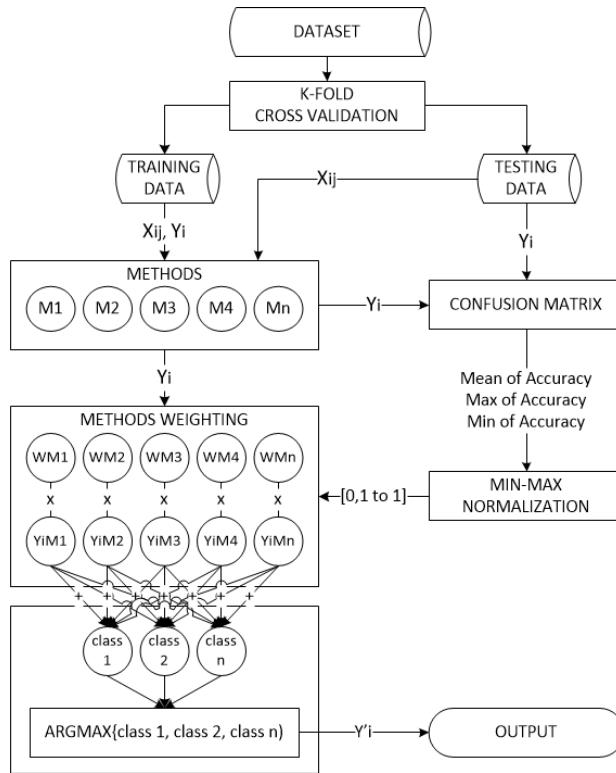
$$w_i = \frac{m_i}{\sum_{i=1}^n m_i} \quad (26)$$

7. Hitung y_{ic} menggunakan Persamaan (27) berikut ini berdasarkan benar atau salahnya hasil prediksi/klasifikasi metode *machine learning* ke- i untuk label *class* ke- $c = (1, 2, \dots, k)$, yang mana k adalah banyaknya label *class*.

$$y_{ic} = \begin{cases} 0; & \text{if classification is false} \\ 1; & \text{if classification is true} \end{cases} \quad (27)$$

8. Dapatkan label *class* keputusan klasifikasi y' menggunakan Persamaan (28) berikut ini.

$$y' = \underset{y_c}{\operatorname{argmax}} \sum_{i=1}^n y_{ic} w_i \quad (28)$$



Gambar 3.6 Weighted Vote Model

Contoh 3.14 Weighted Vote untuk Klasifikasi (Manual)

	K=10	SVM	ANN	KNN	NB	DT	Jml
10-Fold Cross Validation	1	75	64	99	72	71	
	2	45	77	43	91	89	
	3	90	49	88	86	63	
	4	66	74	76	77	79	
	5	46	61	52	62	73	
	6	81	43	57	54	79	
	7	86	46	83	91	94	
	8	96	78	46	94	51	
	9	57	71	63	76	66	
	10	78	52	60	90	71	
		Mean	72	61.5	66.7	79.3	73.6
		Max	96	78	99	94	94
		Min	45	43	43	54	51
		Norm (0:1:1)	0.53	0.53	0.42	0.63	0.53
			0.47	0.47	0.58	0.37	0.47
		Weight	0.20	0.20	0.24	0.16	0.20
WV Pred	Class 0	0	0	1	1	1	
	Class 1	1	1	0	0	0	
WV	Class 0	0.00	0.00	0.24	0.16	0.20	0.60
	Class 1	0.20	0.20	0.00	0.00	0.00	0.40
					Jml		1.00

Class 0 = 0,60 > Class 1 = 0,40, maka keputusan klasifikasi adalah Class 0.

3.7 Data Validation

Dalam melakukan *data validation*, sampel harus mewakili populasi. Data yang digunakan dalam pengujian model harusnya dapat mewakili populasi yang ada. Terkadang sulit untuk memperoleh sampel yang benar-benar valid, maka metode-metode *data validation*, seperti *Holdout*, *Leave One Out*, dan *K-Fold Cross Validation* dapat digunakan.

Strategi pada *data validation* bukan hanya untuk sekedar memperoleh akurasi yang tinggi dari model metode *machine learning* yang digunakan. *Data validation* seharusnya dapat menjamin bahwa semua data digunakan untuk pelatihan dan pengujian model [18]. Pada masalah klasifikasi, setiap label *class* mestinya dapat terwakili secara merata pada pelatihan model.

Metode *Holdout* merupakan strategi validasi data yang simpel. Data latih dan data uji dapat dibagi secara random dengan komposisi 50% -50%, 70% - 30%, atau 80% - 20% [18]. Namun metode ini tidak dapat menjamin bahwa sampel dapat mewakili populasi sehingga dapat mengakibatkan bias.

Metode *Leave One Out Cross Validation* merupakan strategi validasi data yang paling dapat menjamin data latih dan data uji digunakan secara merata, karena pada prinsipnya metode ini menggunakan seluruh data untuk pelatihan dan pengujian model secara bergantian pada setiap data. Metode ini menggunakan $n-1$ data untuk pelatihan dan 1 data untuk pengujian di setiap iterasinya hingga semua data telah digunakan untuk pengujian [18]. Dengan demikian, tentu saja metode ini memiliki kompleksitas komputasi yang tinggi dengan waktu proses yang lama. Jadi jika *dataset* berukuran kecil, maka metode ini efisien untuk digunakan.

Metode lainnya yang dapat menjamin data latih dan data uji terbagi secara merata adalah *K-Fold Cross Validation*. Metode ini membagi data secara merata sebanyak K , yang mana $K - 1$ data digunakan untuk pelatihan dan sisanya untuk pengujian secara bergantian di setiap iterasi K [18]. Oleh karena itu, metode ini juga dapat menjamin bahwa setiap data telah digunakan untuk pelatihan dan pengujian, namun dengan kompleksitas komptasi yang lebih rendah dan waktu proses yang lebih cepat dari pada *Leave One Out Cross Validation*. Jadi jika *dataset* berukuran besar, maka metode ini efisien untuk digunakan.

80% Training (80 Instances)	20% Testing (20 Instances)
e.g. 100 instances dengan partisi 80% - 20%: 80 instances untuk training dan 20 instances untuk testing yang dipartisi secara random dengan komposisi class yang merata. Tidak ada iterasi.	

Gambar 3.7 Holdout Validation

1	99 Instances for Training						1st Instance for Testing		
2	99 Instances for Training						2nd Instance for Testing		
...		
100	99 Instances for Training						Last Instance for Testing		

e.g. 100 instances: Instance pertama untuk testing di iterasi pertama, instance kedua untuk testing di iterasi kedua, dst hingga, instance terakhir (ke-100) untuk testing di iterasi terakhir (ke-100). Instances yang tidak digunakan untuk testing digunakan untuk training di setiap iterasi.

Gambar 3.8 Leave-One-Out Cross Validation

1	10% Testing	10% Training								
2	10% Training	10% Testing	10% Training							
3	10% Training	10% Training	10% Testing	10% Training						
4	10% Training	10% Training	10% Training	10% Testing	10% Training					
5	10% Training	10% Training	10% Training	10% Training	10% Testing	10% Training				
6	10% Training	10% Testing	10% Training	10% Training	10% Training	10% Training				
7	10% Training	10% Testing	10% Training	10% Training	10% Training					
8	10% Training	10% Testing	10% Training	10% Training						
9	10% Training	10% Testing	10% Training							
10	10% Training	10% Testing								

e.g. 100 instances dengan K=10 (10 iterasi dan 10% testing): 10 instances untuk testing dan 90 instances untuk training yang dipartisi secara random di setiap iterasi dengan komposisi class yang merata pula untuk setiap iterasi.

Gambar 3.9 K-Fold Cross Validation

Contoh 3.15 K-Fold Cross Validation (Matlab)

Dataset : *dsHeartDiseaseCleveland* – Class: {0,1,2,3,4} (terlampir)
Data validation : *10-Fold Cross Validation*

```
clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dataset = dsHeartDisease(:,2:15); %ID dan Output 0,1 tidak digunakan
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K); %kolom 14 = class
for i = 1:K
    %% Buat Data Latih dan Data Uji berdasarkan indeks dari K-Fold
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
    subDataLatihInput = dataset(latih,1:13); %input training
    subDataLatihOutput = dataset(latih,14); %output training
    subDataUjiInput = dataset(uji,1:13); % input testing
    subDataUjiOutput = dataset(uji,14); % output testing
end
```

3.8 Soal Latihan Pra Pengolahan Data

Unduh salah satu *dataset* di *UCI Machine Learning Repository* yang memiliki $\text{missing value} \geq 100$, *unbalanced class*, dan atribut ≥ 15 kemudian lakukan beberapa pra pengolahan data berikut ini:

1. Deskripsikan dalam bentuk tabel karakteristik (*instances*, *atribut*, *class*, dsb) *dataset* tersebut?
2. Buatlah suatu *function* menggunakan alat bantu yang anda kuasai (Matlab, Phyton, Netbeans, Visual Studio, dll) untuk *missing value replacement* menggunakan pendekatan *mean/mode* pada *dataset* tersebut!
3. Buatlah suatu *function* menggunakan alat bantu yang anda kuasai (Matlab, Phyton, Netbeans, Visual Studio, dll) untuk *data type transformation* (atribut kategorikal ke numerik menggunakan pendekatan *entropy* atau yang anda kuasai, sedangkan atribut numerik ke kategorikal menggunakan pendekatan *encoding*) pada *dataset* tersebut!
4. Dapatkan anda menentukan *outliers* pada *dataset* tersebut menggunakan metode k-NN atau metode lainnya yang anda kuasai? Jika anda tidak mampu melakukannya, maka buatlah suatu *function* menggunakan alat bantu yang anda kuasai (Matlab, Phyton, Netbeans, Visual Studio, dll) untuk *smoothing noisy data* menggunakan pendekatan *data normalization* (*Min-Max Normalization*, *Z-Score*, dan *Softmax*)!
5. Buatlah suatu model *feature selection* pada *dataset* tersebut menggunakan metode PCA, SVD, GA, *Forward Selection*, dan *Backward Selection*. Sedangkan untuk *data validation* gunakan metode *10-Fold Cross Validation*. Gunakan alat bantu Rapidminer atau yang anda lebih kuasai!
6. Buatlah suatu model *unbalanced class reduction* pada *dataset* tersebut menggunakan metode AdaBoost dan Bagging. Sedangkan untuk *data validation* gunakan metode *10-Fold Cross Validation*. Gunakan alat bantu Rapidminer atau yang anda lebih kuasai!

4. Evaluasi Model

No.	Materi	Tujuan Pembelajaran
1.	Apa itu Evaluasi Model	Anda mampu memahami dan menjelaskan pengertian dan tujuan evaluasi model.
2.	Kompleksitas Algoritma	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi kompleksitas suatu algoritma terhadap waktu dan ruang.
3.	Evaluasi Model Klasifikasi	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi suatu model klasifikasi menggunakan pendekatan <i>Confusion Matrix</i> .
4.	Interval Kepercayaan Akurasi	Anda mampu memahami, menjelaskan, dan mengukur interval kepercayaan akurasi dua model klasifikasi.
5.	Evaluasi Model Regresi	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi suatu model regresi/estimasi menggunakan pendekatan MSE, RMSE, SEE, PE, MAPE, dan PNSR.
6.	Evaluasi Model Klasterisasi	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi suatu model klasterisasi menggunakan pendekatan internal, eksternal, dan evaluasi aplikasi.
7.	Evaluasi Model Asosiasi	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi suatu model asosiasi menggunakan pendekatan <i>Lift Ratio</i> .
8.	Uji Korelasi Variabel	Anda mampu memahami, menjelaskan, dan mengukur/mengevaluasi korelasi antar variabel menggunakan pendekatan koefisien korelasi dan <i>T-Test</i> .

4.1 Apa itu Evaluasi Model?

Suatu model dari metode *machine learning* dapat diukur atau dievaluasi atau diuji kinerjanya menggunakan satu atau lebih metode evaluasi model. Terdapat beragam pendekatan dalam mengevaluasi model, antara lain pengukuran kompleksitas algoritma, akurasi, *error estimasi*, kepadatan data, komparasi data, tingkat kepercayaan akurasi, dsb. Tentunya tidak semua pendekatan tersebut digunakan untuk menguji kinerja suatu model, namun tergantung dari tujuan pemodelannya. Misalnya pada model klasifikasi biasanya menggunakan *Confusion Matrix*, model regresi/estimasi biasanya menggunakan *error estimasi*, pengembangan suatu algoritma dapat diuji melalui kompleksitas algoritmanya, dsb.

Biasanya, tingkat kompleksitas algoritma terhadap waktu berkorelasi terbalik dengan kompleksitas algoritma terhadap ruang maupun akurasi, estimasi *error*, dll. Logikanya, suatu algoritma biasanya dapat menyelesaikan persoalan dengan lebih cepat, namun dengan konswekuensi memori yang besar. Sebaliknya, dengan sedikit memori tapi lambat. Begitupun apabila akurasi yang tinggi, biasanya dengan konswekuensi waktu yang lambat dan bahkan dengan memori yang besar. Secara rinci, metode-metode untuk mengevaluasi model akan dibahas selanjutnya.

4.2 Kompleksitas Algoritma

Kompleksitas algoritma (*algorithm complexity*) mengukur kinerja suatu metode/algoritma berdasarkan kompleksitas terhadap waktu $O(n)$ dan kompleksitas terhadap ruang $S(n)$, yang mana n adalah ukuran masukan yang diproses oleh algoritma. Sebagai contoh, apabila suatu algoritma dapat melaksanakan suatu proses tertentu yang lebih cepat daripada algoritma yang lain, maka dapat dikatakan bahwa algoritma tersebut yang diterapkan pada suatu struktur data lebih efisien dalam hal waktu daripada algoritma lainnya. Apabila suatu algoritma dalam suatu struktur data membutuhkan memori yang lebih sedikit daripada algoritma lainnya, maka dapat dikatakan bahwa algoritma tersebut lebih efisien dalam hal ruang yang digunakan daripada algoritma lainnya.

Dengan demikian, idealnya tingkat efisiensi suatu algoritma yang baik diukur dari sisi waktu dan ruang pula. Namun pada kenyataanya, ada korelasi terbalik antara kompleksitas waktu dengan ruang. Biasanya, suatu persoalan dapat dipecahkan dengan lebih cepat namun dengan konswekuensi akan menggunakan banyak memori, atau dapat dipecahkan dengan menggunakan sedikit memori namun lebih lambat.

Notasi *Big O* dapat digunakan untuk mengukur kompleksitas suatu algoritma terhadap waktu [39]. Sedangkan $O(n)$ menyatakan jumlah tahap komputasi yang dilakukan untuk menjalankan suatu algoritma sebagai fungsi dari ukuran masukan n . Suatu algoritma memiliki kompleksitas $O(f(n))$, dibaca orde f terhadap n , jika waktu yang diperlukan oleh algoritma mengikuti laju fungsi $f(n)$ dengan kondisi nilai n yang besar.

Contoh 4.1 Kompleksitas Algoritma

```
Jumlah ← jumlah + 1 diulangi sebanyak n kali.  
For i ← 1 to n  
    Jumlah ← jumlah + 1  
End For
```

Karena perulangannya sebanyak n kali, maka kompleksitasnya = $O(n)$.

```
For i ← 1 To n  
    For j ← 1 To n  
        Jumlah ← jumlah + x(i, j);  
End For
```

Karena diproses sebanyak $n \times n$ kali, maka kompleksitasnya = $O(n^2)$.

```
For i ← 1 To m  
    For j ← 1 To n  
        Jumlah ← jumlah + 1  
End For
```

$O(mn)$.

```
For i ← 1 To n  
    For j ← 1 To i  
        Jumlah ← jumlah + 1  
End For
```

Sama saja dengan $O(n^2)$.

Big O merupakan piranti yang efektif untuk melakukan analisis algoritma (mengukur kompleksitas algoritma terhadap waktu), namun tentu saja belum mewakili efisiensi algoritma secara menyeluruh. Selain itu, pengabaian konstanta bisa membuat *Big O* tidak tepat untuk jumlah data yang kecil. Contohnya:

Algoritma 1 : $k * (n + 100000)$

Algoritma 2 : $k * (n^2)$

Bila *Big O* sebagai pedoman saja dan dengan fakta bahwa k pada kedua algoritma adalah sama, maka Algoritma 1 yang lebih efisien dengan kompleksitasnya = $O(n)$, sedangkan kompleksitas Algoritma 2 = $O(n^2)$. Namun bagaimana jika ternyata data yang digunakan hanya 100, apakah Algoritma 1 tetap lebih efisien? Misal, data = 100; $k = 5$, maka waktu eksekusi Algoritma 1 = 500500, sedangkan pada Algoritma 2 = 50000, maka dengan begitu Algoritma 2 menjadi lebih efisien daripada Algoritma 1.

Terdapat istilah yang dinamakan dengan, *best-case*, *worst-case*, dan *average-case* dalam komposisi data. Perhatikan contoh berikut ini:

```
Posisi ← 0  
For i ← 1 to n  
    If Nilai(i) = Dicari  
        Posisi ← i  
        Break  
    End If  
End For
```

Jika data yang dicari berada pada posisi terakhir, maka terjadi n pencarian, merupakan *worst-case*, sehingga $\text{Big } O = O(n)$. Namun jika data yang dicari berada pada posisi pertama, maka pencarian hanya dilakukan sekali, merupakan *best-case*, dengan $\text{Big } O = O(1)$. Kenyataan yang sering terjadi, data bisa berada di mana saja, sehingga rata-rata pencarian menjadi sebesar $n/2$, merupakan *average-case*, dengan $\text{Big } O = O(n/2)$ yang sebenarnya identik dengan $O(n)$. Kompleksitas waktu yang dinyatakan dengan $\text{Big } O$ umumnya merujuk ke skenario *average-case* [39].

4.3 Evaluasi Model Klasifikasi

Evaluasi kinerja dari suatu model klasifikasi dapat dilakukan menggunakan pendekatan *Confusion Matrix* untuk memperoleh *accuracy*, *precision*, *recall* (*sensitivity* dan *specificity*), dan *F-Measure* yang ditunjukkan pada Tabel 4.1. *Sensitivity* merupakan *recall* pada *class (-)*, sedangkan *specificity* adalah *recall* pada *class (+)*.

Tabel 4.1 Confusion Matrix

	Actual (+)	Actual (-)	Precision
Predicted (+)	True (+) (TP)	False (+) (FP)	$TP / (TP + FP) *$
Predicted (-)	False (-) (FN)	True (-) (TN)	$TN / (TN + FN)$
Recall	$TP / (TP + FN)$	$TN / (TN + FP)$	
F-Measure	$(2 * \text{Precision} * \text{Sensitivity}) / (\text{Precision} + \text{Sensitivity})$		
Accuracy	$(TP + TN) / (TP + TN + FN + FP)$		

Contoh 4.2 Confusion Matrix (Manual)

Misalnya diketahui hasil klasifikasi 10 data uji sebagai berikut:

Data ke-	Actual Output	Prediksi
1	Kambuh (+)	Kambuh (+)
2	Kambuh (+)	Kambuh (+)
3	Kambuh (+)	Tidak Kambuh (-)
4	Kambuh (+)	Kambuh (+)
5	Kambuh (+)	Kambuh (+)
6	Kambuh (+)	Kambuh (+)
7	Kambuh (+)	Kambuh (+)
8	Tidak Kambuh (-)	Kambuh (+)
9	Tidak Kambuh (-)	Kambuh (+)
10	Tidak Kambuh (-)	Tidak Kambuh (-)

Maka *Confusion Matrix* dari hasil klasifikasi tersebut adalah:

	Actual (+)	Actual (-)	Precision (%)
Predicted (+)	6	2	$(6 / (6 + 2)) * 100 = 75,0000$
Predicted (-)	1	1	$(1 / (1 + 1)) * 100 = 50,0000$
Recall (%)	85,7143	33,3333	
Specificity (%)			$(6 / (6 + 1)) * 100 = 85,7143$
Sensitivity (%)			$(1 / (1 + 2)) * 100 = 33,3333$
F-Measure (%)			$(2 * 75,0000 * 33,3333) / (75,0000 + 33,3333) = 46,1538$
Accuracy (%)			$((6+1) / (6 + 1 + 2 + 1)) * 100 = 70,0000$

4.4 Interval Kepercayaan Akurasi

Data yang digunakan antara suatu penelitian dengan penelitian lainnya yang juga menggunakan metode yang sama biasanya berbeda. Untuk memperkirakan interval kepercayaan akurasi perlu digunakan distribusi probabilitas yang mengatur ukuran akurasi. Pendekatan distribusi normal dapat digunakan untuk mengukur tingkat kepercayaan akurasi.

$$P\left(-Z_{\alpha/2} \leq \frac{acc - p}{\sqrt{\frac{p(1-p)}{n}}} \leq Z_{1-\alpha/2}\right) = 1 - \alpha \quad (29)$$

Keterangan:

- n : jumlah data uji.
- p : akurasi sebenarnya.
- acc : akurasi dari model.
- $Z_{\alpha/2}$: batas atas kepercayaan akurasi.
- $Z_{1-\alpha/2}$: batas bawah kepercayaan akurasi.

Tabel 4.2 Interval Kepercayaan $Z_{\alpha/2}$ dalam Distribusi Normal

$1-\alpha$	0,5	0,7	0,8	0,85	0,9	0,95	0,98	0,99	0,998	0,999
$Z_{\alpha/2}$	0,67	1,04	1,282	1,440	1,645	1,967	2,326	2,576	3,090	3,291

Dengan menyederhanakan Persamaan (29), dihasilkan interval kepercayaan untuk p sebagai berikut:

$$\frac{2 * n * acc + Z_{\alpha/2}^2 \pm Z_{\alpha/2} \sqrt{Z_{\alpha/2}^2 + 4 * n * acc - 4 * n * acc^2}}{2(n + Z_{\alpha/2}^2)} \quad (30)$$

Contoh 4.3 Tingkat Kepercayaan Akurasi

Misalnya, akurasi dari model k-NN = 96,67% dengan data uji $n = 150$, berapa interval kepercayaan akurasi tersebut pada tingkat kepercayaan 95%?

$$\frac{2 * 150 * 0,9767 + 1,967^2 \pm 1,967 \sqrt{1,967^2 + 4 * 150 * 0,9667 - 4 * 150 * 0,9667^2}}{2(150 + 1,967^2)}$$

Didapatkan batas atas = 0,985741, dan batas bawah = 0,924188. Jadi untuk tingkat kepercayaan 95%, di dapatkan interval kepercayaan akurasi dari 92,42% s/d 98,57%.

4.5 Evaluasi Model Regresi

Pengukuran kinerja dari suatu model regresi/estimasi dapat dilakukan dengan mengukur estimasi *error*-nya. Semakin tinggi estimasi *error* suatu model metode *machine learning* yang digunakan untuk regresi/estimasi, maka semakin rendah kinerjanya. Beberapa metode yang dapat digunakan, antara lain *Mean Square Error*

(MSE), Root Mean Squared Error (RMSE), Standard Error Estimation (SEE), Percentage Error (PE), Mean Absolute Percentage Error (MAPE), PNSR, dll.

Nilai kesalahan prediksi/estimasi suatu data dapat ditentukan dengan menggunakan Persamaan (31). Selanjutnya estimasi *error* suatu model metode *machine learning* yang digunakan untuk melakukan estimasi dapat diukur dengan menggunakan MSE (32), RMSE (33), SEE (34), PE (35), MAPE (36), dan PNSR (37). Estimasi aktual dari data ke-*i* dapat dinotasikan dengan y_i dan estimasi oleh metode yang digunakan dari data ke-*i* dapat dinotasikan dengan y'_i . Pada SEE, terdapat parameter f yang merupakan derajat kebebasan, yang mana $f = 1$ untuk data konstan, $f = 2$ untuk data linier, $f = 3$ untuk data kwadratis, dan $f = 4$ untuk data siklis.

$$e_i = y_i - y'_i \quad (31)$$

$$mse = \frac{\sum_{i=1}^n (y_i - y'_i)^2}{n} \quad (32)$$

$$rmse = \sqrt{mse} \quad (33)$$

$$see = \sqrt{\frac{\sum_{i=1}^n (y_i - y'_i)^2}{n - f}} \quad (34)$$

$$pe_i = \left(\frac{y_i - y'_i}{y_i} \right) * 100\% \quad (35)$$

$$mape = \frac{\sum_{i=1}^n |pe_i|}{n} \quad (36)$$

$$pnsr = 10 * \log_{10} 256^2 / mse \quad (37)$$

Contoh 4.4 Error Estimation (Manual)

Data ke-	Actual (y)	Predicted (y')	Error	SE
1	7,0000	7,0210	-0,0210	0,0004
2	2,0000	1,9000	0,1000	0,0100
3	3,0000	2,8900	0,1100	0,0121
4	9,0000	9,0100	-0,0100	0,0001
5	11,0000	9,9900	1,0100	1,0201
6	9,5000	9,1000	0,4000	0,1600
7	2,7000	2,6000	0,1000	0,0100
8	7,2500	6,5000	0,7500	0,5625
9	8,8700	8,9800	-0,1100	0,0121
10	8,0000	8,5000	-0,5000	0,2500
Jumlah				2,0373
MSE	2,0373 / 10 =			0,2037
RMSE	SQRT(0,2037) =			0,4514
SEE (f=1)	SQRT(2,0373 / (10 - 1)) =			0,4758
SEE (f=2)	SQRT(2,0373 / (10 - 2)) =			0,5046
SEE (f=3)	SQRT(2,0373 / (10 - 3)) =			0,5395
SEE (f=4)	SQRT(2,0373 / (10 - 4)) =			0,5827
PNSR	10 * Log10(256 ^ 2 / 0,2037) =			55,0742

4.6 Evaluasi Model Klasterisasi

Evaluasi kinerja suatu model klasterisasi dapat dilakukan menggunakan metode *Silhouette Coefficients*, *Davies-Bouldin Index*, *Dunn Index*, *F-Score*, pendekatan *similarity*, atau bahkan implementasinya langsung, dsb. Evaluasi pada suatu model klasterisasi sebanarnya agak *tricky* (rumit). Terdapat beberapa kategori evaluasi model klasterisasi, yaitu evaluasi internal, evaluasi eksternal, evaluasi oleh pakar, dan dengan mengimplementasikan hasil klaster pada suatu aplikasi (mengaplikasikan langsung hasil klaster).

Evaluasi internal dilakukan berdasarkan data dan hasil klaster. Metode-metode yang dapat digunakan untuk evaluasi internal klasterisasi, yaitu *Silhouette Coefficients*, *Davies-Bouldin Index*, dan *Dunn Index*. Metode-metode ini biasanya memberikan skor terbaik untuk algoritma klasterisasi yang menghasilkan klaster dengan kemiripan tinggi dalam suatu klaster namun kemiripan yang rendah antar klaster-klaster. Perlu diketahui bahwa kinerja dengan skor tinggi dari evaluasi internal tidak selalu menghasilkan aplikasi pengambilan informasi yang efektif [40], belum tentu merupakan hasil yang paling bermanfaat pada aplikasi nyatanya. Selain itu, evaluasi ini bias terhadap algoritma klasterisasi yang digunakan [40], misalnya metode *Silhouette Coefficients* cenderung lebih menguntungkan algoritma klasterisasi yang menggunakan pendekatan *centroid*, seperti *K-Means*. Dengan demikian, evaluasi internal lebih tepat untuk komparasi algoritma klasterisasi, namun informasi tersebut bukan berarti bahwa algoritma dengan kinerja terbaik akan lebih valid daripada algoritma klasterisasi lainnya. Metode-metode evaluasi internal klasterisasi biasanya didasarkan pada intuisi bahwa item dalam klaster yang sama harus lebih mirip daripada item dalam klaster yang berbeda [40].

4.6.1 Evaluasi Internal

Silhouette Coefficients menghitung jarak rata-rata elemen dalam *klaster* yang sama dengan jarak rata-rata elemen di klaster lain. Objek dengan nilai *silhouette* tinggi dianggap terkelompok dengan baik, objek dengan nilai rendah mungkin *outlier*. Metode ini berfungsi baik dengan metode klasterisasi *K-Means* [40].

Davies-Bouldin Index (38) menghitung jarak (*dissimilarity*) antar klaster. Algoritma klasterisasi yang menghasilkan klaster dengan jarak intra-klaster rendah (kesamaan intra-klaster tinggi) dan jarak antar klaster tinggi (kesamaan antar klaster rendah) akan memiliki *Davies-Bouldin Index* yang rendah [40]. Semakin rendah nilai *Davies-Bouldin Index* suatu algoritma *machine learning* yang digunakan untuk klasterisasi, maka semakin tinggi kinerjanya.

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right) \quad (38)$$

Keterangan:

n : jumlah klaster.

c_x : *centroid* dari *cluster* (x).

σ_x : jarak rata-rata semua elemen dalam *cluster* (x) untuk *centroid* (c_x).

$d(c_i, c_j)$: jarak antara *centroid* (c_i) dengan (c_j).

Dunn Index (39) bertujuan untuk mengidentifikasi klaster yang padat dan terpisah dengan baik. Metode ini menghitung rasio jarak antar-klaster minimal dengan jarak intra-klaster maksimal. Karena evaluasi internal mencari klaster dengan kesamaan intra-klaster yang tinggi dan kesamaan antar-klaster yang rendah, maka semakin tinggi nilai *Dunn Index* suatu algoritma *machine learning* yang digunakan untuk klasterisasi, maka semakin tinggi pula kinerjanya. Untuk setiap klaster, *Dunn Index* dapat didefinisikan dengan rumus berikut ini.

$$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)} \quad (39)$$

Keterangan:

$d(i,j)$: mewakili jarak antara kluster i dan j .

$d'(k)$: mengukur jarak intra-klaster k .

4.6.2 Evaluasi Eksternal

Sebaliknya dengan evaluasi internal, dalam evaluasi eksternal, hasil klasterisasi dievaluasi berdasarkan data yang tidak digunakan. Metode-metode yang dapat digunakan untuk evaluasi eksternal klasterisasi, yaitu *Purity*, *Rand Index*, *F-Measure*, *Jaccard Index*, *Dice Index*, *Fowlkes – Mallows Index*, *Normalized Mutual Information* (NMI), dan *Confusion Matrix*. Metode-metode tersebut membandingkan hasil klasterisasi dengan label *class* yang biasanya ditentukan secara manual oleh para ahli (*gold standard* atau *ground truth*) [40] atau label *class* dari informasi awal sebelum dilakukan klasterisasi. Evaluasi eksternal dapat digunakan untuk mengatasi kelemahan-kelemahan evaluasi internal. Perlu diketahui bahwa metode evaluasi *F-Measure* dan NMI pada model klasterisasi berbeda dengan pada model klasifikasi. Pada model klasterisasi, *F-Measure* dan MMI digunakan pendekatan *pairwise*.

Purity (40) mengukur sejauh mana klaster berisis suatu *class*. Metode evaluasi ini sangat sensitif terhadap masalah *unbalanced class* [40]. Algoritmanya adalah sebagai berikut:

1. Untuk setiap klaster, hitung jumlah titik data dari *class* yang paling umum di klaster tersebut; dan
2. Hitung jumlah semua klaster dan bagi dengan jumlah data.

Jika diberikan set klaster M dengan beberapa *class* D , dan N partisi titik data, maka *Purity* dapat didefinisikan dengan rumus berikut ini.

$$P = \frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d| \quad (40)$$

Rand Index (41) menghitung seberapa mirip klaster (dikembalikan oleh algoritma klasterisasi yang digunakan) dengan klasifikasi *benchmark*. *Rand Index* juga dapat dilihat sebagai ukuran persentase keputusan yang benar yang dihasilkan oleh algoritma klasterisasi yang digunakan. *Rand Index* dapat didefinisikan dengan rumus yang sama seperti menghitung akurasi pada model klasifikasi menggunakan *Confusion Matrix*, yaitu sebagai berikut.

$$\frac{TP + TN}{TP + FP + TN + FN} \quad (41)$$

F-Measure (44) merupakan salah satu metode evaluasi eksternal klasterisasi yang populer. Salah satu masalah *Rand Index* adalah bahwa *FP* (*False Positive*) dan *FN* (*False Negative*) sama-sama dihitung, padahal hal ini mungkin tidak diinginkan pada beberapa aplikasi klasterisasi [40]. Metode evaluasi *F-Measure* dapat mengatasi masalah tersebut. Seperti penjelasan sebelumnya, *F-Measure* pada model klasifikasi berbeda dengan *F-Measure* pada model klasterisasi. *F-Measure* dapat menyeimbangkan kontribusi *FN* dengan menghitung *Recall* melalui parameter $\beta \geq 0$. Dengan menggunakan *Confusion Matrix*, *Precision* (42) dan *Recall* (43) dapat didefinisikan sebagai berikut.

$$P = \frac{TP}{TP + FP} \quad (42)$$

$$R = \frac{TP}{TP + FN} \quad (43)$$

Maka *F-Measure* dapat didefinisikan dengan rumus berikut ini.

$$F_\beta = \frac{(\beta^2 + 1). P. R}{\beta^2. P + R} \quad (44)$$

Ketika $\beta = 0$, maka $F_0 = P$. Dengan kata lain, *Recall* tidak memiliki dampak pada *F-Measure* ketika $\beta = 0$. Peningkatan β mengalokasikan nilai *F-Measure* yang meningkat. *F-Measure* untuk klasterisasi tidak memperhitungkan *TN*.

Jaccard Index (45) digunakan untuk mengukur *similarity* (kesamaan) antara dua *dataset*. Perlu diketahui bahwa fungsi *cosine* yang sering digunakan pada klasterisasi (biasanya saat data teks) sebenarnya bukanlah *distance* (*dissimilarity*), tapi lebih tepat disebut sebagai ukuran kedekatan (*similarity*), karena fungsi *cosine* tidak memenuhi sifat ketidaksamaan segitiga. *Jaccard Index* bernilai {0, 1}, yang mana indeks 1 berarti bahwa kedua *dataset* identik, sedangkan indeks 0 berarti bahwa *dataset* tidak memiliki elemen umum [40]. *Jaccard Index* tidak memperhitungkan *TN*. *Jaccard Index* dapat didefinisikan dengan rumus berikut ini.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \quad (45)$$

Dice Index (46) menggandakan nilai *TP* dan mengabaikan *TN*. *Dice Index* dapat didefinisikan dengan rumus berikut ini.

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (46)$$

Fowlkes – Mallows Index (47) menghitung *similarity* antar klaster yang berdasarkan algoritma klasterisasi dan klasifikasi *benchmark*. Metode ini dikenal pula sebagai *G-Measure*, sedangkan *F-Measure* adalah rata-rata harmoniknya [40]. Metode ini sangat terkait dengan *Kappa* [40]. *Fowlkes – Mallows Index* dapat didefinisikan dengan rumus berikut ini.

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}} \quad (47)$$

Mutual Information (MI) menunjukkan ukuran teori informasi tentang seberapa banyak informasi dibagi antara hasil klaster dengan *ground truth* yang dapat mendeteksi kesamaan non-linier antara 2 pengelompokan. Sedangkan *Normalized Mutual Information* (NMI) adalah varian dari MI yang dikoreksi untuk mengurangi bias pada klaster.

4.6.3 Evaluasi Manual & Evaluasi Aplikasi

Walaupun evaluasi eksternal mampu menjawab masalah yang dihadapi evaluasi internal, namun jika evaluasi eksternal tidak memiliki label *class ground truth* (kebenaran dasar), maka untuk apa dilakukan klasterisasi? Sementara dalam aplikasi praktis, biasanya label *class* seperti itu tidak dimiliki [40]. Selain itu, evaluasi internal dan eksternal membutuhkan kompleksitas komputasi yang besar, terlebih pada persoalan *big data*. Oleh karena itu, dibutuhkan evaluasi secara manual oleh manusia (ahli/pakar). Walaupun evaluasi secara manual oleh manusia (ahli/pakar) cenderung subyektif, namun bisa sangat informatif dalam mengidentifikasi klasterisasi yang buruk [40]. Selain evaluasi manual oleh manusia, dapat pula dilakukan evaluasi implementasi, yaitu mengaplikasikan langsung hasil klasterisasi. Akhir-akhir ini, evaluasi implementasi lebih diminati.

4.7 Evaluasi Model Asosiasi

Kinerja suatu model asosiasi dapat diukur berdasarkan nilai *Lift Ratio* dari aturan-aturan asosiasi yang diperoleh berdasarkan *Support* dan *Confidence*. *Lift Ratio* mengukur seberapa penting suatu aturan asosiasi, seberapa erat set item A terjadi secara bersamaan dengan set item B. Jika nilai *Support* item A didefinisikan dengan Persamaan (48) dan *Support* item B didefinisikan dengan Persamaan (49), yang mana *n* adalah jumlah transaksi yang terjadi (jumlah data), maka *Support* set item $A \rightarrow B$ dapat didefinisikan dengan Persamaan (50), *Confidence* $A \rightarrow B$ pada Persamaan (51), dan *Lift* $A \rightarrow B$ pada Persamaan (52) atau (53).

$$supp(A) = \frac{freq(A)}{n} \quad (48)$$

$$supp(B) = \frac{freq(B)}{n} \quad (49)$$

$$supp(A \rightarrow B) = \frac{freq(A \cup B)}{n} \quad (50)$$

$$conf(A \rightarrow B) = \frac{supp(A \rightarrow B)}{supp(B)} \text{ atau } \frac{freq(A \cup B)}{freq(A)} \quad (51)$$

$$lift(A \rightarrow B) = \frac{supp(A \rightarrow B)}{supp(A) \cdot supp(B)} \quad (52)$$

Atau jika A dan B independen:

$$\text{lift}(A \rightarrow B) = \frac{\text{conf}(A \rightarrow B)}{\text{supp}(B)} \quad (53)$$

Nilai *Lift* dari 0 hingga positif tak terhingga. Jika nilai *Lift* mendekati 1, maka hubungan antara set item A dan B pada aturan asosiasi tersebut mungkin independen. Ketika dua set item saling independen pada suatu aturan asosiasi, maka sebenarnya tidak ada aturan yang dapat ditarik atas kedua set item tersebut. Namun jika nilai $\text{Lift} > 1$, maka aturan asosiasi tersebut berpotensi berguna. Artinya semakin > 1 nilai *Lift*, maka aturan asosiasi tersebut semakin dapat dipercaya [26].

Contoh 4.5 Lift Ratio (Manual)

Misalnya diketahui:

$$\begin{aligned} \text{supp}(A) &= 0,30; \\ \text{supp}(B) &= 0,80; \\ \text{supp}(A \rightarrow B) &= 0,30; \text{ dan} \\ \text{conf}(A \rightarrow B) &= 0,38; \end{aligned}$$

Maka:

$$\text{lift}(A \rightarrow B) = \frac{0,30}{0,30 * 0,80} = 1,250$$

Atau jika A dan B dianggap independen, maka:

$$\text{lift}(A \rightarrow B) = \frac{0,38}{0,80} = 0,475$$

4.8 Uji Korelasi Variabel

Uji korelasi antar variabel pada prinsipnya merupakan nilai yang menunjukkan tentang adanya hubungan antara dua variabel atau lebih serta besarnya hubungan tersebut. Namun dalam hal ini, korelasi tidak menunjukkan hubungan sebab akibat. Seandainya dipahami sebagai suatu hubungan sebab akibat, hal itu bukan karena diketahuinya koefisien korelasi antar variabel tersebut, melainkan karena rujukan teori atau logika yang memaknai hasil perhitungan. Oleh karena itu biasanya analisis korelasi mensyaratkan acuan teori yang mendukung adanya hubungan sebab akibat dalam variabel-variabel yang dianalisa hubungannya.

Nilai koefisien korelasi antara variabel bebas/independen (X) terhadap variabel terikat/dependen (Y) dengan jumlah data sebesar n , dapat didefinisikan dengan menggunakan rumus yang dikembangkan oleh Karl Pearson, yaitu [41]:

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{n(\sum X^2) - (\sum X)^2} \sqrt{n(\sum Y^2) - (\sum Y)^2}} \quad (54)$$

Nilai koefisien korelai parsial dua atau lebih variabel bebas (X) terhadap variabel terikat (Y) merupakan hubungan korelasi antara salah satu variabel bebas (X) terhadap variabel terikat (Y) yang mana variabel bebas (X) lainnya dianggap konstan. Atau dengan kata lain, koefisien korelasi parsial menunjukkan kekuatan

hubungan salah satu variabel bebas (X) terhadap variabel terikat (Y) secara parsial, tidak simultan atau bersama-sama. Untuk itu Persamaan (54) di atas menjadi:

$$r_{x_i y} = \frac{n(\sum X_i Y) - (\sum X_i)(\sum Y)}{\sqrt{\{n(\sum X_i^2) - (\sum X_i)^2\}\{n(\sum Y^2) - (\sum Y)^2\}}} \quad (55)$$

Sedangkan nilai koefisien korelasi simultan (bersama-sama) dua variabel bebas (X) terhadap variabel terikat (Y) dapat didefinisikan sebagai berikut:

$$R_{yx_1 x_2} = \sqrt{\frac{r^2 yx_1 + r^2 yx_2 - 2ryx_1 \cdot ryx_2 \cdot rx_1 x_2}{1 - r^2 x_1 x_2}} \quad (56)$$

Keterangan:

ryx_1 : koefisien korelasi antara variabel bebas (X_1) terhadap variabel terikat (Y).

ryx_2 : koefisien korelasi antara variabel bebas (X_2) terhadap variabel terikat (Y).

Jika variabel bebas (X) > 2, dengan pola yang sama seperti Persamaan (56), maka koefisien korelasi simultan dapat didefinisikan sebagai berikut:

$$R_{x_i y} = \sqrt{\frac{\sum(b_i(\sum x_i y))}{\sum y^2}} \quad (57)$$

Nilai $\sum y^2$ dan $\sum x_i y$ diperoleh melalui persamaan berikut ini.

$$\sum y^2 = \sum Y^2 - \frac{(\sum Y)^2}{n} \quad (58)$$

$$\sum x_i y = \sum X_i Y - \frac{(\sum X_i)(\sum Y)}{n} \quad (59)$$

Untuk kekuatan hubungannya, nilai koefisien korelasi berada di antara -1 sampai 1, sedangkan untuk arah dinyatakan dalam bentuk positif (+) dan negatif (-) [42].

Koefisien korelasi menunjukkan seberapa besar varian total suatu variabel berhubungan dengan varian variabel lain. Hal ini berarti bahwa tiap nilai r perlu ditafsirkan posisinya dalam keterkaitan tersebut. Untuk memberikan tafsiran pada nilai koefisien korelasi, dapat digunakan referensi *Guilford Empirical Rules* yang ditunjukkan pada Tabel 4.3 berikut ini.

Tabel 4.3 Guilford Empirical Rules untuk Tafsiran Koefisien Korelasi

Besar r_{yx}	Penafsiran
$0.00 - < 0.20$	Hubungan sangat lemah (diabaikan, dianggap tidak ada).
$>= 0.20 - < 0.40$	Hubungan rendah atau lemah.
$>= 0.40 - < 0.70$	Hubungan sedang atau cukup.
$>= 0.70 - < 0.90$	Hubungan kuat.
$>= 0.90 - < 1.00$	Hubungan sangat kuat.

Setelah nilai koefisien korelasi diperoleh, nilai koefisien determinasi juga dapat diperoleh dengan menggunakan Persamaan (60) untuk mengukur seberapa besar dua variabel bebas (X) mempengaruhi variabel terikat (Y), yang mana sisanya ditentukan oleh variabel bebas (X) lainnya yang tidak ikut dianalisis.

$$KP = (R_{x_1 x_2 y})^2 * 100\% \quad (60)$$

Nilai KP pada Persamaan (60) di atas menunjukkan seberapa besar nilai variabel bebas (X_1) dan (X_2) mempengaruhi nilai variabel terikat (Y).

Nilai $(1 - KP)$ akan menunjukkan persentase besarnya pengaruh faktor-faktor lain di luar faktor yang ada pada variabel bebas, dalam mempengaruhi variabel terikat (Y). Berdasarkan nilai determinasi, dapat dilakukan uji multikolinieritas, yang mana jika $1 - KP \geq 0,1$ dan $1 / (1 - KP) < 10$, maka tidak terjadi korelasi antara variabel bebas (X_1) dan (X_2).

Selain uji koefisien korelasi, dapat pula dilakukan uji regresi seperti *error estimasi* (telah dibahas pada sub pokok bahasan sebelumnya), *T-Test* (uji regresi parsial) dan *F-Test* (uji regresi simultan). Uji *T-Test* digunakan untuk menguji apakah ada pengaruh secara signifikan suatu variabel bebas (X) terhadap variabel terikat (Y). Misalnya (X_1) terhadap (Y) yang mana (X_2) dianggap konstan, atau (X_2) terhadap (Y) yang mana (X_1) dianggap konstan, atau (X_1) terhadap (X_2) yang mana (Y) dianggap konstan. Sedangkan *F-Test* berguna untuk menguji apakah populasi tempat sampel diambil memiliki korelasi nol atau adanya relasi yang signifikan antara variabel bebas (X_1, X_2, \dots, X_n) secara simultan terhadap variabel terikat (Y).

Algoritma uji *T-Test* sebagai berikut:

1. Tentukan hipotesis H_0 (secara parsial tidak ada pengaruh signifikan antar variabel) dan H_a (secara parsial ada pengaruh signifikan antar variabel).
2. Tentukan tingkat signifikansi, biasanya/standarnya $\alpha = 5\%$.
3. Tentukan t hitung.
4. Tentukan t tabel.
5. Tentukan kriteria pengujian, misalnya:
 Ho diterima jika $-t$ tabel $< t$ hitung $< t$ tabel.
 Ho ditolak jika $-t$ hitung $< -t$ tabel atau t hitung $> t$ tabel.
6. Bandingkan t hitung dengan t tabel
7. Simpulkan, misalnya t hitung $> t$ tabel maka H_0 diterima, artinya secara parsial tidak ada pengaruh signifikan antar variabel.

Korelasi variabel bebas (X_1) terhadap variabel terikat (Y) yang mana variabel bebas (X_2) dianggap konstan menggunakan uji *T-Test* didefinisikan sebagai berikut.

$$r_{y1.2} = \frac{r_{y1} - (r_{y2} r_{12})}{\sqrt{(1 - r_{y2}^2)(1 - r_{12}^2)}} \quad (61)$$

Korelasi variabel bebas (X_2) terhadap variabel terikat (Y) yang mana variabel bebas (X_1) dianggap konstan menggunakan uji *T-Test* didefinisikan sebagai berikut.

$$r_{y2.1} = \frac{r_{y2} - (r_{y1} r_{12})}{\sqrt{(1 - r_{y1}^2)(1 - r_{12}^2)}} \quad (62)$$

Korelasi variabel bebas (X_1) terhadap variabel bebas (X_2) yang mana variabel terikat (Y) dianggap konstan menggunakan uji *T-Test* didefinisikan sebagai berikut.

$$r_{12,y} = \frac{r_{12} - (r_{y1}r_{y2})}{\sqrt{(1 - r_{y1}^2)(1 - r_{y2}^2)}} \quad (63)$$

Nilai r_{y1} , r_{y2} , dan r_{12} diperoleh melalui persamaan berikut ini.

$$r_{y1} = \frac{n \sum X_1 Y - (\sum Y \sum X_1)}{\sqrt{[(n \sum y^2) - (\sum y)^2][(n \sum X_1^2) - (\sum X_1)^2]}} \quad (64)$$

$$r_{y2} = \frac{n \sum X_2 Y - (\sum Y \sum X_2)}{\sqrt{[(n \sum y^2) - (\sum y)^2][(n \sum X_2^2) - (\sum X_2)^2]}} \quad (65)$$

$$r_{12} = \frac{n \sum X_1 X_2 - (\sum X_1 \sum X_2)}{\sqrt{[(n \sum X_1^2) - (\sum X_1)^2][(n \sum X_2^2) - (\sum X_2)^2]}} \quad (66)$$

$$r_{y12} = \sqrt{\frac{(r_{y1}^2 + r_{y2}^2) - (2r_{y1}r_{y2}r_{12})}{(1 - r_{12})^2}} \quad (67)$$

4.9 Soal Latihan Evaluasi Model

Buatlah tabel dengan variabel input, output, dan 7 instances (semester ke-) sbb:

Variabel input:

- | | |
|--|---|
| 1. Semester ke- | = {1, 2, ..., 7} (sebagai ID) |
| 2. Matematika | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 3. Bhs. Inggris | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 4. Komputasi | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 5. Bhs. Pemrograman | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 6. Kepribadian | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 7. Spritual | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 8. Bekerja | = {Tidak (0), Ya (1)} |
| 9. Motivasi belajar | = {Memburuk (-1), Tetap (0), Membaik (1)} |
| 10. Rata-rata jam belajar dalam sehari | = {0, 1, ... 24} |
| 11. Jumlah buku yang telah dibaca | = {0, 1, ... n} |

Variabel output:

- | | |
|---------------------------------|---|
| 1. IPK aktual | = {0,00 – 4,00} |
| 2. IPK (prediksi) | = {0,00 – 4,00} (random atau perkiraan) |
| 3. Prestasi akademik (aktual) | = {Menurun (0), Meningkat (1)} |
| 4. Prestasi akademik (prediksi) | = {Menurun (0), Meningkat (1)} (sda) |

Mintalah data tabel tersebut dari 9 teman dekat anda, sehingga anda memiliki sebuah dataset dengan $7 * 10 = 70$ instances, 10 variabel input, dan 4 variabel output. Kemudian lakukan evaluasi model berikut ini:

1. Confusion Matrix (gunakan output 3 dan 4).
2. RMSE, SEE, dan PNSR (gunakan output 1 dan 2).
3. Uji koefisien korelasi dan *T-Test* pada input 10 dan input 11 terhadap output 1.

5. Fuzzy Logic

No.	Materi	Tujuan Pembelajaran
1.	Karakteristik <i>Fuzzy Logic</i>	Anda mampu memahami dan menjelaskan karakteristik algoritma <i>Fuzzy Logic</i> .
2.	<i>Fuzzification</i>	Anda mampu memahami, menjelaskan, dan menerapkan <i>Membership Function</i> <i>trimf</i> , <i>smf</i> , <i>zmf</i> , <i>pimf</i> , <i>trapmf</i> , <i>sigmf</i> , <i>gaussmf</i> , dan <i>gbellmf</i> dalam proses <i>Fuzzification Fuzzy Logic</i> .
3.	<i>Knowledge Base</i>	Anda mampu memahami, menjelaskan, dan merancang <i>rules</i> dalam proses <i>Knowledge Base Fuzzy Logic</i> .
4.	<i>Machine Inference</i>	Anda mampu memahami, menjelaskan, dan menerapkan operasi irisan, gabungan, dan komplemen dalam proses <i>Machine Inference Fuzzy Logic</i> .
5.	<i>Defuzzification</i>	Anda mampu memahami, menjelaskan, dan menerapkan teknik <i>average</i> dan <i>centroid</i> dalam proses <i>Defuzzification Fuzzy Logic</i> .
6.	Penerapan <i>Fuzzy Logic</i>	Anda mampu memahami dan menerapkan metode <i>Fuzzy Logic</i> <i>Tsukamoto</i> , <i>Mamdani</i> , dan <i>Sugeno</i> secara manual maupun menggunakan <i>tools</i> .

5.1 Karakteristik Fuzzy Logic

Metode *Fuzzy Logic* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada tahun 1962 [43]. *Fuzzy Logic* atau *Fuzzy Inference System* (FIS) merupakan salah satu pendekatan *reasoning* dalam AI. Jika pendekatan *searching* kesulitan dalam menentukan apakah aturan-aturan sudah tepat dan lengkap karena masalah yang dihadapi cukup kompleks sehingga representasi masalah ke dalam *state* menjadi tidak efisien, maka pendekatan *reasoning* dengan representasi *logic* (bahasa formal) merupakan solusinya. Awalnya metode-metode *reasoning* digunakan pada masalah yang memiliki kepastian, bagaimana jika masalah mengandung ketidakpastian? Pendekatan seperti teori probabilitas dan *Fuzzy Logic* merupakan solusinya. Metode-metode dengan pendekatan probabilitas untuk masalah yang mengandung ketidakpastian bersifat peluang. Bagaimana jika masalah mengandung ketidakpastian yang besifat samar? *Fuzzy Logic* merupakan solusinya.

Fuzzy Logic mampu memodelkan fungsi-fungsi nonlinier, mampu mengatasi masalah yang sangat kompleks, didasarkan pada bahasa formal/alami, memiliki toleransi terhadap data yang tidak tepat, dan mampu merepresentasikan pengetahuan pakar ke dalam basis pengetahuannya sebagai aturan-aturan yang berlaku sehingga tidak memerlukan proses *learning*. Namun terkadang aturan-aturan tidak selalu bisa didefinisikan secara tepat dan lengkap, hal ini merupakan kekuarangan metode *Fuzzy Logic*. Oleh karena itu, metode ini membutuhkan pengetahuan pakar untuk diterapkan sebagai aturan-aturan.

Segala sesuatu pada logika konfensional/klasik dianggap hanya memiliki dua kemungkinan, benar atau salah, ya atau tidak, baik atau buruk, dll sehingga hanya memiliki nilai keanggotaan 0 dan 1. Sedangkan nilai keanggotaan pada *Logika Fuzzy* berada dalam jangkauan $[0; 1]$. Itulah mengapa disebut Logika Samar.

Himpunan *Fuzzy Logic* memiliki dua jenis atribut, yaitu [43]:

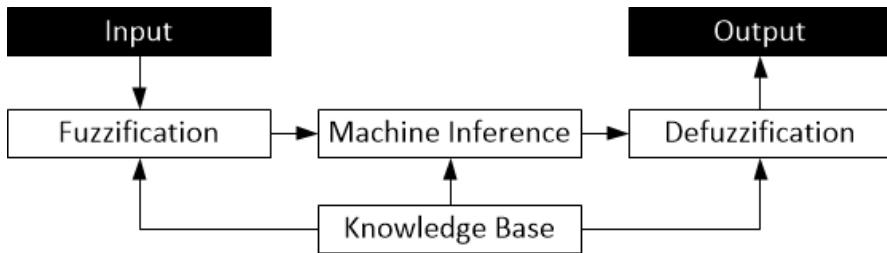
1. Linguistik, merupakan nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya label dingin, sejuk, dan panas mewakili variabel temperatur.
2. Numerik, merupakan suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 2, 7, ..., dll.

Terdapat beberapa elemen yang harus dipahami dalam *Fuzzy Logic*, yaitu [43]:

1. Variabel *Fuzzy*, merupakan variabel-variabel yang ditangani *Fuzzy Logic*, misalnya variabel permintaan, penghasilan, temperatur, umur, dll.
2. Himpunan *Fuzzy*, merupakan suatu kelompok yang mewakili keadaan tertentu dalam suatu variabel *Fuzzy*, misalnya variabel permintaan terbagi dua himpunan *Fuzzy*, yaitu himpunan naik dan turun.
3. Semesta pembicaraan, merupakan seluruh nilai yang diizinkan untuk dioperasikan dalam suatu variabel *Fuzzy*, misalnya semesta pembicaraan variabel temperatur = $[-10; 90]$ (dari -10 hingga 90).
4. Domain himpunan *Fuzzy*, merupakan seluruh nilai yang diizinkan dalam suatu semesta pembicaraan, misalnya domain himpunan turun pada variabel permintaan = $[0; 5000]$ dan naik = $[1000; +\infty]$.

Fuzzy Logic terdiri dari beberapa proses berikut ini (struktur *Fuzzy Logic*) [43]:

1. *Fuzzification*, merupakan proses untuk merubah nilai tegas (*crisp*) inputan menjadi nilai *Fuzzy* (derajat keanggotaan) menggunakan suatu *Membership Function* (Fungsi Keanggotaan).
2. *Knowledge Base*, merupakan kumpulan aturan-aturan (*rules*) dalam bentuk pernyataan *IF ... THEN ...*.
3. *Machine Inference*, merupakan proses untuk merubah inputan menjadi output (y_i), yang mana y_i adalah output dari *rule ke-i* dalam *Knowledge Base*. *Machine Inference* menggunakan fungsi implikasi *Max-Min* atau *Dot-Product* untuk memperoleh α -predikat dari *rule ke-i* (α_i) dan output *rule ke-i* (y_i).
4. *Defuzzification*, merupakan proses merubah output-output y_i yang diperoleh pada *Machine Inference* menjadi satu nilai *crisp* output (y'). Terdapat dua pendekatan/metode yang biasanya digunakan pada proses *Defuzzification*, yaitu *Average* (82) dan *Centroid* (83).



Gambar 5.1 Struktur Fuzzy Logic

Terdapat tiga algoritma *Fuzzy Logic* yang dapat digunakan, yaitu:

1. *Fuzzy Logic Tsukamoto*
 - *Knowledge Base* menggunakan model *rule*:

$$IF (x_1 \text{ IS } a_1) AND/OR (x_2 \text{ IS } a_2) \dots AND/OR (x_n \text{ IS } a_n) THEN (y \text{ IS } b)$$
 - Secara standar, *Machine Inference* menggunakan fungsi implikasi *Min* untuk memperoleh α_i yang digunakan untuk memperoleh y_i .
 - Secara standar, *Defuzzification* menggunakan metode *Average*.
2. *Fuzzy Logic Mamdani*
 - *Knowledge Base* menggunakan model *rule* yang sama seperti *Tsukamoto*.
 - Secara standar, *Machine inference* menggunakan fungsi implikasi *Min* dan komposisi antar *rule* menggunakan fungsi *Max*, menghasilkan himpunan *Fuzzy* baru.
 - Secara standar, *Defuzzification* menggunakan metode *Centroid*.
3. *Fuzzy Logic Sugeno*
 - *Knowledge Base* menggunakan model *rule*:

$$IF (x_1 \text{ IS } a_1) AND/OR (x_2 \text{ IS } a_2) \dots AND/OR (x_n \text{ IS } a_n) THEN y = f(x,y)$$

$f(x,y)$ adalah fungsi *crisp* yang biasanya merupakan fungsi linier dari x dan y . Dengan demikian output setiap rule berupa konstanta atau persamaan linier.
 - Secara standar, *Machine Inference* menggunakan fungsi implikasi *Min* untuk memperoleh α_i yang digunakan untuk memperoleh y_i .
 - Secara standar, *Defuzzification* menggunakan metode *Average*.

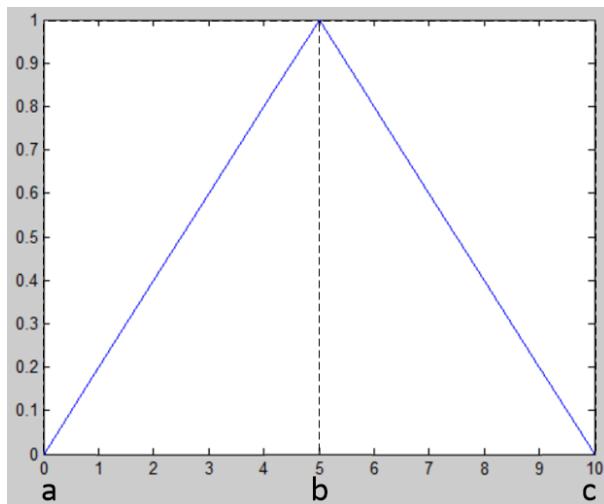
5.2 Fuzzification

Fuzzification merupakan proses untuk merubah nilai tegas (*crisp*) inputan menjadi nilai *Fuzzy* (derajat keanggotaan) menggunakan suatu *Membership Function* (Fungsi Keanggotaan). *Membership Function* merupakan grafik yang mewakili nilai derajat keanggotaan (berada dalam interval 0 dan 1) tiap-tiap variabel input, dapat disimbolkan $f(x)$. Terdapat beberapa tipe *Membership Function* yang biasa/sering digunakan, yaitu *trimf*, *smf*, *zmf*, *pimf*, *trapmf*, *sigmf*, *dsigmf*, *psigmf*, *gaussmf*, *gauss2mf*, *gbellmf*, dan *mf2mf*.

5.2.1 Triangular Membership Function

Triangular Membership Function (trimf) merupakan grafik kurva linier yang terdiri dari dua garis lurus, yaitu kurva linier naik dan kurva linier turun. *Trimf* memiliki tiga parameter a , b , dan c yang didefinisikan sebagai berikut.

$$f(x) = \begin{cases} 0; & x \leq a \text{ or } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (c - x)/(c - b); & b \leq x \leq c \\ 1; & x = b \end{cases} \quad (68)$$



Gambar 5.2 Grafik Kurva Trimf

Contoh 5.1 Trimf (Matlab & Manual)

```
x = 0:10; %variabel
h = [0 5 10]; %[a b c]
y = trimf(x,h); %fungsi trimf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva trimf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0000	0,2000	0,4000	0,6000	0,8000	1,0000	0,8000	0,6000	0,4000	0,2000	0,0000

Penyelesaian secara manual untuk $x = 2, 5$, dan 8 adalah:

$$f(2) = \frac{x - a}{b - a} = \frac{2 - 0}{5 - 0} = 0,4$$

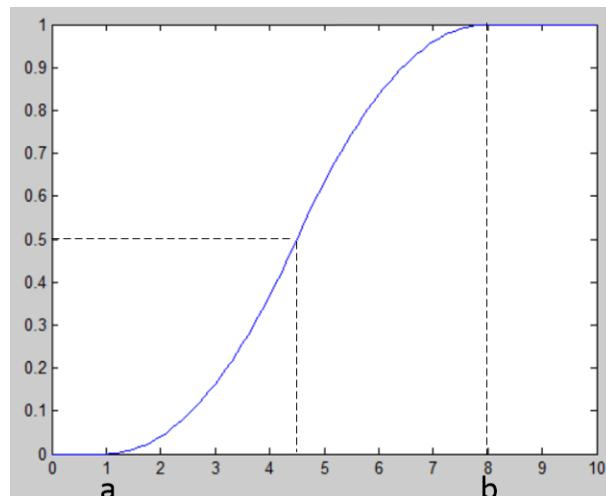
$$f(5) = x = b = 1$$

$$f(8) = \frac{c - x}{c - b} = \frac{10 - 8}{10 - 5} = 0,4$$

5.2.2 S & Z Shaped Membership Function

S-Shaped Membership Function (smf) memiliki kurva berbentuk seperti huruf S yang memiliki dua parameter a dan b yang didefinisikan sebagai berikut.

$$f(x) = \begin{cases} 0; & x \leq a \\ 2((x - a)/(b - a))^2; & a \leq x \leq (a + b)/2 \\ 1 - 2((x - b)/(b - a))^2; & (a + b)/2 \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (69)$$



Gambar 5.3 Grafik Kurva smf

Contoh 5.2 Smf (Matlab & Manual)

```
x = 0:0.1:10; %variabel
h = [1 8]; %[a b]
y = smf(x,h); %fungsi smf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva smf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0000	0,0000	0,0408	0,1633	0,3673	0,6327	0,8367	0,9592	1,0000	1,0000	1,0000

Penyelesaian secara manual untuk $x = 1, 3, 6$, dan 8 adalah:

$$f(1) = 1 \leq a = 0$$

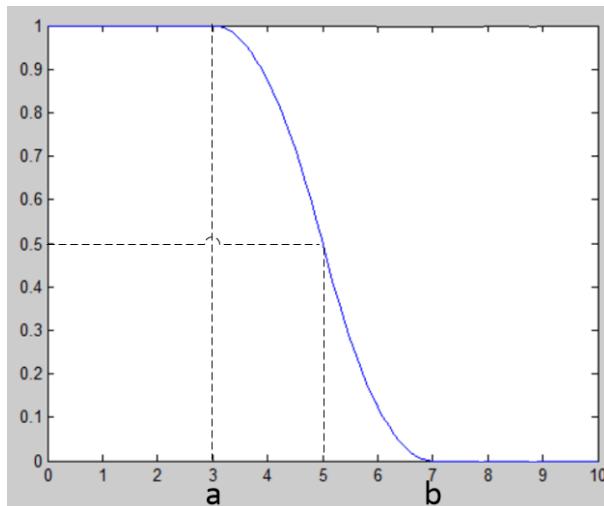
$$f(3) = 2 \left(\frac{x-a}{b-a} \right)^2 = 2 \left(\frac{3-1}{8-1} \right)^2 = 0,1633$$

$$f(6) = 1 - 2 \left(\frac{x-b}{b-a} \right)^2 = 1 - 2 \left(\frac{6-8}{8-1} \right)^2 = 0,8367$$

$$f(8) = 8 \geq b = 1$$

Z-Shaped Membership Function (zmf) merupakan kebalikan dari *smf*, memiliki kurva berbentuk seperti huruf Z yang memiliki dua parameter a dan b yang didefinisikan sebagai berikut.

$$f(x) = \begin{cases} 1; & x \leq a \\ 1 - 2((x-a)/(b-a))^2; & a \leq x \leq (a+b)/2 \\ 2((x-b)/(b-a))^2; & (a+b)/2 \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (70)$$



Gambar 5.4 Grafik Kurva zmf

Contoh 5.3 Zmf (Matlab & Manual)

```
x = 0:0.1:10; %variabel
h = [3 7]; %[a b]
y = zmf(x,h); %fungsi zmf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva zmf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	1,0000	1,0000	1,0000	1,0000	0,8750	0,5000	0,1250	0,0000	0,0000	0,0000	0,0000

Penyelesaian secara manual untuk $x = 3, 4, 6$, dan 7 adalah:

$$f(3) = 3 \leq a = 1$$

$$f(4) = 1 - 2 \left(\frac{x-a}{b-a} \right)^2 = 1 - 2 \left(\frac{4-3}{7-3} \right)^2 = 0,8750$$

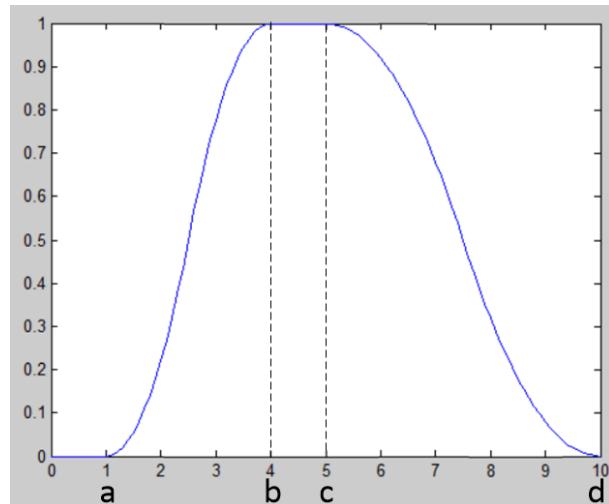
$$f(6) = 2 \left(\frac{x - b}{b - a} \right)^2 = 2 \left(\frac{6 - 7}{7 - 3} \right)^2 = 0,1250$$

$$f(7) = 7 \geq b = 0$$

5.2.3 PI Membership Function

PI Membership Function (pimf) merupakan gabungan *smf* dan *zmf* yang terdiri dari 4 parameter a , b , c , dan d yang didefinisikan sebagai berikut.

$$f(x) = \begin{cases} 0; & x \leq a \\ 2((x - a)/(b - a))^2; & a \leq x \leq (a + b)/2 \\ 1 - 2((x - b)/(b - a))^2; & (a + b)/2 \leq x \leq b \\ 1; & b \leq x \leq c \\ 1 - 2((x - c)/(d - c))^2; & c \leq x \leq (c + d)/2 \\ 2((x - d)/(d - c))^2; & (c + d)/2 \leq x \leq d \\ 0; & x \geq d \end{cases} \quad (71)$$



Gambar 5.5 Grafik Kurva pimf

Contoh 5.4 Pimf (Matlab & Manual)

```
x = 0:0.1:10; %variabel
h = [1 4 5 10]; %[a b c d]
y = pimf(x,h); %fungsi pimf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva pimf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0000	0,0000	0,2222	0,7778	1,0000	1,0000	0,9200	0,6800	0,3200	0,0800	0,0000

Penyelesaian secara manual untuk $x = 1, 2, 3, 4, 6, 8$, dan 10 adalah:

$$f(1) = 1 \leq a = 0$$

$$f(2) = 2 \left(\frac{x-a}{b-a} \right)^2 = 2 \left(\frac{2-1}{4-1} \right)^2 = 0,2222$$

$$f(3) = 1 - 2 \left(\frac{x-b}{b-a} \right)^2 = 1 - 2 \left(\frac{3-4}{4-1} \right)^2 = 0,7778$$

$$f(4) = b \leq 4 \leq c = 1$$

$$f(6) = 1 - 2 \left(\frac{x-c}{d-c} \right)^2 = 1 - 2 \left(\frac{6-5}{10-5} \right)^2 = 0,9200$$

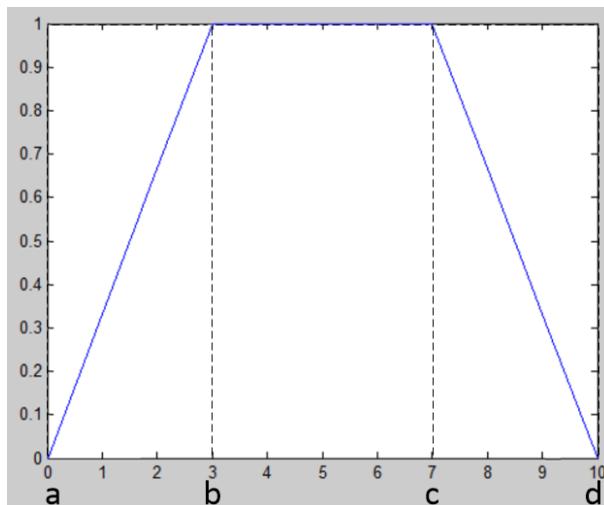
$$f(8) = 2 \left(\frac{x-d}{d-c} \right)^2 = 2 \left(\frac{8-10}{10-5} \right)^2 = 0,3200$$

$$f(10) = 10 \geq d = 0$$

5.2.4 Trapezoidal Membership Function

Pada prinsipnya *Trapezoidal Membership Function (trapmf)* sama dengan *trimf*, namun beberapa titik pada *trapmf* memiliki derajat keanggotaan = 1, sehingga memiliki empat parameter a , b , c , dan d yang didefinisikan sebagai berikut.

$$f(x) = \begin{cases} 0; & x \leq a \text{ or } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \end{cases} \quad (72)$$



Gambar 5.6 Grafik Kurva trapmf

Contoh 5.5 Trapmf (Matlab & Manual)

```
x = 0:10; %variabel
h = [0 3 7 10]; %[a b c d]
y = trapmf(x,h); %fungsi trapmf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva trapmf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0000	0,3333	0,6667	1,0000	1,0000	1,0000	1,0000	1,0000	0,6667	0,3333	0,0000

Penyelesaian secara manual untuk $x = 2, 5$, dan 8 adalah?

$$f(2) = \frac{x - a}{b - a} = \frac{2 - 0}{3 - 0} = 0,6667$$

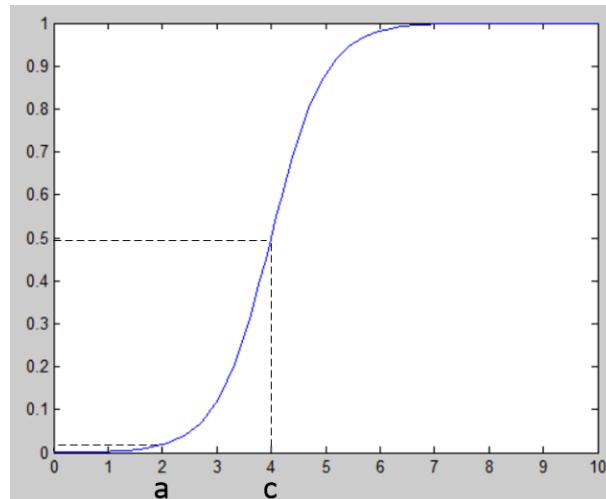
$$f(5) = b \leq x \leq c = 1$$

$$f(8) = \frac{d - x}{d - x} = \frac{10 - 8}{10 - 7} = 0,6667$$

5.2.5 Sigmoidal Membership Function

Sigmoidal Membership Function (sigmf) sesuai untuk merepresentasikan konsep/nilai-nilai yang sangat besar atau terlalu negatif. *Sigmf* memiliki dua parameter a dan c yang didefinisikan sebagai berikut.

$$f(x) = \frac{1}{1 + \exp(-a(x - c))} \quad (73)$$



Gambar 5.7 Grafik Kurva sigmf

Contoh 5.6 Sigmf (Matlab & Manual)

```

x = 0:0.1:10; %variabel
h = [2 4]; %[a c]
y = sigmf(x,h); %fungsi sigmf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva sigmf
    
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0003	0,0025	0,0180	0,1192	0,5000	0,8808	0,9820	0,9975	0,9997	1,0000	1,0000

Penyelesaian secara manual untuk $x = 2, 4$, dan 8 adalah?

$$f(2) = \frac{1}{1 + \exp(-2(2 - 4))} = 0,0180$$

$$f(4) = \frac{1}{1 + \exp(-2(4 - 4))} = 0,5000$$

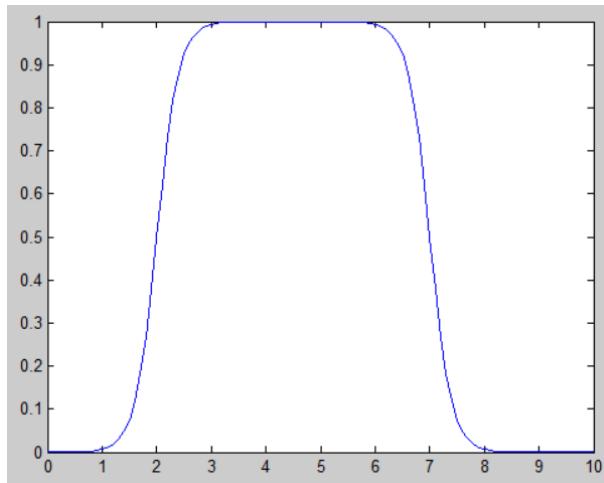
$$f(8) = \frac{1}{1 + \exp(-2(8 - 4))} = 0,9997$$

Selain *sigmf*, ada pula *Difference Between Two Sigmoidal Membership Function (dsigmf)* dan *Product of Two Sigmoidal Membership Function (psigmf)*. *Dsigmf* dan *psigmf* merupakan gabungan dua *sigmf* (*sigmf* pertumbuhan dan *sigmf* Penyusutan), sehingga memiliki empat parameter $a1, c1, a2, c2$ yang didefinisikan sebagai berikut.

$$f(x; a, c) = \frac{1}{1 + \exp(-a(x - c))}$$

$$f(x) \text{ Dsigmf} = f1(x1; a1, c1) - f2(x2; a2, c2) \quad (74)$$

$$f(x) \text{ Psigmf} = f1(x1; a1, c1) * f2(x2; a2, c2) \quad (75)$$



Gambar 5.8 Grafik Kurva dsigmf

Contoh 5.7 Dsigmf (Matlab & Manual)

```

x = 0:0.1:10;           %variabel
h = [5 2 5 7];          %[a1 c1 a2 c2]
y = dsigmf(x,h);        %fungsi dsigmf
hasil = [x;y];           %input output
plot(x,y)                %grafik kurva dsigmf

```

x	0	1	2	3	4	5	6	7	8	9	10
y	4,54E-05	0,007	0,500	0,993	1,000	1,000	0,993	0,500	0,007	4,54E-05	3,06E-07

Penyelesaian secara manual untuk $x = 8$ adalah?

$$f_1(8; 5,2) = \frac{1}{1 + \exp(-5(8 - 2))} = 0.9999999999999991$$

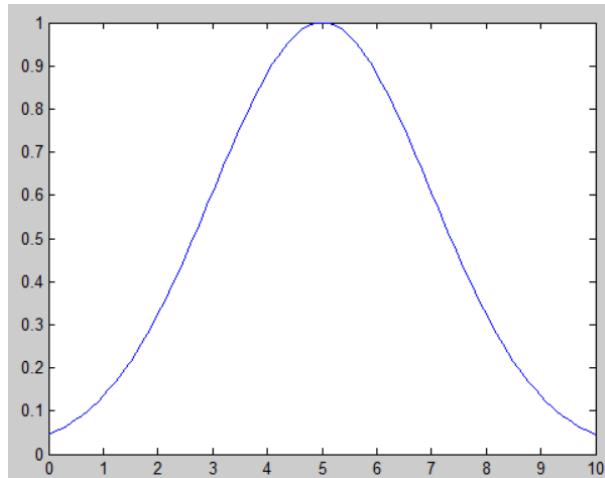
$$f_2(8; 5,7) = \frac{1}{1 + \exp(-5(8 - 7))} = 0,9933$$

$$f(8) = 0.9999999999999991 - 0,9933 = 0,007$$

5.2.6 Gaussian Membership Function

Gaussian Membership Function (gaussmf) memiliki parameter σ dan c , yang mana σ adalah *variance (standard deviation ^ 2)* dan c adalah *mean*. *Gaussmf* didefinisikan sebagai berikut.

$$f(x) = \exp\left(\frac{-(x - c)^2}{2\sigma^2}\right) \quad (76)$$



Gambar 5.9 Grafik Kurva gaussmf

Contoh 5.8 Gaussmf (Matlab & Manual)

```

x = 0:0.1:10;           %variabel
h = [2 5];              %[varian c]
y = gaussmf(x,h);      %fungsi gaussmf
hasil = [x;y];          %input output
plot(x,y)               %grafik kurva gaussmf

```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0439	0,1353	0,3247	0,6065	0,8825	1,0000	0,8825	0,6065	0,3247	0,1353	0,0439

Penyelesaian secara manual untuk $x = 5$ adalah?

$$f(5) = \exp\left(\frac{-(5 - 5)^2}{2\sigma^2}\right) = 1$$

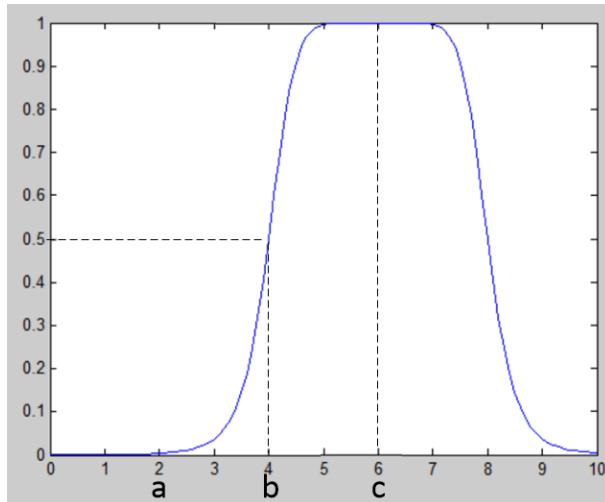
Selain *gaussmf*, terdapat pula *Combination of Two Gaussian Membership Function* (*gauss2mf*) yang merupakan kombinasi dari 2 *gaussmf*, sehingga memiliki empat parameter $\sigma_1, c_1, \sigma_2, c_2$. σ_1 dan c_1 adalah *variance* dan *mean* kurva sebelah kiri, sedangkan σ_2 dan c_2 adalah *variance* dan *mean* kurva sebelah kanan. *Gauss2mf* didefinisikan sebagai berikut.

$$f_k(x) = \exp\left(\frac{-(x - c_k)^2}{2\sigma_k^2}\right); k = 1,2 \quad (77)$$

5.2.7 Generalized Bell-Shaped Membership Function

Generalized Bell-Shaped Membership Function (*gbellmf*) memiliki tiga parameter a, b (berada di tengah kurva), dan c yang didefinisikan sebagai berikut.

$$f(x) = \frac{1}{1 + \left|\frac{x + c}{a}\right|^{2b}} \quad (78)$$



Gambar 5.10 Grafik Kurva gbellmf

Contoh 5.9 Gbellmf (Matlab & Manual)

```
x = 0:0.1:10; %variabel
h = [2 4 6]; %[a b c]
y = gbellmf(x,h); %fungsi gbellmf
hasil = [x;y]; %input output
plot(x,y) %grafik kurva gbellmf
```

x	0	1	2	3	4	5	6	7	8	9	10
y	0,0002	0,0007	0,0039	0,0376	0,5000	0,9961	1,0000	0,9961	0,5000	0,0376	0,0039

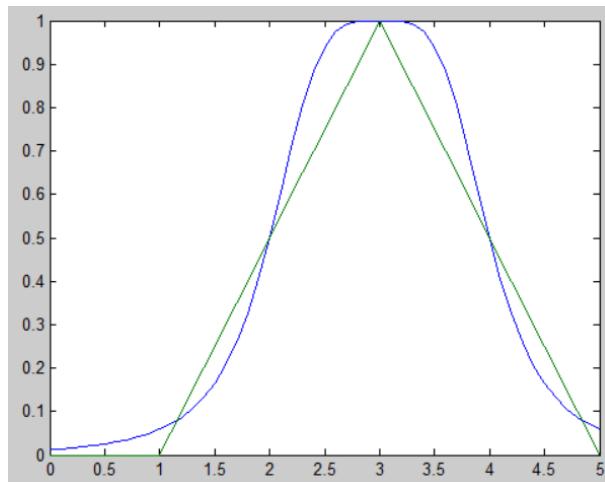
Penyelesaian secara manual untuk $x = 3$ dan 6 adalah?

$$f(3) = \frac{1}{1 + \left| \frac{3+6}{2} \right|^{(2*4)}} = 0,0376$$

$$f(6) = \frac{1}{1 + \left| \frac{6+6}{2} \right|^{(2*4)}} = 1$$

5.2.8 Translate Parameters Between Membership Function

Translate Parameters Between Membership Function (mf2mf) merupakan *Membership Function* yang dapat melakukan transfer parameter-parameter antara dua *Membership Function*. *Mf2mf* memiliki parameter *inParams*, *inType*, dan *outType* yang menghasilkan *outParams* (parameter-parameter output).



Gambar 5.11 Grafik Kurva mf2mf

Contoh 5.10 Mf2mf (Matlab)

```
x = 0:0.1:5; %variabel
mfp1 = [1 2 3]; %parameter yang akan ditransfer ke mfp2
mfp2 = mf2mf(mfp1,'gbellmf','trimf'); %=mfp1
plot(x,gbellmf(x,mfp1),x,trimf(x,mfp2)); %grafik kurva mf2mf
```

5.3 Knowledge Base

Knowledge Base merupakan kumpulan aturan-aturan (*rules*) dalam bentuk pernyataan *IF ... THEN* Biasanya *rules* diperoleh melalui pengetahuan pakar. Dalam *Knowledge Base*, *rules* tersebut dibentuk sebagai berikut.

IF x IS a THEN y IS b

Notasi x dan y adalah *scalar*, sedangkan a dan b adalah himpunan *Fuzzy*. Proposisi setelah *IF* disebut anteseden, sedangkan proposisi setelah *THEN* disebut konsekuensi. Dengan menggunakan operator AND/OR, bentuk rules dapat diperluas menjadi.

IF (x_1 IS a_1) AND/OR (x_2 IS a_2) AND/OR ... AND/OR (x_n IS a_n) *THEN* y IS b

Pada algoritma *Fuzzy Sugeno*, biasanya y IS b merupakan suatu fungsi linier atau konstanta, sehingga dapat ditulis menjadi $y = f(x,y)$, yang mana $f(x,y)$ merupakan fungsi linier dari x dan y .

Secara umum, terdapat tiga fungsi implikasi yang dapat diterapkan pada *rules*, yaitu *MIN*, *MAX*, dan *Dot-Product*. Fungsi implikasi *MIN* digunakan untuk mendapatkan nilai α -predikat dengan cara memotong output himpunan *Fuzzy* sesuai dengan derajat keanggotaan yang terkecil. Fungsi implikasi *MAX* merupakan kebalikan dari *MIN*. Sedangkan *Dot-Product* digunakan untuk mendapatkan nilai α -predikat dengan cara menskalakan output himpunan *Fuzzy* sesuai dengan derajat keanggotaan yang terkecil. Hal ini akan dibahas lebih dalam pada sub pokok bahasan berikutnya (*Machine Inference*).

5.4 Machine Inference

Machine Inference, merupakan proses untuk merubah inputan menjadi output (y_i), yang mana y_i adalah output dari *rule ke-i* berdasarkan nilai α_i. *Machine Inference* menggunakan fungsi implikasi *Max-Min* atau *Dot-Product* untuk memperoleh α-predikat dari *rule ke-i* (α_i) dan output *rule ke-i* (y_i), yang mana y_i merupakan nilai *crisp*. Proses inilah yang merupakan proses penalaran, itulah mengapa *Fuzzy Logic* merupakan kelompok algoritma *reasoning* dalam AI.

Derajat keanggotaan dua atau lebih himpunan *Fuzzy* ($f(x)$ AND/OR/NOT $f(x)$) disebut *fire strength* atau α -predikat. Terdapat beberapa operasi dasar yang sering digunakan untuk memperoleh nilai α -predikat, yaitu *AND* (*intersection/iris*), *OR* (*union/gabungan*), dan *NOT* (*complement*). Nilai α -predikat dari *rule ke-i* (α_i) inilah yang akan digunakan untuk memperoleh nilai *crisp* output *rule ke-i* (y_i).

5.4.1 Operasi Irisan (Intersection)

Operasi irisan menggunakan operator *AND*, yang mana dua himpunan *Fuzzy* A AND B dapat dinyatakan dengan $A \cap B$. Untuk memperoleh α -predikat dari $A \cap B$ digunakan fungsi *MIN* (79) yang berarti bahwa derajat keanggotaan dari $A \cap B$ adalah derajat keanggotaan yang terkecil antara A atau B.

$$\alpha_i = f_{A \cap B} = \min \{f_A(x), f_B(x)\} \quad (79)$$

Contoh 5.11 Machine Inference Fuzzy: Operasi MIN (Manual)

Misalnya diketahui $f(2)$ dari $A = 0,4$ dan $f(7)$ dari $B = 0,3$ pada *rule* pertama, maka α -predikat $A \cap B$ pada *rule ke-1* adalah:

$$\alpha_1 = f_{A \cap B} = \min\{f_A(2), f_B(7)\} = \min\{0,4; 0,3\} = 0,3$$

5.4.2 Operasi Gabungan (Union)

Operasi gabungan menggunakan operator *OR*, yang mana dua himpunan *Fuzzy A OR B* dapat dinyatakan dengan $A \cup B$. Untuk memperoleh α -predikat dari $A \cup B$ digunakan fungsi *MAX* (80) yang berarti bahwa derajat keanggotaan dari $A \cup B$ adalah derajat keanggotaan yang terbesar antara A atau B.

$$\alpha_i = f_{A \cup B} = \max \{f_A(x), f_B(x)\} \quad (80)$$

Contoh 5.12 Machine Inference Fuzzy: Operasi MAX (Manual)

Misalnya diketahui $f(2)$ dari $A = 0,4$ dan $f(7)$ dari $B = 0,3$ pada *rule* pertama, maka α -predikat $A \cup B$ pada *rule ke-1* adalah:

$$\alpha_i = f_{A \cup B} = \max\{f_A(2), f_B(7)\} = \max\{0,4; 0,3\} = 0,4$$

5.4.3 Operasi Komplemen (Complement)

Operasi komplemen menggunakan operator *NOT*, dapat dinyatakan dengan *NOT A* (A^c). Nilai α -predikat dari A^c diperoleh menggunakan persamaan berikut ini.

$$\alpha_i = f_{A^c} = 1 - f_A(x) \quad (81)$$

Contoh 5.13 Machine Inference Fuzzy: Operasi NOT (Manual)

Misalnya diketahui $f(2)$ dari $A = 0,4$ maka α -predikat untuk *NOT A* adalah:

$$\alpha_i = f_{A^c} = 1 - f_A(2) = 1 - 0,4 = 0,6$$

5.5 Defuzzification

Defuzzification, merupakan proses merubah output-output y_i yang diperoleh berdasarkan α_i pada *Machine Inference* menjadi satu nilai *crisp output* (y'). Terdapat dua pendekatan/metode yang biasanya digunakan pada proses *Defuzzification*, yaitu *Average* (82) dan *Centroid* (83).

$$y' = \frac{\sum \alpha_i y_i}{\sum \alpha_i} \quad (82)$$

$$y' = \frac{\int \alpha(y) y dy}{\int \alpha(y) dy} \quad (83)$$

Fuzzy Logic Tsukamoto dan *Fuzzy Logic Sugeno* menggunakan metode *Average*, sedangkan *Fuzzy Logic Mamdani* menggunakan metode *Centroid*.

5.6 Penerapan Fuzzy Logic

Agar dapat memahami cara kerja *Fuzzy Logic*, berikut ini diberikan beberapa contoh penerapan *Fuzzy Logic* yang diselesaikan secara manual dan menggunakan alat bantu Matlab.

5.6.1 Penerapan Fuzzy Logic Tsukamoto

Struktur algoritma *Fuzzy Logic Tsukamoto* adalah sebagai berikut:

- *Fuzzification* menggunakan *Membership Function*.
- *Knowledge Base* menggunakan model *rule*:
IF (x_1 IS a_1) *AND/OR* (x_2 IS a_2) ... *AND/OR* (x_n IS a_n) *THEN* (y IS b)
- Secara standar, *Machine Inference* menggunakan fungsi implikasi *Min* untuk memperoleh a_i , yang digunakan untuk memperoleh y .
- Secara standar, *Defuzzification* menggunakan metode *Average* (82).

Algoritma *Fuzzy Logic Tsukamoto* tidak terdapat pada Rapidminer dan bahkan Matlab, sehingga contoh penerapannya hanya disajikan dalam bentuk manual.

Contoh 5.14 Fuzzy Logic Tsukamoto (Manual)

Diketahui:

Variabel	Satuan	Tipe	Himpunan	Membership Function
kecepatan	[1000 5000] rpm	Input (x1)	lambat cepat	trimf[1000 1000 5000] trimf[1000 5000 5000]
suhu ruangan	[100 600] kelvin	Input (x2)	rendah tinggi	trimf[100 100 600] trimf[100 600 600]
frekuensi putar	[2000 7000] rpm	Output (y)	kecil besar	trimf[2000 2000 7000] trimf[2000 7000 7000]

Berapa sumber frekuensi putar kipas angin (y) saat kecepatan ($x1$) = 4000 rpm dan suhu ruangan ($x2$) = 300 kelvin?

$f_{x1,lambat}(4000)$: trimf(4000, [1000 1000 5000]), gunakan Persamaan (68)

$$f_{x1,lambat}(4000) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{5000 - 4000}{5000 - 1000} = \frac{1000}{4000} = 0,25$$

$f_{x1,cepat}(4000)$: trimf(4000, [1000 5000 5000]), gunakan Persamaan (68)

$$f_{x1,cepat}(4000) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{4000 - 1000}{5000 - 1000} = \frac{3000}{4000} = 0,75$$

$f_{x2,rendah}(300)$: trimf(300, [100 100 600]), gunakan Persamaan (68)

$$f_{x2,rendah}(300) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{600 - 300}{600 - 100} = \frac{300}{500} = 0,60$$

$f_{x2,tinggi}(300)$: trimf(300, [100 600 600]), gunakan Persamaan (68)

$$f_{x2,tinggi}(300) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{300 - 100}{600 - 100} = \frac{200}{500} = 0,40$$

$f_{y,\text{kecil}}(y)$: trimf(y , [2000 2000 7000]), gunakan Persamaan (68)

$$f_{y,\text{kecil}}(y) = \begin{cases} 0; & y \leq 2000 \\ (7000 - y)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

$f_{y,\text{besar}}(y)$: trimf(y , [2000 7000 7000]), gunakan Persamaan (68)

$$f_{y,\text{besar}}(y) = \begin{cases} 0; & y \leq 2000 \\ (y - 2000)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

R1: IF kecepatan IS lambat AND suhu IS rendah THEN frekuensi IS kecil

R2: IF kecepatan IS lambat AND suhu IS tinggi THEN frekuensi IS kecil

R3: IF kecepatan IS cepat AND suhu IS rendah THEN frekuensi IS besar

R4: IF kecepatan IS cepat AND suhu IS tinggi THEN frekuensi IS besar

$$\begin{aligned} \alpha_1 &= f_{x1,\text{lambat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,25; 0,60\} = 0,25 \end{aligned}$$

$$y_1 = f_{y,\text{kecil}}(y) = \frac{7000 - y}{7000 - 2000} = 0,25 \rightarrow 7000 - 5000 * 0,25 = 5750$$

$$\begin{aligned} \alpha_2 &= f_{x1,\text{lambat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,25; 0,40\} = 0,25 \end{aligned}$$

$$y_2 = f_{y,\text{kecil}}(y) = \frac{7000 - y}{7000 - 2000} = 0,25 \rightarrow 7000 - 5000 * 0,25 = 5750$$

$$\begin{aligned} \alpha_3 &= f_{x1,\text{cepat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,75; 0,60\} = 0,60 \end{aligned}$$

$$y_3 = f_{y,\text{besar}}(y) = \frac{y - 2000}{7000 - 2000} = 0,60 \rightarrow 2000 + 5000 * 0,60 = 5000$$

$$\begin{aligned} \alpha_4 &= f_{x1,\text{cepat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,75; 0,40\} = 0,40 \end{aligned}$$

$$y_4 = f_{y,\text{besar}}(y) = \frac{y - 2000}{7000 - 2000} = 0,40 \rightarrow 2000 + 5000 * 0,40 = 4000$$

Defuzzification Fuzzy Logic Tsukamoto menggunakan Persamaan (82).

$$\begin{aligned} y' &= \frac{\sum \alpha_i y_i}{\sum \alpha_i} = \frac{0,25 * 5750 + 0,25 * 5750 + 0,60 * 5000 + 0,40 * 4000}{0,25 + 0,25 + 0,60 + 0,40} \\ &= 4983,33 \end{aligned}$$

Dengan demikian, frekuensi putar kipas angin yang dihasilkan = 4983 rpm.

5.6.2 Penerapan Fuzzy Logic Mamdani

Struktur algoritma *Fuzzy Logic Mamdani* adalah sebagai berikut:

- *Fuzzification* menggunakan *Membership Function*.
- *Knowledge Base* menggunakan model *rule* yang sama seperti *Fuzzy Logic Tsukamoto*.
 $IF (x_1 \text{ IS } a_1) AND/OR (x_2 \text{ IS } a_2) \dots AND/OR (x_n \text{ IS } a_n) THEN (y \text{ IS } b)$
- Secara standar, *Machine inference* menggunakan fungsi implikasi *Min* dan komposisi antar *rule* menggunakan fungsi *Max*, menghasilkan himpunan *Fuzzy* baru.
- Secara standar, *Defuzzification* menggunakan metode *Centroid* (83).

Contoh 5.15 Fuzzy Logic Mamdani (Manual & Matlab)

Diketahui:

Variabel	Satuan	Tipe	Himpunan	Membership Function
kecepatan	[1000 5000] rpm	Input (x1)	lambat cepat	trimf[1000 1000 5000] trimf[1000 5000 5000]
suhu ruangan	[100 600] kelvin	Input (x2)	rendah tinggi	trimf[100 100 600] trimf[100 600 600]
frekuensi putar	[2000 7000] rpm	Output (y)	kecil besar	trimf[2000 2000 7000] trimf[2000 7000 7000]

Berapa sumber frekuensi putar kipas angin (y) saat kecepatan (x_1) = 4000 rpm dan suhu ruangan (x_2) = 300 kelvin?

Tahap *Fuzzification* sama dengan *Fuzzy Logic Tsukamoto*. Tahap *Machine Inference* sama dengan *Fuzzy Logic Tsukamoto* untuk nilai α_i , namun nilai y_i diperoleh melalui pendekatan komposisi *rules* menggunakan fungsi *MAX*. Sedangkan tahap *Defuzzification* menggunakan metode *centroid*.

$f_{x1,\text{lambat}}(4000)$: trimf(4000, [1000 1000 5000]), gunakan Persamaan (68)

$$f_{x1,\text{lambat}}(4000) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{5000 - 4000}{5000 - 1000} = \frac{1000}{4000} = 0,25$$

$f_{x1,\text{cepat}}(4000)$: trimf(4000, [1000 5000 5000]), gunakan Persamaan (68)

$$f_{x1,\text{cepat}}(4000) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{4000 - 1000}{5000 - 1000} = \frac{3000}{4000} = 0,75$$

$f_{x2,\text{rendah}}(300)$: trimf(300, [100 100 600]), gunakan Persamaan (68)

$$f_{x2,\text{rendah}}(300) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{600 - 300}{600 - 100} = \frac{300}{500} = 0,60$$

$f_{x2,\text{tinggi}}(300)$: trimf(300, [100 600 600]), gunakan Persamaan (68)

$$f_{x2,\text{tinggi}}(300) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{300 - 100}{600 - 100} = \frac{200}{500} = 0,40$$

$f_{y,\text{kecil}}(y)$: trimf(y, [2000 2000 7000]), gunakan Persamaan (68)

$$f_{y,\text{kecil}}(y) = \begin{cases} 0; & y \leq 2000 \\ (7000 - y)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

$f_{y,\text{besar}}(y)$: trimf(y, [2000 7000 7000]), gunakan Persamaan (68)

$$f_{y,\text{besar}}(y) = \begin{cases} 0; & y \leq 2000 \\ (y - 2000)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

R1: IF kecepatan IS lambat AND suhu IS rendah THEN frekuensi IS kecil

R2: IF kecepatan IS lambat AND suhu IS tinggi THEN frekuensi IS kecil

R3: IF kecepatan IS cepat AND suhu IS rendah THEN frekuensi IS besar

R4: IF kecepatan IS cepat AND suhu IS tinggi THEN frekuensi IS besar

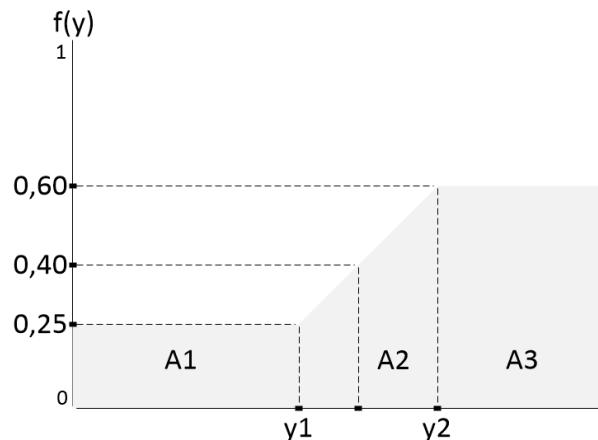
$$\begin{aligned} \alpha_1 &= f_{x1,\text{lambat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,25; 0,60\} = 0,25 \end{aligned}$$

$$\begin{aligned} \alpha_2 &= f_{x1,\text{lambat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,25; 0,40\} = 0,25 \end{aligned}$$

$$\begin{aligned} \alpha_3 &= f_{x1,\text{cepat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,75; 0,60\} = 0,60 \end{aligned}$$

$$\begin{aligned} \alpha_4 &= f_{x1,\text{cepat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,75; 0,40\} = 0,40 \end{aligned}$$

Komposisi *rules* menggunakan fungsi *MAX* adalah:



$$y_1 = \frac{y - 2000}{7000 - 2000} = 0,25 \rightarrow 2000 + 5000 * 0,25 = 3250$$

$$y_2 = \frac{y - 2000}{7000 - 2000} = 0,60 \rightarrow 2000 + 5000 * 0,60 = 5000$$

Dengan demikian, *Membership Function* untuk himpunan *Fuzzy* yang baru adalah:

$$f(y) = \begin{cases} 0,25; & y \leq 3250 \\ (y - 2000)/(7000 - 2000); & 3250 \leq y \leq 5000 \\ 0,60; & y \geq 5000 \end{cases}$$

Defuzzification Fuzzy Logic Mamdani menggunakan Persamaan (83).

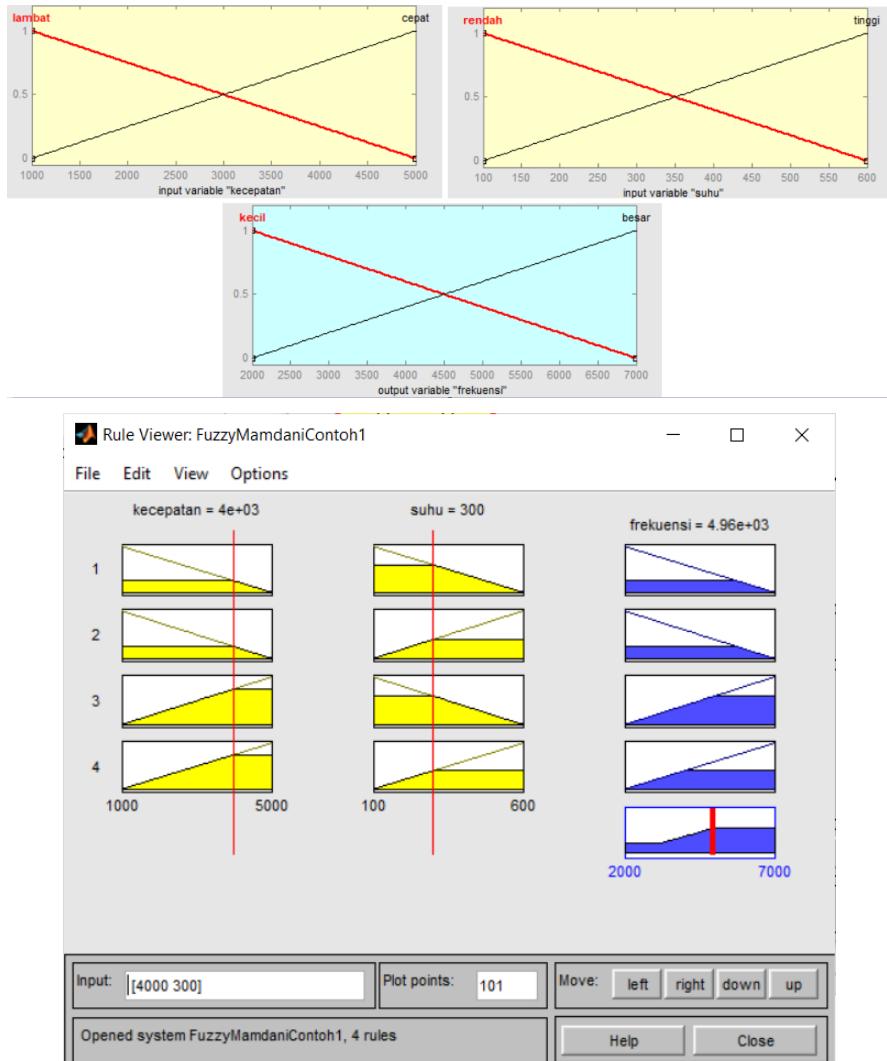
$$y' = \frac{\int \alpha(y)y dy}{\int \alpha(y) dy} = \frac{\int_0^{3250} 0,25y dy + \int_{3250}^{5000} \frac{y - 2000}{(7000 - 2000)} y dy + \int_{5000}^{7000} 0,6y dy}{\int_0^{3250} 0,25y dy + \int_{3250}^{5000} \frac{y - 2000}{(7000 - 2000)} dy + \int_{5000}^{7000} 0,6y dy}$$

$$y' = \frac{1320312,5 + 3187515,625 + 7200000}{812,5 + 743,75 + 1200} = 4247,74$$

Dengan demikian, frekuensi putar kipas angin yang dihasilkan = 4248 rpm.

Berikut ini penyelesaiannya menggunakan alat bantu Matlab:

```
[System]
Name='FuzzyMamdaniContoh1'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=4
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'
[Input1]
Name='kecepatan'
Range=[1000 5000]
NumMFs=2
MF1='lambat':'trimf',[1000 1000 5000]
MF2='cepat':'trimf',[1000 5000 5000]
[Input2]
Name='suhu'
Range=[100 600]
NumMFs=2
MF1='rendah':'trimf',[100 100 600]
MF2='tinggi':'trimf',[100 600 600]
[Output1]
Name='frekuensi'
Range=[2000 7000]
NumMFs=2
MF1='kecil':'trimf',[2000 2000 7000]
MF2='besar':'trimf',[2000 7000 7000]
[Rules]
1 1, 1 (1) : 1
1 2, 1 (1) : 1
2 1, 2 (1) : 1
2 2, 2 (1) : 1
```

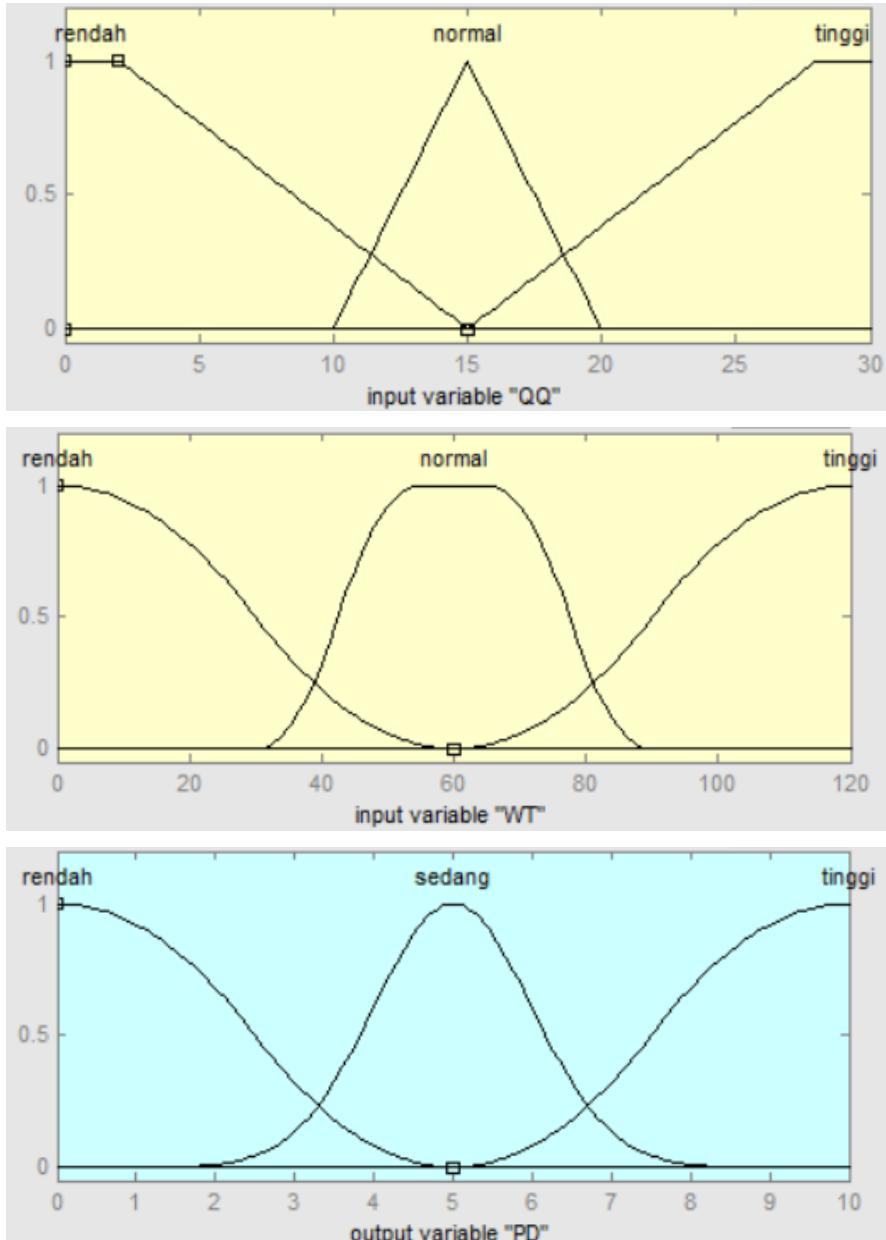


Contoh 5.16 Fuzzy Logic Mamdani: Traffic Light Control (Matlab)

Misalnya variabel setiap ruas pada simpang 4 adalah sebagai berikut:

Variabel	Satuan	Tipe	Himpunan	Membership Function
QQ (Queues Quantity)	[0 30]	Kendaran	Input (x1)	rendah normal tinggi
WT (Waiting Times)	[0 120]	Detik	Input (x2)	rendah normal tinggi
PD (Priority Degree)	[0 10]	-	Output (y)	rendah sedang tinggi

Lampu hijau menyala pada ruas yang memiliki nilai output (PD) tertinggi.



```
function [allPD] = FuzzyMamdaniaa(inputRuasA, inputRuasB, inputRuasC,
inputRuasD)
a = newfis('TLC','mamdani','min','max','min','max','centroid');
%% Input1: QQ (Queues Quantity), antrean kendaraan
a = addvar(a,'input','QQ',[0 30]);
a = addmf(a,'input',1,'Rendah','trapmf',[0 0 2 15]);
a = addmf(a,'input',1,'Normal','trimf',[10 15 20]);
a = addmf(a,'input',1,'Tinggi','trapmf',[15 28 30 30]);
```

```

%% Input2: WT (Waiting Times), waktu tunggu dalam detik
a = addvar(a,'input','WT',[0 120]);
a = addmf(a,'input',2,'Rendah','zmf',[0 60]);
a = addmf(a,'input',2,'Normal','pimf',[30 55 65 90]);
a = addmf(a,'input',2,'Tinggi','smf',[60 120]);
%% Output" PD (Priority Degree), prioritas lampu hijau menyala
a = addvar(a,'output','PD',[0 10]);
a = addmf(a,'output',1,'Rendah','zmf',[0 5]);
a = addmf(a,'output',1,'Sedang','gaussmf',[1 5]);
a = addmf(a,'output',1,'Tinggi','smf',[5 10]);
%% Set Rules: input1 input2 output weight OR=2/AND=1
ruleList=[...
    1 1 1 1 1
    1 2 1 1 1
    1 3 2 1 1
    2 1 1 1 1
    2 2 2 1 1
    2 3 3 1 1
    3 1 2 1 1
    3 2 3 1 1
    3 3 3 1 1];
a = addrule(a,ruleList);
%% Output tiap-tiap ruas
PD1 = evalfis(inputRuasA, a);
PD2 = evalfis(inputRuasB, a);
PD3 = evalfis(inputRuasC, a);
PD4 = evalfis(inputRuasD, a);
allPD = [1,PD1; 2,PD2; 3,PD3; 4,PD4];
end

```

Berikut ini *script* untuk menggunakan fungsi di atas:

```

clc; clear; close all; warning off all;
[PD] = FuzzyMamdaniaa([30 1], [30 60], [0 0], [1 60]);
lampaHijau = sortrows(PD,2);
disp(lampaHijau)
disp('Maka lampu hijau menyala pada ruas:')
disp(lampaHijau(4,1))

```

Inputan pada *script* di atas adalah sebagai berikut:

Ruas 1 = [30 1]: QQ = 30 antrian kendaraan, WT = 1 detik.

Ruas 2 = [30 60]: QQ = 30 antrian kendaraan, WT = 60 detik.

Ruas 3 = [0 0]: QQ = 0 antrian kendaraan, WT = 0 detik.

Ruas 4 = [1 60]: QQ = 1 antrian kendaraan, WT = 60 detik.

Hasilnya adalah:

PD ruas 1 = 5,0000

PD ruas 2 = 8,5706

PD ruas 3 = 1,4294

PD ruas 4 = 1,4294

Dengan demikian, lampu hijau menyala pada ruas 2 (PD tertinggi).

5.6.3 Penerapan Fuzzy Logic Sugeno

Struktur algoritma *Fuzzy Logic Sugeno* adalah sebagai berikut:

- *Fuzzification* menggunakan *Membership Function*.
- *Knowledge Base* menggunakan model *rule*:
 $IF (x_1 \text{ IS } a_1) \text{ AND/OR } (x_2 \text{ IS } a_2) \dots \text{ AND/OR } (x_n \text{ IS } a_n) \text{ THEN } y = f(x,y)$
 $f(x,y)$ adalah fungsi *crisp* yang biasanya merupakan fungsi linier dari x dan y . Dengan demikian, output setiap *rule* berupa konstanta atau persamaan linier.
- Secara standar, *Machine Inference* menggunakan fungsi implikasi *Min* untuk memperoleh α_i yang digunakan untuk memperoleh y_i .
- Secara standar, *Defuzzification* menggunakan metode *Average* (82).

Contoh 5.17 Fuzzy Logic Sugeno (Manual & Matlab)

Diketahui:

Variabel	Satuan	Tipe	Himpunan	Membership Function
kecepatan	[1000 5000] rpm	Input (x1)	lambat cepat	trimf[1000 1000 5000] trimf[1000 5000 5000]
suhu ruangan	[100 600] kelvin	Input (x2)	rendah tinggi	trimf[100 100 600] trimf[100 600 600]
frekuensi putar	[2000 7000] rpm	Output (y)	constant1 constant2 constant3 constant4	2*Kecepatan-4000 0,5*Kecepatan+1700 Kecepatan+700 0,5*Kecepatan+2000

Berapa sumber frekuensi putar kipas angin (y) saat kecepatan ($x1$) = 4000 rpm dan suhu ruangan ($x2$) = 300 kelvin?

Pada prinsipnya algoritma *Fuzzy Logic Sugeno* sama dengan *Fuzzy Logic Tsukamoto*, perbedaannya pada proses *Machine Learning*, yang mana output dari setiap *rule* bersifat linier atau konstan (bukan menggunakan *Membership Function* dari variabel output yang digunakan pada tahap *Fuzzification*).

$f_{x1,\text{lambat}}(4000)$: trimf(4000, [1000 1000 5000]), gunakan Persamaan (68)

$$f_{x1,\text{lambat}}(4000) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{5000 - 4000}{5000 - 1000} = \frac{1000}{4000} = 0,25$$

$f_{x1,\text{cepat}}(4000)$: trimf(4000, [1000 5000 5000]), gunakan Persamaan (68)

$$f_{x1,\text{cepat}}(4000) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{4000 - 1000}{5000 - 1000} = \frac{3000}{4000} = 0,75$$

$f_{x2,\text{rendah}}(300)$: trimf(300, [100 100 600]), gunakan Persamaan (68)

$$f_{x2,\text{rendah}}(300) = \frac{c - x}{c - b}; b \leq x \leq c = \frac{600 - 300}{600 - 100} = \frac{300}{500} = 0,60$$

$f_{x2,\text{tinggi}}(300)$: trimf(300, [100 600 600]), gunakan Persamaan (68)

$$f_{x2,\text{tinggi}}(300) = \frac{x - a}{b - a}; a \leq x \leq b = \frac{300 - 100}{600 - 100} = \frac{200}{500} = 0,40$$

$f_{y,\text{kecil}}(y)$: trimf(y, [2000 2000 7000]), gunakan Persamaan (68)

$$f_{y,\text{kecil}}(y) = \begin{cases} 0; & y \leq 2000 \\ (7000 - y)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

$f_{y,\text{besar}}(y)$: trimf(y, [2000 7000 7000]), gunakan Persamaan (68)

$$f_{y,\text{besar}}(y) = \begin{cases} 0; & y \leq 2000 \\ (y - 2000)/(7000 - 2000); & 2000 \leq y \leq 7000 \\ 1; & y \geq 7000 \end{cases}$$

R1: IF kecepatan IS lambat AND suhu IS rendah THEN frekuensi IS constant1

R2: IF kecepatan IS lambat AND suhu IS tinggi THEN frekuensi IS constant2

R3: IF kecepatan IS cepat AND suhu IS rendah THEN frekuensi IS constant3

R4: IF kecepatan IS cepat AND suhu IS tinggi THEN frekuensi IS constant4

$$\begin{aligned} \alpha_1 &= f_{x1,\text{lambat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,25; 0,60\} = 0,25 \end{aligned}$$

$$\begin{aligned} \alpha_2 &= f_{x1,\text{lambat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{lambat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,25; 0,40\} = 0,25 \end{aligned}$$

$$\begin{aligned} \alpha_3 &= f_{x1,\text{cepat} \cap x2,\text{rendah}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{rendah}}(300)\} \\ &= \min\{0,75; 0,60\} = 0,60 \end{aligned}$$

$$\begin{aligned} \alpha_4 &= f_{x1,\text{cepat} \cap x2,\text{tinggi}} = \min\{f_{x1,\text{cepat}}(4000); f_{x2,\text{tinggi}}(300)\} \\ &= \min\{0,75; 0,40\} = 0,40 \end{aligned}$$

$$\text{constant1} = 2 * \text{kecepatan} - 4000 \rightarrow y_1 = 2 * 4000 - 4000 = 4000$$

$$\text{constant2} = 0,5 * \text{kecepatan} + 1700 \rightarrow y_2 = 0,5 * 4000 + 1700 = 3700$$

$$\text{constant3} = \text{kecepatan} + 700 \rightarrow y_3 = 4000 + 700 = 4700$$

$$\text{constant4} = 0,5 * \text{kecepatan} + 2000 \rightarrow y_4 = 0,5 * 4000 + 2000 = 4000$$

Defuzzification Fuzzy Logic Sugeno menggunakan Persamaan (82).

$$\begin{aligned} y' &= \frac{\sum \alpha_i y_i}{\sum \alpha_i} = \frac{0,25 * 4000 + 0,25 * 3700 + 0,60 * 4700 + 0,40 * 4000}{0,25 + 0,25 + 0,60 + 0,40} \\ &= 4230 \end{aligned}$$

Dengan demikian, frekuensi putar kipas angin yang dihasilkan = 4230 rpm.

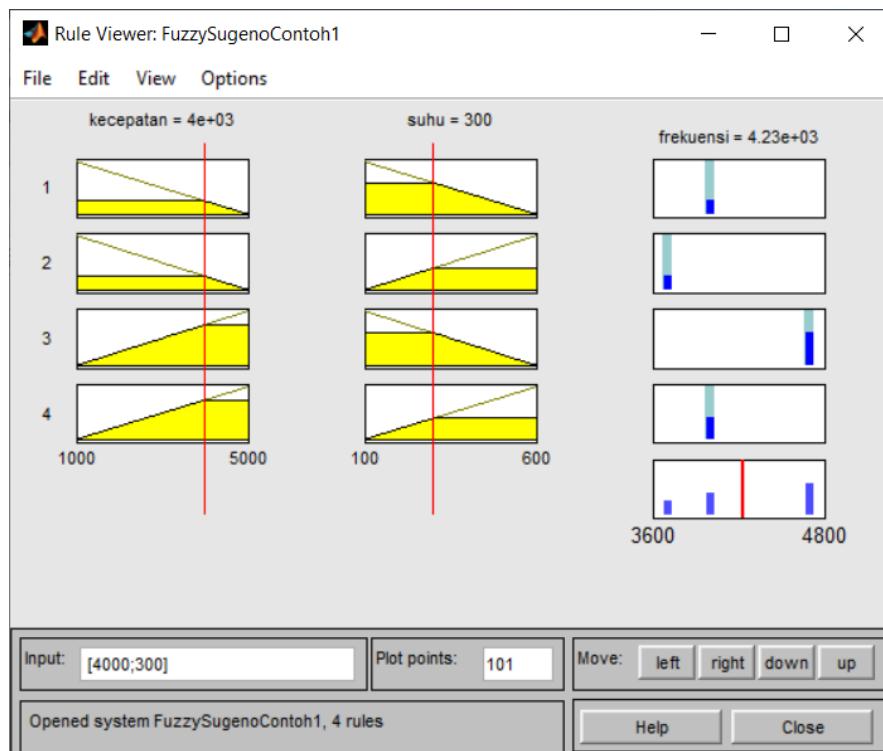
Berikut ini penyelesaiannya menggunakan alat bantu Matlab:

```
[System]
Name='FuzzyMamdaniContoh1'
Type='sugeno'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=4
AndMethod='min'
OrMethod='max'
ImpMethod='prod'
```

```

AggMethod='sum'
DefuzzMethod='wtaver'
[Input1]
Name='kecepatan'
Range=[1000 5000]
NumMFs=2
MF1='lambat':'trimf',[1000 1000 5000]
MF2='cepat':'trimf',[1000 5000 5000]
[Input2]
Name='suhu'
Range=[100 600]
NumMFs=2
MF1='rendah':'trimf',[100 100 600]
MF2='tinggi':'trimf',[100 600 600]
[Output1]
Name='frekuensi'
Range=[0 1]
NumMFs=4
MF1='constant1':'constant',[4000]
MF2='constant2':'constant',[3700]
MF3='constant3':'constant',[4700]
MF4='constant4':'constant',[4000]
[Rules]
1 1, 1 (1) : 1
1 2, 2 (1) : 1
2 1, 3 (1) : 1
2 2, 4 (1) : 1

```



5.7 Soal Latihan Fuzzy Logic

Kumpulkan 50 data mahasiswa seperti berikut ini (variabel output tidak perlu diisi):

Variabel	Satuan	Tipe	Himpunan	Membership Function
Nilai Toelf	[0 600]	Toelf	Rendah	trapmf[0 0 200 300]
			Sedang	trimf[250 300 350]
			Tinggi	trapmf[300 400 600 600]
Nilai IPK	[0 4]	IPK	Buruk	trapmf[0 0 1 2]
			Cukup	trimf[1,5 2 2,5]
			Bagus	trapmf[2 3 4 4]
Penghasilan Orang Tua	[0 16]	Juta	Sedikit	trapmf[0 0 6 8]
			Sedang	trimf[7 8 9]
			Banyak	trapmf[8 10 16 16]
Beasiswa	[0 10]	-	Output (y)	Tidak Ya
				zmf[0 8] smf[7 10]

Rules:

IF (x1 IS rendah) AND (x2 IS buruk) AND (x3 IS sedikit) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS buruk) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS buruk) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS cukup) AND (x3 IS sedikit) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS cukup) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS cukup) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS bagus) AND (x3 IS sedikit) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS bagus) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS rendah) AND (x2 IS bagus) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS buruk) AND (x3 IS sedikit) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS buruk) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS buruk) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS cukup) AND (x3 IS sedikit) THEN y IS ya
 IF (x1 IS sedang) AND (x2 IS cukup) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS cukup) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS bagus) AND (x3 IS sedikit) THEN y IS ya
 IF (x1 IS sedang) AND (x2 IS bagus) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS sedang) AND (x2 IS bagus) AND (x3 IS banyak) THEN y IS tidak
 IF (x1 IS tinggi) AND (x2 IS buruk) AND (x3 IS sedikit) THEN y IS tidak
 IF (x1 IS tinggi) AND (x2 IS buruk) AND (x3 IS sedang) THEN y IS tidak
 IF (x1 IS tinggi) AND (x2 IS buruk) AND (x3 IS banyak) THEN y IS tidak

IF (x1 IS tinggi) AND (x2 IS cukup) AND (x3 IS sedikit) THEN y IS ya
IF (x1 IS tinggi) AND (x2 IS cukup) AND (x3 IS sedang) THEN y IS tidak
IF (x1 IS tinggi) AND (x2 IS cukup) AND (x3 IS banyak) THEN y IS tidak
IF (x1 IS tinggi) AND (x2 IS bagus) AND (x3 IS sedikit) THEN y IS ya
IF (x1 IS tinggi) AND (x2 IS bagus) AND (x3 IS sedang) THEN y IS ya
IF (x1 IS tinggi) AND (x2 IS bagus) AND (x3 IS banyak) THEN y IS tidak

Soal:

1. Berdasarkan nilai derajat keanggotaan, siapa saja mahasiswa yang memiliki nilai Toelf (x1) = Tinggi, IPK (x2) = Bagus, dan Penghasilan Orang Tua (x3) = Sedikit (gunakan *Fuzzy Logic Tsukamoto* secara manual)?
2. Berdasarkan soal nomor 1, tentukan mahasiswa yang berhak menerima beasiswa, yang mana jika α -predikat dari ($x_1 = \text{Tinggi}$ AND $x_2 = \text{Bagus}$ AND $x_3 = \text{Sedikit}$) > 0 , maka mahasiswa tersebut berhak menerima beasiswa.
3. Tentukan nilai output (beasiswa) setiap mahasiswa menggunakan algoritma *Fuzzy Logic Mamdani* (gunakan Matlab atau alat bantu lainnya yang anda kuasai).

6. ANN, SVM, & Fuzzy

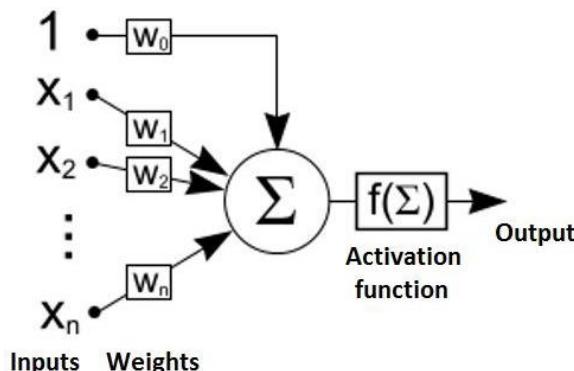
No.	Materi	Tujuan Pembelajaran
1.	ANN - Backpropagation	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>Backpropagation</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.
2.	Adaptive Neuro Fuzzy Inference System	Anda mampu memahami, menjelaskan, dan menerapkan algoritma ANFIS secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.
3.	Support Vector Machine	Anda mampu memahami, menjelaskan, dan menerapkan algoritma <i>Binary SVM</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.
4.	Multi Class SVM	Anda mampu memahami, menjelaskan, dan menerapkan pendekatan 1V1 dan 1VR pada SVM secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.
5.	Fuzzy SVM	Anda mampu memahami, menjelaskan, dan menerapkan algoritma FSVM secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.

6.1 ANN- Backpropagation

Artificial Neural Network (ANN) merupakan salah satu algoritma *machine learning* yang dapat digunakan untuk estimasi/regresi dan klasifikasi. ANN bekerja meniru cara kerja otak manusia dari sisi: (1) Pengetahuan yang diperoleh oleh *network* dari lingkungan, melalui suatu proses pembelajaran; (2) Kekuatan koneksi antar unit yang disebut *synaptic weights*, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh *network* tersebut [43]. Pada tahun 1943, Mc Culloch dan Pitts memperkenalkan model matematika yang merupakan penyederhanaan dari struktur sel saraf yang sebenarnya [43] yang ditunjukkan pada Persamaan (84). Berawal dari diperkenalkannya model matematika *neuron* tersebut, ANN berkembang cukup pesat, dan mencapai puncak keemasan pertama pada era tahun 60-an, dan puncak kedua pada pertengahan tahun 80-an [43].

$$y = f(\sum_{i=1}^n x_i w_i) \quad (84)$$

Korelasi antara ketiga komponen pada persamaan di atas yaitu: Signal (x) berupa vektor berdimensi n (x_1, x_2, \dots, x_n) akan mengalami penguatan oleh *synapse* w (w_1, w_2, \dots, w_n). Selanjutnya akumulasi dari penguatan tersebut akan mengalami transformasi oleh fungsi aktifasi f . Fungsi f ini akan memonitor, bila akumulasi penguatan signal itu telah melebihi batas tertentu, maka sel *neuron* yang semula berada dalam kondisi “0”, akan mengeluarkan signal “1”. Berdasarkan nilai *output* tersebut (y), sebuah *neuron* dapat berada dalam dua status: “0” atau “1”. *Neuron* disebut dalam kondisi *firing* bila menghasilkan output bernilai “1”.



Gambar 6.1 Arsitektur Jaringan ANN

Gambar 6.1 menunjukkan bahwa suatu jaringan ANN memiliki tiga komponen, yaitu *synapse* (w_1, w_2, \dots, w_n), alat penambah (*adder*), dan fungsi aktifasi (f). Berdasarkan arsitekturnya, ANN dapat dikategorikan menjadi 3 jenis, antara lain *singlelayer neural network*, *multilayer neural network*, dan *recurrent neural network*. Terdapat banyak pengembangan metode ANN seperti yang ditunjukkan pada Gambar 2.2. Salah satu metode ANN yang umum digunakan adalah *Backpropagation*.

Ciri algoritma *Backpropagation* adalah berusaha meminimalkan *error* pada output yang dihasilkannya. Pada prinsipnya, pelatihan metode *Backpropagation* terdiri dari tiga langkah, yaitu:

1. Data dimasukkan ke *input* jaringan (*feedforward*);
2. Perhitungan dan propagasi balik dari *error* yang dihasilkan; dan
3. Pembaharuan (*adjustment*) bobot dan bias.

Notasi yang digunakan *Backpropagation* dalam tahap pelatihan, yaitu:

x_i	: Unit input ke- <i>i</i> .
z_j	: <i>Hidden</i> unit ke- <i>j</i> .
y_k	: Unit output data ke- <i>k</i> .
v_{0j}	: Bias untuk <i>hidden</i> unit ke- <i>j</i> .
v_{ij}	: Bobot antara unit <i>input</i> ke- <i>i</i> dengan <i>hidden</i> unit ke- <i>j</i> .
w_{0k}	: Bias untuk unit <i>output</i> ke- <i>k</i> .
w_{jk}	: Bobot antara <i>hidden</i> unit ke- <i>j</i> dengan unit <i>output</i> ke- <i>k</i> .
δ_k	: Faktor koreksi <i>error</i> untuk bobot w_{jk} .
δ_j	: Faktor koreksi <i>error</i> untuk bobot v_{ij} .
α	: <i>Learning rate</i> , mengontrol perubahan bobot selama pelatihan.
m	: Momentum.

Tahap-tahap pelatihan metode *Backpropagation*, yaitu:

1. Inisialisasi bobot dan bias. Baik bobot maupun bias dapat diset acak dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif).
2. Jika *stopping condition* masih belum terpenuhi, jalankan tahap 4 – 10.
3. Untuk setiap data latih, lakukan tahap 4 – 9.
4. Setiap unit input (x_i , $i = 1, 2, \dots, n$) menerima sinyal input x_i dan menyebarkan sinyal tersebut pada seluruh unit pada *hidden layer*.
5. Setiap *hidden unit* (z_j , $j = 1, 2, \dots, p$) akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya.

$$z_{in_j} = V_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (85)$$

Gunakan fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari *hidden unit* yang bersangkutan.

$$z_j = f(z_{in_j}) \quad (86)$$

Selanjutnya mengirim sinyal output ini ke seluruh unit pada unit output.

6. Setiap unit output (y_k , $k = 1, 2, \dots, m$) menjumlahkan sinyal-sinyal input yang sudah berbobot termasuk biasnya.

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (87)$$

Gunakan fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari unit output yang bersangkutan.

$$z_j = f(z_{in_j}) \quad (88)$$

7. Propagasi balik *error* (*backpropagation of error*). Setiap unit output (y_k , $k=1, 2, \dots, m$) menerima suatu target (output yang diharapkan) yang akan dibandingkan dengan output yang dihasilkan.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (89)$$

Faktor δ_k ini digunakan untuk menghitung koreksi *error* (Δw_{jk}) yang nantinya akan dipakai untuk memperbarui w_{jk} .

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (90)$$

Selain itu juga dihitung koreksi bias Δw_{0k} yang nantinya akan dipakai untuk memperbarui w_{0k} .

$$\Delta w_{0k} = \alpha \delta_k \quad (91)$$

Faktor δ_k ini kemudian dikirimkan ke *layer* di depannya.

8. Setiap *hidden unit* (z_j , $j = 1, 2, \dots, p$) menjumlahkan input delta (yang dikirim dari *layer* pada tahap 6) yang sudah berbobot.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (92)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi δ_j .

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (93)$$

Faktor δ_j ini digunakan untuk menghitung koreksi *error* (Δv_{ij}) yang nantinya akan dipakai untuk memperbarui v_{ij} .

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (94)$$

Selain itu juga dihitung koreksi bias Δv_{0j} yang nantinya akan dipakai untuk memperbarui v_{0j} .

$$\Delta v_{0j} = \alpha \delta_j \quad (95)$$

9. Pembaharuan bobot dan bias: Setiap unit output (y_k , $k=1, \dots, m$) akan memperbarui bias dan bobotnya dengan setiap *hidden unit*.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (96)$$

Demikian pula untuk setiap *hidden unit* akan memperbarui bias dan bobotnya dengan setiap unit input.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (97)$$

10. Memeriksa *stopping condition*. Jika *stop condition* telah terpenuhi, maka pelatihan jaringan dapat dihentikan. Untuk menentukan *stopping condition* terdapat dua cara yang biasa dipakai, yaitu: (1) Membatasi iterasi yang ingin dilakukan. Misalkan jaringan akan dilatih sampai iterasi yang ke-500. Yang

dimaksud dengan satu iterasi adalah tahap 4 sampai tahap 9 untuk semua data latih yang ada; (2) Membatasi *error*. Misalnya menentukan besar antara output yang dikehendaki dan output yang dihasilkan oleh jaringan menggunakan metode estimasi *error* MSE.

Cara mendapatkan output dari model metode *Backpropagation* yang telah dilatih yaitu dengan mengimplementasikan model metode *Backpropagation* yang sama seperti proses pelatihan, tetapi hanya pada bagian umpan majunya saja. Notasi yang digunakan dalam tahap pengujian maupun prediksi data baru, yaitu:

- x_i : Unit input ke-*i*.
- z_j : *Hidden unit* ke-*j*.
- y_k : Unit output ke-*k*.
- v_{0j} : Bias untuk *hidden unit* ke-*j*.
- v_{ij} : Bobot antara unit input ke-*i* dengan *hidden unit* ke-*j*.
- w_{0k} : Bias untuk unit output ke-*k*.
- w_{jk} : Bobot antara *hidden unit* ke-*j* dengan unit output ke-*k*.

Tahap-tahap prediksi pada model *Backpropagation*, yaitu:

1. Inisialisasi bobot sesuai dengan bobot yang telah dihasilkan pada proses pelatihan.
2. Untuk setiap input, lakukan tahap 3 – 5.
3. Untuk setiap input $i = 1, 2, \dots, n$ skalakan bilangan dalam jangkauan fungsi aktivasi seperti yang dilakukan pada proses pelatihan.
4. Tahap 3: untuk $j = 1, 2, \dots, p$.

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (98)$$

$$z_j = f(z_{in_j}) \quad (99)$$

5. Tahap 4: Untuk $k = 1, 2, \dots, m$.

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (100)$$

$$y_k = f(y_in_k) \quad (101)$$

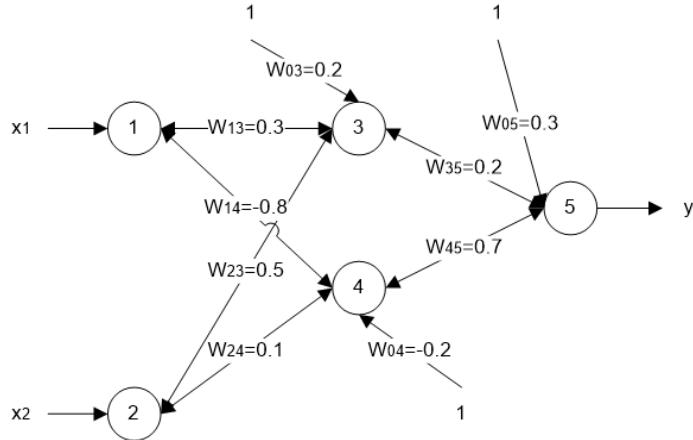
Variabel y_k adalah output yang masih dalam skala menurut jangkauan fungsi aktivasi. Untuk mendapatkan nilai output yang sesungguhnya, y_k harus dikembalikan seperti semula.

Berikut ini merupakan contoh penyelesaian manual *Backpropagation*. Data yang digunakan terdiri dari 4 *instances*, 2 atribut input, dan 2 label *class* (0 dan 1). Penyelesaian manual *Backpropagation* cukup kompleks, itulah mengapa jumlah data yang digunakan hanya sedikit.

Contoh 6.1 ANN Backpropagation (Manual) – 1

Data ke	X1	X2	Class
1.	2	3	0
2.	2	6	1
3.	1	4	1
4.	1	2	0

Kondisi bobot saat inisialisasi ditunjukkan pada gambar berikut ini:



Jumlah *neuron* = 2 pada *hidden layer*;

Laju pembelajaran (η) = 0.1;

Fungsi aktivasi yang digunakan adalah *Sigmoid Biner*;

Target error = 0.0001 dengan kriteria SSE;

Momentum yang digunakan = 0.95; dan

Maksimum jumlah iterasi pelatihan = 1,000 kali.

- Nilai *neuron* 3 dan *neuron* 4 pada *hidden layer*:

$$v_j(p) = \sum_{i=1}^n x_i(p)w_{ij}(p)$$

$$y_j(p) = \frac{1}{1 + e^{-v_j(p)}}$$

(p) adalah iterasi ke-*i*.

$$v_3(1) = x1w_{13} + x2w_{23} + 1w_{03} = 2 * 0.3 + 3 * 0.5 + 1 * 0.2 = 2.3$$

$$y_3(1) = \frac{1}{1 + e^{-2.3}} = \frac{1}{1 + 2.71828^{-2.3}} = 0.908877$$

$$v_4(1) = x1w_{14} + x2w_{24} + 1w_{04} = 2 * (-0.8) + 3 * 0.1 + 1 * (-0.2) = 1.7$$

$$y_4(1) = \frac{1}{1+e^{-1.7}} = 0.845535$$

2. Nilai pada *neuron* di *output layer*:

$$v_k(p) = \sum_{j=1}^m x_j(p)w_{jk}(p)$$

$$y_k(p) = \frac{1}{1 + e^{-v_k(p)}}$$

$$v_5(1) = Y_3(1)w_{35} + Y_4(1)w_{45} + 1w_{05} = 0.908877 * 0.2 + 0.845535 * 0.7 + 1 * 0.3 = 2.054411496$$

$$y_5(1) = \frac{1}{1+e^{-2.054411496}} = 0.886392478$$

3. *Gradient error* untuk *neuron* pada *output layer*:

$$e_k(p) = y_{dk}(p) - y_k(p)$$

$$\delta_k(p) = y_k(p) * [1 - y_k(p)] * e_k(p)$$

Untuk data pertama, target nilai yang diharapkan adalah $y_d = 0$, sedangkan keluaran yang didapatkan $y_5(1) = 0.886392478$. Maka hitung error pada iterasi pertama untuk data pertama pada *neuron* 5 di *output layer*:

$$e_5^1(1) = y_d - y_5(1) = 0 - 0.886392478 = -0.886392478$$

$$\delta_5(1) = y_5(1) * (1 - y_5(1)) * e_5^1(1) = 0.886392478 * (1 - 0.886392478) * (-0.886392478) = -0.089260478$$

4. Koreksi bobot untuk *output layer*, Δw_{35} , Δw_{45} , dan Δw_{05} :

$$\Delta w_{jk}(p) = \eta \cdot y_j(p) * \delta_k(p)$$

$$\Delta w_{35} = \eta * y_3(1) * \delta_5(1) = 0.1 * 0.91 * (-0.089) = -0.008112679$$

$$\Delta w_{45} = \eta * y_4(1) * \delta_5(1) = 0.1 * 0.85 * (-0.089) = -0.007547282$$

$$\Delta w_{05} = \eta * 1 * \delta_5(1) = 0.1 * 1 * (-0.089) = -0.0089266048$$

5. *Gradient error* pada *hidden layer* $\delta_3(1)$ dan $\delta_4(1)$:

$$\delta_j(p) = y_j(p) * [(1 - y_j(p))] + \sum_{k=1}^1 \delta_k(p) * w_{jk}(p)$$

$$\begin{aligned} \delta_3(1) &= y_3(1) * (1 - y_3(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{3k}(1) \\ &= y_3(1) * (1 - y_3(1)) * \delta_5(1) \cdot w_{35}(1) \end{aligned}$$

$$= 0.91 * (1 - 0.91) * (-0.089) * 0.2 = -0.001478505$$

$$\begin{aligned}
 \delta_4(1) &= y_4(1) * (1 - y_4(1)) * \sum_{k=1}^1 \delta_k(1) \cdot w_{4k}(1) \\
 &= y_4(1) * (1 - y_4(1)) * \delta_5(1) \cdot w_{45}(1) \\
 &= 0.85 * (1 - 0.85) * (-0.089) * 0.7 = -0.008160558
 \end{aligned}$$

6. Koreksi bobot untuk *hidden layer* $\Delta w_{13}, \Delta w_{23}, \Delta w_{03}, \Delta w_{14}, \Delta w_{24}, \Delta w_{04}$
 $w_{ij}(p) = \eta * x_i(p) * \delta_j(p)$

$$\Delta w_{13} = \eta * x_1 * \delta_3(1) = 0.1 * 2 * (-0.0015) = -0.0003$$

$$\Delta w_{23} = \eta * x_2 * \delta_3(1) = 0.1 * 3 * (-0.0015) = -0.00044$$

$$\Delta w_{03} = \eta * x_1 * \delta_3(1) = 0.1 * 1 * (-0.0015) = -0.00015$$

$$\Delta w_{14} = \eta * x_1 * \delta_4(1) = 0.1 * 2 * (-0.0082) = -0.00163$$

$$\Delta w_{24} = \eta * x_2 * \delta_4(1) = 0.1 * 3 * (-0.0082) = -0.00245$$

$$\Delta w_{04} = \eta * x_1 * \delta_4(1) = 0.1 * 1 * (-0.0082) = -0.00082$$

7. Perbaharui bobot untuk neuron pada hidden layer w_{35}, w_{45}, w_{05} :

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

$$w_{35}(2) = w_{35}(1) + \Delta w_{35} = 0.2 + (-0.008112679) = 0.191887321$$

$$w_{45}(2) = w_{45}(1) + \Delta w_{45} = 0.7 + (-0.007547282) = 0.692452718$$

$$w_{05}(2) = w_{05}(1) + \Delta w_{05} = 0.3 + (-0.0089266048) = 0.291073952$$

8. Perbaharui bobot pada layer tersembunyi, $w_{13}, w_{23}, w_{03}, w_{14}, w_{24}, w_{04}$:

$$w_{13}(2) = w_{13}(1) + \Delta w_{13} = 0.3 + (-0.0003) = 0.299704$$

$$w_{23}(2) = w_{23}(1) + \Delta w_{23} = 0.5 + (-0.00044) = 0.499556$$

$$w_{03}(2) = w_{03}(1) + \Delta w_{03} = 0.2 + (-0.00015) = 0.199852$$

$$w_{14}(2) = w_{14}(1) + \Delta w_{14} = 0.8 + (-0.00163) = -0.798368$$

$$w_{24}(2) = w_{24}(1) + \Delta w_{24} = 0.1 + (-0.00245) = 0.097552$$

$$w_{04}(2) = w_{04}(1) + \Delta w_{04} = (-0.2) + (-0.00082) = -0.20082$$

9. Naikkan 1 langkah iterasi (p), kembali ke langkah 2 dan ulangi proses tersebut sampai kriteria error tercapai.

10. Dengan demikian, didapatkan bobot akhir pada iterasi pertama untuk data pertama [2, 3]:

$$w_{13} = 0.299704;$$

$$w_{23} = 0.499556;$$

$$w_{03} = 0.199852;$$

$$w_{14} = -0.798368;$$

$$w_{24} = 0.097552;$$

$$w_{04} = -0.20082;$$

$$w_{35} = 0.191887321;$$

$$w_{45} = 0.692452718;$$

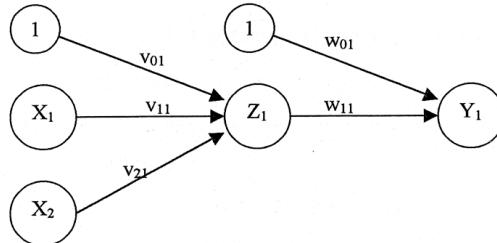
$$w_{05} = 0.291073952$$

Error yang didapatkan = -0.886392478. Selanjutnya proses di atas dilanggi untuk data ke-dua [2, 6], data ke-tiga [1, 4], dan data ke-empat [1, 2] sehingga iterasi pertama selesai dilakukan. Pada setiap data yang diproses dalam satu iterasi, error keluaran disimpan untuk dihitung dengan kriteria *error*, seperti SSE, MSE, dsb. Apanila hasil MSE < *error* yang didapatkan maka iterasi berhenti, sebaliknya dilakukan perambatan terus hingga batas iterasi/epoch.

Contoh 6.2 ANN Backpropagation (Manual) – 2

Misalnya, jaringan terdiri dari 2 unit input, 1 *hidden unit* (dengan 1 *hidden layer*), dan 1 unit output. Jaringan akan dilatih untuk memecahkan fungsi *XOR*. Fungsi aktivasi yang digunakan adalah *sigmoid biner* dengan nilai *learning rate* (α) = 0,01 dan nilai $\sigma = 1$.

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0



Sebelum pelatihan, harus ditentukan terlebih dahulu *stopping conditionnya*. Misalnya dihentikan jika *error* telah mencapai 0,41.

Tahap pelatihan:

1. Tahap 0: Misalnya inisialisasi bobot dan bias adalah:

$$v01 = 1,718946$$

$$v11 = -1,263178$$

$$v21 = -1,083092$$

$$w01 = -0,541180$$

$$w11 = 0,543960$$

2. Tahap 1: Dengan bobot di atas, tentukan *error* untuk *training data* secara keseluruhan dengan *Mean Square Error*:

$$z_in11 = 1,718946 + \{(0 \times -1,263178) + (0 \times -1,083092)\} = 1,718946$$

$$z11 = f(z_in11) = 0,847993$$

$$z_in12 = 1,718946 + \{(0 \times -1,263178) + (1 \times -1,083092)\} = 0,635854$$

$z_{12} = f(z_{in12}) = 0,653816$
 $z_{in13} = 1,718946 + \{(1x - 1,263178) + (0x - 1,083092)\} = 0,455768$
 $z_{13} = f(z_{in13}) = 0,612009$
 $z_{in14} = 1,718946 + \{(1x - 1,263178) + (1x - 1,083092)\} = -0,627324$
 $z_{14} = f(z_{in14}) = 0,348118$
 yang mana indeks z_{jn} adalah *hidden unit* ke-j dan *training data* ke-n.
 $y_{in11} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$
 $y_{11} = f(y_{in11}) = 0,480034$
 $y_{in12} = -0,541180 + (0,653816 \times 0,543960) = -0,185530$
 $y_{12} = f(y_{in12}) = 0,453750$
 $y_{in13} = -0,541180 + (0,612009 \times 0,543960) = 0,208271$
 $y_{13} = f(y_{in13}) = 0,448119$
 $y_{in14} = -0,541180 + (0,348118 \times 0,543960) = -0,351818$
 $y_{14} = f(y_{in14}) = 0,412941$
 maka $E = 0,5 \times \{(0-0,480034)^2 + (1-0,453750)^2 + (1-0,448119)^2 + (0-0,412941)^2\} = 0,501957$

3. Tahap 2: Karena *error* masih lebih besar dari 0,41 maka langkah 3-8 dijalankan
4. Tahap 3: $x1=0$; $x2=0$ (*iterasi* pertama, *training data* pertama)
5. Tahap 4:
 $z_{in1} = 1,718946 + \{(0x-1,263126)+(0x-1,083049)\} = 1,718946$.
 $z_1 = f(z_{in1}) = 0,847993$
6. Tahap 5:
 $y_{in11} = -0,541180 + (0,847993 \times 0,543960) = 0,079906$
 $y_{11} = f(y_{in11}) = 0,480034$
7. Tahap 6:
 $\delta_1 = (0-0,480034)f'(0,079906) = -0,119817$
 $\Delta w_{11} = 0,01x - 0,119817 \times 0,847993 = -0,001016$
 $\Delta w_{01} = 0,01x - 0,119817 = -0,00119817$
8. Tahap 7:
 $\delta_{in1} = -0,00119817 \times 0,543960 = -0,00065176$
 $\delta_1 = -0,00065176 \times f'(1,718946) = -0,00008401$
 $\Delta v_{11} = 0,01x - 0,00008401 \times 0 = 0$
 $\Delta v_{21} = 0,01x - 0,00008401 \times 0 = 0$
 $\Delta v_{01} = 0,01x - 0,00008401 = -0,0000008401$
9. Tahap 8:
 $w_{01}(\text{baru}) = -0,541180 + (-0,00119817) = -0,542378$
 $w_{11}(\text{baru}) = 0,543960 + (-0,001016) = 0,542944$
 $v_{01}(\text{baru}) = 1,718946 + (-0,0000008401) = 1,718862$
 $v_{11}(\text{baru}) = -1,263178 + 0 = -1,263178$
 $v_{21}(\text{baru}) = -1,083092 + 0 = -1,083092$

Saat ini v_{11} dan v_{21} masih belum berubah karena kedua *inputnya* = 0. Nilai v_{01} dan v_{02} baru berubah pada *iterasi* pertama untuk *training data* yang kedua. Setelah tahap 3-8 untuk *training data* pertama dijalankan, selanjutnya kembali lagi ke tahap 3 untuk *training data* yang kedua ($x1=0$ dan $x2=1$). Langkah yang sama dilakukan sampai pada *training data* yang keempat. Bobot yang dihasilkan pada *iterasi* pertama, *training data* ke-2,3, dan 4 adalah:

- Pelatihan data ke-2:

w₀₁ = -0,541023

w₁₁ = 0,543830

v₀₁ = 1,718862

v₁₁ = -1,263178

v₂₁ = -1,083092

- Pelatihan data ke-3:

w₀₁ = -0,539659

w₁₁ = 0,544665

v₀₁ = 1,719205

v₁₁ = -1,263002

v₂₁ = -1,082925

- Pelatihan data ke-4:

w₀₁ = -0,540661

w₁₁ = 0,544316

v₀₁ = 1,719081

v₁₁ = -1,263126

v₂₁ = -1,083049

Setelah sampai pada pelatihan data ke-4, maka *iterasi* pertama selesai. Berikutnya, pelatihan sampai pada tahap 9, yaitu memeriksa *stopping condition* dan kembali pada tahap 2. Demikian seterusnya sampai *stopping condition* yang ditentukan terpenuhi. Setelah pelatihan selesai, bobot yang didapatkan adalah:

v₀₁= 12,719601

v₁₁= -6,779127

v₂₁= -6,779127

w₀₁= -5,018457

w₁₁= 5,719889

Jika ada input baru, misalnya $x_1 = 0,2$ dan $x_2 = 0,9$ maka outputnya dapat dicari dengan langkah umpan maju sebagai berikut:

1. Tahap 0: Bobot yang dipakai adalah bobot hasil pelatihan.
2. Tahap 1: Perhitungan dilakukan pada tahap 2-4.
3. Tahap 2: Dalam contoh ini, bilangan telah berada dalam interval 0 sampai dengan 1, jadi tidak perlu diskalakan lagi.
4. Tahap 3:

$$z_{in1} = 12,719601 + \{(0,2x-6,779127) + (0,9x-6,779127)\} = 5,262561$$

$$z_1 = f(5,262561) = 0,994845$$
5. Tahap 4:

$$y_{in1} = -5,018457 + (0,994845 \times 5,719889) = 0,671944$$

$$y_1 = f(0,671944) = 0,661938$$

Jadi, jika input $x_1 = 0,2$ dan $x_2 = 0,9$; *output* yang dihasilkan = 0,661938.

Contoh 6.3 ANN Backpropagation: Klasifikasi (Matlab)

<i>Dataset</i>	: <i>dsHeartDiseaseCleveland – Class: {0,1,2,3,4}</i> (terlampir)
<i>Data validation</i>	: <i>10-Fold Cross Validation</i>
<i>Neuron HD</i>	: <i>50</i>
<i>Train Function</i>	: <i>traingdx</i>
<i>Evaluation</i>	: <i>Confusion Matrix</i>

Fungsi “*ANNaa*”:

```
function [output, lamaProses, akurasi] = ANNaa(dataLatihInput,
dataLatihOutput, dataUjiInput, dataUjiOutput)
tic;
%% ANN Network:
trnFcn = 'traingdx';
hd = 50; %neuron hd
ol = 5; %output layer = 1 dng neuron = 5 label class
ANNnetwork = feedforwardnet([hd ol],trnFcn);
%% ANN training:
[ANNmodel,tr,Y,E] = train(ANNnetwork, dataLatihInput,
dataLatihOutput);
save ANNbackpropagation.mat ANNmodel
[nilaiNeuron, outputTest] = max(sim(ANNmodel,
dataUjiInput)); %tidak perlu max jika estimasi
output = outputTest;
%% ANN Evaluasi:
conMat = confusionmat(dataUjiOutput, output);
jmlData = sum(conMat(:));
hasilBenar = sum(diag(conMat));
akurasi = 100 * (hasilBenar / jmlData);
lamaProses = toc;
plot(dataUjiInput,dataUjiOutput,'o',dataUjiInput,output','+')
end
```

Script “*ANN*”:

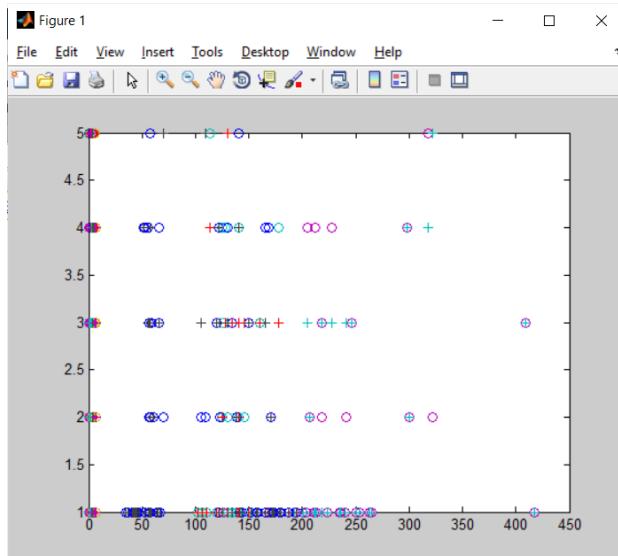
```
clc; clear; close all; warning off all;
%% data (Kolom: 1 ID, 2-14 Input, 15 Output 0-4, 16 Output 0-1)
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dataset = dsHeartDisease(:,2:15); % ID dan Output 0-1 tidak diproses
[N,M] = size(dataset); % m baris, n kolom
%% pada ANN dimensi data dibalik & jika klasifikasi class diencoding
kelas = dataset(:,14);
kelas_baru = zeros(N,5); %5 label class
kelas_0 = find(kelas==0);
kelas_1 = find(kelas==1);
kelas_2 = find(kelas==2);
kelas_3 = find(kelas==3);
kelas_4 = find(kelas==4);
kelas_baru(kelas_0,1) = 1; %encode to 1 0 0 0 0
kelas_baru(kelas_1,2) = 1; %encode to 0 1 0 0 0
kelas_baru(kelas_2,3) = 1; %encode to 0 0 1 0 0
kelas_baru(kelas_3,4) = 1; %encode to 0 0 0 1 0
kelas_baru(kelas_4,5) = 1; %encode to 0 0 0 0 1
dataANN = [dataset kelas_baru]; %gabunkan ke dataset
dataANN(:,14) = dataANN(:,14) + 1; %class 0 = 1, dst
dataANN = dataANN'; %balik kolom jadi baris, cara kerja ANN
%% Variabel untuk hasil ANN
ANNhasilSub = []; %kolom: 1 lama proses, 2 akurasi
ANNoutputAll.x = []; %Output ANN di setiap K dr K-Fold
```

```

ANNhasil = []; %kolom: 1 lama proses, 2 acc, 3 acc max, 4 acc min
%% K-Fold Cross Validation
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K); % kolom 14 = class
for i = 1:K
    %% Buat Data Latih dan Data Uji berdasarkan indeks dari K-Fold
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
    subDataLatihInput = dataANN(1:13,latih); %input training
    subDataLatihOutput = dataANN(15:19,latih); %output training
    subDataUjiInput = dataANN(1:13,uji); %input testing
    subDataUjiOutput = dataANN(14,uji); %output testing
    %% ANN Modelling di tiap K
    [ANNsubOutput, ANNsubLamaProses, ANNsubAkurasi] =
    ANNaa(subDataLatihInput, subDataLatihOutput, subDataUjiInput,
    subDataUjiOutput);
    ANNhasilSub(i,1) = ANNsubLamaProses; %hasil lama proses dalam K
    ANNhasilSub(i,2) = ANNsubAkurasi; %hasil akurasi dalam K
    subDataUjiOutput(1,:) = subDataUjiOutput(1,:)-1; %output asli
    ANNsubOutput(1,:) = ANNsubOutput(1,:)-1; % output ANN asli
    ANNoutputAll(i).x = [subDataUjiInput' subDataUjiOutput'
    ANNsubOutput'];
end
%% Hasil Akhir ANN
ANNhasil(1)=mean(ANNhasilSub(:,1)); % rata2 lama proses
ANNhasil(2)=mean(ANNhasilSub(:,2)); % akurasi akhir
ANNhasil(3)=max(ANNhasilSub(:,2)); % akurasi max
ANNhasil(4)=min(ANNhasilSub(:,2)); % akurasi min

```

Hasilnya ketika *K-Fold Cross Validation* di K = 10:



$$\text{Akurasi} = 60,5586\%$$

$$\text{Akurasi maksimum} = 73,3333\%$$

$$\text{Akurasi minimum} = 43,3333\%$$

$$\text{Lama proses} = 0,4847 \text{ detik}$$

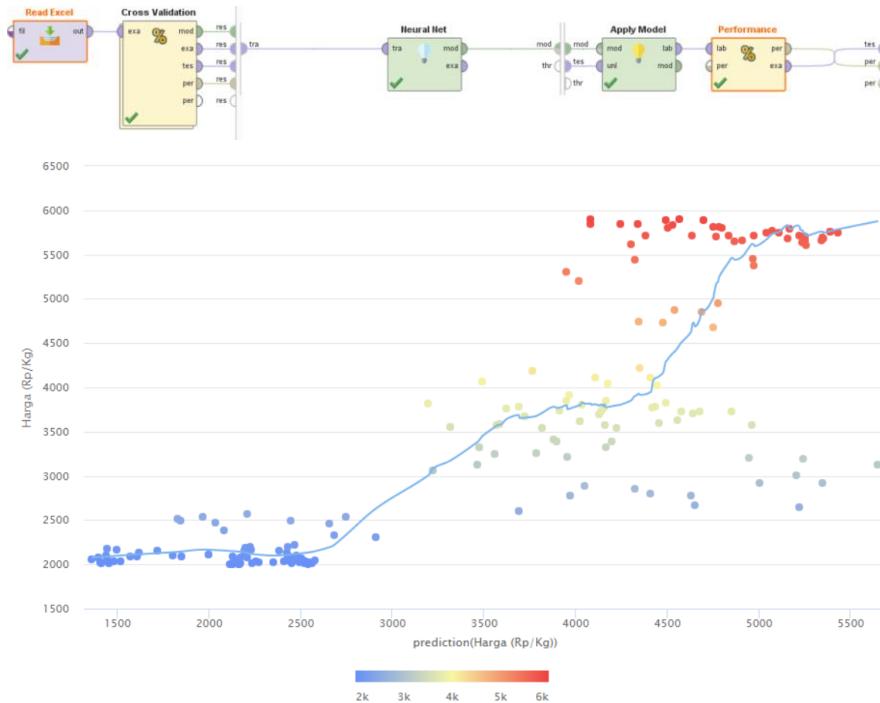
Contoh 6.4 ANN Backpropagation: Regresi (Rapidminer)

Dataset : *dsPanganTimeSeries* (terlampir)

Data validation : *10-Fold Cross Validation*

Neuron HD : 50

Evaluation : *RMSE*



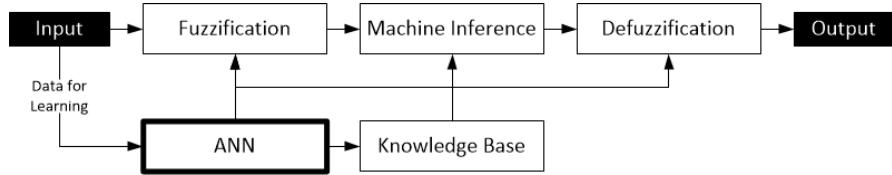
Root Mean Squared Error: 788.458 +/- 202.506 (micro average: 811.525 +/- 0.000)

6.2 Adaptive Neuro Fuzzy Inference System

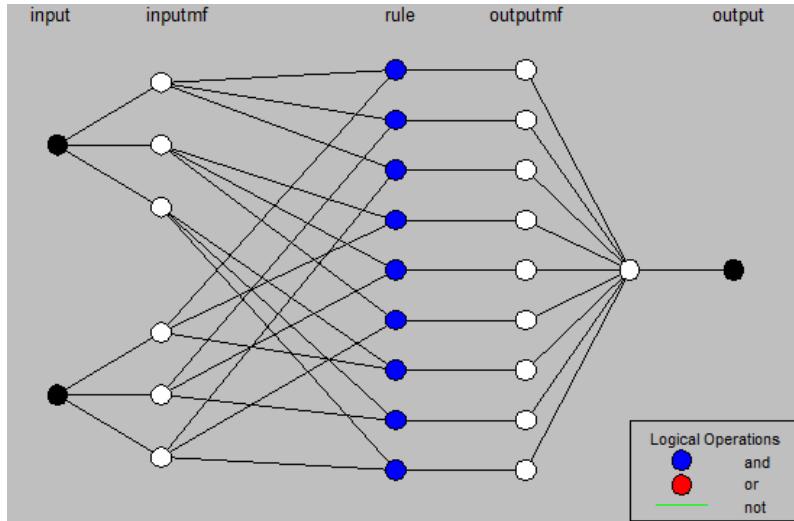
Pada prinsipnya, *Fuzzy Logic* tidak mampu melakukan pembelajaran sendiri, maka dengan menerapkan pendekatan *machine learning*, *Fuzzy Logic* akan dapat melakukan pendekatan *reasoning* dan *learning*. Salah satu hasil dari pengembangan pendekatan ini adalah *Adaptive Neuro Fuzzy Inference System (ANFIS)*, yang mana *Artificial Neural Network (ANN)* bertugas untuk melakukan *learning* berdasarkan data latih untuk mengoptimalkan proses-proses *Fuzzy Logic* dalam melakukan *reasoning*, sehingga algoritma ini menjadi dinamis. Lebih jelasnya, struktur ANFIS dapat ditunjukkan pada Gambar 6.2.

Algoritma *Fuzzy Logic* yang digunakan pada ANFIS adalah *Fuzzy Logic Sugeno*. Sedangkan algortima ANN yang digunakan pada ANFIS adalah *Backpropagation* atau *Hybrid*, dengan parameter *error tolerance* dan *epoch* (iterasi). Suatu jaringan ANFIS dapat terdiri dari lima *layer*, yaitu *layer input*, *layer Fuzzification (Membership Function* dari inputan), *layer Knowledge Base (rules)*,

layer Machine Inference (Membership Function dari output), dan layer Defuzzification (output) seperti ditunjukkan pada Gambar 6.3.



Gambar 6.2 Struktur ANFIS



Gambar 6.3 Arsitektur Jaringan ANFIS

Contoh 6.5 ANFIS (Matlab)

<i>Dataset</i>	:	<i>dsTrafficLight: dataAWT</i> (terlampir)
<i>Data validation</i>	:	<i>10-Fold Cross Validation</i>
<i>Membership Function</i>	:	<i>trimf, trapmf, gaussmf</i>
<i>Num of Membership</i>	:	<i>3, 4, 5, 6</i>
<i>Epoch</i>	:	<i>50</i>
<i>Error Tolerance</i>	:	<i>1e-6</i>
<i>Evaluation</i>	:	<i>RMSE & AWT (Average Waiting Times) reduction</i>

Variabel	Tipe	Jenis	Satuan	Keterangan
AT (Arrival Times)	Integer	Input	Detik	Waktu kedatangan
TT (Transportation Type)	Ordinal	Input	2 (Roda 2) 4 (Roda 4) 7 (Roda 6 - 8) 10 (Roda >=10)	Ukuran kendaraan berdasarkan jumlah roda
GJ (Goal Junction)	Ordinal	Input	1 (ke ruas dekat) 2 (ke ruas sedang) 3 (ke ruas jauh)	Kategori jarak dari ruas asal ke ruas tujuan suatu kendaraan
WT (Waiting Times)	Integer	Output	Detik	Waktu tunggu (waktu tiba hingga sampai di ruas tujuan)

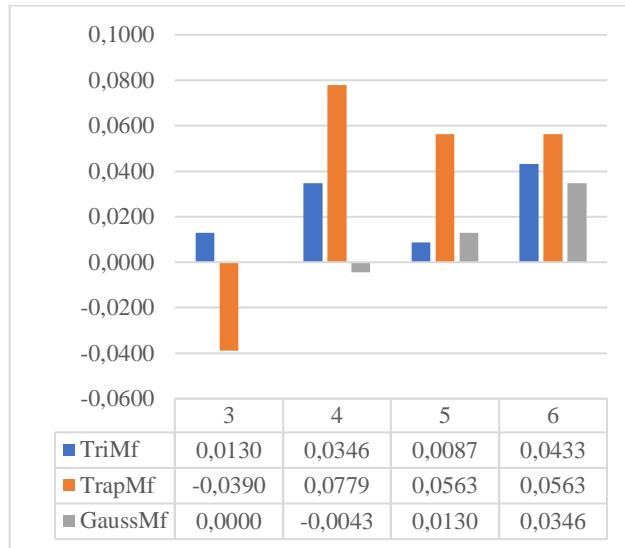
```

clc; clear; close all; warning off all;
tic;
%% set dataset
ds = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\dsTrafficLight.xlsx','dataAWT');
dsInput = ds(:,1:3);
dsOutput = ds(:,4);
%% set parameter ANFIS (setiap ruas menggunakan parameter yang sama)
numMFs = [4 4 4]; % membership setiap variabel input (uji coba ini)
mfType = 'trapmf'; %membership function (uji coba ini)
epoch = 50;
errorTolerance = 1e-6; % atau 0.01
trnOpt(1) = epoch;
trnOpt(2) = errorTolerance;
%% set arsitektur ANFIS untuk ksetiap ruas
fismat = genfis1([dsInput, dsOutput], numMFs, mfType);
%% pelatihan ANFIS untuk setiap ruas
[trnfismat, rmse] = anfis([dsInput, dsOutput], fismat, trnOpt);
%% Plot Membership Function
[x,mf] = plotmf(trnfismat,'input',1);
subplot(1,3,1), plot(x,mf) % subplot(baris, kolom, posisi)
xlabel('AT (Arrival Times)')
[x,mf] = plotmf(trnfismat,'input',2);
subplot(1,3,2), plot(x,mf)
xlabel('TT (Transportation Type)')
[x,mf] = plotmf(trnfismat,'input',3);
subplot(1,3,3), plot(x,mf)
xlabel('GJ (Goal Junction)')
%% Menyimpan variabel trnfismat hasil pelatihan ANFIS
writefis(trnfismat,'trnANFISaa');
%% Testing --> gunakan kembali data latih
outputANFIS = evalfis(dsInput, trnfismat);
%% AWT dan AGT
waitingTimes = zeros(length(dsOutput),1);
for i=1:length(dsOutput)
    waitingTimes(i) = dsOutput(i) - round(outputANFIS(i));
end
AWT = mean(waitingTimes);
%% Membandingkan hasil ANFIS vs Actual Output
ANFISvsACTUAL = [round(outputANFIS), dsOutput, waitingTimes];
%% Error Estimasi
err = dsOutput - round(outputANFIS);
SE = err.^2;
MSE = mean(SE);
RMSE = sqrt(MSE);
PSNR = 10*log10(256^2/MSE); %bagus jika dibawah 10
%SSE = sqrt(SE/(n-f)); --> n = banyaknya data; f = derajat kebebasan
%(1 = data konstant, 2 = data linier, 3 = data kuadratis atau siklis)
lamaProses = toc;

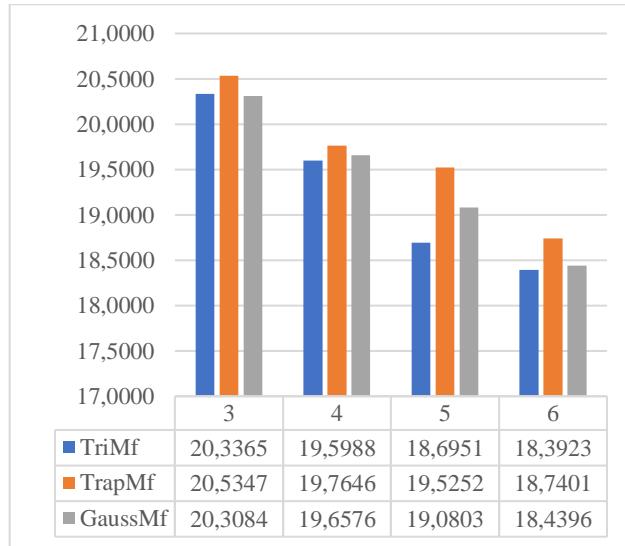
```

Berdasarkan hasil evaluasi, kinerja terbaik ANFIS dalam mereduksi AWT diperoleh ketika menggunakan *Membership Function* = *TrapMf* dan jumlah keanggotaan = 4, yang mana AWT yang mampu direduksi sebesar 0,0779 detik dengan RMSE sebesar 19,7646. AWT yang mampu direduksi menunjukkan bahwa fungsi keanggotaan *TrapMf* selalu memberikan hasil yang lebih baik pada setiap jumlah keanggotaan yang diuji coba, kecuali pada jumlah keanggotaan = 3. Sedangkan RMSE yang dihasilkan menunjukkan bahwa semakin rendah jumlah keanggotaan, maka semakin rendah pula nilai RMSE, baik pada *Membership Function* = *TriMf*, *TrapMf*, maupun *GaussMf*.

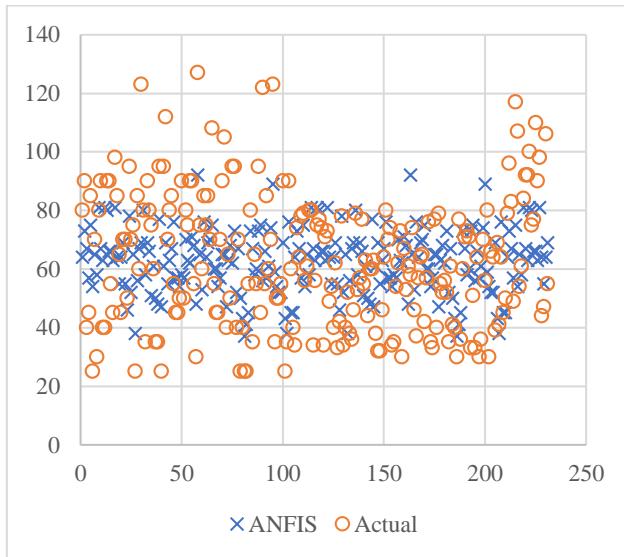
Berikut ini adalah hasil *AWT reduction*:



Berikut ini adalah hasil RMSE:



Berikut ini adalah WT aktual vs WT ANFIS (*4 trapmf*)



6.3 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma *machine learning* yang dapat melakukan klasifikasi dan regresi (*supervised learning*). SVM pertama kali diperkenalkan oleh Vapnik bersama Boser & Guyon yang pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory, yang mana konsep dasar SVM sebenarnya merupakan kombinasi/rangkaian harmonis dari teori-teori komputasi unggulan yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda & Hart pada tahun 1973, Cover pada tahun 1965, Vapnik 1964, dsb.), *Kernel* yang diperkenalkan oleh Aronszajn pada tahun 1950, dan begitupun dengan konsep-konsep yang lainnya. Pada prinsipnya, dapat dikatakan bahwa SVM merupakan pengembangan dari ANN.

SVM merupakan metode yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*, seperti yang dikemukakan oleh Cortes and Vapnik, bahwa: “*Support Vector Machine (SVM) originally separates the binary classes (k=2) with a maximized margin criterion* [44].” Pada ANN, semua data latih akan dipelajari selama proses pelatihan, namun pada SVM berbeda, kerana hanya sejumlah data terpilih saja yang berkontribusi untuk membentuk model yang akan digunakan dalam klasifikasi/prediksi yang akan dipelajari. Data-data yang berkontribusi tersebut disebut *support vector* sehingga metodenya disebut *Support Vector Machine*. Hal ini menjadi kelebihan SVM, karena tidak semua data latih akan dipandang untuk dilibatkan dalam setiap iterasi pelatihannya. Dengan demikian SVM dianggap bisa lebih cepat daripada ANN.

Berbagai upaya terus dilakukan untuk mengembangkan SVM, salah satu contohnya adalah usaha menemukan *hyperplane* yang terbaik pada *input space* yang dapat bekerja pada masalah *non-linear* dengan cara memasukkan konsep *Kernel Trick* pada ruang kerja berdimensi tinggi, sehingga dewasa ini SVM termasuk berhasil diaplikasikan pada *real-world problem* dan secara umum memberikan solusi yang lebih baik dibandingkan metode lainnya seperti ANN. Vapnik mampu membuktikan bahwa SVM merupakan metode yang tepat untuk digunakan dalam memecahkan masalah berdimensi tinggi dan dari keterbatasan sampel data yang ada. Tidak seperti ANN yang memberikan solusi yang *local optimal*, SVM mampu memberikan solusi yang *global optimal*. Maka tidak heran bila kita menjalankan ANN solusi dari setiap *training* selalu berbeda. Hal ini disebabkan solusi *local optimal* yang dicapai tidak selalu sama. Sedangkan SVM selalu mencapai solusi yang sama untuk setiap *running*. Untuk lebih jelasnya, berikut ini adalah beberapa karakteristik SVM [20]:

1. SVM sebenarnya bisa dikatakan sebagai teknik klasifikasi yang *semi-eiger learner* karena selain memerlukan proses pelatihan, SVM juga menyimpan sebahagian kecil data latih untuk digunakan kembali pada saat proses prediksi. Sebahagian data yang masih disimpan ini adalah *support vector*.
2. Untuk parameter yang sama yang digunakan dalam klasifikasi, SVM memberikan model klasifikasi yang solusinya adalah *global optima*. Hal ini berarti SVM selalu memberikan model yang sama dan solusi dengan margin maksimal.
3. Proses pelatihan yang dilakukan SVM tidak sebanyak metode lainnya seperti ANN, tetapi sering kali memberikan kinerja yang lebih baik.
4. Tidak membutuhkan pemilihan parameter-parameter yang banyak. Dalam SVM kita hanya perlu menentukan fungsi *Kernel* yang harus digunakan (untuk kasus data yang distirbusi kelasnya tidak dapat dipisahkan secara linier).
5. SVM membutuhkan komputasi pelatihan dan prediksi yang rumit karena dimensi data yang digunakan dalam proses pelatihan dan prediksi lebih besar daripada dimensi data yang sesungguhnya. Hal ini bertentangan dengan metode lain yang pada umumnya mengurangi dimensi untuk memberikan kinerja yang lebih cepat dan akurasi yang lebih baik. Hal ini dapat mengakibatkan masalah *curse of dimensionality*.
6. Untuk *dataset* yang berjumlah besar, SVM membutuhkan memori yang sangat besar untuk alokasi matriks *Kernel* yang digunakan. Misalnya, data latih dengan ukuran 1000 data dan 10 kolom fitur akan berubah menjadi matriks *Kernel* berukuran 1000*1000.
7. Penggunaan matriks *Kernel* mempunyai keuntungan lain, yaitu pada *dataset* dengan dimensi besar tetapi jumlah datanya sedikit akan lebih cepat karena ukuran data pada dimensi baru berkurang banyak. Misalnya, data latih berukuran 10 data dan 1000 kolom fitur akan berubah menjadi matriks *Kernel* berukuran 10*10 saja.

Berikut ini beberapa kelebihan SVM, yaitu:

1. Generalisasi. Generalisasi didefinisikan sebagai kemampuan suatu metode untuk mengklasifikasikan suatu *pattern*, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu. Vapnik menjelaskan bahwa *generalization error* dipengaruhi oleh dua faktor: *error* terhadap *training set*, dan satu faktor

lagi yang dipengaruhi oleh dimensi VC (Vapnik-Chervokinensis). Strategi pembelajaran pada ANN dan umumnya metode *machine learning* difokuskan pada usaha untuk meminimalkan *error* pada *training-set*. Strategi ini disebut *Empirical Risk Minimization* (ERM). Adapun SVM selain meminimalkan *error* pada *training-set*, juga meminimalkan faktor kedua. Strategi ini disebut *Structural Risk Minimization* (SRM), dan dalam SVM diwujudkan dengan memilih *hyperplane* dengan margin terbesar. Berbagai studi empiris menunjukkan bahwa pendekatan SRM pada SVM memberikan *error generalisasi* yang lebih kecil daripada yang diperoleh dari strategi ERM pada ANN maupun metode *machine learning* yang lainnya.

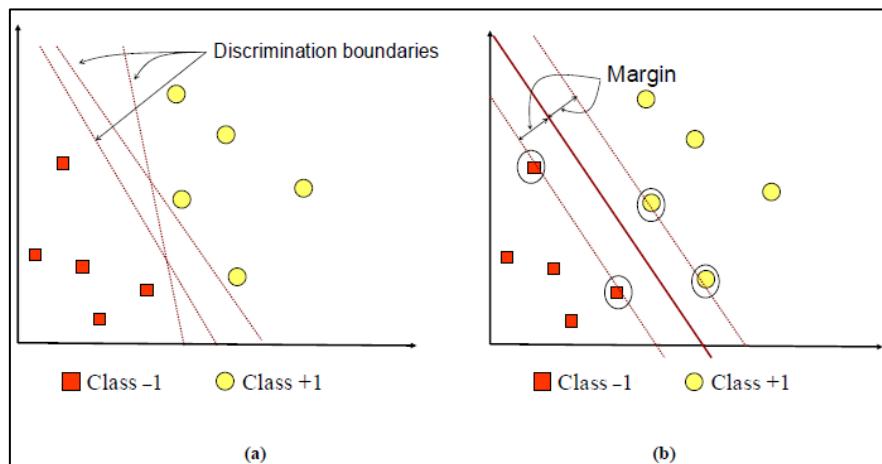
2. *Curse of dimensionality.* *Curse of dimensionality* didefinisikan sebagai masalah yang dihadapi suatu metode *machine learning* dalam mengestimasikan parameternya, dikarenakan jumlah sampel data yang relatif sedikit dibandingkan dimensional ruang vektor data tersebut. Semakin tinggi dimensi dari ruang vektor informasi yang diolah, membawa konsekuensi dibutuhkannya jumlah data dalam proses pembelajaran. Pada kenyataannya seringkali terjadi, data yang diolah berjumlah terbatas, dan untuk mengumpulkan data yang lebih banyak tidak mudah dilakukan. Dalam kondisi tersebut, jika metode itu terpaksa harus bekerja pada data yang berjumlah relatif sedikit dibandingkan dimensinya, akan membuat proses estimasi parameter metode menjadi sangat sulit. *Curse of dimensionality* sering dialami dalam aplikasi yang ketersediaan datanya sangat terbatas dan penyediaannya memerlukan biaya tinggi. Vapnik membuktikan bahwa tingkat generalisasi yang diperoleh oleh SVM tidak dipengaruhi oleh dimensi dari *input vector*. Hal ini merupakan alasan mengapa SVM merupakan salah satu metode yang tepat dipakai untuk memecahkan masalah berdimensi tinggi, dalam keterbatasan sampel data yang ada.
3. Landasan teori yang jelas. Sebagai metode yang berbasis statistik, SVM memiliki landasan teori yang dapat dianalisa dengan jelas, dan tidak bersifat *black box*.
4. Feasibility. SVM dapat diimplementasikan relatif mudah, karena proses penentuan *support vector* dapat dirumuskan dalam *QP problem*. Dengan demikian jika kita memiliki *library* untuk menyelesaikan *QP problem*, dengan sendirinya SVM dapat diimplementasikan dengan mudah. Selain itu dapat diselesaikan dengan metode sekuensial.

Disamping kelebihannya, SVM memiliki kelemahan pula, antara lain:

1. Sulit digunakan dalam masalah berskala besar. Skala besar dalam hal ini dimaksudkan dengan jumlah *sample* data yang diolah.
2. SVM secara teoritik dikembangkan untuk *problem* klasifikasi dengan dua *class*. Dewasa ini SVM telah dimodifikasi agar dapat menyelesaikan masalah dengan *class* lebih dari dua. Namun demikian, masing-masing strategi dari berbagai penelitian masih memiliki kelemahan, sehingga dapat dikatakan penelitian dan pengembangan SVM pada *multi-class problem* merupakan tema penelitian yang masih terbuka.

Seperti yang telah dijelaskan sebelumnya, bahwa konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 7.1(a) memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class*: +1

dan -1 . *Pattern* yang tergabung pada *class* -1 disimbolkan kotak, sedangkan *pattern* pada *class* $+1$ disimbolkan lingkaran. Masalah klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada Gambar 7.1(a). *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 7.1(b) menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik kotak dan lingkaran yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.



Gambar 6.4 Cara Kerja Hyperplane SVM

SVM mempelajari pemetaan input (X) \rightarrow output (Y), yang mana (x_{ij}, y_i) adalah *instance ke-i* = {1, 2, 3, ..., n} dan atribut (variabel input) *ke-j* = {1, 2, 3, ..., m}; $y \in \{\pm 1\}$ adalah label/class/output; dan $y_i = \{y_1, y_2, \dots, y_n\}$, merupakan *label class* dari *instance ke-i*. Langkah-langkah pelatihan dan pengujian SVM adalah sebagai berikut:

1. *Input/Output sets*: X, Y ;
2. *Training set*: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$;
3. *Generalization*: dari $x \in X$ temukan kecocokannya dengan $y \in Y$;
4. Misal, mempelajari/melatih sebuah pengklasifikasi $y = f(x, \alpha)$, dimana α sebuah parameter pada fungsi tersebut;
5. Jika kita memilih model dari pelatihan tersebut dengan menset pemisah (*hyperplane*) dalam R^n , maka kita mempunyai $f(x, \{w, b\}) = sign(w.x+b)$; dan
6. Fungsi pemisah tersebut digunakan untuk menentukan output, yang mana dimensi dan kapasitas fungsi sebaiknya: $test\ error \leq (training\ error + complexity\ of\ set\ of\ models)$.
- 7.

Fungsi *hyperplane* SVM dapat didefinisikan sebagai berikut:

$$f(x) = w \cdot x + b \quad (102)$$

Variabel w dan b adalah variabel yang ingin dicari nilainya dengan meminimumkan:

$$\frac{1}{2} \|\vec{w}\|^2 \quad (103)$$

Di bawah kendala:

$$y_i(w \cdot x_i + b) \geq 1 \quad (104)$$

Pada formulasi di atas, data diasumsikan 100% dapat terklasifikasikan dengan benar oleh *hyperplane*. Namun umumnya data tidak dapat terklasifikasikan 100% benar, sehingga proses optimalisasi tidak dapat diselesaikan karena tidak ada w dan b yang memenuhi Pertidaksamaan (104). Dengan demikian, masalah ini diatasi dengan *Soft Margin*, yaitu dengan memasukkan variabel *Slack*, dan C yg merupakan parameter yang mengontrol *trade off* antara *margin* dan *error* klasifikasi. Hal ini disebut dengan *minimize training error* yang didefinisikan sebagai berikut:

$$P(w, b) = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (105)$$

Sebelah kiri operator $+$ adalah *maximize margin*, sedangkan sebelah kanan adalah *minimize training error*, di bawah kendala berikut ini:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (106)$$

Atau:

$$y_i(w \cdot x_i + b) + \xi_i \geq 1; \xi_i \geq 0; (i = 1, 2, \dots, n) \quad (107)$$

Sehingga selanjutnya untuk menyelesaikan/menyederhanakan Persamaan (103) dan Pertidaksamaan (104), maka digunakan pendekatan *Lagrange Multiplier* sebagai berikut:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i(w \cdot x_i + b) - 1); \alpha_i \geq 0 \quad (108)$$

Persamaan ini harus diturunkan pada variabel w dan b , sehingga:

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (109)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (110)$$

Karena w dan b tidak bisa langsung didapatkan nilainya, maka Persamaan (109) dan (110) disubtitusikan ke Persamaan (108) sehingga diperoleh *dualitas optimalitas*, ini disebut *quadratic programming* yang didefinisikan sebagai berikut:

$$L(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (111)$$

Di bawah kendala:

$$\alpha_i \geq 0; (i = 1, 2, \dots, n) \quad (112)$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (113)$$

Instance x_i dari *training set* yang α_i tidak bernilai 0 itulah yang disebut *support vector*. Apabila α telah diperoleh, maka w (110) dapat diperoleh, sedangkan b diperoleh melalui persamaan berikut ini.

$$b = -\frac{1}{2}(w \cdot x_{-1} + w \cdot x_{+1}) \quad (114)$$

Dengan demikian, klasifikasi data dapat dihitung dengan menggunakan persamaan berikut ini.

$$f(\vec{t}) = \text{sign} \left(\sum_{i=1, x_i \in SV}^n \alpha_i y_i < \vec{t}, x_i > + b \right) \quad (115)$$

Untuk menangani masalah non linier, maka pendekatan *Kernel* dapat diterapkan. Fungsi *Kernel SVM* dapat didefinisikan sebagai berikut.

$$< \varphi(x_i) \cdot \varphi(x) > = K(x_i, x) \quad (116)$$

Tipe *Kernel* yang dapat digunakan adalah *dot* atau *linear* (117), *polynomial* (118), *gaussian* (119), *sigmoid* (120), RBF (121), dll.

$$K(x, y) = x \cdot y \quad (117)$$

$$K(x, y) = < x, y >^d \quad (118)$$

$$K(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\sigma^2} \right); \sigma > 0 \quad (119)$$

$$K(x, y) = \tanh(K < x, y > + \delta); K > 0; \delta < 0 \quad (120)$$

$$K(x, y) = \exp(-\gamma \|x, y\|^2); \gamma > 0 \quad (121)$$

Contoh 6.6 SVM: Klasifikasi Biner (Manual)

Contoh 1:

X1	X2	Y
1	1	+1
-1	1	-1
1	-1	-1
-1	-1	-1

$$P(w, b) = \frac{1}{2}(w_1^2 + w_2^2) + C(\xi_1 + \xi_2 + \xi_3 + \xi_4)$$

$$w_1 + w_2 + b + \xi_1 \geq 1 \rightarrow w_1 = 1(1+b) + \xi_i \geq 1; w_2 = 1(1+b) + \xi_i \geq 1$$

$$w_1 - w_2 - b + \xi_2 \geq 1 \rightarrow w_1 = -1(-1-b) + \xi_i \geq 1; w_2 = -1(1-b) + \xi_i \geq 1$$

$$-w_1 + w_2 - b + \xi_3 \geq 1 \rightarrow w_1 = -1(1-b) + \xi_i \geq 1; w_2 = -1(-1-b) + \xi_i \geq 1$$

$$w_1 + w_2 - b + \xi_4 \geq 1 \rightarrow w_1 = -1(-1-b) + \xi_i \geq 1; w_2 = -1(-1-b) + \xi_i \geq 1$$

Karena kasus linier, maka bisa dipastikan variabel *slack* = 0 dan $C = 0$, maka diperoleh:

$$w_1 = 1; w_2 = 1; b = -1$$

Sehingga fungsi pemisahnya (*hyperplane*) adalah:

$$f(x) = x_1 + x_2 - 1$$

Dengan demikian:

$$g(x) = \text{sign}(x); \text{if } f(x) \geq 1 \text{ Then } +1; \text{ Else } -1$$

$$\text{Class instance ke-1} : (1+1)-1 = 1 \rightarrow +1$$

$$\text{Class instance ke-2} : (-1+1)-1 = -1 \rightarrow -1$$

$$\text{Class instance ke-3} : (1-1)-1 = -1 \rightarrow -1$$

$$\text{Class instance ke-4} : (-1-1)-1 = -3 \rightarrow -1$$

Contoh 2:

X1	X2	X3	Y
-1	-1	-2	-1
2	-2	-1	-1
-2	-2	2	+1

Misanya diketahui fungsi *hyperplane* sebagai berikut:

$$f(x) = x_1 + 2x_2 + 2x_3 + 3$$

Maka:

$$w = (1, 2, 2); b = 3$$

Margin:

$$M = \frac{2}{\sqrt{1^2 + 2^2 + 2^2}} = 2/3$$

$$\text{Instance ke-1} : -1*((1*(-1)+2*(-1)+2*(-2))+3) \rightarrow 4$$

$$\text{Instance ke-2} : -1*((1*2+2*(-2)+2*(-1))+3) \rightarrow 1$$

$$\text{Instance ke-3} : 1*((1*(-2)+2*(-2)+2*2))+3 \rightarrow 1$$

Output = 1 adalah data yang masuk *support vector*, yaitu *instance* ke-2 dan ke-3.

Contoh 6.7 SVM: Klasifikasi (Matlab)

<i>Dataset</i>	: <i>dsHeartDiseaseCleveland</i> – Class: {0,1} (terlampir)
<i>Data validation</i>	: <i>10-Fold Cross Validation</i>
<i>Kernel Function</i>	: <i>RBF</i>
<i>Evaluation</i>	: <i>Confusion Matrix</i>

Fungsi “SVMaa”:

```

function [output, lamaProses, akurasi] = SVMaa(dataLatihInput,
dataLatihOutput, dataUjiInput, dataUjiOutput)
tic;
    %% SVM Training:
    SVMmodel = svmtrain(dataLatihInput, dataLatihOutput,
'Kernel_Function', 'rbf');
    %% SVM Testing:
    output = svmclassify(SVMmodel, dataUjiInput);
    %% SVM Evaluasi:
    %MSE = perform(ANNmodel, output, dataUjiOutput);
    conMat = confusionmat(dataUjiOutput, output); % actual vs model
    jmlData = sum(conMat(:));
    hasilBenar = sum(diag(conMat));
    akurasi = 100 * (hasilBenar / jmlData);
    lamaProses = toc;
    plot(dataUjiInput,dataUjiOutput,'o',dataUjiInput,output,'+')
end

```

Script “SVM”:

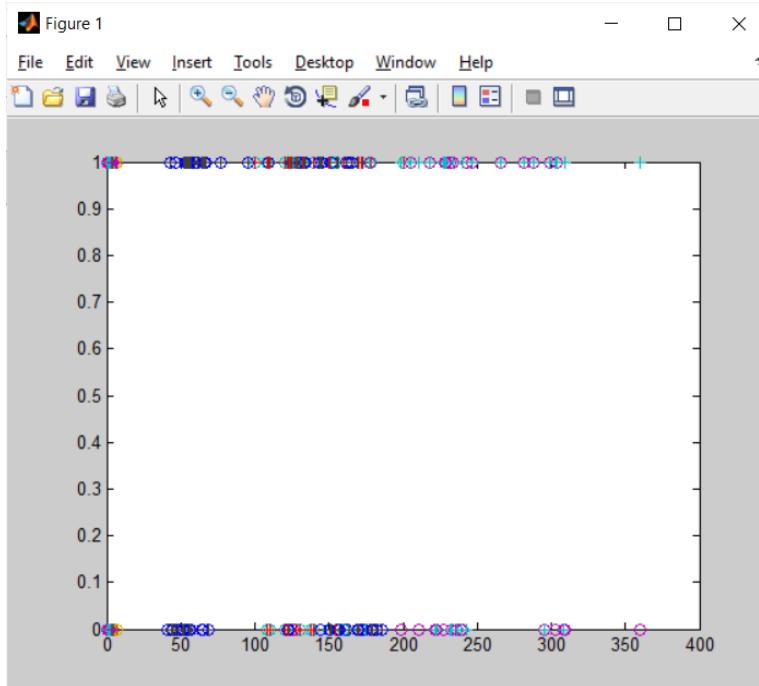
```

clc; clear; close all; warning off all;
%% data (Kolom: 1 ID, 2-14 Input, 15 Output 0-4, 16 Output 0-1)
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dsInput = dsHeartDisease(:,2:14); % inputan: ID tidak digunakan
dsOutput = dsHeartDisease(:,16); % output: yang binary classification
dataset = [dsInput, dsOutput];
%% Variabel untuk hasil SVM
SVMhasilSub = []; %kolom: 1 lama proses, 2 akurasi
SVMoutputAll.x = []; %Output SVM di setiap K dr K-Fold
SVMhasil = []; %kolom: 1 lama proses, 2 acc, 3 acc max, 4 acc min
%% K-Fold Cross Validation
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K); % kolom 14 = class
for i = 1:K
    %% Buat Data Latih dan Data Uji berdasarkan indeks dari K-Fold
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
    subDataLatihInput = dataset(latih,1:13); %input training
    subDataLatihOutput = dataset(latih,14); %output training
    subDataUjiInput = dataset(uji,1:13); % input testing
    subDataUjiOutput = dataset(uji,14); % output testing
    %% SVM Modelling di tiap K
    [SVMsubOutput, SVMsubLamaProses, SVMsubAkurasi] =
SVMaa(subDataLatihInput, subDataLatihOutput, subDataUjiInput,
subDataUjiOutput);
    SVMhasilSub(i,1) = SVMsubLamaProses; % hasil lama proses dalam K
    SVMhasilSub(i,2) = SVMsubAkurasi; % hasil akurasi dalam K
    SVMoutputAll(i).x = [subDataUjiInput subDataUjiOutput
SVMsubOutput];
end
%% Hasil Akhir SVM

```

```
SVMhasil(1)=mean(SVMhasilSub(:,1)); % rata2 lama proses
SVMhasil(2)=mean(SVMhasilSub(:,2)); % akurasi akhir
SVMhasil(3)=max(SVMhasilSub(:,2)); % akurasi max
SVMhasil(4)=min(SVMhasilSub(:,2)); % akurasi min
```

Hasilnya ketika *K-Fold Cross Validation* di K = 10:



Akurasi = 75,9032% (lebih baik daripada ANN)

Akurasi max = 87,0968%

Akurasi min = 66,6667%

Lama proses = 0,0412 detik (lebih cepat daripada ANN)

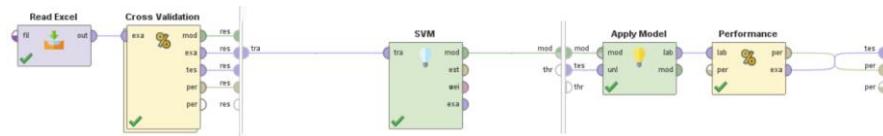
Contoh 6.8 SVM: Regresi (Rapidminer)

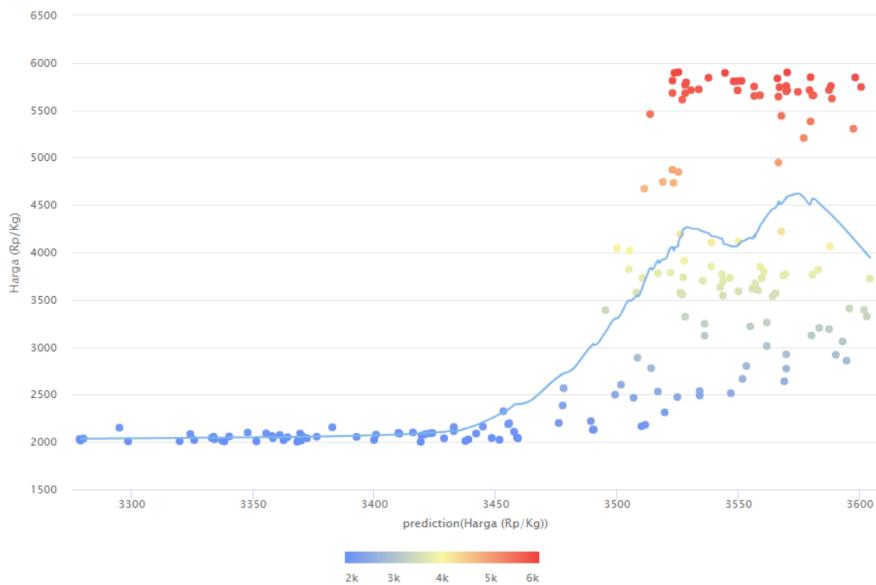
Dataset : dsPanganTimeSeries (terlampir).

Data validation : 10-Fold Cross Validation.

Kernel Function : dot

Evaluation : RMSE





Root Mean Squared Error (RMSE): 1347.703 +/- 110.793 (micro average: 1351.796 +/- 0.000), menunjukkan hasil yang tidak lebih baik daripada ANN.

Pada masalah klasifikasi penyakit jantung, SVM menunjukkan akurasi yang lebih baik daripada ANN. Namun pada masalah estiamsi harga pangan, SVM menunjukkan akurasi yang tidak lebih baik daripada ANN.

6.4 Multi-Class SVM

Seperti telah disinggung di awal bab ini, bahwa pada masalah klasifikasi, pada prinsipnya SVM diperuntukkan pada kasus *binary classification*. Terdapat banyak pendekatan yang dapat digunakan agar SVM dapat menangani klasifikasi dengan *multi class*, misalnya seperti: *One-Versus-Rest* (1VR) SVM oleh Vapnik pada tahun 1998, *One-Versus-One* (1V1) SVM oleh Krebel pada tahun 1999, *Weston & Watkins* (W&W) SVM pada tahun 1999, *Multi-Class Kernel-Based Vector Machine* pada tahun 2001 oleh Crammer & Singer, dsb [45].

Pada prinsipnya, ada dua pilihan untuk mengimplementasikan *Multi-Class SVM*, yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa *class* ke dalam bentuk optimasi [46]. Dalam praktiknya, masalah *multi-class* umumnya didekomposisi menjadi serangkaian masalah biner sehingga standar SVM dapat langsung diterapkan [44], seperti pada 1VR dan 1V1. Sedangkan W&W dan *Multi-Class Kernel-Based Vector Machine* mengkombinasikan masalah *multiple binary-class optimization* menjadi satu fungsi tujuan tunggal, namun berakibat kompleksitas komputasi yang besar karena ukuran pada masalah *Quadratic Programming* (QP) [44], [46].

Suykens & Vandewalle pada tahun 1999 memperluas metode LS-SVM menjadi salah satu pilihan untuk *multi-class* SVM, namun kelemahan dari LS-SVM

adalah solusinya dibangun dari sebahagian besar data latih dimana hal ini disebut sebagai masalah *non-sparsenes* [44]. Namun pada tahun 2008, Xia & Lee menghadirkan sebuah metode LS-SVM yang baru untuk menangani masalah *multi-class* SVM, yang mana solusinya merupakan *sparse* dalam koefisien *weight* pada *support vector*, pendekatan tersebut (LS-SVM) berkaitan erat dengan pendekatan 1VR [44].

Pendekatan 1V1 atau dekomposisi berpasangan diperkenalkan oleh Knerr et. al., pada tahun 1990. Pendekatan ini mengevaluasi semua *classifiers* yang mungkin berpasangan dan dengan demikian menginduksi *binary classification*. Setiap *classifier* diterapkan ke data uji yang akan memberikan satu nilai untuk *class pemenang*. Ukuran pengklasifikasi yang diciptakan oleh pendekatan 1V1 jauh lebih besar dibandingkan dengan pendekatan 1VR. Namun, ukuran QP di setiap *classifier* lebih kecil, yang memungkinkan untuk lebih cepat dalam pelatihan. Selain itu, dibandingkan dengan pendekatan 1VR, 1V1 yang lebih simetris [44]. Merujuk pula pada penelitian yang dilakukan oleh Hsu & Lin pada tahun 2002 yang memberikan komparasi berbagai metode untuk masalah *multi-class* SVM dan menyatakan bahwa 1V1 yang terbaik saat itu [46].

1V1 SVM melakukan dekomposisi *class* dengan menggunakan persamaan berikut ini [45]:

$$c = \text{cls}(\text{cls} - 1)/2 \quad (122)$$

Misalnya terdapat 4 *label class* (*c1*, *c2*, *c3*, dan *c4*) yang akan didekomposisi berpasangan:

$$c = \frac{4 * (4 - 1)}{2} = 6$$

Maka terdapat 6 *class biner* sebagai berikut:

Class +1	V	Class -1
<i>c1</i>	V	<i>c2</i>
<i>c1</i>	V	<i>c3</i>
<i>c1</i>	V	<i>c4</i>
<i>c2</i>	V	<i>c3</i>
<i>c2</i>	V	<i>c4</i>
<i>c3</i>	V	<i>c4</i>

Sebenarnya fungsi ini sama saja dengan pendekatan kombinasi. Selanjutnya untuk menentukan klasifikasi, maka digunakan pendekatan *mode* (hasil klasifikasi yang terbanyak dari tiap-tiap *class biner*). Ilustrasinya sebagai berikut:

Data ke-	Multi-Class						Binary Class
	<i>c1Vc2</i>	<i>c1Vc3</i>	<i>c1Vc4</i>	<i>c2Vc3</i>	<i>c2Vc4</i>	<i>c4Vc5</i>	
1	<i>c1</i>	<i>c1</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c1</i>
2	<i>c1</i>	<i>c1</i>	<i>c2</i>	<i>c2</i>	<i>c2</i>	<i>c4</i>	<i>c2</i>
...
n	<i>c3</i>	<i>c1</i>	<i>c4</i>	<i>c2</i>	<i>c3</i>	<i>c3</i>	<i>c3</i>

Contoh 6.9 1V1 SVM: Multi Classification (Matlab)

Dataset Input : *dsHeartDiseaseCleveland(I)* (terlampir)
 Dataset Output : *dsHeartDiseaseCleveland(O1)* (terlampir)
 Class = {positif, resiko rendah, resiko sedang, resiko tinggi, positif}
 Data validation : *Holdout (10%)*
 Kernel Function : *Polynomial*
 Evaluation : *Confusion Matrix*

```

clc; clear; close all; warning off all;
tic;
%% set data input
[~, ~, raw] = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland(I)');
R = cellfun(@(x) ~isnumeric(x) || isnan(x),raw); %Find non-numeric
raw(R) = {0.0}; % Replace non-numeric cells
data = cell2mat(raw); % Create data
clearvars raw R; % clear temporary variables
%% Import Class
[~, ~, kelas] = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland(O1)');
kelas(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),kelas)) =
{''};
%% nominal class to numeric
% g --> kelas yang diubah ke numerik
% gn --> kelas asli yg berjumlah 5
[g gn] = grp2idx(kelas);
%% split training/testing sets
[trainIdx testIdx] = crossvalind('HoldOut', kelas, 0.1);
%% 1-vs-1 pairwise models --> c(c-1)/2
pairwise = nchoosek(1:length(gn),2);
svmModel = cell(size(pairwise,1),1);
predTest = zeros(sum(testIdx),numel(svmModel));
%% Klasifikasi menggunakan 1v1 to SVM dengan 3rd degree polynomial
for k=1:numel(svmModel)
    idx = trainIdx & any( bsxfun(@eq, g, pairwise(k,:)) , 2 );
    svmModel{k} = svmtrain(data(idx,:), g(idx), ...
        'BoxConstraint',2e-1, 'Kernel_Function','polynomial',
        'polyorder',3);
    predTest(:,k) = svmclassify(svmModel{k}, data(testIdx,:));
    XsupportVector{k} = svmModel{k}.SupportVectors;
    XindexSupportVector{k} = svmModel{k}.SupportVectorIndices;
    xxx = data(idx,:);
    XdataSupportVector{k} = xxx(XindexSupportVector{k}',:);
    Xalpha{k} = svmModel{k}.Alpha;
    Xbias{k} = svmModel{k}.Bias;
    XkernelFun{k} = svmModel{k}.KernelFunction;
    XkernelFunArgs{k} = svmModel{k}.KernelFunctionArgs;
end
%% Evaluation
pred = mode(predTest,2); % voting: clasify as the class receiving
cmat = confusionmat(g(testIdx),pred);
jmlDataTest = sum(cmat(:));
hasilBenar = sum(diag(cmat));
hasilSalah = jmlDataTest - hasilBenar;
acc = 100*sum(diag(cmat))/sum(cmat(:));
lajuError = 100 * (hasilSalah / jmlDataTest);
kelasTestAsli = g(testIdx);

```

```

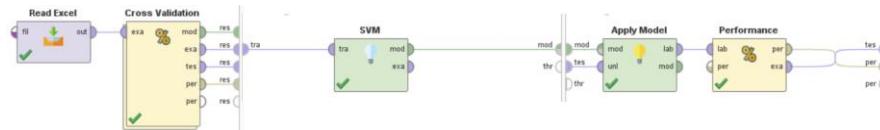
lamaProses = toc;
%% Print
disp(' ');
disp('-----');
fprintf('Confusion Matrix:\n'), disp(cmat);
fprintf('Jumlah Data Test = %.0f\n', jmlDataTest);
fprintf('Hasil Benar = %.0f\n', hasilBenar);
fprintf('Hasil Salah = %.0f\n', hasilSalah);
fprintf('Training & Testing Time (Seconds) = %.4f\n', lamaProses);
fprintf('Akurasi = %.2f%%\n', acc);
fprintf('Laju Error = %.2f%%\n', lajuError);
disp('-----');
disp(' ');

-----
Confusion Matrix:
  13      0      2      1      0
  2      1      0      0      0
  4      1      0      0      0
  0      0      2      0      1
  0      0      0      1      0

Jumlah Data Test = 28
Hasil Benar = 14
Hasil Salah = 14
Training & Testing Time (Seconds) = 1.8268
Akurasi = 50.00%
Laju Error = 50.00%
-----
```

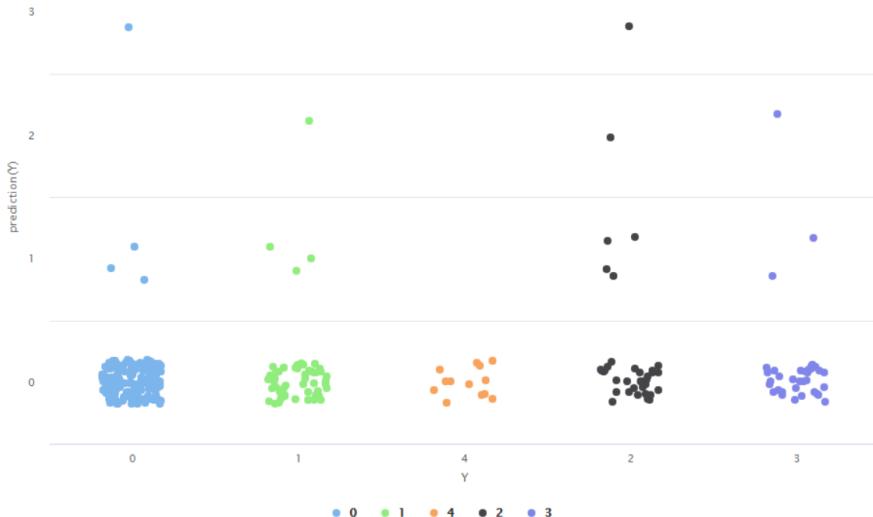
Contoh 6.10 LibSVM: Multi Classification (Rapidminer)

- Dataset : *dsHeartDiseaseCleveland – Class: {0,1,2,3,4}* (terlampir)
 Data validation : *10-Fold Cross Validation*
 Kernel Function : *RBF*
 Evaluation : *Confusion Matrix*



accuracy: 54.12% +/- 2.47% (micro average: 54.13%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred. 0	160	30	51	32	13	55.94%
pred. 2	0	1	1	1	0	33.33%
pred. 1	3	4	3	2	0	25.00%
pred. 3	1	1	0	0	0	0.00%
pred. 4	0	0	0	0	0	0.00%
class recall	97.56%	2.78%	5.45%	0.00%	0.00%	



Hasil dari berbagai contoh ini mengindikasikan bahwa prediksi Penyakit Jantung yang memiliki masalah *unbalanced class* yang sangat tinggi menggunakan metode SVM dengan pendekatan *unbalanced class* secara manual (menyatukan *class 0 = 0* dan *class 1 = 1, 2, 3, dan 4*) sekaligus agar SVM dapat menangani data tersebut, memberikan akurasi sebesar 75,9032%. Sedangkan menggunakan metode 1V1 SVM tanpa mereduksi *unbalanced class* secara manual, memberikan akurasi sebesar 50%. Apabila menggunakan metode LibSVM tanpa mereduksi *unbalanced class* pula, memberikan akurasi sebesar 54,13%.

Dengan demikian, mungkin dapat saja kita tarik suatu kesimpulan bahwa masalah *unbalanced class* memang sangat mempengaruhi kinerja model, sehingga lebih baik direduksi lebih dahulu, mungkin dengan pendekatan tertentu, daripada memaksakan metode *machine learning* yang digunakan untuk menyelesaiakannya.

6.5 Fuzzy SVM

Fuzzy SVM (FSVM) sebenarnya merupakan pengembangan SVM untuk dapat menangani masalah *multi-class classification*, sama seperti metode-metode yang telah dijelaskan sebelumnya, 1V1 SVM dan LibSVM, namun dengan pendekatan yang berbeda. FSVM menggunakan pendekatan *Membership Function* pada *class* yang tidak dapat diklasifikasikan *binary SVM*.

FSVM menggunakan fungsi *hyperplane* (102) yang diperoleh SVM, yang mana terdapat data yang tidak dapat diklasifikasikan pada fungsi tersebut dalam masalah *multi-class classification*, sehingga digunakan *Membership Function* untuk mengklasifikasikan data yang tidak dapat diklasifikasikan oleh fungsi *hyperplane* SVM ($f_{ij}(x)=w_{ij}x+b_{ij}$) untuk *class i* dan *class j*, yang didefinisikan sebagai berikut.

$$m_{ij} = \begin{cases} 1; & f_{ij}(x) \geq 1 \\ f_{ij}(x); & \text{lainnya} \end{cases} \quad (123)$$

Dengan menggunakan Persamaan (123), maka dapat didefinisikan *Membership Function* x terhadap *class i* sebagai berikut:

$$\begin{aligned} m_i(x) &= \min_{j=1,2,\dots,n} m_{ij}(x) \\ m_i(x) &= \min (1, \min_{j \neq i, j=1,2,\dots,n} f_{ij}(x)) \\ m_i(x) &= \min_{i \neq j, j=1,2,\dots,n} f_{ij}(x) \end{aligned} \quad (124)$$

Dengan begitu, *instance ke-x* dapat diklasifikasikan berdasarkan derajat keanggotaan yang paling tinggi.

6.6 Soal Latihan ANN, SVM, & Fuzzy

1. Unduh salah satu *dataset binary classification* pada *UCI Machine Learning Repository* kemudian lakukan analisis klasifikasi menggunakan algoritma ANN dan SVM, mana yang terbaik untuk *dataset* yang anda gunakan?
2. Unduh salah satu *dataset* regresi pada *UCI Machine Learning Repository* kemudian lakukan analisis regresi menggunakan algoritma ANN dan SVM, mana yang terbaik untuk *dataset* yang anda gunakan?
3. Unduh salah satu *dataset multi-class classification* pada *UCI Machine Learning Repository* kemudian lakukan analisis klasifikasi menggunakan algoritma 1V1 SVM, LibSVM, dan Fuzzy SVM.
4. Kumpulkan data arus lalu lintas jangka pendek pada salah satu *traffic light* di persimpangan yang berada di lokasi anda, gunakan variabel input dan output seperti pada contoh ANFIS dan lakukan analisis regresi menggunakan ANFIS.

7. K-Means & Fuzzy C-Means

No.	Materi	Tujuan Pembelajaran
1.	K-Means	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>K-Means</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasterisasi.
2.	Fuzzy C-Means (FCM)	Anda mampu memahami, menjelaskan, dan menerapkan algortima FCM secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasterisasi.

7.1 K-Means

K-Means merupakan salah satu algoritma *machine learning* untuk menangani masalah klasterisasi (*unsupervised learning*). *K-Means* merupakan salah satu metode klasterisasi non hirarki. Algoritma ini berusaha menemukan pusat dari kelompok dalam data sebanyak iterasi perbaikan yang dilakukan.

K-Means mempartisi data ke dalam dua atau lebih kelompok data, yang mana data yang memiliki karakteristik yang sama dimasukkan dalam suatu kelompok yang sama, sedangkan data memiliki karakteristik berbeda dimasukkan dalam kelompok lainnya. Karakteristik tersebut berdasarkan jarak (*dissimilarity*) data dengan pusat kelompok data (*centroid*) yang telah ditentukan di awal proses dan diperbarui setiap proses iterasi *K-Means* selama masih ada data yang berpindah kelompok atau apabila perubahan centroid/fungsi obyektif yang digunakan masih melebihi ambang batas yang ditentukan.

Parameter K pada *K-Means* menunjukkan jumlah kelompok data. Pemilihan nilai K yang optimal merupakan hal sulit yang dilakukan. Seandainya ada informasi mengenai kelompok-kelompok data dalam *dataset* yang diolah, seperti jumlah partisi yang secara alami menggambarkan *dataset*, maka informasi tersebut dapat digunakan sebagai nilai K yang optimal. Jika tidak ada, maka cara lainnya dengan menggunakan solusi yang naif, yaitu dengan mencoba beberapa nilai K berbeda dan memilih *clustering* yang nilai fungsi objektifnya terkecil (minimal). Sayangnya nilai yang diberikan oleh fungsi objektif tidak cukup informatif untuk digunakan sebagai harapan penyelesaian masalah ini. Misalnya biasa solusi optimal terhadap peningkatan K menurun sampai menjadi 0 ketika jumlah *cluster* sama dengan jumlah titik data berbeda. Cara lainnya adalah dengan melakukan evaluasi klasterisasi menggunakan pendekatan evaluasi internal, eksternal, maupun evaluasi aplikasi seperti yang telah dibahas sebelumnya.

K-Means mencapai kondisi konvergen ketika pengalokasian kembali titik data (dan juga lokasi *centroid*) tidak lagi berubah. Proses dari iterasi ke iterasi hingga dicapai kondisi konvergen juga dapat diamati dari nilai fungsi obyektif yang didapatkan, semakin konvergen maka nilai fungsi obyektif semakin menurun. Pemilihan K titik data sebagai *centroid* juga mempengaruhi hasil *cluster*, sehingga hasilnya bisa berbeda-beda di tiap percobaan. Kondisi seperti itu dikenal dengan solusi yang *local optima*. Dengan demikian, *K-Means* sangat sensitif terhadap *centroid* awal. Inisialisasi yang jelek dapat mengakibatkan hasil *cluster* yang jelek pula. Masalah *local optima* ini dapat diselesaikan dengan menjalankan algoritma beberapa kali dengan inisial *centroid* yang berbeda kemudian memilih hasil yang terbaik berdasarkan hasil evaluasinya.

K-Means mengelompokkan secara tegas data hanya pada satu *cluster*, maka nilai suatu data pada semua *cluster*, hanya salah satu *cluster* yang bernilai 1, sedangkan *cluster* yang lainnya 0, seperti dinyatakan pada persamaan berikut ini.

$$a_{ij} = \begin{cases} 1; \min\{d(x_i, C_j)\} \\ 0; \quad \text{Lainnya} \end{cases} \quad (125)$$

$d(x_i, C_j)$ menyatakan jarak (*dissimilarity*) data (x_i) ke *cluster* (C_j).

Relokasi *centroid* untuk memperoleh titik *centroid* (C), diperoleh dengan menghitung rata-rata setiap fitur/atribut dari semua data yang tergabung dalam setiap *cluster* yang didefinisikan sebagai berikut.

$$C_j = \frac{1}{Nk} \sum_{i=1}^{Nk} x_j \quad (126)$$

Nk adalah jumlah data dalam suatu *cluster*.

Suatu data akan selalu memilih suatu *cluster* dengan jarak yang terdekat, maka sebenarnya *K-Means* berusaha meminimalkan fungsi obyektif yang dapat didefinisikan sebagai berikut.

$$f(x) = \sum_{i=1}^N \sum_{j=1}^K a_{ij} d(x_i, C_j)^2 \quad (127)$$

N adalah jumlah data, K adalah jumlah *cluster*, $d(x_i, C_j)$ menyatakan jarak (*dissimilarity*) data ke- i ke *cluster* C_j , $a_{ij} = 1$ menyatakan bahwa data tersebut merupakan anggota suatu *cluster*, sementara $a_{ij} = 0$ menyatakan sebaliknya.

Ada beberapa pendekatan perhitungan jarak (*dissimilarity*) yang dapat digunakan untuk mengukur jarak suatu data ke *centroid*, antara lain *Euclidean* (Bezdek, 1981), *Manhattan* atau *Cityblock* (Miyamoto & Agusta, 1995), *Minkowski* (Miyamoto & Agusta, 1995), dll [20].

Jarak antara dua buah data x' dan x menggunakan *Euclidean Distance* dapat didefinisikan sebagai berikut.

$$D(x', x) = \|x' - x\|_2 = \sqrt{\sum_{i=1}^n (x'_i - x_i)^2} \quad (128)$$

Jarak antara dua buah data (x') dan (x) menggunakan *Manhattan Distance* dapat didefinisikan sebagai berikut.

$$D(x', x) = \|x' - x\|_1 = \sum_{i=1}^n \text{abs}(x'_i - x_i) \quad (129)$$

Jarak antara dua buah data (x') dan (x) menggunakan *Minkowski Distance* dapat didefinisikan sebagai berikut.

$$D(x', x) = \|x' - x\|_\lambda = \sqrt[\lambda]{\sum_{i=1}^n (x'_i - x_i)^\lambda} \quad (130)$$

$\lambda = 1$ adalah ruang jarak yang sama dengan *Manhattan*, $\lambda = 2$ adalah ruang jarak yang sama dengan *Euclidean*, dan $\lambda = \infty$ adalah ruang jarak yang sama dengan *Chebyshev*.

Euclidean dan *Manhattan* merupakan metode yang paling umum digunakan. *Euclidean* menjadi pilihan untuk jarak terdekat (garis lurus), sedangkan *Manhattan* untuk jarak terjauh. Jika data mengandung banyak *outlier*, maka *Manhattan* menjadi pilihan yang lebih baik.

Algoritma *K-Means* adalah sebagai berikut:

1. Inisialisasi K (jumlah *cluster*), $f(x)$ (fungsi obyektif) awal, *centroid* awal (K data sebagai *centroid* awal), dan tetapkan ambang batas untuk perubahan fungsi objektif dan perubahan *centroid*;
2. Alokasikan tiap-tiap data ke *centroid* yang terdekat menggunakan salah satu metode pengukuran jarak (*Euclidean*, *Manhattan*, atau lainnya);
3. Hitung kembali *centroid* (C) dan $f(x)$ (fungsi obyektif) berdasarkan data yang mengikuti *cluter*-nya masing-masing.
4. Ulangi langkah 2 dan 3 hingga kondisi konvergen tercapai, yaitu apabila perubahan $f(x)$ (fungsi obyektif) \leq ambang batas, atau apabila perubahan *centroid* \leq ambang batas, atau sudah tidak ada data yang berpindah *cluster*.

Contoh 7.1 K-Means: Klasterisasi (Manual)

Data Ke-	X1	X2
1	1	1
2	4	1
3	6	1
4	1	2
5	2	3
6	5	3
7	2	5
8	3	5
9	2	6
10	3	8

Inisialisasi:

$$\begin{aligned}
 K &= 3 \\
 \text{Ambang batas} &= 0,1 \\
 f(x) &= 1000 \\
 \text{Distance measure} &= \text{Euclidean} \\
 \text{Centroid awal} &:
 \end{aligned}$$

Data Ke-	X1	X2	Centroid
2	4	1	1
4	1	2	2
6	5	3	3

Iterasi ke-1:

Jarak data ke-1 ($x_1 = [1, 1]$) ke *centroid/cluster* 1 ($c_1 = [4, 1]$) adalah:

$$D(c_1, x_1) = \sqrt{(1-4)^2 + (1-1)^2} = \sqrt{9} = 3$$

Lengkapnya, jarak setiap data ke *centroid* dan *cluster* yang diikutinya ditunjukkan pada tabel berikut ini.

Data Ke-	X1	X2	Jarak ke Centroid			a (Cluster yang diikuti)		
			1	2	3	1	2	3
1	1	1	3,0000	1,0000	4,4721	0	1	0
2	4	1	0,0000	3,1623	2,2361	1	0	0
3	6	1	2,0000	5,0990	2,2361	1	0	0
4	1	2	3,1623	0,0000	4,1231	0	1	0
5	2	3	2,8284	1,4142	3,0000	0	1	0
6	5	3	2,2361	4,1231	0,0000	0	0	1
7	2	5	4,4721	3,1623	3,6056	0	1	0
8	3	5	4,1231	3,6056	2,8284	0	0	1
9	2	6	5,3852	4,1231	4,2426	0	1	0
10	3	8	7,0711	6,3246	5,3852	0	0	1
			Jumlah			2	5	3

Banyaknya data yang masuk di $c_1 = 2$, $c_2 = 5$, dan $c_3 = 3$.

Hitung nilai setiap *centroid* baru, yaitu jumlah nilai data dibagi banyaknya data dalam *centroid/cluster* tersebut.

$$c_{1,1} = \frac{10}{2} = 5,0000; c_{1,2} = \frac{2}{2} = 1,0000$$

$$c_{2,1} = \frac{8}{5} = 1,6000; c_{2,2} = \frac{17}{5} = 3,4000$$

$$c_{3,1} = \frac{11}{3} = 3,6667; c_{3,2} = \frac{16}{3} = 5,3333$$

Selanjutnya hitung jarak setiap data ke *cluster*-nya (*centroid* baru tersebut) masing-masing.

$D(c_1, x_1) = 0$; karena data x_1 tidak masuk dalam *cluster* 1 (c_1), tapi masuk di c_2 .

$$D(c_2, x_1) = \sqrt{(1 - 1,6000)^2 + (1 - 3,4000)^2} = \sqrt{6,12} = 2,4739$$

Lengkapnya, ditunjukkan pada tabel berikut ini.

Selanjutnya hitung $f(x)$ (fungsi obyektif) baru dengan menjumlahkan jarak seluruh *cluster*.

$$\text{Jumlah } D(c_1, x_{1,2,\dots,n}) = D(c_1, x_2) + D(c_1, x_3) = 1 + 1 = 2$$

$$\begin{aligned} \text{Jumlah } D(c_2, x_{1,2,\dots,n}) &= D(c_2, x_1) + D(c_2, x_4) + D(c_2, x_5) + D(c_2, x_7) + D(c_2, x_9) \\ &= 2,4739 + 1,5232 + 0,5657 + 1,6492 + 2,6306 = 8,8425 \end{aligned}$$

$$\begin{aligned} \text{Jumlah } D(c_3, x_{1,2,\dots,n}) &= D(c_3, x_6) + D(c_3, x_8) + D(c_3, x_{10}) \\ &= 2,6874 + 0,7454 + 2,7487 = 6,1815 \end{aligned}$$

$$\begin{aligned} f(x) &= D(c_1, x_{1,2,\dots,n}) + D(c_2, x_{1,2,\dots,n}) + D(c_3, x_{1,2,\dots,n}) = 2 + 8,8425 + 6,1815 \\ &= 17,0240 \end{aligned}$$

Perubahan terhadap $f(x)$ (fungsi obyektif), yaitu $f'(x)$ lama - $f(x)$ baru.

$$\text{Perubahan } f(x) = f(x)\text{lama} - f(x)\text{baru} = 1000 - 17,0240 = 982,9760$$

Sehingga diperoleh perubahan $f(x) \geq$ Ambang Batas $\rightarrow 982,9760 \geq 0,1$ atau masih ada data yang berpindah cluster, maka iterasi dilanjutkan.

Data Ke-	Cluster						Jarak ke Centroid		
	1		2		3		1	2	3
	X1	X2	X1	X2	X1	X2			
1	0	0	1	1	0	0	0,0000	2,4739	0,0000
2	4	1	0	0	0	0	1,0000	0,0000	0,0000
3	6	1	0	0	0	0	1,0000	0,0000	0,0000
4	0	0	1	2	0	0	0,0000	1,5232	0,0000
5	0	0	2	3	0	0	0,0000	0,5657	0,0000
6	0	0	0	0	5	3	0,0000	0,0000	2,6874
7	0	0	2	5	0	0	0,0000	1,6492	0,0000
8	0	0	0	0	3	5	0,0000	0,0000	0,7454
9	0	0	2	6	0	0	0,0000	2,6306	0,0000
10	0	0	0	0	3	8	0,0000	0,0000	2,7487
Jumlah	10	2	8	17	11	16	2,0000	8,8425	6,1815
Centroid Baru	5,0000	1,0000	1,6000	3,4000	3,6667	5,3333			
f(x) Baru	Jumlah seluruh jarak semua centroid						17,0240		
Perubahan f(x)	f(x) Lama - f(x) Baru						982,9760		
Keterangan	Perubahan f(x) > Ambang Batas dan data masih berpindah cluster, maka dilanjutkan								

Lengkapnya, keseluruhan proses ditunjukkan pada tabel-tabel berikut ini.

Iterasi ke-2:

Data Ke-	X1	X2	Jarak ke Centroid			a (Cluster yang diikuti)		
			1	2	3	1	2	3
1	1	1	4,0000	2,4739	5,0881	0	1	0
2	4	1	1,0000	3,3941	4,3461	1	0	0
3	6	1	1,0000	5,0120	4,9216	1	0	0
4	1	2	4,1231	1,5232	4,2687	0	1	0
5	2	3	3,6056	0,5657	2,8674	0	1	0
6	5	3	2,0000	3,4234	2,6874	1	0	0
7	2	5	5,0000	1,6492	1,6997	0	1	0
8	3	5	4,4721	2,1260	0,7454	0	0	1
9	2	6	5,8310	2,6306	1,7951	0	0	1
10	3	8	7,2801	4,8083	2,7487	0	0	1
			Jumlah			3	4	3

Data Ke-	Cluster						Jarak ke Centroid				
	1		2		3		1	2	3		
	X1	X2	X1	X2	X1	X2					
1	0	0	1	1	0	0	0,0000	1,8200	0,0000		
2	4	1	0	0	0	0	1,2019	0,0000	0,0000		
3	6	1	0	0	0	0	1,2019	0,0000	0,0000		
4	0	0	1	2	0	0	0,0000	0,9014	0,0000		
5	0	0	2	3	0	0	0,0000	0,5590	0,0000		
6	5	3	0	0	0	0	1,3333	0,0000	0,0000		
7	0	0	2	5	0	0	0,0000	2,3049	0,0000		
8	0	0	0	0	3	5	0,0000	0,0000	1,3744		
9	0	0	0	0	2	6	0,0000	0,0000	0,7454		
10	0	0	0	0	3	8	0,0000	0,0000	1,6997		
Jumlah	15	5	6	11	8	19	3,7370	5,5853	3,8194		
Centroid Baru	5,0000	1,6667	1,5000	2,7500	2,6667	6,3333					
f(x) Baru	Jumlah seluruh jarak semua centroid						13,1418				
Perubahan f(x)	f(x) Lama - f(x) Baru						3,8823				
Keterangan	Perubahan f(x) > Ambang Batas dan data masih berpindah cluster, maka dilanjutkan										

Iterasi ke-3:

Data Ke-	X1	X2	Jarak ke Centroid			a (Cluster yang diikuti)		
			1	2	3	1	2	3
1	1	1	4,0552	1,8200	5,5877	0	1	0
2	4	1	1,2019	3,0516	5,4975	1	0	0
3	6	1	1,2019	4,8283	6,2893	1	0	0
4	1	2	4,0139	0,9014	4,6428	0	1	0
5	2	3	3,2830	0,5590	3,3993	0	1	0
6	5	3	1,3333	3,5089	4,0689	1	0	0
7	2	5	4,4845	2,3049	1,4907	0	0	1
8	3	5	3,8873	2,7042	1,3744	0	0	1
9	2	6	5,2705	3,2882	0,7454	0	0	1
10	3	8	6,6416	5,4601	1,6997	0	0	1
Jumlah						3	3	4

Data Ke-	Cluster						Jarak ke Centroid		
	1		2		3		1	2	3
	X1	X2	X1	X2	X1	X2			
1	0	0	1	1	0	0	0,0000	1,0541	0,0000
2	4	1	0	0	0	0	1,2019	0,0000	0,0000
3	6	1	0	0	0	0	1,2019	0,0000	0,0000
4	0	0	1	2	0	0	0,0000	0,3333	0,0000
5	0	0	2	3	0	0	0,0000	1,2019	0,0000
6	5	3	0	0	0	0	1,3333	0,0000	0,0000
7	0	0	0	0	2	5	0,0000	0,0000	1,1180
8	0	0	0	0	3	5	0,0000	0,0000	1,1180
9	0	0	0	0	2	6	0,0000	0,0000	0,5000
10	0	0	0	0	3	8	0,0000	0,0000	2,0616
Jumlah	15	5	4	6	10	24	3,7370	2,5893	4,7976
Centroid Baru	5,0000	1,6667	1,3333	2,0000	2,5000	6,0000			
f(x) Baru	Jumlah seluruh jarak semua centroid						11,1239		
Perubahan f(x)	f(x) Lama - f(x) Baru						2,0178		
Keterangan	Perubahan f(x) > Ambang Batas dan data masih berpindah cluster, maka dilanjutkan								

Iterasi ke-4:

Data Ke-	X1	X2	Jarak ke Centroid			a (Cluster yang diikuti)		
			1	2	3	1	2	3
1	1	1	4,0552	1,0541	5,2202	0	1	0
2	4	1	1,2019	2,8480	5,2202	1	0	0
3	6	1	1,2019	4,7726	6,1033	1	0	0
4	1	2	4,0139	0,3333	4,2720	0	1	0
5	2	3	3,2830	1,2019	3,0414	0	1	0
6	5	3	1,3333	3,8006	3,9051	1	0	0
7	2	5	4,4845	3,0732	1,1180	0	0	1
8	3	5	3,8873	3,4319	1,1180	0	0	1
9	2	6	5,2705	4,0552	0,5000	0	0	1
10	3	8	6,6416	6,2272	2,0616	0	0	1
Jumlah						3	3	4

Data Ke-	Cluster						Jarak ke Centroid				
	1		2		3		1	2	3		
	X1	X2	X1	X2	X1	X2					
1	0	0	1	1	0	0	0,0000	1,0541	0,0000		
2	4	1	0	0	0	0	1,2019	0,0000	0,0000		
3	6	1	0	0	0	0	1,2019	0,0000	0,0000		
4	0	0	1	2	0	0	0,0000	0,3333	0,0000		
5	0	0	2	3	0	0	0,0000	1,2019	0,0000		
6	5	3	0	0	0	0	1,3333	0,0000	0,0000		
7	0	0	0	0	2	5	0,0000	0,0000	1,1180		
8	0	0	0	0	3	5	0,0000	0,0000	1,1180		
9	0	0	0	0	2	6	0,0000	0,0000	0,5000		
10	0	0	0	0	3	8	0,0000	0,0000	2,0616		
Jumlah	15	5	4	6	10	24	3,7370	2,5893	4,7976		
Centroid Baru	5,0000	1,6667	1,3333	2,0000	2,5000	6,0000					
f(x) Baru	Jumlah seluruh jarak semua centroid						11,1239				
Perubahan f(x)	f(x) Lama - f(x) Baru						0,0000				
Keterangan	Perubahan f(x) < Ambang Batas atau data masih berpindah cluster, maka selesai										

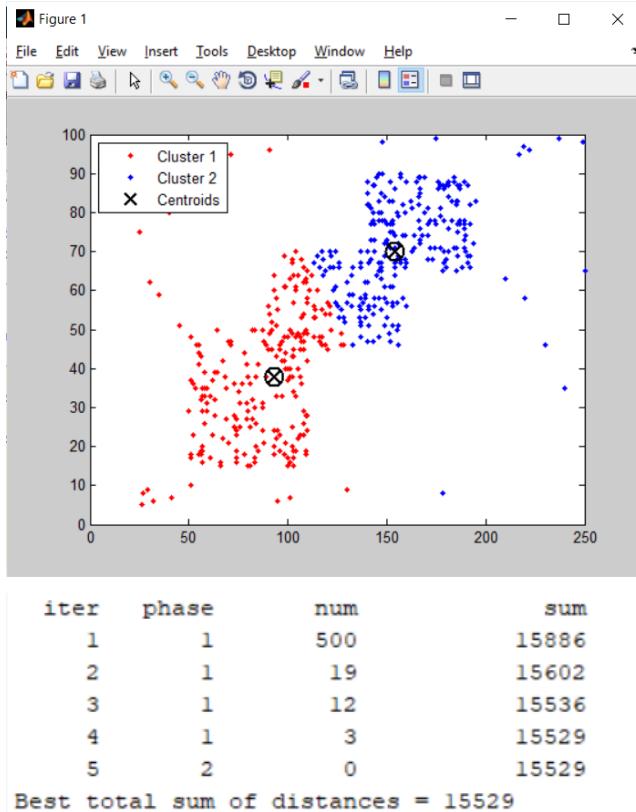
Karena perubahan $f(x) \leq$ Ambang Batas $\rightarrow 0 \leq 0,1$ serta tidak ada lagi data yang berpindah cluster, maka iterasi dihentikan dan proses berakhir. Hasil klasterisasi ditunjukkan pada tabel berikut ini.

Data Ke-	X1	X2	Klaster
1	1	1	2
2	4	1	1
3	6	1	1
4	1	2	2
5	2	3	2
6	5	3	1
7	2	5	3
8	3	5	3
9	2	6	3
10	3	8	3

Contoh 7.2 K-Means: Klasterisasi (Matlab)

Dataset : dsTinggiBeratBadan – Column 2, 3 (terlampir)
 k : 2
Distance Measure : Cityblock

```
clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing Methods\Book2\dataset\datasets.xlsx','dsTinggiBeratBadan');
dataset = dsHeartDisease(:,2:3); % tinggi dan berat badan
%% k-Means Clustering
k=2;
[idx,center] =
kmeans(dataset,k,'Distance','cityblock','Display','iter');
%% Grafik Cluster
plot(dataset(idx==1,1),dataset(idx==1,2),'r.','MarkerSize',12)
hold on
plot(dataset(idx==2,1),dataset(idx==2,2),'b.','MarkerSize',12)
plot(center(:,1),center(:,2),'kx',...
'MarkerSize',12,'LineWidth',2)
plot(center(:,1),center(:,2),'ko',...
'MarkerSize',12,'LineWidth',2)
legend('Cluster 1','Cluster 2','Centroids',...
'Location','NW')
```



7.2 Fuzzy C-Means

Fuzzy C-Means (FCM) sebenarnya merupakan pengembangan dari *K-Means* (versi *Fuzzy* dari *K-Means* dengan beberapa perubahan). *Membership Function* untuk menentukan derajat keanggotaan suatu variabel pada *Fuzzy* diadopsi, sehingga nilai keanggotaan suatu data tidak diberi nilai secara tegas lagi (seperti 1 adalah anggota dan 0 adalah bukan anggota), melainkan nilai keanggotaan suatu data berdasarkan derajat keanggotaan dalam interval 0 hingga 1.

Data dalam suatu *dataset* $(x_i, i = 1, 2, \dots, m)$, yang mana m adalah banyaknya data memiliki fitur/atribut n dimensi, $x_{i1}, x_{i2}, \dots, x_{in}$. Data tersebut dapat dikelompokkan dalam sejumlah *cluster* $(c_i, i = 1, 2, \dots, k)$, yang mana k adalah banyaknya *cluster*. Setiap data (x_i) memiliki derajat keanggotaan pada setiap *cluster* (c_i) yang dinyatakan dengan u_{ij} , bernilai 0 hingga 1, yang mana i adalah data (x_i) dan j adalah *cluster* (c_i) . Nilai derajat keanggotaan setiap data (x_i) selalu sama dengan 1, yang didefinisikan sebagai berikut.

$$\sum_{j=1}^k u_{ij} = 1 \quad (131)$$

Setiap *cluster* (c_i) memiliki paling sedikit satu data dengan nilai keanggotaan $\neq 0$, tapi tidak pula 1 pada semua data, didefinisikan sebagai berikut.

$$0 < \sum_{i=1}^m u_{ij} < m \quad (132)$$

Nilai derajat keanggotaan data (x_i) pada *cluster* (c_i) didefinisikan sebagai berikut.

$$u_{ij} = \frac{d(x_i, c_i)^{\frac{2}{w-1}}}{\sum_{l=1}^k d(x_i, c_l)^{\frac{2}{w-1}}} \quad (133)$$

Notasi $d(x_i, c_i)$ adalah jarak antara data (x_i) ke *centroid/cluster* (c_i). w adalah parameter bobot pangkat (*weighting exponent*) yang bernilai > 1 , umumnya = 2.

Perhitungan *centroid* (c_i) pada fitur ke- j dapat didefinisikan sebagai berikut.

$$c_{ij} = \frac{\sum_{l=1}^m (u_{il})^w x_{lj}}{\sum_{l=1}^m (u_{il})^w} \quad (134)$$

Notasi m adalah banyaknya data, w adalah bobot pangkat, dan u_{ij} adalah nilai derajat keanggotaan data (x_i) ke *cluster* (c_i).

Fungsi obyektif yang digunakan didefinisikan sebagai berikut.

$$f(x) = \sum_{i=1}^m \sum_{j=1}^k (u_{ij})^w d(x_i, c_j)^2 \quad (135)$$

Algoritma FCM mirip dengan algoritma *K-Means* karena memang merupakan versi *Fuzzy* dari *K-Means*, yaitu sebagai berikut:

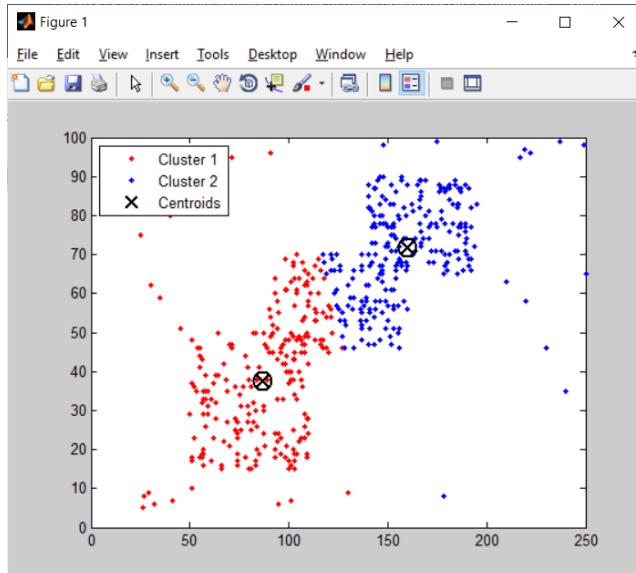
1. Inisialisasi K (jumlah *cluster*), $f(x)$ (fungsi obyektif) awal, *centroid* awal (K data sebagai *centroid* awal), dan tetapkan ambang batas (*threshold*), pembobot (w), dan maksimum iterasi;
2. Hitung nilai *centroid* tiap-tiap *cluster* menggunakan Persamaan (134);
3. Hitung nilai derajat keanggotaan setiap data ke setiap *cluster*.
4. Hitung $f(x)$ (fungsi obyektif) menggunakan Persamaan (135), yang mana metode pengukuran jarak yang digunakan FCM adalah *Euclidean*.
5. Perbaiki derajat keanggotaan setiap data pada setiap *cluster* dengan menggunakan Persamaan (133);
6. Ulangi langkah 2 dan 5 hingga kondisi konvergen tercapai, yaitu apabila nilai derajat keanggotaan \leq ambang batas, atau perubahan $f(x)$ (fungsi obyektif) \leq ambang batas, atau apabila perubahan *centroid* \leq ambang batas, atau telah mencapai iterasi maksimum yang ditentukan di awal.

Contoh 7.3 FCM: Klasterisasi (Matlab)

<i>Dataset</i>	: <i>dsTinggiBeratBadan – Column 2, 3</i> (terlampir)
<i>k</i>	: 2
<i>w</i>	: 2 (<i>default</i>)
<i>Max Iteration</i>	: 100 (<i>default</i>)
<i>Threshold</i>	: 0,0001

```

clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsTinggiBeratBadan');
dataset = dsHeartDisease(:,2:3); % tinggi dan berat badan
%% Fuzzy C-Means Clustering
k=2;
w=2;
iterasi=100;
t=0.0001;
info=1;
[center,U,obj_fcn] = fcm(dataset,k,[w,iterasi,t,info]);
maxU = max(U); %nilai keanggotaan terbesar (cluster yang diikuti)
index1 = find(U(1,:) == maxU); % indeks cluster 1
index2 = find(U(2, :) == maxU); % indeks cluster 2
%% Grafik Cluster
plot(dataset(index1,1),dataset(index1,2),'r.','MarkerSize',12)
hold on
plot(dataset(index2,1),dataset(index2,2),'b.','MarkerSize',12)
plot(center(:,1),center(:,2), 'kx',...
    'MarkerSize',12,'LineWidth',2)
plot(center(:,1),center(:,2), 'ko',...
    'MarkerSize',12,'LineWidth',2)
legend('Cluster 1','Cluster 2','Centroids',...
    'Location','NW')
    
```



Iteration count = 1, obj. fcn = 714361.433443
 Iteration count = 2, obj. fcn = 587612.360652
 Iteration count = 3, obj. fcn = 586847.115372
 Iteration count = 4, obj. fcn = 578952.804357
 Iteration count = 5, obj. fcn = 521386.027506
 Iteration count = 6, obj. fcn = 382659.633807
 Iteration count = 7, obj. fcn = 325960.035248
 Iteration count = 8, obj. fcn = 321246.861751
 Iteration count = 9, obj. fcn = 321050.812442

*Iteration count = 10, obj. fcn = 321042.880359
Iteration count = 11, obj. fcn = 321042.284216
Iteration count = 12, obj. fcn = 321042.181793
Iteration count = 13, obj. fcn = 321042.158157
Iteration count = 14, obj. fcn = 321042.152428
Iteration count = 15, obj. fcn = 321042.151030
Iteration count = 16, obj. fcn = 321042.150689
Iteration count = 17, obj. fcn = 321042.150605*

7.3 Soal Latihan K-Means & FCM

1. Kumpulkan 30 data tinggi dan berat badan rekan-rekan masasiswa anda, kemudian lakukan analisis klasterisasi secara manual menggunakan *K-Means*.
2. Unduh salah satu *dataset* klasterisasi pada *UCI Machine Learning Repository* kemudian lakukan analisis klasterisasi menggunakan algoritma *K-Means* dan FCM. Gunakan alat bantu Matlab, Rapidminer, atau yang anda kuasai.
3. Gunakan salah satu metode evaluasi internal dan eksternal untuk menentukan kinerja dari *K-Means* dan FCM, mana yang lebih baik untuk dataset yang anda gunakan?

8. Naïve Bayes, k-NN, & Fuzzy

No.	Materi	Tujuan Pembelajaran
1.	Naïve Bayes	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>Naïve Bayes</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
2.	Gaussian Naïve Bayes	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>Gaussian Naïve Bayes</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
3.	Absolute Correlation Weighted Naïve Bayes	Anda mampu memahami, menjelaskan, dan menerapkan algortima AC W-NB secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
4.	k-Nearest Neighbor	Anda mampu memahami, menjelaskan, dan menerapkan algortima k-NN secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
5.	Weighted k-NN	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>Weighted k-NN</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
6.	Fuzzy k-NN	Anda mampu memahami, menjelaskan, dan menerapkan algortima FkNN secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
7.	Fuzzy k-NN in every class	Anda mampu memahami, menjelaskan, dan menerapkan algortima FkNNC secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
8.	KFACWNB-NN	Anda mampu memahami dan menjelaskan konsep dasar algortima KFACWNB-NN.

8.1 Naïve Bayes

Algoritma *Naïve Bayes* (NB) merupakan penerapan dari *Bayes Theorem*. Ide dasar dari *Bayes Theorem* adalah bahwa hasil dari hipotesis (H) dapat diperkirakan berdasarkan pada beberapa *evidence* (E) yang diamati. Teorema ini berbasis pendekatan probabilistik yang secara umum didefinisikan sebagai berikut.

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)} \quad (136)$$

P menyatakan probabilitas, H menyatakan hipotesis, dan E menyatakan bukti (*evidence*). $P(H|E)$ adalah probabilitas akhir bersyarat, suatu hipotesis (H) terjadi jika diberikan *evidence* (E) terjadi. $P(E|H)$ adalah probabilitas *evidence* (E) terjadi akan mempengaruhi hipotesis (H). $P(H)$ adalah probabilitas awal (priori) hipotesis (H) terjadi tanpa memandang *evidence* apapun. $P(E)$ adalah probabilitas awal (priori) *evidence* E terjadi tanpa memandang hipotesis/bukti yang lain.

Bayes Theorem dapat menangani banyak *evidence*, misalnya terdapat *evidence* (E_1, E_2, \dots, E_n), sehingga Persamaan (136) di atas dapat didefinisikan menjadi seperti berikut.

$$P(H|E_1, E_2, \dots, E_n) = \frac{P(E_1, E_2, \dots, E_n|H) * P(H)}{P(E_1, E_2, \dots, E_n)} \quad (137)$$

Bayes Theorem menganggap bahwa setiap *evidence* (dalam NB dapat dianggap sebagai fitur) bersifat independen (tidak saling berketergantungan), atau setiap fitur tidak berelasi dengan fitur lainnya, atau setiap fitur sama pentingnya, atau setiap fitur bernilai 0. Sebenarnya hal ini merupakan salah satu kelemahan NB karena kenyataannya pada banyak kasus, asumsi tersebut tidak selalu tepat. Kerena asumsi tersebut, maka Persamaan (137) di atas dapat diubah menjadi seperti berikut.

$$P(H|E_1, E_2, \dots, E_n) = \frac{P(E_1|H) * P(E_2|H) * \dots * P(E_n|H) * P(H)}{P(E_1) * P(E_2) * \dots * P(E_n)} \quad (138)$$

Jika dikaitkan antara *Bayes Theorem* dengan NB *classifier*, maka hipotesis (H) menyatakan label *class*, sedangkan *evidence* (E) menyatakan fitur/atribut (variabel input), maka probabilitas suatu label *class* (y) berdasarkan/dipengaruhi fitur-fitur ($x_i, i = 1, 2, \dots, m$), yang mana m adalah banyaknya fitur, dapat didefinisikan sebagai berikut.

$$P(y|x_i) = \frac{P(y) \prod_{i=1}^m P(x_i|y)}{P(x_i)} \quad (139)$$

Nilai $P(x_i)$ selalu tetap (konstan), sehingga label *class* hasil keputusan klasifikasi algoritma NB dapat didefinisikan sebagai berikut.

$$y' = \operatorname{argmax}_{y_k} P(y_k) \prod_{i=1}^m P(x_i|y_k) \quad (140)$$

Keterangan:

y' : label *class* hasil keputusan klasifikasi suatu data/*instance*.

$P(y_k)$: Probabilitas label *class* ($y_k, k = 1, 2, \dots, j$), yang mana j adalah banyaknya label *class*.

$P(x_i|y_k)$: Probabilitas atribut ($x_i, i = 1, 2, \dots, m$) pada label *class* (y_k), yang mana m adalah banyaknya atribut.

NB merupakan salah satu algoritma *lazy learning* (melakukan *learning/pelatihan* saat ada data yang akan diprediksi) yang dapat digunakan pada masalah klasifikasi. NB merupakan algoritma yang cukup sederhana, namun algoritma ini masih masuk dalam *top ten* metode-metode *machine learning* justru karena kesederhanaan, efisiensi, dan kinerjanya [47]. Salah satu kelebihan NB adalah kinerjanya yang tidak terlalu berubah ketika ukuran data menjadi besar, sehingga sangat baik pada data yang berukuran besar.

Contoh 8.1 Naïve Bayes: Klasifikasi (Manual)

Data ke-	X1	X2	X3	Y
1.	Pria	Kurus	Balita	-
2.	Wanita	Ideal	Muda	-
3.	Wanita	Gemuk	Dewasa	-
4.	Pria	Gemuk	Tua	-
5.	Pria	Kurus	Dewasa	-
6.	Pria	Ideal	Muda	+
7.	Wanita	Gemuk	Muda	+
8.	Pria	Gemuk	Tua	+
9.	Wanita	Kurus	Tua	+
10.	Wanita	Gemuk	Balita	+
Data uji (prediksi)	Wanita	Gemuk	Tua	?

Jumlah data (m) = 10

Jumlah label class (y_-) = 5

Jumlah label class (y_+) = 5

Jumlah atribut ($x_{1(Wanita)}$) pada label class (y_-) = 2

Jumlah atribut ($x_{1(Wanita)}$) pada label class (y_+) = 3

Jumlah atribut ($x_{2(Gemuk)}$) pada label class (y_-) = 2

Jumlah atribut ($x_{2(Gemuk)}$) pada label class (y_+) = 3

Jumlah atribut ($x_{3(Tua)}$) pada label class (y_-) = 1

Jumlah atribut ($x_{3(Tua)}$) pada label class (y_+) = 2

Probabilitas setiap label *class* ($P(y_k)$):

$$P(y_-) = \frac{5}{10} = 0,5$$

$$P(y_+) = \frac{5}{10} = 0,5$$

Probabilitas setiap atribut pada setiap label *class* ($P(x_i|y_k)$):

$$P(x_{1(Wanita)}|y_-) = \frac{2}{5} = 0,4$$

$$P(x_{1(Wanita)}|y_+) = \frac{3}{5} = 0,6$$

$$P(x_{2(Gemuk)}|y_-) = \frac{2}{5} = 0,4$$

$$P(x_{2(\text{Gemuk})}|y_+) = \frac{3}{5} = 0,6$$

$$P(x_{3(\text{Tua})}|y_-) = \frac{1}{5} = 0,2$$

$$P(x_{3(\text{Tua})}|y_+) = \frac{2}{5} = 0,4$$

Probabilitas setiap label *class* berdasarkan setiap atribut ($P(y_k|x_i)$):

$$P(y_-|x_{1(\text{Wanita})}, x_{2(\text{Gemuk})}, x_{3(\text{Tua})}) = 0,5 * 0,4 * 0,4 * 0,2 = 0,016$$

$$P(y_+|x_{1(\text{Wanita})}, x_{2(\text{Gemuk})}, x_{3(\text{Tua})}) = 0,5 * 0,6 * 0,6 * 0,4 = 0,072$$

Dengan demikian hasil klasifikasi/prediksi data uji adalah:

$$\text{Argmax} (P(y_{-,+}|x_1, x_2, x_3)) = \{0,016; 0,072\} = +$$

8.2 Gaussian Naïve Bayes

Standarnya, NB bekerja pada data kategorikal, namun pendekatan distribusi normal (*Gaussian*) dapat diterapkan pada NB untuk menentukan probabilitas pada atribut numerik. Pendekatan lainnya yang dapat diterapkan pada NB agar dapat bekerja pada data numerik adalah *Kernel*, sehingga dinamakan *Kernel Naive Bayes*. $P(x_i|y_k)$, yaitu probabilitas dari setiap atribut numerik (x_i) pada setiap label *class* (y_k) menggunakan distribusi *Gaussian* dapat didefinisikan sebagai berikut.

$$P(x_i|y_k) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (141)$$

Mean (μ) diperoleh melalui Persamaan (11) dan *standard deviation* (σ) diperoleh melalui persamaan (12) yang telah dijelaskan sebelumnya. $P(x_i|y_k)$ dari distribusi *Gaussian* kemudian dapat diterapkan pada Persamaan NB (140), yang mana penentuan $P(y_k)$ sama seperti standar NB.

Contoh 8.2 Gaussian Naïve Bayes: Klasifikasi (Manual)

No.	X1	X2	X3	X4	X5	X6	Y
1.	113	9,9	3,1	2	5,9	Rendah	A
2.	127	12,9	2,4	1,4	0,6	Sedang	B
3.	109	5,3	1,6	1,4	1,5	Tinggi	A
4.	105	7,3	1,5	1,5	-0,1	Rendah	B
5.	105	6,1	2,1	1,4	7	Sedang	B
6.	110	10,4	1,6	1,6	2,7	Tinggi	A
7.	114	9,9	2,4	1,5	5,7	Rendah	C
8.	106	9,4	2,2	1,5	0	Tinggi	B
9.	107	13	1,1	0,9	3,1	Tinggi	C
10.	106	4,2	1,2	1,6	1,4	Tinggi	C
Data uji (prediksi)	107	10,1	2,2	0,9	2,7	Tinggi	?

Probabilitas setiap label *class* adalah:

$$P(y_A) = \frac{3}{10} = 0,3$$

$$P(y_B) = \frac{4}{10} = 0,4$$

$$P(y_C) = \frac{3}{10} = 0,3$$

Nilai μ dan σ atribut (x_1) pada label *class* (y_A) adalah:

$$\mu(x_1|y_A) = \frac{1}{3} 113 + 109 + 110 = \frac{332}{3} = 110,67$$

$$\begin{aligned} \sigma(x_1|y_A) &= \left(\frac{1}{3-1} ((113 - 110,67)^2 + (109 - 110,67)^2 \right. \\ &\quad \left. + (110 - 110,67)^2) \right)^{1/2} = \left(\frac{1}{3-1} 8,67 \right)^{1/2} = 2,08 \end{aligned}$$

Lengkapnya, μ dan σ setiap atribut pada setiap label *class* ditunjukkan pada tabel berikut ini.

	μ			σ		
	y_A	y_B	y_C	y_A	y_B	y_C
x_1	110,67	110,75	109,00	2,08	10,84	4,36
x_2	8,53	8,93	9,03	2,81	2,98	4,46
x_3	2,10	2,05	1,57	0,87	0,39	0,72
x_4	1,67	1,45	1,33	0,31	0,06	0,38
x_5	3,37	1,88	3,40	2,27	3,43	2,17
x_6	No	No	No	No	No	No

Probabilitas atribut (x_1) pada label *class* (y_A) menggunakan Persamaan (141) yang mana $\exp(1) = 2,7183$ adalah:

$$\begin{aligned} P(x_{1(107)}|y_A) &= \frac{1}{2,08\sqrt{2 * 3,14}} 2,7183^{-\frac{(107-110,67)^2}{2 * 2,08^2}} \\ &= \frac{1}{2,08 * 2,5060} 2,7183^{-1,5513} = \frac{1}{5,2166} 0,2120 = 0,0406 \end{aligned}$$

Probabilitas atribut (x_6) pada label *class* (y_A) adalah:

$$P(x_{6(\text{Tinggi})}|y_A) = \frac{2}{3} = 0,6667$$

Lengkapnya, probabilitas setiap atribut pada setiap label *class* ditunjukkan pada tabel berikut ini.

	y_A	y_B	y_C
x_1	0,0406	0,0347	0,0824
x_2	0,1215	0,1239	0,0869
x_3	0,4577	0,9559	0,3760
x_4	0,0560	1,3598E-19	0,5475
x_5	0,1681	0,1130	0,1749
x_6	0,6667	0,2500	0,6667

Sehingga probabilitas setiap label *class* berdasarkan setiap atribut adalah:

$$\begin{aligned} P(y_A | x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}, x_{6(\text{Tinggi})}) \\ = 0,3 * 0,0406 * 0,1215 * 0,4577 * 0,0560 * 0,1681 * 0,6667 \\ = 0,0000042579 \end{aligned}$$

$$\begin{aligned} P(y_B | x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}, x_{6(\text{Tinggi})}) \\ = 0,4 * 0,0347 * 0,1239 * 0,9559 * 1,3598E - 19 * 0,1130 \\ * 0,2500 = 6,3072E - 24 \end{aligned}$$

$$\begin{aligned} P(y_C | x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}, x_{6(\text{Tinggi})}) \\ = 0,3 * 0,0824 * 0,0869 * 0,3760 * 0,5475 * 0,1749 * 0,6667 \\ = 0,0000515501 \end{aligned}$$

Dengan demikian, hasil klasifikasi/prediksi data uji adalah:

$$\begin{aligned} \text{Argmax} (P(y_{A,B,C} | x_1, x_2, x_3, x_4, x_5, x_6)) \\ = \{0,0000042579; 6,3072E - 24; 0,0000515501\} = C \end{aligned}$$

Contoh 8.3 Kernel Naïve Bayes: Klasifikasi (Matlab)

Dataset : *dsHeartDiseaseCleveland – Class = {0,1,2,3,4}* (terlampir)
Distribution : *Kernel*

Fungsi “NBaa”:

```
function [output, lamaProses, akurasi] = NBaa(dataLatihInput,
dataLatihOutput, dataUjiInput, dataUjiOutput)
    tic;
    %ganti 'kernel' menjadi 'normal' jika menggunakan Gaussian
    NBmodel = NaiveBayes.fit(dataLatihInput, dataLatihOutput,
'Distribution','kernel');
    output = predict(NBmodel, dataUjiInput); %NB Testing
    conMat = confusionmat(dataUjiOutput, output); %Confusion Matrix
    jmlData = sum(conMat(:));
    hasilBenar = sum(diag(conMat));
    akurasi = 100 * (hasilBenar / jmlData);
    lamaProses = toc;
    plot(dataUjiInput,dataUjiOutput,'o',dataUjiInput,output','+')
end
```

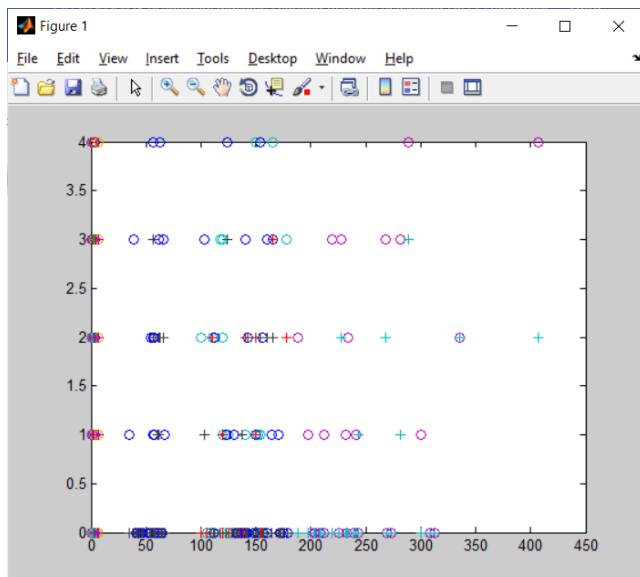
Script “NB”:

```
clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dataset = dsHeartDisease(:,2:15); % ID dan Output 0-1 tidak digunakan
%% Variabel untuk hasil NB
NBhasilSub = []; %kolom: 1 lama proses, 2 akurasi
NBoutputAll.x = []; %Output NB di setiap K dr K-Fold
NBhasil = []; %kolom: 1 lama proses, 2 acc, 3 acc max, 4 acc min
%% K-Fold Cross Validation
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K); % kolom 14 = class
for i = 1:K
    %% Buat Data Latih dan Data Uji berdasarkan indeks dari K-Fold
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
```

```

subDataLatihInput = dataset(latih,1:13); %input training
subDataLatihOutput = dataset(latih,14); %output training (hasil
encoding)
subDataUjiInput = dataset(uji,1:13); % input testing
subDataUjiOutput = dataset(uji,14); % output testing
%% NB Modelling di tiap K
[NBsubOutput, NBsubLamaProses, NBsubAkurasi] =
NBaa(subDataLatihInput, subDataLatihOutput, subDataUjiInput,
subDataUjiOutput);
NBhasilSub(i,1) = NBsubLamaProses; % hasil lama proses dalam K
NBhasilSub(i,2) = NBsubAkurasi; % hasil akurasi dalam K
NOutputAll(i).x = [subDataUjiInput subDataUjiOutput
NBsubOutput];
end
%% Hasil Akhir NB
NBhasil(1)=mean(NBhasilSub(:,1)); % rata2 lama proses
NBhasil(2)=mean(NBhasilSub(:,2)); % akurasi akhir
NBhasil(3)=max(NBhasilSub(:,2)); % akurasi max
NBhasil(4)=min(NBhasilSub(:,2)); % akurasi min

```



$$\text{Lama proses} = 0,0841 \text{ detik}$$

$$\text{Akurasi minimum} = 43,3333\%$$

$$\text{Akurasi maksimum} = 60,0000\%$$

$$\text{Akurasi} = 54,4451\%$$

8.3 Absolute Correlation Weighted Naïve Bayes

Seperti yang telah dijelaskan sebelumnya bahwa NB menganggap bahwa setiap atribut bersifat independen (tidak saling berketergantungan), atau setiap atribut tidak berelasi dengan atribut lainnya, atau setiap atribut sama pentingnya, atau setiap atribut bernilai 0. Namun kenyataannya pada banyak kasus, asumsi tersebut tidak selalu tepat, sehingga hal ini merupakan salah satu kelemahan NB. Untuk mengatasi masalah ini, pendekatan *attribute weighting* atau *feature selection* dapat diterapkan.

Salah satu metode *attribute weighting* yang telah dijelaskan sebelumnya adalah *Absolute Correlation Coefficient* (ACC) yang didefinisikan pada Persamaan (25). Metode ini bekerja pada data numerik dan dapat menentukan kekuatan hubungan/korelasi antar atribut, sehingga dapat diterapkan pada *Gaussian Naïve Bayes*. Pengembangan algoritma NB ini dinamakan *Absolute Correlation Weighted Naïve Bayes* (AC W-NB) [28].

Pembobotan atribut untuk NB disebut *Weighted Naive Bayes* (WNB), sehingga Persamaan (140) dapat diubah menjadi persamaan berikut ini.

$$y' = \underset{y_k}{\operatorname{argmax}} P(y_k) \prod_{i=1}^m P(x_i|y_k)^{w_i} \quad (142)$$

Dengan demikian notasi w_i (*weight*) pada Persamaan (142) di atas dapat menggunakan metode ACC (25), sehingga menjadi AC W-NB. Secara rinci, algoritma AC W-NB adalah sebagai berikut:

1. Hitung $P(x_i|y_k)$, yaitu probabilitas setiap atribut (x_i) pada setiap label *class* (y_k) menggunakan distribusi *Gaussian* yang didefinisikan pada Persamaan (141).
2. Hitung L_k , yaitu *weight likelihood* setiap label *class* menggunakan Persamaan (143) berikut ini, yang mana w_i diperoleh menggunakan metode ACC yang didefinisikan pada Persamaan (25).

$$L_k = \prod_{i=1}^m P(x_i|y_k)^{w_i} \quad (143)$$

3. Hitung $P(y_k)$, yaitu probabilitas setiap label *class* (y_k) menggunakan Persamaan (144) berikut ini, yang mana L_k menyatakan *weight likelihood* label *class* ke- k , sedangkan $L_{\bar{k}}$ menyatakan *weight likelihood* label *class* lainnya. $P(y_k)$ inilah yang mengganti $P(y_k)$ standar NB.

$$P(y_k) = \frac{L_k}{L_k + L_{\bar{k}}} \quad (144)$$

4. Akhirnya $P(x_i|y_k)$ yang diperoleh dari distribusi *Gaussian* (proses/langkah 1) dan $P(y_k)$ yang diperoleh dari proses/langkah 3 dapat diterapkan pada Persamaan NB (140).

Contoh 8.4 AC W-NB: Klasifikasi (Manual)

No.	X1	X2	X3	X4	X5	Y
1.	113	9,9	3,1	2	5,9	1
2.	127	12,9	2,4	1,4	0,6	2
3.	109	5,3	1,6	1,4	1,5	1
4.	105	7,3	1,5	1,5	-0,1	2
5.	105	6,1	2,1	1,4	7	2
6.	110	10,4	1,6	1,6	2,7	1
7.	114	9,9	2,4	1,5	5,7	3
8.	106	9,4	2,2	1,5	0	2
9.	107	13	1,1	0,9	3,1	3
10.	106	4,2	1,2	1,6	1,4	3
Data uji (prediksi)	107	10,1	2,2	0,9	2,7	?

Langkah 1, hitung $P(x_i|y_k)$, yaitu probabilitas setiap atribut (x_i) pada setiap label *class* (y_k) menggunakan distribusi *Gaussian* (141).

Nilai μ dan σ atribut (x_i) pada label *class* (y_1) adalah:

$$\mu(x_1|y_1) = \frac{1}{3} 113 + 109 + 110 = \frac{332}{3} = 110,67$$

$$\begin{aligned} \sigma(x_1|y_1) &= \left(\frac{1}{3-1} ((113 - 110,67)^2 + (109 - 110,67)^2 \right. \\ &\quad \left. + (110 - 110,67)^2) \right)^{1/2} = \left(\frac{1}{3-1} 8,67 \right)^{1/2} = 2,08 \end{aligned}$$

Lengkapnya, μ dan σ setiap atribut pada setiap label *class* ditunjukkan pada tabel berikut ini.

	μ			σ		
	y_1	y_2	y_3	y_1	y_2	y_3
x_1	110,67	110,75	109,00	2,08	10,84	4,36
x_2	8,53	8,93	9,03	2,81	2,98	4,46
x_3	2,10	2,05	1,57	0,87	0,39	0,72
x_4	1,67	1,45	1,33	0,31	0,06	0,38
x_5	3,37	1,88	3,40	2,27	3,43	2,17

Probabilitas atribut (x_i) pada label *class* (y_1) menggunakan Persamaan (141) yang mana $\exp(1) = 2,7183$ adalah:

$$\begin{aligned} P(x_{1(107)}|y_1) &= \frac{1}{2,08\sqrt{2 * 3,14}} 2,7183^{-\frac{(107-110,67)^2}{2 * 2,08^2}} \\ &= \frac{1}{2,08 * 2,5060} 2,7183^{-1,5513} = \frac{1}{5,2166} 0,2120 = 0,0406 \end{aligned}$$

Lengkapnya, probabilitas setiap atribut pada setiap label *class* ($P(x_i|y_k)$) ditunjukkan pada tabel berikut ini.

	y_1	y_2	y_3
x_1	0,0406	0,0347	0,0824
x_2	0,1215	0,1239	0,0869
x_3	0,4577	0,9559	0,3760
x_4	0,0560	1,3598E-19	0,5475
x_5	0,1681	0,1130	0,1749

Langkah 2, hitung *weight likelihood* setiap label *class* (L_k) menggunakan Persamaan (143), yang mana w_i diperoleh menggunakan metode ACC yang didefinisikan pada Persamaan (25).

Weight atribut x_I (w_I) adalah sebagai berikut:

$$w_1 = \frac{\mu_{x_1,y_1} - \mu_{x_1,y_2} - \mu_{x_1,y_3}}{\sigma_{x_1,y_1} - \sigma_{x_1,y_2} - \sigma_{x_1,y_3}} = \frac{110,67 - 110,75 - 109,00}{2,08 - 10,84 - 4,36} = -6,31$$

Weight likelihood atribut (x_I) pada label *class* (y_I) adalah sebagai berikut:

$$P(x_1|y_1)^{w_1} = P(x_1|y_1) * w_1 = 0,0406 * (-6,31) = -0,26$$

Lengkapnya, *weight* setiap atribut (w_i) dan *weight likelihood* setiap atribut (x_i) pada setiap label *class* (y_k) ditunjukkan pada tabel berikut ini.

	w	y_1	y_2	y_3
x_1	-6,31	-0,26	-0,22	-0,52
x_2	-0,92	-0,11	-0,11	-0,08
x_3	-0,77	-0,35	-0,73	-0,29
x_4	-1,51	-0,08	-2,05E-19	-0,82
x_5	-0,24	-0,04	-0,03	-0,04

Sehingga *weight likelihood* setiap label *class* (L_k) adalah sebagai berikut:

$$L_1 = (-0,26) * (-0,11) * (-0,35) * (-0,08) * (-0,04) = -3,46E - 05$$

$$L_2 = (-0,22) * (-0,11) * (-0,73) * (-2,05E - 19) * (-0,03) = -1,02E - 22$$

$$L_3 = (-0,52) * (-0,08) * (-0,29) * (-0,82) * (-0,04) = -4,19E - 04$$

Langkah 3, hitung probabilitas setiap label *class* ($P(y_k)$) menggunakan Persamaan (144).

$$P(y_1) = \frac{-3,46E - 05}{(-3,46E - 05) + (-1,02E - 22) + (-4,19E - 04)} = 0,0763$$

$$P(y_2) = \frac{-1,02E - 22}{(-3,46E - 05) + (-1,02E - 22) + (-4,19E - 04)} = 2,2604E - 19$$

$$P(y_3) = \frac{-4,19E - 04}{(-3,46E - 05) + (-1,02E - 22) + (-4,19E - 04)} = 0,9237$$

Langkah 4, hitung probabilitas setiap label *class* (y_k) berdasarkan setiap atribut (x_i), kemudian tentukan hasil keputusan klasifikasi.

$$\begin{aligned} P(y_1|x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}) \\ = 0,0763 * 0,0406 * 0,1215 * 0,4577 * 0,0560 * 0,1681 \\ = 0,0000016 \end{aligned}$$

$$\begin{aligned} P(y_2|x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}) \\ = (2,2604E - 19) * 0,0347 * 0,1239 * 0,9559 \\ * (1,3598E - 19) * 0,1130 = 1,4257174E - 41 \end{aligned}$$

$$\begin{aligned}
 P(y_3|x_{1(107)}, x_{2(10,1)}, x_{3(2,2)}, x_{4(0,9)}, x_{5(2,7)}) \\
 = 0,9237 * 0,0824 * 0,0869 * 0,3760 * 0,5475 * 0,1749 \\
 = 0,0002381
 \end{aligned}$$

Dengan demikian, hasil klasifikasi/prediksi data uji adalah:

$$\begin{aligned}
 \text{Argmax} (P(y_{1,2,3}|x_1, x_2, x_3, x_4, x_5)) \\
 = \{0,0000016; 1,4257174E - 41; 0,0002381\} = 3
 \end{aligned}$$

8.4 k-Nearest Neighbor

Algoritma *k-Nearest Neighbor* (*k*-NN) merupakan salah satu algoritma *lazy learning* untuk klasifikasi data yang berkerja secara lokal pada data numerik menggunakan pendekatan ukuran jarak (*dissimilarity*). Salah satu kelebihan K-NN adalah kesederhanaannya sehingga mudah diimplementasikan, namun secara umum mampu memberikan kinerja yang relatif tinggi untuk model klasifikasi data. Walaupun begitu, *k*-NN memiliki beberapa kelemahan, yaitu:

1. Sensitif terhadap ukuran ketetanggan *k*. Masalah ini dapat diatasi melalui optimalisasi nilai *k* untuk penentuan otomatis nilai *k* yang optimal.
2. Sensitif terhadap *outliers (noisy data)* ketika nilai *k* terlalu kecil, namun sensitif terhadap distorsi data ketika nilai *k* terlalu besar. Masalah ini dapat diatasi melalui *distance weighted* untuk estimasi probabilitas label *class*.
3. Sensitif terhadap fitur-fitur yang kurang relevan karena menganggap setiap fitur/atribut bersifat independen. Masalah ini dapat diatasi melalui pendekatan *feature selection* atau *attribute weighting*.
4. Kompleksitas waktu dan memori yang relatif besar karena melakukan *learning* (pelatihan) di setiap kali melakukan prediksi (sama seperti *Naïve Bayes*). Masalah ini dapat diatasi melalui perbaikan terhadap struktur data k-NN.

Oleh karena itu, K-NN termasuk salah satu metode *machine learning* yang masih sering/menarik dimodifikasi/diperbaiki oleh para ahli karena kinerjanya, cara kerjanya yang sederhana, dan kelemahan-kelemahannya.

Data yang akan diprediksi atau data uji dapat dinotasikan dengan $z = (x'_i, y')$, yang mana vektor data yang akan diprediksi ($x'_i, i = 1, 2, \dots, n$) memiliki sejumlah n atribut dan label *class* (y') yang belum diketahui (akan diprediksi). Sedangkan suatu data latih dapat dinotasikan dengan $L = (x_i, y)$, yang mana vektor suatu data latih ($x_i, i = 1, 2, \dots, n$) memiliki sejumlah n atribut dan label *class* (y). Jarak antara data prediksi/uji (z) ke setiap data latih ($L_j, j = 1, 2, \dots, m$), yang mana m menyatakan banyaknya data dapat dinotasikan dengan $d(x'_i, x_j)$, simpan dalam D . k-NN memilih $D_z \in D$ dalam k tetangga dari z , kemudian menghitung jumlah data yang mengikuti label *class* dalam k tetangga, yang mana label *class* dengan data terbanyak yang mengikutinya menjadi hasil prediksi y' dari z yang dapat didefinisikan sebagai berikut.

$$y' = \underset{v}{\text{argmax}} \sum_{D \in D_z}^k I(v = y_c) \quad (145)$$

Notasi v menyatakan jumlah data (*class*) yang masuk dalam *class* (y_c , $c = 1, 2, \dots, p$), yang mana p menyatakan banyaknya *class*. Sedangkan $d(x', x_i)$ merupakan jarak antara data prediksi/uji (z) ke setiap data latih (L_i) yang disimpan dalam D dapat dihitung menggunakan salah satu metode pengukuran jarak (*dissimilarity*), umumnya menggunakan *Euclidean* (128) untuk memperoleh jarak yang lebih dekat antar dua data atau menggunakan *Manhattan* (129) untuk memperoleh jarak yang lebih jauh antara dua data.

Penentuan data latih sebanyak k yang masuk sebagai tetangga terdekat dapat menggunakan salah satu aturan (*vote rule*) *Nearest*, *Random*, dan *Consensus*. *Nearest* menetapkan tetangga terdekat dari jarak terdekat, *Random* dari jarak yang ditentukan secara acak, dan *Consensus* dari jarak berdasarkan aturan kesepakatan. Metode vote rule yang umum (*default*) digunakan adalah *Nearest*. Selanjutnya pada proses klasifikasi/prediksi, k-NN menggunakan pendekatan *majority vote*, yaitu keputusan klasifikasi berdasarkan label *class* yang memiliki data tetangga terbanyak.

Contoh 8.5 k-NN: Klasifikasi (Manual)

No.	Atribut		Class y	Euclidean Distance	k=5	Prediction y'	
	x1	x2				+	-
L	1.	2	1	+	$\sqrt{(2-1)^2 + (1-5)^2} = 4,12$	No	
	2.	2	3	+	$\sqrt{(2-1)^2 + (3-5)^2} = 2,24$	Yes	1 0
	3.	2	1	+	$\sqrt{(2-1)^2 + (1-5)^2} = 4,12$	No	
	4.	1	1	+	$\sqrt{(1-1)^2 + (1-5)^2} = 4,00$	Yes	1 0
	5.	2	2	+	$\sqrt{(2-1)^2 + (2-5)^2} = 3,16$	Yes	1 0
	6.	1	7	-	$\sqrt{(1-1)^2 + (7-5)^2} = 2,00$	Yes	0 1
	7.	2	6	-	$\sqrt{(2-1)^2 + (6-5)^2} = 1,41$	Yes	0 1
z	8.	1	5	?		Jumlah	3 2

Dengan demikian prediksi/klasifikasi data uji/prediksi ke-8 adalah:

$$\text{Argmax}(y_c) = \{3, 2\} = +$$

Contoh 8.6 k-NN: Klasifikasi (Matlab)

Dataset : *dsHeartDiseaseCleveland – Class = {0,1,2,3,4}* (terlampir)
 k : 10
 Distance Measure : *Euclidean*
 Vote Rule : *Nearest*
 Data Validation : *10-Fold Cross Validation*
 Evaluation : *Confusion Matrix*

Fungsi “*KNNaam*”

```

function [output, lamaProses, akurasi] = KNNaam(dataLatihInput,
dataLatihOutput, dataUjiInput, dataUjiOutput)
tic;
output = knnclassify(dataUjiInput, dataLatihInput,
dataLatihOutput, 7, 'euclidean', 'nearest'); %KNN model
conMat = confusionmat(dataUjiOutput, output); %evaluasi
jmlData = sum(conMat(:));
hasilBenar = sum(diag(conMat));
akurasi = 100 * (hasilBenar / jmlData);
lamaProses = toc;

```

```

        plot(dataUjiInput,dataUjiOutput,'o',dataUjiInput,output','+')
    end

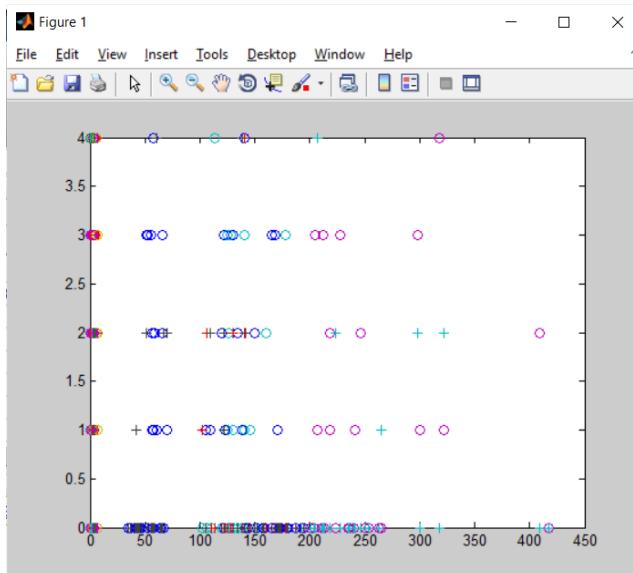
```

Script "KNN"

```

clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dataset = dsHeartDisease(:,2:15); % ID dan Output 0-1 tidak digunakan
%% Variabel untuk hasil KNN
KNNhasilSub = []; %kolom: 1 lama proses, 2 akurasi
KNNoutputAll.x = []; %Output KNN di setiap K dari K-Fold
KNNhasil = []; %kolom: 1 lama proses, 2 acc, 3 acc max, 4 acc min
%% K-Fold Cross Validation
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K);
for i = 1:K
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
    subDataLatihInput = dataset(latih,1:13); %input training
    subDataLatihOutput = dataset(latih,14); %output training
    subDataUjiInput = dataset(uji,1:13); % input testing
    subDataUjiOutput = dataset(uji,14); % output testing
    %% KNN Modelling di tiap K
    [KNNsubOutput, KNNsubLamaProses, KNNsubAkurasi] =
    KNNaa(subDataLatihInput, subDataLatihOutput, subDataUjiInput,
    subDataUjiOutput);
    KNNhasilSub(i,1) = KNNsubLamaProses; % hasil lama proses dalam K
    KNNhasilSub(i,2) = KNNsubAkurasi; % hasil akurasi dalam K
    KNNoutputAll(i).x = [subDataUjiInput subDataUjiOutput
    KNNsubOutput];
end
%% Hasil Akhir KNN
KNNhasil(1)=mean(KNNhasilSub(:,1)); % rata-rata lama proses
KNNhasil(2)=mean(KNNhasilSub(:,2)); % akurasi akhir
KNNhasil(3)=max(KNNhasilSub(:,2)); % akurasi max
KNNhasil(4)=min(KNNhasilSub(:,2)); % akurasi min

```



Lama proses = 0,0701 detik

Akurasi minimum = 37,9310%

Akurasi maksimum = 53,3333%

Akurasi = 46,8228%

8.5 Weighted k-NN

Dalam proses *majority vote* secara tegas pada k-NN, keputusan klasifikasi didasarkan pada *class* yang memiliki tetangga terbanyak yang dihitung secara tegas. Pendekatan seperti ini dapat mengakibatkan distorsi data yang besar ketika nilai k terlalu besar. Namun sebaliknya apabila menggunakan nilai k yang terlalu kecil, maka bisa jadi menyebabkan k-NN sensitif terhadap data yang *noise*. Masalah tersebut dapat terjadi karena setiap tetangga dianggap sama pentingnya, mempunyai bobot yang sama terhadap data yang akan diprediksi. Masalah tersebut dapat diatasi dengan melakukan pembobotan pada fungsi jarak untuk data tetangga (*neighborhood weighting* atau *distance weighting*), yang mana *weight* (w_j) dari setiap tetangga dapat didefinisikan sebagai berikut.

$$w_j = \frac{1}{d(x', x_j)^2} \quad (146)$$

Notasi $d(x', x_j)$ menyatakan jarak data uji/prediksi (x') ke data latih (x_j , $j = 1, 2, \dots, m$), yang mana m menyatakan banyaknya data tetangga atau sama dengan k .

Dengan demikian keputusan klasifikasi k-NN yang didefinisikan pada Persamaan (145) dapat diubah menjadi.

$$y' = \operatorname{argmax}_v \sum_{D \in D_z}^k w_j I(v = y_c) \quad (147)$$

Contoh 8.7 Weighted k-NN: Klasifikasi (Manual)

No.	Atribut			Euclidean Distance	k=5	w	Prediction Y*			
	X1	X2	Y				+	-	+	-
L	1.	2	1	+	4,12	No				
	2.	2	3	+	2,24	Yes	$\frac{1}{2,24^2} = 0,20$	1	0	$1*0,20=0,20$
	3.	2	1	+	4,12	No				
	4.	1	1	+	4,00	Yes	$\frac{1}{4,00^2} = 0,06$	1	0	$1*0,06=0,06$
	5.	2	2	+	3,16	Yes	$\frac{1}{3,16^2} = 0,10$	1	0	$1*0,10=0,10$
	6.	1	7	-	2,00	Yes	$\frac{1}{2,00^2} = 0,25$	0	1	$0*0,25=0$
	7.	2	6	-	1,41	Yes	$\frac{1}{1,41^2} = 0,50$	0	1	$0*0,50=0$
z	8.	1	5	?			Jumlah		0,36	0,75

Dengan demikian, prediksi/klasifikasi data uji/prediksi ke-8 adalah:

$$\text{Argmax}(y_c) = \{0,36, 0,75\} = -$$

Perhatikan bahwa contoh soal yang sama digunakan untuk k-NN dan *Weighted k-NN*, namun memiliki jawaban yang berbeda. k-NN memutuskan output data uji/prediksi ke-8 adalah label *class* (+), sedangkan *Weighted k-NN* memutuskan output data uji/prediksi ke-8 adalah label *class* (-). Jika bengitu, mana yang lebih baik?

Dari 5 tetangga terdekat, 3 tetangga milik label *class* (+), sedangkan 2 tetangga milik label *class* (-). Jika menggunakan standar k-NN, maka keputusan klasifikasi tentu saja adalah label *class* (+). Namun sebenarnya 3 tetangga milik label *class* (+) merupakan semua tetangga yang lebih jauh jaraknya {2,24; 4,00; dan 3,16} dibandingkan milik label *class* (-) {2,00 dan 1,41}. Dengan melakukan pembobotan terhadap setiap tetangga (*neighborhood weighted* atau *distance weighted*), yang mana tetangga yang jaraknya lebih dekat diberikan bobot yang lebih besar, maka keputusan klasifikasi akan lebih *smooth*, sehingga dapat mereduksi terjadinya distorsi data (seperti contoh ini karena data uji/prediksi ke-8 diprediksi ke label *class* (+), pedahal seharusnya label *class* (-)) ketika nilai *k* terlalu besar dan mereduksi *noisy data* ketika nilai *k* terlalu kecil. *Weighted k-NN* mampu mengatasi masalah tersebut, sehingga pada contoh ini, output data uji/prediksi ke-8 adalah label *class* (-) dengan nilai = 0,75 yang lebih besar daripada label *class* (-) dengan nilai = 0,36.

8.6 Fuzzy k-NN

Mirip seperti *Weighted K-NN*, *Fuzzy K-NN* (FkNN) juga bertujuan untuk mereduksi *noisy data* dan distorsi data melalui pendekatan *distance weighting* untuk memperoleh probabilitas setiap label *class*, namun dengan cara yang berbeda. Dalam FkNN, pendekatan derajat keanggotaan *Fuzzy* digunakan untuk menentukan derajat keanggotaan setiap label *class* [48]. FkNN mendefinisikan derajat keanggotaan setiap label *class* berdasarkan nilai jarak setiap data tetangga yang didefinisikan sebagai berikut.

$$u(x', y'_c) = \frac{\sum_{j=1}^k u(x_j, y_c) d(x', x_j)^{-2}}{\sum_{j=1}^k d(x', x_j)^{-2}} \quad (148)$$

Keterangan:

- k : menyatakan banyaknya tetangga terdekat.
- $u(x', y'_c)$: menyatakan derajat keanggotaan label *class* (y') ke- c untuk data prediksi/uji (x').
- $u(x_j, y_c)$: menyatakan derajat keanggotaan data tetangga (x) ke- j pada label *class* (y) ke- c . Nilainya = 1 jika data x_j milik *class* y_c (*class* dari $x_j = y_c$), selain itu = 0.
- $d(x', x_j)$: menyatakan jarak data prediksi/uji (x') ke data latih tetangga (x) ke- j .
- m : menyatakan bobot pangkat (*weight exponent*), nilainya > 1 .

Keputusan klasifikasi/prediksi FkNN adalah label *class* yang memiliki probabilitas terbesar, didefinisikan sebagai berikut.

$$y' = \operatorname{argmax}_c(u(x', y'_c)) \quad (149)$$

Contoh 8.8 FkNN: Klasifikasi (Manual)

Dengan menggunakan contoh manual yang sama seperti pada k-NN dan *Weighted k-NN*, maka penyelesaian manual FkNN adalah sebagai berikut.

Data latih (L) yang paling dekat jaraknya dengan data uji (z) menggunakan metode pengukuran jarak *Euclidean* adalah data latih ke-7 (L_7), yaitu sebagai berikut:

$$d(x', x_7) = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2} = \sqrt{(2-1)^2 + (6-5)^2} = 1,41$$

Dengan dimikian tentu saja data latih ke-7 (L_7) masuk dalam tetangga ($k=5$) dari data uji (z) karena memiliki jarak yang paling dekat dengan data uji (z). Label *class* (output) dari L_7 adalah (-), sehingga *class* prediksi (y') untuk label (+) = 0 dan (-) = 1. Jika menggunakan *Weighted k-NN*, maka bobot (w) dari L_7 , yang mana L_7 dalam k setelah diurutkan dari jarak terdekat, menjadi data latih/tetangga pertama (L_1), maka nilai w_1 (bobot dari data latih ke-7 atau data tetangga pertama) adalah sebagai berikut:

$$w_1 = \frac{1}{d(x', x_1)^2} = \frac{1}{1,41^2} = 0,5$$

Sehingga *class* prediksi (y') untuk label (+) = 0 * 0,5 = 0, sedangkan untuk label (-) = 1 * 0,5 = 0,5. Jika menggunakan FkNN, yang mana nilai parameter $m = 2$ (*default*), maka penyelesaiannya adalah sebagai berikut:

$$\begin{aligned} u(x', y'_{(+)}) &= \frac{\sum_{j=1}^k u(x_j, y_c) d(x', x_j)^{-2}}{\sum_{j=1}^k d(x', x_j)^{-2}} \\ &= \frac{\left(0 * 1,41^{\frac{-2}{2-1}}\right) + \left(0 * 2,00^{\frac{-2}{2-1}}\right) + \left(1 * 2,24^{\frac{-2}{2-1}}\right) + \left(1 * 3,16^{\frac{-2}{2-1}}\right) + \left(1 * 4,00^{\frac{-2}{2-1}}\right)}{1,41^{\frac{-2}{2-1}} + 2,00^{\frac{-2}{2-1}} + 2,24^{\frac{-2}{2-1}} + 3,16^{\frac{-2}{2-1}} + 4,00^{\frac{-2}{2-1}}} \\ &= \frac{0,3625}{1,1125} = 0,3258 \end{aligned}$$

$$\begin{aligned} u(x', y'_{(-)}) &= \frac{\sum_{j=1}^k u(x_j, y_c) d(x', x_j)^{-2}}{\sum_{j=1}^k d(x', x_j)^{-2}} \\ &= \frac{\left(1 * 1,41^{\frac{-2}{2-1}}\right) + \left(1 * 2,00^{\frac{-2}{2-1}}\right) + \left(0 * 2,24^{\frac{-2}{2-1}}\right) + \left(0 * 3,16^{\frac{-2}{2-1}}\right) + \left(0 * 4,00^{\frac{-2}{2-1}}\right)}{1,41^{\frac{-2}{2-1}} + 2,00^{\frac{-2}{2-1}} + 2,24^{\frac{-2}{2-1}} + 3,16^{\frac{-2}{2-1}} + 4,00^{\frac{-2}{2-1}}} \\ &= \frac{0,7500}{1,1125} = 0,6742 \end{aligned}$$

Sehingga hasil prediksi (y') adalah (-) dengan nilai probabilitas = 0,6742 yang lebih besar daripada label *class* (+) = 0,3258.

Lengkapnya, penyelesaian setiap proses k-NN, *Weighted k-NN*, dan FkNN ditunjukkan pada tabel berikut ini.

No.	Variabel			Euclidean Distance	k=5		Prediction y'					
							Standar k-NN		Weighted k-NN		Fuzzy k-NN (m=2)	
	x1	x2	y		Y/N	Asc	(+)	(-)	(+)	(-)	(+)	(-)
L	1.	2	1	(+)	4,12	0						
	2.	2	3	(+)	2,24	1	3	1	0	0,20	0,00	0,20
	3.	2	1	(+)	4,12	0						
	4.	1	1	(+)	4,00	1	5	1	0	0,06	0,00	0,06
	5.	2	2	(+)	3,16	1	4	1	0	0,10	0,00	0,10
	6.	1	7	(-)	2,00	1	2	0	1	0,00	0,25	0,25
	7.	2	6	(-)	1,41	1	1	0	1	0,00	0,50	0,50
z	8.	1	5	?			Jumlah	3	2	0,36	0,75	1,11
										0,33		0,67
					y'		(+)	(-)			(+)	(-)

Perhatikan bahwa contoh soal yang sama digunakan untuk k-NN, *Weighted k-NN*, dan FkNN namun memiliki jawaban yang berbeda-beda. Seperti yang telah dijelaskan sebelumnya bahwa sebenarnya 2 tetangga milik label *class* (-) sebenarnya tetangga yang paling dekat jaraknya dibandingkan semua (3) tetangga milik label *class* (+), sehingga seharusnya hasil prediksi adalah (-). k-NN memutuskan hasil prediksi adalah (+), sedangkan *Weighted k-NN* dan FkNN memutuskan hasil prediksi adalah (-), namun selisih nilai antara label *class* (+) dengan (-) dari FkNN lebih kecil daripada dari *Weighted k-NN*. Dengan demikian, dapat disimpulkan bahwa hasil keputusan yang diberikan FkNN lebih *smooth* daripada *Weighted k-NN*, sedangkan hasil keputusan yang diberikan k-NN sebenarnya kurang tepat (terjadi distorsi data).

8.7 Fuzzy k-NN in every class

Pengembangan dari FkNN adalah *Fuzzy K-Nearest Neighbor in every class* (FkNNC). FkNNC sedikit memodifikasi FkNN dengan memberikan sejumlah k tetangga pada setiap label *class* (setiap label *class* memiliki jumlah tetangga yang sama sebanyak k) [20]. Dengan demikian, cara kerja FkNNC sebenarnya berbeda dengan k-NN, *Weighted k-NN*, bahkan FkNN karena setiap label *class* pada FkNNC memiliki tetangga terdekat sebanyak k , yang mana tetangga yang paling dekat untuk suatu label *class* adalah data latih milik label *class* tersebut yang jaraknya paling dekat dengan data uji/prediksi, bukan data latih yang paling dekat jaraknya dengan data uji seperti pada standar k-NN dan lainnya. Dengan kata lain, label *class* pada FkNNC sangat berpengaruh. Mungkin metode ini bertujuan pula untuk mereduksi *unbalanced class*. Namun apakah lebih baik daripada FkNN dalam mereduksi distorsi data? Mungkin jawabannya bisa didapatkan melalui Contoh 8.9.

Perhitungan jarak data prediksi/uji ($z = (x'_i, y')$) ke data latih tetangga ($L = (x_i, y)$) ke- j menggunakan metode yang sama seperti standar k-NN, seperti *Euclidean*, *Manhattan*, dll yang secara umum dapat didefinisikan sebagai berikut.

$$d(x', x_j) = \left(\sum_{i=1}^n |x_i - x'_i|^p \right)^{\frac{1}{p}} \quad (150)$$

$d(x', x_j)$ menyatakan jarak antara data uji/prediksi ($z = (x', y')$) ke data latih ($L = (x_i, y_i)$) ke- j . Notasi n menyatakan banyaknya atribut. Jika $p = 1$ menyatakan bahwa perhitungan jarak sama dengan metode *Manhattan*, $p = 2$ menyatakan bahwa perhitungan jarak sama dengan metode *Euclidean*, dan $p = \infty$ menyatakan bahwa perhitungan jarak sama dengan metode *Chebyshev*.

Selanjutnya jarak data prediksi/uji ($z = (x', y')$) ke data latih tetangga ($L = (x_i, y_i)$) ke- j untuk setiap label *class* (y) ke- c dijumlahkan, didefinisikan sebagai berikut.

$$S_c = \sum_{j=1}^k u(x_j, y_c) d(x', x_j)^{\frac{-2}{(m-1)}} \quad (151)$$

Selanjutnya nilai S_c setiap label *class* (y) ke- c digabungkan, didefinisikan sebagai berikut.

$$D = \sum_{c=1}^p S_c \quad (152)$$

Notasi p menyatakan banyaknya *class*.

Derajat keanggotaan data prediksi/uji (z) untuk label *class* (y') ke- c didefinisikan sebagai berikut.

$$u(x', y'_c) = \frac{S_c}{D} \quad (153)$$

Akhirnya keputusan klasifikasi/prediksi adalah label *class* yang memiliki probabilitas terbesar yang didefinisikan pada Persamaan (149).

Contoh 8.9 FkNN: Klasifikasi (Manual)

No.	Variabel			d	k-NN		FkNN (m=2)			k=3	FkNN (m=2)				
	x1	x2	y		k=3	(+)	(-)	m	(+)	(-)	m	(+)	(-)		
L	1.	10	9	(+)	7,21	0					0				
	2.	9	10	(+)	7,07	0					0				
	3.	9	9	(+)	6,40	0					1	0,0244	1	0,0244	
	4.	7	2	(+)	4,24	1	1	0	0,0556	0,0556	0,0000	1	0,0556	1	0,0556
	5.	3	8	(+)	3,16	1	1	0	0,1000	0,1000	0,0000	1	0,1000	1	0,1000
	6.	10	9	(-)	7,21	0					0				
	7.	9	10	(-)	7,07	0					0				
	8.	8	8	(-)	5,00	0					1	0,0400	0	0,0400	
	9.	8	2	(-)	5,00	0					1	0,0400	0	0,0400	
	10.	3	6	(-)	1,41	1	0	1	0,5000	0,0000	0,5000	1	0,5000	0	0,5000
z	11.	4	5	?	Jumlah	2	1	0,6556	0,1556	0,5000	0,7599	0,1799	0,5800		
								0,2373	0,7627			0,2368	0,7632		
y'					(+)			(-)			(-)				

Perhatikan bahwa data-data tetangga milik label *class* (-) sebenarnya lebih dekat jaraknya daripada data-data tetangga milik label *class* (+). Nilai k yang digunakan

adalah 3, yang mana 1 tetangga milik label *class* (-) yang jaraknya paling dekat dengan data uji (z) daripada 2 tetangga milik label *class* (-). Namun standar k-NN tentu saja memutuskan hasil prediksi adalah (+). Kesalahan prediksi seperti ini diperbaiki oleh FkNN, yang mana FkNN berhasil memprediksi data uji (z) dengan tepat, yaitu (-). Cara kerja yang berbeda dilakukan oleh FkNNC, yang mana setiap label *class* memiliki k tetangga, sehingga setiap label *class* memiliki 3 tetangga terdekat masing-masing. Namun ternyata hasil prediksi FkNNC tidak jauh berbeda dengan FkNN. Probabilitas label *class* (-) dari FkNNC sedikit lebih besar daripada FkNN, selisih 0,0005.

8.8 KFACWNB-NN

Apa itu algoritma KFACWNB-NN (*k Fuzzy Absolute Correlation Weighted Naïve Bayes – Nearest Neighbor*)? Hingga buku ini ditulis, algoritma tersebut sebenarnya belum ada, masih sementara proses penelitian yang penulis lakukan. Konsepnya, KFACWNB-NN merupakan pengembangan dari k-NN dengan menerapkan *Fuzzy* dan AC W-NB untuk memperbaiki kelemahan k-NN yang sensitif terhadap atribut-atribut yang kurang relevan, distorsi data, dan *noisy data*.

k-NN melakukan klasifikasi/prediksi menggunakan teknik *majority vote* dari k data latih yang jaraknya terdekat dengan data yang akan diprediksi menggunakan salah satu metode pengukuran jarak (*dissimilarity*). Metode-metode *dissimilarity*, seperti *Euclidean*, *Manhattan*, dll mengukur jarak antara data yang akan diprediksi ke setiap data latih berdasarkan setiap fitur/atribut yang dianggap independen (tidak saling berelasi) atau setiap atribut dianggap sama pentingnya. Padahal dalam banyak kasus, asumsi tersebut tidak selalu tepat. Hal ini berarti bahwa k-NN sensitif terhadap atribut-atribut yang kurang relevan. Seperti yang telah dikemukakan sebelumnya, pendekatan *Absolute Correlation Coefficient* (ACC) dapat menentukan kekuatan korelasi antar atribut dan bekerja pada atribut bertipe numerik. Penerapan ACC pada NB yang diistilahkan AC W-NB terbukti berhasil meningkatkan kinerja NB. Oleh karena itu, dengan menggunakan konsep dan tujuan yang sama, pendekatan tersebut dapat pula diterapkan pada k-NN.

Selain itu, metode *Weighted k-NN*, *FkNN*, dan *FkNNC* terbukti mampu menangani kelemahan k-NN yang sensitif terhadap distorsi data dan data yang *noise*. Idenya, probabilitas setiap *label class* yang diberikan ketika metode tersebut berdasarkan *distance weighted* dapat diintegrasikan dengan probabilitas setiap label *class* yang diberikan AC W-NB berdasarkan *attribute weighted*. Logikanya, penyatuhan pendekatan *distance weighted* menggunakan *Fuzzy Weighted k-NN* dengan *attribute weighted* menggunakan AC W-NB akan mereduksi distorsi data, data yang *noise*, dan atribut-atribut yang kurang relevan sehingga kinerja k-NN dalam klasifikasi data dapat meningkat. Sayangnya, metode tersebut belum dapat penulis jelaskan lebih jauh karena masih dalam tahap penelitian. Semoga pada buku edisi selanjutnya, penelitian yang penulis lakukan tersebut telah berakhir sehingga dapat penulis bahas dalam buku berikutnya.

8.9 Soal Latihan Naïve Bayes, k-NN, & Fuzzy

1. Unduh salah satu *dataset classification* pada *UCI Machine Learning Repository* kemudian lakukan analisis klasifikasi menggunakan algoritma *Naïve Bayes* dan *k-NN*, mana yang terbaik untuk *dataset* yang anda gunakan?
2. Gunakan kembali *dataset* tersebut untuk melakukan analisis komparasi terhadap algoritma *Naïve Bayes* dan AC W-NB.
3. Gunakan kembali *dataset* tersebut untuk melakukan analisis komparasi terhadap algoritma *k-NN*, *Weighted k-NN*, *FkNN*, dan *FkNNC*.
4. Dapatkah anda mengintegrasikan algoritma *Weighted k-NN* dengan *FkNN*? Buatlah prosedurnya dan uji cobanya menggunakan *dataset* tersebut.
5. Buatlah *dataset* buatan anda sendiri (minimal 5 atribut numerik, 10 data latih, dan 1 data uji/prediksi), kemudian tentukan hasil prediksi dari data uji/prediksi tersebut menggunakan algoritma *Naïve Bayes*, *k-NN*, AC W-NB, *Weighted k-NN*, *FkNN*, dan *FkNNC* secara manual.

9. Bonus: C4.5, Linear Regression, & A-Priori

No.	Materi	Tujuan Pembelajaran
1.	Decision Tree (C4.5)	Anda mampu memahami, menjelaskan, dan menerapkan algortima C4.5 secara manual maupun menggunakan <i>tools</i> dalam menangani masalah klasifikasi.
2.	Linear Regression	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>Linear Regression</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah regresi/estimasi.
3.	A-Priori	Anda mampu memahami, menjelaskan, dan menerapkan algortima <i>A-Priori</i> secara manual maupun menggunakan <i>tools</i> dalam menangani masalah asosiasi.

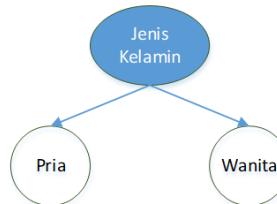
9.1 Decision Tree (C4.5)

Algoritma C4.5 dapat digunakan untuk menangani masalah-masalah klasifikasi data. Algoritma ini diperkenalkan oleh Quinlan (1996) untuk mengatasi kelemahan dari ID3 yang tidak dapat menangani atribut bertipe numerik. Pada prinsipnya, cara kerja algoritma C4.5 sama dengan ID3, yaitu menggunakan nilai *Gain* dalam menentukan fitur/atribut yang menjadi pemecah node pada pohon keputusan. Perbedaannya adalah algoritma C4.5 dapat menangani atribut bertipe numerik dengan cara melakukan diskretisasi data terhadap atribut bertipe numerik.

Pendekatan untuk menyatakan syarat pengujian pada node terdiri atas:

1. Pemecahan pada atribut binominal/biner

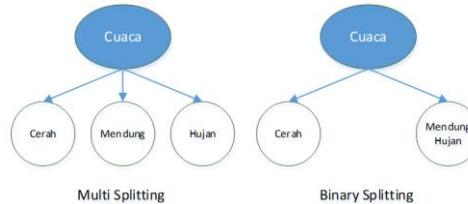
Pemecahan terdiri dari 2 cabang (*binary splitting*), seperti ditunjukkan pada Gambar 9.1 berikut ini.



Gambar 9.1 Binary Splitting pada Atribut Binominal/Biner

2. Pemecahan pada atribut nominal (kategorikal)

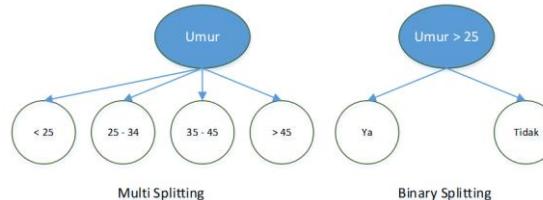
Pemecahan terdiri dari *multi splitting* dan *binary splitting*, seperti ditunjukkan pada Gambar 9.2 berikut ini.



Gambar 9.2 Binary dan Multi Splitting pada Atribut Nominal

3. Pemecahan pada atribut numerik

Pemecahan terdiri dari *multi splitting* dan *binary splitting*, seperti ditunjukkan pada Gambar 9.3 berikut ini.



Gambar 9.3 Bianry dan Multi Splitting pada Atribut Numerik

Algoritma C4.5 menggunakan nilai *Entropy* (E), *Information Gain* (IG), *Gain* (G), dan *Gain Ratio* (GR) dalam melakukan klasifikasi data. Nilai *Entropy* dari suatu variabel maupun suatu label variabel didefinisikan sebagai berikut.

$$E(S) = - \sum_{i=1}^m P(W_i|S) \log_2 P(W_i|S) \quad (154)$$

- $E(S)$: menyatakan *Entropy* dari objek/variabel S.
- m : menyatakan banyaknya label atau banyaknya nilai yang berbeda pada objek/variabel S.
- $P(W_i|S)$: menyatakan proporsi label ke- i (W_i) dalam semua data yang diproses untuk objek/variabel S, yaitu jumlah data milik label ke- i (W_i) dibagi dengan jumlah seluruh data yang diproses untuk objek/variabel S.

Nilai *Information Gain* (IG) dari suatu variabel didefinisikan sebagai berikut.

$$IG(S) = \sum_{i=1}^m P(W_i|S) E(S_i) \quad (155)$$

- $IG(S)$: menyatakan *Information Gain* dari objek/variabel S.
- m : menyatakan label atau banyaknya nilai yang berbeda pada objek/variabel S.
- $P(W_i|S)$: menyatakan proporsi label ke- i (W_i) dalam semua data yang diproses untuk objek/variabel S, yaitu jumlah data milik label ke- i (W_i) dibagi dengan jumlah seluruh data yang diproses untuk objek/variabel S.
- $E(S_i)$: menyatakan *Entropy* dari label ke- i objek/variabel S.

Nilai *Gain* (G) dari suatu variabel didefinisikan sebagai berikut.

$$G(S) = E(S) - IG(S_i) \quad (156)$$

- $G(S)$: menyatakan *Gain* dari objek/variabel S.
- $E(S)$: menyatakan *Entropy* dari objek/variabel S.
- $IG(S_i)$: menyatakan *Information Gain* dari objek/variabel S label ke- i .

Nilai *Gain Ratio* (GR) suatu variabel didefinisikan sebagai berikut.

$$GR(S) = \frac{G(S)}{E(S)} \quad (157)$$

- $GR(S)$: menyatakan *Gain Ratio* dari objek/variabel S.
- $G(S)$: menyatakan *Gain* dari objek/variabel S.
- $E(S)$: menyatakan *Entropy* dari objek/variabel S.

Diskritisasi data pada atribut bertipe numerik/kontinyu bertujuan untuk mentransformasi data dari numerik ke diskrit (kategorikal). Beberapa teknik yang

umum digunakan, yaitu *Boolean Reasoning*, *Entropy*, *Equal Frequency Binning*, dan dapat pula dengan teknik normalisasi seperti mean dan standar deviasi.

Teknik diskretisasi yang umum digunakan algoritma C4.5 adalah *Equal Frequency Binning* dan *Entropy*. *Equal Frequency Binning* memecah data numerik menjadi beberapa *bin*, misalnya domain suatu atribut numerik adalah {0; 100}, dapat dibagi menjadi empat bin (0..24, 25..49, 50..74, 75..100). Setiap nilai atribut akan dikonversi menjadi atribut nominal yang berkorespondensi dengan salah satu bin. Oleh karena itu metode ini merupakan *unsupervised discretization method*. Namun metode ini dapat menyebabkan banyak informasi yang bisa hilang.

Contoh 9.1 Diskretisasi Binning pada C4.5

No	Suhu	Class
1.	85	Tidak
2.	80	Tidak
3.	83	Ya
4.	70	Ya
5.	68	Ya
6.	65	Tidak
7.	64	Ya
8.	72	Tidak
9.	69	Ya
10.	75	Ya
11.	75	Ya
12.	72	Ya
13.	81	Ya
14.	71	Tidak

$$\text{Jumlah data} = 14$$

$$\text{Jumlah data Suhu dengan label class 'Ya'} = 9$$

$$\text{Jumlah data Suhu dengan label class 'Tidak'} = 5$$

$$\text{Jumlah data Suhu untuk pemecahan } \leq 70 = 5$$

$$\text{Jumlah data Suhu untuk pemecahan } > 70 = 9$$

$$E(\text{Suhu}) = - \left(\left(\left(\frac{9}{14} \log_2 \right) \frac{9}{14} \right) + \left(\left(\frac{5}{14} \log_2 \right) \frac{5}{14} \right) \right) = 0,9403$$

$$E(\text{Suhu}_{\leq 70}) = - \left(\left(\left(\frac{4}{5} \log_2 \right) \frac{4}{5} \right) + \left(\left(\frac{1}{5} \log_2 \right) \frac{1}{5} \right) \right) = 0,7219$$

$$E(\text{Suhu}_{>70}) = - \left(\left(\left(\frac{5}{9} \log_2 \right) \frac{5}{9} \right) + \left(\left(\frac{4}{9} \log_2 \right) \frac{4}{9} \right) \right) = 0,9911$$

$$IG(\text{Suhu}) = \left(\frac{5}{14} 0,7219 \right) + \left(\frac{9}{14} 0,9911 \right) = 0,8950$$

$$G(\text{Suhu}) = 0,9403 - 0,8950 = 0,0453$$

Lengkapnya, ditunjukkan pada tabel berikut ini.

Suhu	v = 70		v = 75		v = 80	
	<=70	>70	<=75	>75	<=80	>80
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Jumlah	5	9	10	4	11	3
	14		14		14	
E(Suhu)	0.7219	0.9911	0.8813	1.0000	0.9457	0.9183
IG(Suhu)	0,8950		0,9152		0,9398	
G(Suhu)	0,0453		0,0251		0,0005	

Dengan demikian, objek/variabel/atribut Suhu didiskretisasi menjadi 2 label (kategori) ≤ 70 dan > 70 .

Sedangkan diskretisasi berbasis *Entropy* melakukan pemecahan berdasarkan nilai *Gain*. Algoritmanya adalah sebagai berikut:

1. Urutkan data secara *ascending*.
 2. Hitung *Split Info* (SP) ke- i , yaitu rata-rata nilai per 2 item data yang bersebelahan, lakukan pada seluruh item data.
- $$SP_i = \frac{a_i + a_{i+1}}{2} \quad (158)$$
3. Hitung *Entropy* dari setiap label *Split Info*.
 4. Hitung *Information Gain* dari setiap *Split Info*.
 5. Hitung *Gain* dari setiap *Split Info*.
 6. Gunakan *Split Info* yang memiliki nilai *Gain* tertinggi.

Contoh 9.2 Diskretisasi Entropy pada C4.5

Umur (telah terurut secara ascending)	Class
35	Tinggi
35	Sedang
37	Tinggi
38	Sedang
41	Sedang
42	Tinggi
47	Sedang

Nilai yang berbeda atribut umur adalah $\{35, 37, 38, 41, 42, 47\}$.

$$SP_{35} = \frac{35+37}{2} = 36, \text{ sehingga } SP_{35} \text{ adalah } \leq 36 \text{ dan } > 36.$$

Lengkapnya ditunjukkan pada tabel berikut.

Split Info	Sedang	Tinggi	Jumlah	Entropy	Info Gain	Gain
35	<=36	1	1	2	1	0,97925
	>36	3	2	5	0,970951	
37	<=37,5	1	2	3	0,918296	0,857143
	>37,5	3	1	4	0,811278	
38	<=39,5	2	2	4	1	0,964984
	>39,5	2	1	3	0,918296	
41	<=41,5	3	2	5	0,970951	0,97925
	>41,5	1	1	2	1	
42	<=44,5	3	3	6	1	0,857143
	>44,5	1	0	1	0	
47	Jumlah	4	3	7	0,985228	

Maka atribut Umur didiskretisasi menjadi 2 label (kategori) $\leq 44,5$ dan $> 44,5$.

Contoh 9.3 C4.5: Klasifikasi (Manual)

No	Cuaca	Suhu	Kelembaban	Angin	Bermain (Class)
1.	Cerah	85	85	Biasa	Tidak
2.	Cerah	80	90	Kencang	Tidak
3.	Mendung	83	78	Biasa	Ya
4.	Hujan	70	96	Biasa	Ya
5.	Hujan	68	80	Biasa	Ya
6.	Hujan	65	70	Kencang	Tidak
7.	Mendung	64	65	Kencang	Ya
8.	Cerah	72	95	Biasa	Tidak
9.	Cerah	69	70	Biasa	Ya
10.	Hujan	75	80	Biasa	Ya
11.	Cerah	75	70	Kencang	Ya
12.	Mendung	72	90	Kencang	Ya
13.	Mendung	81	75	Biasa	Ya
14.	Hujan	71	80	Kencang	Tidak

Diskretisasi atribut Suhu ditunjukkan pada tabel berikut ini.

Suhu	$v = 70$		$v = 75$		$v = 80$	
	≤ 70	> 70	≤ 75	> 75	≤ 80	> 80
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Jumlah	5	9	10	4	11	3
			14			14
Entropy	0.7219	0.9911	0.8813	1.0000	0.9457	0.9183
Gain	0.0453		0.0251		0.0005	

Diskretisasi atribut Kelembaban ditunjukkan pada tabel berikut ini.

Kelembaban	$v = 70$		$v = 75$		$v = 80$		$v = 85$	
	≤ 70	> 70	≤ 75	> 75	≤ 80	> 80	≤ 85	> 85
Ya	3	6	4	5	7	2	7	2
Tidak	1	4	1	4	2	3	3	2
Jumlah	4	10	5	9	9	5	10	4
			14		14		14	
Entropy	0.8113	0.9710	0.7219	0.9911	0.7642	0.9710	0.8813	1.0000
Gain	0.0150		0.0453		0.1022		0.0251	

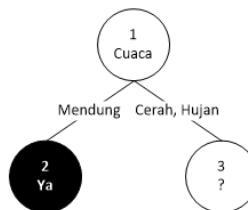
Penyelesaian pada node akar ditunjukkan pada tabel berikut ini.

Node Akar	Jumlah	Ya	Tidak	Entropy	Gain
Class	14	9	5	0.9403	
Cuaca	Cerah	5	2	0.9710	0.2467
	Mendung	4	4	0.0000	
	Hujan	5	3	0.9710	
Suhu	≤ 70	5	4	0.7219	0.0453
	> 70	9	5	0.9911	
Kelembaban	≤ 80	9	7	0.7642	0.1022
	> 80	5	2	0.9710	
Angin	Biasa	8	6	0.8113	0.0481
	Kencang	6	3	1.0000	

Atribut yang memiliki *Gain* tertinggi adalah Cuaca, sehingga menjadi node akar. Karena terdiri dari 3 label, maka lakukan *splitting*.

Cuaca	Jumlah	Entropy	Gain Ratio
Cerah	5		
Mendung	4	1.5774	0.16
Hujan	5		
Cerah	5	0.9403	0.26
Mendung & Hujan	9		
Cerah & Mendung	9	0.9403	0.26
Hujan	5		
Cerah & Hujan	10	0.8631	0.29
Mendung	4		

Dengan demikian, pohon keputusan yang dibentuk pada node akar ditunjukkan pada gambar berikut ini.



Selanjutnya proses untuk node 3 menggunakan data yang atribut *Cuaca IS Cerah, Hujan*. Diskretisasi atribut Suhu ditunjukkan pada tabel berikut ini.

Suhu	v = 70		v = 75		v = 80	
	<=70	>70	<=75	>75	<=80	>80
Ya	3	2	5	0	5	0
Tidak	1	4	3	2	4	1
Jumlah	4	6	8	2	9	1
	10		10		10	
Entropy	0.8113	0.9183	0.9544	0.0000	0.9911	0.0000
Gain	0.1245		0.2365		0.1080	

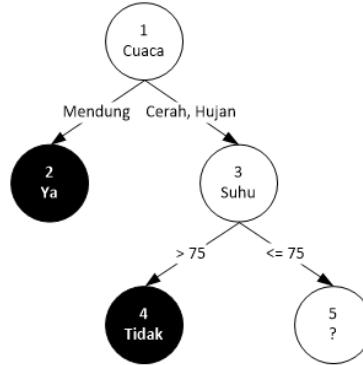
Diskretisasi atribut Kelembaban ditunjukkan pada tabel berikut ini.

Kelembaban	v = 70		v = 75		v = 80	
	<=70	>70	<=75	>75	<=80	>80
Ya	2	3	2	3	4	1
Tidak	1	4	1	4	2	3
Jumlah	3	7	3	7	6	4
	10		10		10	
Entropy	0.9183	0.9852	0.9183	0.9852	0.9183	0.8113
Gain	0.0349		0.0349		0.1245	

Penyelesaian pada node 3 ditunjukan pada tabel berikut ini.

Node 3	Jumlah	Ya	Tidak	Entropy	Gain
Class	10	5	5	1.0000	
Cuaca	Cerah	5	2	0.9710	0.0290
	Hujan	5	3	0.9710	
Suhu	<=75	8	5	0.9544	0.2365
	>75	2	0	0.0000	
Kelembaban	<=80	6	4	0.9183	0.1245
	>80	4	1	0.8113	
Angin	Biasa	6	4	0.9183	0.1245
	Kencang	4	1	0.8113	

Atribut yang memiliki *Gain* tertinggi adalah Suhu, sehingga menjadi node 3. Semua *Suhu IS > 75* merupakan label *class* Ya, maka menjadi daun. Pohon keputusan hingga node 3 ditunjukkan pada gambar berikut ini.



Selanjutnya proses untuk node 5 menggunakan data yang atribut *Cuaca IS Cerah, Hujan* dan *Suhu IS ≤ 75*. Diskretisasi atribut Suhu ditunjukkan pada tabel berikut ini.

Suhu	$v = 70$		$v = 75$	
	≤ 70	> 70	≤ 75	> 75
Ya	3	2	5	0
Tidak	1	2	3	0
Jumlah	4	4	8	0
	8		8	
Entropy	0.8113	1.0000	0.9544	0.0000
Gain	0.0488		0.0000	

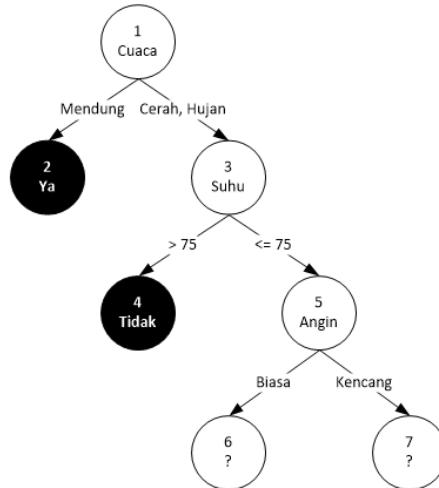
Diskretisasi atribut Kelembaban ditunjukkan pada tabel berikut ini.

Kelembaban	$v = 70$		$v = 75$		$v = 80$	
	≤ 70	> 70	≤ 75	> 75	≤ 80	> 80
Ya	2	3	2	3	4	1
Tidak	1	2	1	2	2	1
Jumlah	3	5	3	5	6	2
	8		8		8	
Entropy	0.9183	0.9710	0.9183	0.9710	0.9183	1.0000
Gain	0.0032		0.0032		0.0157	

Penyelesaian pada node 5 ditunjukkan pada tabel berikut ini.

		Jumlah	Ya	Tidak	Entropy	Gain
Class		8	5	3	0.9544	
Cuaca	Cerah	3	2	1	0.9183	0.0032
	Hujan	5	3	2	0.9710	
Suhu	≤ 70	4	3	1	0.8113	0.0488
	> 70	4	2	2	1.0000	
Kelembaban	≤ 80	6	4	2	0.9183	0.0157
	> 80	2	1	1	1.0000	
Angin	Biasa	5	4	1	0.7219	0.1589
	Kencang	3	1	2	0.9183	

Atribut yang memiliki *Gain* tertinggi adalah Angin, sehingga menjadi node 5. Pohon keputusan hingga node 5 ditunjukkan pada gambar berikut ini.



Selanjutnya proses untuk node 6 menggunakan data yang atribut *Cuaca IS Cerah, Hujan* dan *Suhu IS* ≤ 75 dan *Angin IS Biasa*. Diskretisasi atribut Suhu ditunjukkan pada tabel berikut ini.

Suhu	$v = 70$		$v = 75$		
	≤ 70	> 70	≤ 75	> 75	
Ya	3	1	4	0	
Tidak	0	1	1	0	
Jumlah	3	2	5	0	
Entropy	0.0000	1.0000	0.7219	0.0000	
Gain	0.3219		0.0000		

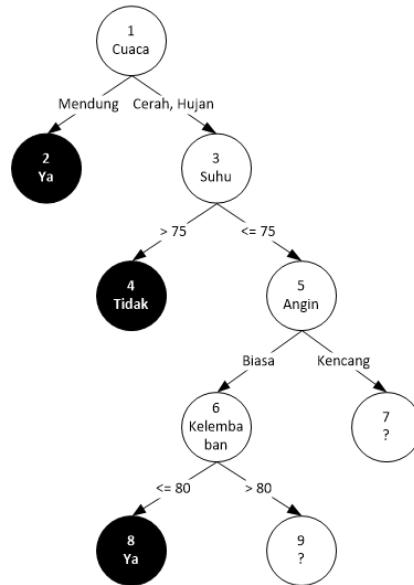
Diskretisasi atribut Kelembaban ditunjukkan pada tabel berikut ini.

Kelembaban	$v = 75$		$v = 80$		$v = 85$	
	≤ 75	> 75	≤ 80	> 80	≤ 85	> 85
Ya	1	3	3	1	3	1
Tidak	0	1	0	1	0	1
Jumlah	1	4	3	2	3	2
	5		5		5	
Entropy	0.0000	0.8113	0.0000	1.0000	0.0000	1.0000
Gain	0.0729		0.3219			0.3219

Penyelesaian pada node 6 ditunjukkan pada tabel berikut ini.

		Jumlah	Ya	Tidak	Entropy	Gain
Class		5	4	1	0.7219	
Cuaca	Cerah	2	1	1	1.0000	0.3219
	Hujan	3	3	0	0.0000	
Suhu	≤ 70	3	3	0	0.0000	0.3219
	> 70	2	1	1	1.0000	
Kelembaban	≤ 80	3	3	0	0.0000	0.3219
	> 80	2	1	1	1.0000	

Ketiga atribut memiliki *Gain* yang sama, maka boleh pilih sembarang atribut. Misalnya Kelembaban yang dipilih, sehingga menjadi node 6. Semua *Kelembaban IS ≤ 80* merupakan label *class Ya*, sehingga menjadi daun. Pohon keputusan hingga node 6 ditunjukkan pada gambar berikut ini.



Selanjutnya proses untuk node 7 menggunakan data yang atribut *Cuaca IS Cerah, Hujan* dan *Suhu IS ≤ 75* dan *Angin IS Kencang*. Diskretisasi atribut Suhu ditunjukkan pada tabel berikut ini.

Suhu	$v = 70$		$v = 75$	
	≤ 70	> 70	≤ 75	> 75
Ya	0	1	1	0
Tidak	1	1	2	0
Jumlah	1	2	3	0
Entropy	0.0000	1.0000	0.9183	0.0000
Gain	0.2516		0.0000	

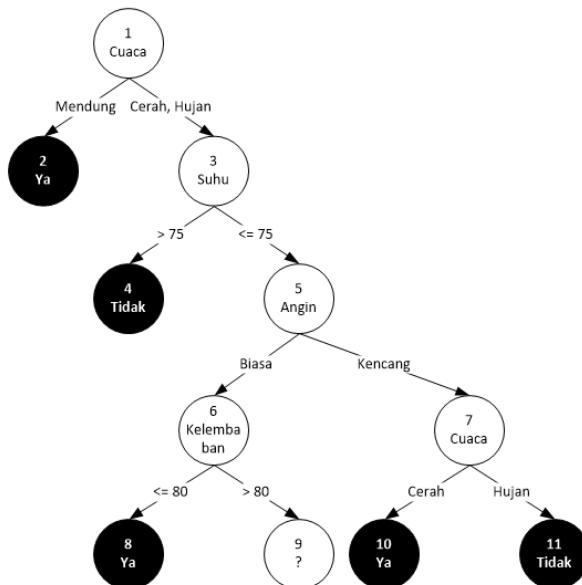
Diskretisasi atribut Kelembaban ditunjukkan pada tabel berikut ini.

Kelembaban	$v = 75$	
	≤ 75	> 75
Ya	1	0
Tidak	1	1
Jumlah	2	1
	3	
Entropy	1.0000	0.0000
Gain	0.2516	

Penyelesaian pada node 7 ditunjukan pada tabel berikut ini.

	Jumlah	Ya	Tidak	Entropy	Gain
Class	3	1	2	0.9183	
Cuaca	Cerah	1	1	0.0000	0.9183
	Hujan	2	0	0.0000	
Suhu	≤ 70	1	0	0.0000	0.2516
	> 70	2	1	1.0000	
Kelembaban	≤ 75	2	1	1.0000	0.2516
	> 75	1	0	0.0000	

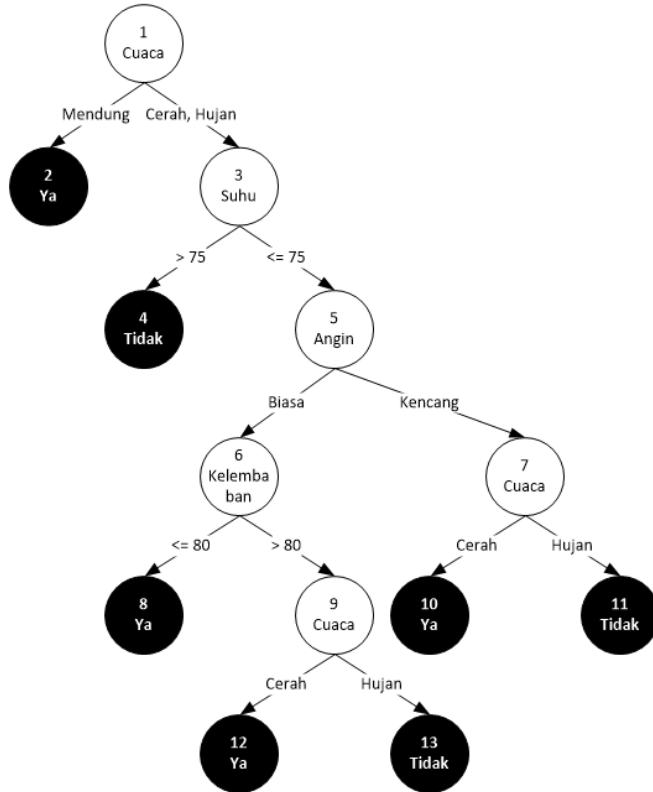
Atribut yang memiliki *Gain* tertinggi adalah Cuaca, sehingga menjadi node 7. Semua *Cuaca IS Cerah* merupakan label *class Ya* dan semua *Cuaca IS Hujan* merupakan label *class Tidak*, sehingga keduanya menjadi daun. Pohon keputusan hingga node 7 ditunjukkan pada gambar berikut ini.



Selanjutnya proses untuk node 9 menggunakan data yang atribut *Cuaca IS Cerah*, *Hujan* dan *Suhu IS ≤ 75* dan *Angin IS Biasa* dan *Kelembaban IS > 80* . Namun sebenarnya node 9 ini menjadi node cabang yang seharusnya tidak perlu dibentuk, sebaiknya cabangnya dipotong karena dapat menyebabkan terjadinya *overfitting*. Akhirnya, karena sudah tidak ada lagi node yang dapat dibentuk (tidak ada lagi data yang dapat diproses), maka induksi C4.5 berakhir dengan hasil (*rule*):

1. IF *Cuaca IS Mendung* THEN *Bermain IS Ya*;
2. IF *Cuaca IS {Cerah, Hujan}* AND *Suhu IS > 75* THEN *Bermain IS Tidak*;
3. IF *Cuaca IS {Cerah, Hujan}* AND *Suhu IS ≤ 75* AND *Angin IS Biasa* AND *Kelembaban ≤ 80* THEN *Bermain IS Ya*;
4. IF *Cuaca IS Cerah* AND *Suhu IS ≤ 75* AND *Angin IS Biasa* AND *Kelembaban IS > 80* THEN *Bermain IS Ya*;
5. IF *Cuaca IS Hujan* AND *Suhu IS ≤ 75* AND *Angin IS Biasa* AND *Kelembaban > 80* THEN *Bermain IS Tidak*;

6. IF Cuaca IS Cerah AND Suhu IS ≤ 75 AND Angin IS Kencang THEN Bermain IS Tidak;
7. IF Cuaca IS Hujan AND Suhu IS ≤ 75 AND Angin IS Kencang THEN Bermain IS Ya;



Contoh 9.4 C4.5: Klasifikasi (Matlab)

Dataset : *dsHeartDiseaseCleveland – Class = {0,1,2,3,4}* (terlampir)
Data Validation : *10-Fold Cross Validation*
Evaluation : *Confusion Matrix*

Fungsi “C45aa.m”

```

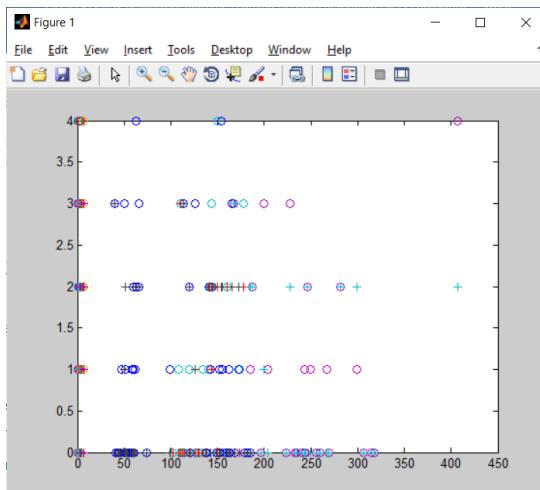
function [output, lamaProses, akurasi] = C45aa(dataLatihInput,
dataLatihOutput, dataUjiInput, dataUjiOutput)
tic;
C45model = ClassificationTree.fit(dataLatihInput,
dataLatihOutput); %C45 Training
output = predict(C45model, dataUjiInput); %C45 Testing
conMat = confusionmat(dataUjiOutput, output); %C45 Evaluasi:
jmlData = sum(conMat(:));
hasilBenar = sum(diag(conMat));
akurasi = 100 * (hasilBenar / jmlData);
lamaProses = toc;
plot(dataUjiInput,dataUjiOutput,'o',dataUjiInput,output','+')
end
  
```

Script "C45.m"

```

clc; clear; close all; warning off all;
dsHeartDisease = xlsread('D:\aa Book\Computing
Methods\Book2\dataset\datasets.xlsx','dsHeartDiseaseCleveland');
dataset = dsHeartDisease(:,2:15);
C45hasilSub = []; %kolom: 1 lama proses, 2 akurasi
C45outputAll.x = []; %Output C45 di setiap K dr K-Fold
C45hasil = []; %kolom: 1 lama proses, 2 acc, 3 acc max, 4 acc min
%% K-Fold Cross Validation
K=10;
indeks = crossvalind('Kfold', dataset(:,14), K);
for i = 1:K
    uji = (indeks == i);
    latih = ~uji; %indeks latih = yg bukan indeks uji
    subDataLatihInput = dataset(latih,1:13); %input training
    subDataLatihOutput = dataset(latih,14); %output training
    subDataUjiInput = dataset(uji,1:13); % input testing
    subDataUjiOutput = dataset(uji,14); % output testing
    %% C45 Modelling di tiap K
    [C45subOutput, C45subLamaProses, C45subAkurasi] =
    C45aa(subDataLatihInput, subDataLatihOutput, subDataUjiInput,
    subDataUjiOutput);
    C45hasilSub(i,1) = C45subLamaProses; % hasil lama proses dalam K
    C45hasilSub(i,2) = C45subAkurasi; % hasil akurasi dalam K
    C45outputAll(i).x = [subDataUjiInput subDataUjiOutput
    C45subOutput];
end
%% Hasil Akhir C45
C45hasil(1)=mean(C45hasilSub(:,1)); % rata2 lama proses
C45hasil(2)=mean(C45hasilSub(:,2)); % akurasi akhir
C45hasil(3)=max(C45hasilSub(:,2)); % akurasi max
C45hasil(4)=min(C45hasilSub(:,2)); % akurasi min

```



$$\text{Lama proses} = 0,0919 \text{ detik}$$

$$\text{Akurasi minimum} = 41,9355\%$$

$$\text{Akurasi maksimum} = 63,3333\%$$

$$\text{Akurasi} = 52,1706\%$$

9.2 Linear Regression

9.2.1 Linear Regression dengan 1 Variabel Bebas

Selain ANN dan SVM, *Linear Regression* juga merupakan salah satu algoritma *machine learning* yang dapat digunakan untuk menangani masalah-masalah estimasi/regresi. Persamaan *Linear Regression* dengan satu atribut (variabel bebas/input) didefinisikan sebagai berikut.

$$y' = a + bX \quad (159)$$

Konstanta a dan koefisien b diperoleh melalui persamaan berikut ini.

$$b = \frac{n \sum XY - \sum X \sum Y}{n \sum X^2 - (\sum X)^2} \quad (160)$$

$$a = \frac{\sum Y - b \sum X}{n} \quad (161)$$

Nilai positif pada koefisien b menunjukkan bahwa antara variabel bebas dengan variabel terikat (variabel output) berjalan satu arah, yang mana setiap penurunan atau peningkatan variabel bebas akan diikuti dengan peningkatan atau penurunan variabel terikatnya. Sementara nilai negatif pada koefisien b menunjukkan bahwa antara variabel bebas dengan variabel terikat berjalan dua arah, yang mana setiap peningkatan variabel bebas akan diikuti dengan penurunan variabel terikatnya, dan sebaliknya.

Contoh 9.5 Linear Regression: Estimasi dengan 1 Variabel Bebas (Manual)

No	X	Y
1.	5	20000
2.	6	25000
3.	3	15000
4.	6	27000
5.	4	17500
6.	2	10000
7.	1	7500

Hitung X^2 , Y^2 , dan XY , dan total dari tiap-tiap variabel

No	X	Y	X^2	Y^2	XY
1.	5	20000	25	400000000	100000
2.	6	25000	36	625000000	150000
3.	3	15000	9	225000000	45000
4.	6	27000	36	729000000	162000
5.	4	17500	16	306250000	70000
6.	2	10000	4	100000000	20000
7.	1	7500	1	56250000	7500
Jumlah	27	122000	127	2441500000	554500

Hitung konstanta a dan koefisien b :

$$b = \frac{n \sum XY - \sum X \sum Y}{n \sum X^2 - (\sum X)^2} = \frac{(7 * 554500) - (27 * 122000)}{(7 * 127) - (27)^2} = 3671,875$$

$$a = \frac{\sum Y - b \sum X}{n} = \frac{(122000) - (3671,875 * 27)}{7} = 3265,625$$

Dengan demikian, model persamaan *Linear Regression* yang diperoleh adalah:

$$y' = 3265,626 + 3671,875(X)$$

Misalnya, berapa nilai Y jika nilai $X = 7$?

$$y' = 3265,626 + (3671,875 * 7) = 28968,75$$

Maka hasil prediksi/estimasi $X = 7$ adalah 28968,75, dibulatkan menjadi 28969.

Selanjutnya, bagaimana jika yang diketahui adalah nilai $Y = 30000$, maka $X = ?$

$$30000 = 3265,626 + 3671,875(X)$$

$$3671,875(X) = 30000 - 3265,626$$

$$X = \frac{26734,37}{3671,875} = 7,280851$$

Maka hasil prediksi/estimasi $Y = 30000$ adalah 7,280851, dibulatkan menjadi 7.

Selanjutnya dapat dilakukan uji koefisien korelasi menggunakan Persamaan (54) dan pengujian estimasi *error* menggunakan salah satu metode pengujianya, seperti MSE (32), RMSE (33), dll. Baca kembali bab tentang evaluasi model, khususnya terkait dengan pengujian korelasi variabel dan estimasi *error*.

9.2.2 Linear Regression dengan 2 Variabel Bebas

Ketika terdapat 2 variabel bebas yang diolah, maka persamaan *Linear Regression* didefinisikan sebagai berikut.

$$y' = a + b_1 X_1 + b_2 X_2 \quad (162)$$

Konstanta a serta koefisien b_1 dan b_2 diperoleh melalui persamaan berikut ini.

$$b_1 = \frac{[(\sum x_2^2 \sum x_1 y) - (\sum x_2 y \sum x_1 x_2)]}{[(\sum x_1^2 \sum x_2^2) - (\sum x_1 x_2)^2]} \quad (163)$$

$$b_2 = \frac{[(\sum x_1^2 \sum x_2 y) - (\sum x_1 y \sum x_1 x_2)]}{[(\sum x_1^2 \sum x_2^2) - (\sum x_1 x_2)^2]} \quad (164)$$

$$a = \frac{(\sum Y) - (b_1 \sum x_1) - (b_2 \sum x_2)}{n} \quad (165)$$

Keterangan:

$$\sum x_1^2 = \sum x_1^2 - \frac{(\sum X_1)^2}{n}$$

$$\sum x_2^2 = \sum x_2^2 - \frac{(\sum X_2)^2}{n}$$

$$\sum y^2 = \sum Y^2 - \frac{(\sum Y)^2}{n}$$

$$\sum x_1y = \sum X_1Y - \frac{\sum X_1 \sum Y}{n}$$

$$\sum x_2y = \sum X_2Y - \frac{\sum X_2 \sum Y}{n}$$

$$\sum x_1x_2 = \sum X_1X_2 - \frac{\sum X_1 \sum X_2}{n}$$

Contoh 9.6 Linear Regression: Estimasi dengan 2 Variabel Bebas (Manual)

Tahun	X ₁	X ₂	Y
1990	4.90	6.47	8300
1991	3.28	3.14	7500
1992	5.05	5.00	8950
1993	4.00	4.75	8250
1994	5.97	6.23	9000
1995	4.24	6.03	8750
1996	8.00	8.75	10000
1997	7.45	7.72	8200
1998	7.47	8.00	8300
1999	12.68	10.40	10900
2000	14.45	12.42	12800
2001	10.50	8.62	9450
2002	17.24	12.07	13000
2003	15.56	5.83	8000
2004	10.85	5.20	6500
2005	16.56	8.53	9000
2006	13.24	7.37	7600
2007	16.98	9.38	10200

Hitung X₁², X₂², Y², X₁X₂, X₁Y, X₂Y, dan total dari tiap-tiap variabel.

Thn	X ₁	X ₂	Y	X ₁ ²	X ₂ ²	Y ²	X ₁ X ₂	X ₁ Y	X ₂ Y
1990	4,90	6,47	8300	24,01	41,86	68890000	31,70	40670,00	53701,00
1991	3,28	3,14	7500	10,76	9,86	56250000	10,30	24600,00	23550,00
1992	5,05	5,00	8950	25,50	25,00	80102500	25,25	45197,50	44750,00
1993	4,00	4,75	8250	16,00	22,56	68062500	19,00	33000,00	39187,50
1994	5,97	6,23	9000	35,64	38,81	81000000	37,19	53730,00	56070,00
1995	4,24	6,03	8750	17,98	36,36	76562500	25,57	37100,00	52762,50
1996	8,00	8,75	10000	64,00	76,56	100000000	70,00	80000,00	87500,00
1997	7,45	7,72	8200	55,50	59,60	67240000	57,51	61090,00	63304,00
1998	7,47	8,00	8300	55,80	64,00	68890000	59,76	62001,00	66400,00
1999	12,68	10,40	10900	160,78	108,16	118810000	131,87	138212,00	113360,00
2000	14,45	12,42	12800	208,80	154,26	163840000	179,47	184960,00	158976,00
2001	10,50	8,62	9450	110,25	74,30	89302500	90,51	99225,00	81459,00
2002	17,24	12,07	13000	297,22	145,68	169000000	208,09	224120,00	156910,00
2003	15,56	5,83	8000	242,11	33,99	64000000	90,71	124480,00	46640,00
2004	10,85	5,20	6500	117,72	27,04	42250000	56,42	70525,00	33800,00
2005	16,56	8,53	9000	274,23	72,76	81000000	141,26	149040,00	76770,00
2006	13,24	7,37	7600	175,30	54,32	57760000	97,58	100624,00	56012,00
2007	16,98	9,38	10200	288,32	87,98	104040000	159,27	173196,00	95676,00
Jml	178,42	135,91	164700	2179,93	1133,11	1557000000	1491,47	1701770,50	1306828,00

Hitung konstanta a dan koefisien b_1 dan b_2 :

$$\sum x_1^2 = 2179,93 - \frac{178,42^2}{18} = 411,39$$

$$\sum x_2^2 = 1133,11 - \frac{135,91^2}{18} = 106,92$$

$$\sum y^2 = 1557000000 - \frac{164700^2}{18} = 49995000$$

$$\sum x_1y = 1701770,50 - \frac{178,42 * 164700}{18} = 69227,50$$

$$\sum x_2y = 1306828,00 - \frac{135,91 * 164700}{18} = 63251,50$$

$$\sum x_1x_2 = 1491,47 - \frac{178,42 * 135,91}{18} = 144,30$$

$$b_1 = \frac{(106,92 * 69227,50) - (63251,50 * 144,30)}{(411,39 * 106,92) - 144,30^2} = -74,49$$

$$b_2 = \frac{(411,39 * 63251,50) - (69227,50 * 144,30)}{(411,39 * 106,92) - 144,30^2} = 692,11$$

$$a = \frac{164700 - (-74,49 * 178,42) - (692,11 * 135,91)}{18} = 4662,55$$

Dengan demikian, model persamaan *Linear Regression* yang diperoleh adalah:

$$y' = 4662,55 - 74,49(X_1) + 692,11(X_2)$$

Misalnya, berapa nilai Y jika nilai $X_1 = 20$ dan $X_2 = 15$?

$$y' = 4662,55 - (74,49 * 20) + (692,11 * 15) = 13554,4$$

Maka hasil prediksi/estimasi $X_1 = 20$ dan $X_2 = 15$ adalah 13554,4.

Nilai $a = 4662,55$, berarti jika $X_1 = 0$ dan $X_2 = 0$, maka $Y = 4662,55$.

Nilai $b_1 = -74,49$, berarti jika X_1 mengalami peningkatan sebesar 1, maka akan terjadi penurunan Y sebesar 74,49, yang mana X_2 dianggap tetap.

Nilai $b_2 = 692,11$ artinya jika X_2 mengalami peningkatan sebesar 1, maka akan terjadi peningkatan Y sebesar 692,11, yang mana X_1 dianggap tetap.

Selanjutnya dapat dilakukan uji koefisien korelasi secara parsial, uji koefisien korelasi secara simultan, uji estimasi *error*, uji regresi secara parsial (*T-Test*), dan uji regresi secara simultan (*F-Test*). Baca kembali bab tentang evaluasi model, khususnya terkait dengan pengujian korelasi variabel dan estimasi *error*.

9.2.3 Linear Regression dengan 3 atau Lebih Variabel Bebas

Ketika variabel bebas yang diolah > 2, maka nilai konstanta dan regresi setiap variabel bebas dapat diperoleh dengan menggunakan pendekatan matriks determinan yang didefinisikan sebagai berikut.

$$A = \begin{bmatrix} n & \sum X_1 & \sum X_2 & \sum X_{...} & \sum X_n \\ \sum X_1 & \sum (X_1 X_1) & \sum (X_1 X_2) & \sum (X_1 X_{...}) & \sum (X_1 X_n) \\ \sum X_2 & \sum (X_2 X_1) & \sum (X_2 X_2) & \sum (X_2 X_{...}) & \sum (X_2 X_n) \\ \sum X_{...} & \sum (X_{...} X_1) & \sum (X_{...} X_2) & \sum (X_{...} X_{...}) & \sum (X_{...} X_n) \\ \sum X_n & \sum (X_n X_1) & \sum (X_n X_2) & \sum (X_n X_{...}) & \sum (X_n X_n) \end{bmatrix} \quad (166)$$

$$b = \begin{bmatrix} a \\ b_1 \\ b_2 \\ b_{...} \\ b_n \end{bmatrix} \quad (167)$$

$$H = \begin{bmatrix} \sum Y \\ \sum (X_1 Y) \\ \sum (X_2 Y) \\ \sum (X_{...} Y) \end{bmatrix} \quad (168)$$

$$A_o = \begin{bmatrix} \sum Y & \sum X_1 & \sum X_2 & \sum X_{...} & \sum X_n \\ \sum (X_1 Y) & \sum (X_1 X_1) & \sum (X_1 X_2) & \sum (X_1 X_{...}) & \sum (X_1 X_n) \\ \sum (X_2 Y) & \sum (X_2 X_1) & \sum (X_2 X_2) & \sum (X_2 X_{...}) & \sum (X_2 X_n) \\ \sum (X_{...} Y) & \sum (X_{...} X_1) & \sum (X_{...} X_2) & \sum (X_{...} X_{...}) & \sum (X_{...} X_n) \\ \sum (X_n Y) & \sum (X_n X_1) & \sum (X_n X_2) & \sum (X_n X_{...}) & \sum (X_n X_n) \end{bmatrix} \quad (169)$$

$$A_1 = \begin{bmatrix} n & \sum Y & \sum X_2 & \sum X_{..} & \sum X_n \\ \sum X_1 & \sum (X_1 Y) & \sum (X_1 X_2) & \sum (X_1 X_{..}) & \sum (X_1 X_n) \\ \sum X_2 & \sum (X_2 Y) & \sum (X_2 X_2) & \sum (X_2 X_{..}) & \sum (X_2 X_n) \\ \sum X_{..} & \sum (X_{..} Y) & \sum (X_{..} X_2) & \sum (X_{..} X_{..}) & \sum (X_{..} X_n) \\ \sum X_n & \sum (X_n Y) & \sum (X_n X_2) & \sum (X_n X_{..}) & \sum (X_n X_n) \end{bmatrix} \quad (170)$$

$$A_2 = \begin{bmatrix} n & \sum X_1 & \sum Y & \sum X_{..} & \sum X_n \\ \sum X_1 & \sum (X_1 X_1) & \sum (X_1 Y) & \sum (X_1 X_{..}) & \sum (X_1 X_n) \\ \sum X_2 & \sum (X_2 X_1) & \sum (X_2 Y) & \sum (X_2 X_{..}) & \sum (X_2 X_n) \\ \sum X_{..} & \sum (X_{..} X_1) & \sum (X_{..} Y) & \sum (X_{..} X_{..}) & \sum (X_{..} X_n) \\ \sum X_n & \sum (X_n X_1) & \sum (X_n Y) & \sum (X_n X_{..}) & \sum (X_n X_n) \end{bmatrix} \quad (171)$$

$$A_{..} = \begin{bmatrix} n & \sum X_1 & \sum X_2 & \sum Y & \sum X_n \\ \sum X_1 & \sum (X_1 X_1) & \sum (X_1 X_2) & \sum (X_1 Y) & \sum (X_1 X_n) \\ \sum X_2 & \sum (X_2 X_1) & \sum (X_2 X_2) & \sum (X_2 Y) & \sum (X_2 X_n) \\ \sum X_{..} & \sum (X_{..} X_1) & \sum (X_{..} X_2) & \sum (X_{..} Y) & \sum (X_{..} X_n) \\ \sum X_n & \sum (X_n X_1) & \sum (X_n X_2) & \sum (X_n Y) & \sum (X_n X_n) \end{bmatrix} \quad (172)$$

$$A_4 = \begin{bmatrix} n & \sum X_1 & \sum X_2 & \sum X_{..} & \sum Y \\ \sum X_1 & \sum (X_1 X_1) & \sum (X_1 X_2) & \sum (X_1 X_{..}) & \sum (X_1 Y) \\ \sum X_2 & \sum (X_2 X_1) & \sum (X_2 X_2) & \sum (X_2 X_{..}) & \sum (X_2 Y) \\ \sum X_{..} & \sum (X_{..} X_1) & \sum (X_{..} X_2) & \sum (X_{..} X_{..}) & \sum (X_{..} Y) \\ \sum X_n & \sum (X_n X_1) & \sum (X_n X_2) & \sum (X_n X_{..}) & \sum (X_n Y) \end{bmatrix} \quad (173)$$

Setelah perhitungan determinasi untuk matriks A , A_0 , A_1 , A_2 , $A_{..}$, dan A_4 dilakukan, selanjutnya dapat dihitung nilai konstanta a dan koefisien $\{b_1, b_2, b_3, b_4\}$ yang didefinisikan sebagai berikut.

$$a = \frac{A_0}{A} \quad (174)$$

$$b_i = \frac{A_i}{A} \quad (175)$$

Contoh 9.7 Linear Regression: Estimasi dengan 3 Variabel Bebas (Manual)

Tahun	Angka Kematian (X1)	Angka Kelahiran (X2)	Migrasi Masuk (X3)	Migrasi Keluar (X4)	Jumlah Penduduk (Y)
2007	82440	77845	3963822	2919062	225600000
2008	88441	82177	4680121	4077420	228500000
2009	81943	84343	5038270	4656599	231400000
2010	99615	85426	5396419	5235778	237600000
2011	85567	85968	5104908	4847978	242000000
2012	84214	86509	4959153	4654077	245400000
2013	84891	88639	4886275	4557127	248800000
2014	84308	89703	4849836	4508652	252200000
2015	83725	90768	4813397	4460177	255500000

$$n = 9$$

$$\sum X_1 = 775144$$

$$\sum X_2 = 771378$$

$$\sum X_3 = 43692201$$

$$\sum X_4 = 39916870$$

$$\sum Y = 2167000000$$

$$\sum (X_1 X_1) = 66993873210$$

$$\sum (X_1 X_2) = 66434564751$$

$$\sum (X_1 X_3) = 3772231486252$$

$$\sum (X_1 X_4) = 3451564485129$$

$$\sum (X_2 X_2) = 66240880938$$

$$\sum (X_2 X_3) = 3752031208261$$

$$\sum (X_2 X_4) = 3434938281957$$

$$\sum (X_3 X_3) = 213339587486789$$

$$\sum (X_3 X_4) = 195800078762567$$

$$\sum (X_4 X_4) = 180395413961304$$

$$\sum Y^2 = 522679020000000000$$

$$\sum(X_1Y) = 186585792200000$$

$$\sum (X_2 Y) = 18605473270000$$

$$\sum (X_3 Y) = 10532711399000000$$

$$\sum (X_4 Y) = 9637580345900000$$

Matriks determinan:

$$\begin{bmatrix} 9 & 775144 & 771378 & 43692201 & 39916870 \\ 775144 & 66993873210 & 66434564751 & 3772231486252 & 3451564485129 \\ 771378 & 66434564751 & 66240880938 & 3752031208261 & 3434938281957 \\ 43692201 & 3772231486252 & 3752031208261 & 213339587486789 & 195800078762567 \\ 39916870 & 3451564485129 & 3434938281957 & 195800078762567 & 180395413961304 \end{bmatrix}$$

2167000000	775144	771378	43692201	39916870
186585792200000	66993873210	66434564751	3772231486252	3451564485129
186054732700000	66434564751	66240880938	3752031208261	3434938281957
10532711399000000	3772231486252	3752031208261	213339587486789	195800078762567
9637580345900000	3451564485129	3434938281957	195800078762567	180395413961304

$$A_1 = 13776306640040900$$

9	2167000000	771378	43692201	39916870
775144	186585792200000	66434564751	3772231486252	3451564485129
771378	186054732700000	66240880938	3752031208261	3434938281957
43692201	10532711399000000	3752031208261	213339587486789	195800078762567
39916870	9637580345900000	3434938281957	195800078762567	180395413961304

$$A_2 = 634212483216400$$

9	775144	2167000000	43692201	39916870
775144	66993873210	186585792200000	3772231486252	3451564485129
771378	66434564751	186054732700000	3752031208261	3434938281957
43692201	3772231486252	10532711399000000	213339587486789	195800078762567
39916870	3451564485129	9637580345900000	195800078762567	180395413961304

9	775144	771378	2167000000	39916870
775144	66993873210	66434564751	186585792200000	3451564485129
771378	66434564751	66240880938	186054732700000	3434938281957
43692201	3772231486252	3752031208261	10532711399000000	195800078762567
39916870	3451564485129	3434938281957	9637580345900000	180395413961304

$$A_4 = 447330647600511000$$

9	775144	771378	43692201	2167000000
775144	66993873210	66434564751	3772231486252	186585792200000
771378	66434564751	66240880938	3752031208261	186054732700000
43692201	3772231486252	3752031208261	213339587486789	10532711399000000
39916870	3451564485129	3434938281957	195800078762567	96375803459000000

Konstanta a dan koefisien b :

Dengan demikian, persamaan *Linear Regression* yang diperoleh adalah:

$$y' = 415313773,86 + 268,39(X_1) + 1235,58(X_2) - 142,15(X_3) + 87,15(X_4)$$

Misalnya, berapa nilai Y jika nilai $X_1 = 70000, X_2 = 60000, X_3 = 2$ juta, $X_4 = 1$ juta?

$$y' = 415313773,86 + (268,39 * 70000) + (1235,58 * 60000) - (142,15 * 2000000) + (87,15 * 1000000) = 879679921$$

Maka hasil prediksi/estimasi jumlah penduduk, jika angka kematian = 70000, angka kelahiran = 60000, migrasi masuk = 2 juta, dan migrasi keluar = 1 juta adalah 879679921 penduduk.

Selanjutnya dapat dilakukan uji koefisien korelasi secara parsial, uji koefisien korelasi secara simultan, uji estimasi *error*, uji regresi secara parsial (*T-Test*), dan uji regresi secara simultan (*F-Test*). Baca kembali bab tentang evaluasi model, khususnya terkait dengan pengujian korelasi variabel dan estimasi *error*.

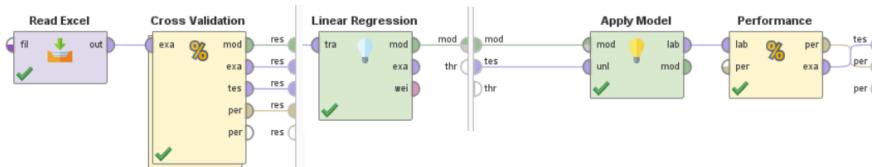
Contoh 9.8 Linear Regression: Estimasi (Rapidminer)

Dataset : *dsPanganTimeSeries* (terlampir)

Data validation : 10-Fold Cross Validation

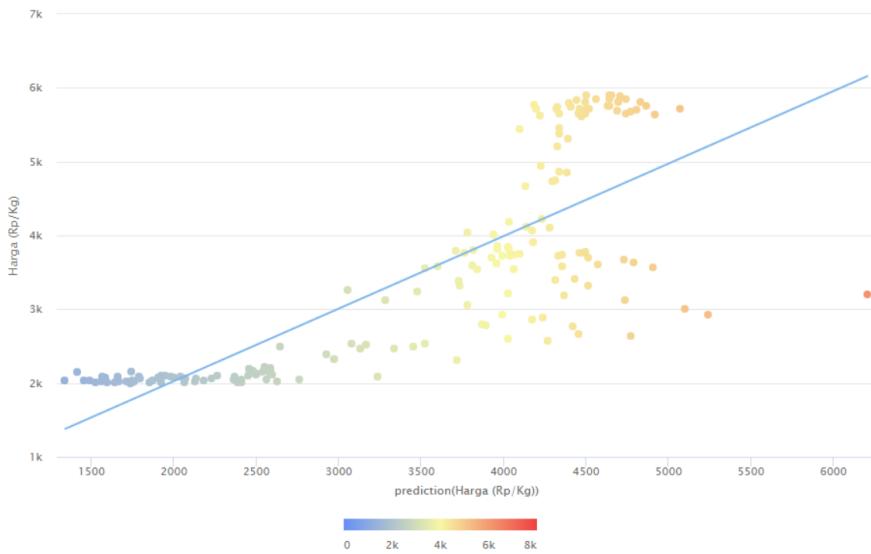
Neuron HD : 50

Evaluation : RMSE



Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat
Luas lahan (Ha)	-0.038	0.008	-0.948	0.049	-4.636
Jml Produksi (Ton)	0.012	0.001	1.692	0.049	8.277
(Intercept)	1864.892	284.784	?	?	6.548

RMSE: 854.333 +/- 154.273 (micro average: 866.779 +/- 0.000)



9.3 A-Priori

Algoritma *A-Priori* merupakan salah satu algoritma *machine learning* yang dapat digunakan untuk menangani masalah asosiasi. Selain *A-Priori*, algoritma yang juga umum digunakan untuk asosiasi adalah *FP-Growth*. Persoalan pada asosiasi hanya memiliki satu atribut dan tidak memiliki variabel output. Analisis asosiasi bekerja dengan cara mencari hubungan asosiasi (bukan korelasi) antar item data dalam suatu atribut yang diolah untuk menghasilkan model aturan asosiasi yang dapat digunakan sebagai pengetahuan untuk mendukung pengambilan keputusan.

Algoritma *A-Priori* merupakan algoritma yang perhitungannya relatif mudah, namun membutuhkan kompleksitas komputasi yang cukup besar. *Support* suatu item data *A* dapat didefinisikan sebagai berikut.

$$S(A) = \frac{F(A)}{n} \quad (176)$$

S(A) menyatakan nilai *Support* item data *A*. *F(A)* menyatakan jumlah (frekuensi) data yang mengandung item data *A*, sedangkan *n* menyatakan jumlah transaksi/data.

Selanjutnya kombinasi dua set item data dapat dibentuk dengan menggunakan pendekatan kombinasi yang didefinisikan pada Persamaan (177), sedangkan *Support* dari kombinasi 2 set item data didefinisikan pada Persamaan (178).

$$\text{comb}(m, r) = \frac{m!}{r! (m - r)!} \quad (177)$$

$$S(A \rightarrow B) = \frac{F(A \cup B)}{n} \quad (178)$$

Notasi m menyatakan banyaknya item data yang akan dikombinasikan, r menyatakan kombinasi berapa yang akan dibentuk, $m!$ menyatakan faktorial dari m . $S(A \rightarrow B)$ menyatakan nilai *Support* dari set item data A dan B , $F(A \cup B)$ menyatakan jumlah (frekuensi) data yang mengandung set item data A dan B , sedangkan n menyatakan jumlah data.

Selanjutnya *Confidence* dari 2 set item data didefinisikan sebagai berikut.

$$C(A \rightarrow B) = \frac{S(A \rightarrow B)}{S(B)}; \text{ OR } \frac{F(A \cup B)}{F(A)} \quad (179)$$

Secara sederhana, kinerja suatu model asosiasi dapat diukur berdasarkan nilai *Lift Ratio* dari aturan-aturan asosiasi yang diperoleh berdasarkan *Support* dan *Confidence*. *Lift Ratio* mengukur seberapa penting suatu aturan asosiasi, seberapa erat asosiasi antara set item data A dan B dalam suatu aturan asosiasi. *Lift* dari 2 set item data didefinisikan sebagai berikut.

$$\text{lift}(A \rightarrow B) = \frac{S(A \rightarrow B)}{S(A)S(B)} \quad (180)$$

Atau jika A dan B dianggap independen, maka:

$$\text{lift}(A \rightarrow B) = \frac{C(A \rightarrow B)}{S(B)} \quad (181)$$

Rentang nilai *Lift* adalah dari 0 hingga positif tak terhingga. Jika nilai *Lift* mendekati 1, maka hubungan antara set item X_1 dan X_2 pada aturan asosiasi tersebut mungkin independen. Ketika dua set item saling independen pada suatu aturan asosiasi, maka sebenarnya tidak ada aturan yang dapat ditarik atas kedua set item tersebut. Namun jika nilai $\text{lift} > 1$, maka aturan asosiasi tersebut berpotensi berguna digunakan untuk prediksi atau untuk mendukung pengambilan keputusan. Artinya, jika *Lift* semakin > 1 , maka aturan asosiasi tersebut semakin dapat dipercaya [26].

Contoh 9.9 A-Priori: Asosiasi (Manual)

Misalnya terdapat 6 item data $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ dengan 10 data transaksi yang terjadi dari item-item tersebut adalah sebagai berikut:

Transaksi (data) ke-1	=	$\{x_1, x_4, x_6\}$
Transaksi (data) ke-2	=	$\{x_1, x_3, x_4, x_6\}$
Transaksi (data) ke-3	=	$\{x_1, x_6\}$
Transaksi (data) ke-4	=	$\{x_2, x_1\}$
Transaksi (data) ke-5	=	$\{x_2, x_3, x_6\}$
Transaksi (data) ke-6	=	$\{x_3, x_4\}$
Transaksi (data) ke-7	=	$\{x_3, x_5, x_6\}$
Transaksi (data) ke-8	=	$\{x_2, x_4, x_5, x_6\}$
Transaksi (data) ke-9	=	$\{x_3, x_6\}$
Transaksi (data) ke-10	=	$\{x_6, x_3, x_4, x_1\}$

Misalnya *minimum Support* dan *minimum Confidence* adalah sebagai berikut:

$$\begin{aligned} \text{Minimum Support} &= 0,3 \\ \text{Minimum Confidence} &= 0,7 \end{aligned}$$

Minimum Support digunakan untuk menghentikan proses iterasi *A-Priori*, yang mana jika tidak ada lagi kombinasi set item data yang memenuhi *minimum Support* maka proses iterasi berhenti dan dilanjutkan ke pembentukan aturan asosiasi. Sedangkan *minimum Confidence* digunakan untuk menentukan aturan asosiasi, yang mana kombinasi set item data yang memenuhi *minimum Confidence* yang digunakan sebagai aturan asosiasi.

No Transaksi	x1	x2	x3	x4	x5	x6
1.	1	0	0	1	0	1
2.	1	0	1	1	0	1
3.	1	0	0	0	0	1
4.	1	1	0	0	0	0
5.	0	1	1	0	0	1
6.	0	0	1	1	0	0
7.	0	0	1	0	1	1
8.	0	1	0	1	1	1
9.	0	0	1	0	0	1
10.	1	0	1	1	0	1

Iterasi pertama:

1L	x1	x2	x3	x4	x5	x6
F	5	3	6	5	2	8
S	0,5	0,3	0,6	0,5	0,2	0,8
Ket	Yes	Yes	Yes	Yes	No	Yes

Set item data yang memenuhi *minimum Support* adalah $\{S(x_1), S(x_2), S(x_3), S(x_4), S(x_6)\}$. Sehingga diperoleh 5 item $\{x_1, x_2, x_3, x_4, x_6\}$ yang akan dikombinasikan. Selanjutnya dilakukan kombinasi 2 dari 5 item tersebut menggunakan Persamaan (177), sehingga diperoleh 10 kombinasi set item data sebagai berikut.

Iterasi kedua:

2L	2L1		2L2		2L3		2L4		2L5		2L6		2L7		2L8		2L9		2L10												
	x1	x2	F	x1	x3	F	x1	x4	F	x1	x6	F	x2	x3	F	x2	x4	F	x2	x6	F	x3	x4	F	x3	x6	F	x4	x6	F	
1	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1			
2	1	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1			
3	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0			
4	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0			
5	0	1	0	0	1	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	0			
6	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0			
7	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	0	1			
8	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	1				
9	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	0	1			
10	1	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1				
F				1			2			3			4			1			1			2			3			5			4
S				0,10			0,20			0,30			0,40			0,10			0,10			0,20			0,30			0,50			0,40
Ket	No	No		Yes		Yes			No			No			No			Yes			Yes			Yes			Yes			Yes	

Kombinasi set item data yang memenuhi *minimum Support* adalah $\{S(x1 \rightarrow x4), S(x1 \rightarrow x6), S(x3 \rightarrow x4), S(x3 \rightarrow x6), S(x4 \rightarrow x6)\}$. Sehingga diperoleh 4 item $\{x1, x3, x4, x6\}$. Selanjutnya dilakukan kombinasi 3 dari 4 item tersebut menggunakan Persamaan (177), sehingga diperoleh 4 kombinasi set item data sebagai berikut.

Iterasi ketiga:

3L	3L1				3L2				3L3				3L4			
	x1	x3	x4	F	x1	x3	x6	F	x1	x4	x6	F	x3	x4	x6	F
1	1	0	1	0	1	0	1	0	1	1	1	1	0	1	1	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0
4	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	1	1	0	0	0	1	0	1	0	1	0
6	0	1	1	0	0	1	0	0	0	1	0	0	1	1	0	0
7	0	1	0	0	0	1	1	0	0	0	1	0	1	0	1	0
8	0	0	1	0	0	0	1	0	0	1	1	0	0	1	1	0
9	0	1	0	0	0	1	1	0	0	0	1	0	1	0	1	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F					2				2				3			2
S						0,2				0,2			0,3			0,2
Ket					No				No				Yes			No

Kombinasi set item data yang memenuhi *minimum Support* adalah hanya $S(x1 \cup x4 \cup x6)$. Sehingga diperoleh 3 item $\{x1, x4, x6\}$. Walaupun masih ada kombinasi set item data yang memenuhi *minimum Support*, namun tidak ada lagi kombinasi set item data yang dapat dibentuk, karena kombinasi 4 dari 3 item tersebut tidak dapat dibentuk, sehingga proses iterasi *A-Priori* berhenti dan dilanjutkan dengan menentukan aturan asosiasi berdasarkan *Confidence* dari semua kombinasi set item data yang memenuhi *minimum Support* dalam iterasi pertama hingga terakhir (iterasi ketiga).

Rule	Items	S(A → B)	S(B)	C(A → B)	Lift(A → B) bebas	Lift(A → B)	Ket
1.	$x1 \rightarrow x4$	0,30	0,50	0,60	1,20	1,20	No
2.	$x4 \rightarrow x1$	0,30	0,50	0,60	1,20	1,20	No
3.	$x1 \rightarrow x6$	0,40	0,80	0,50	0,63	1,00	No
4.	$x6 \rightarrow x1$	0,40	0,50	0,80	1,60	1,00	Yes
5.	$x3 \rightarrow x4$	0,30	0,50	0,60	1,20	1,00	No
6.	$x4 \rightarrow x3$	0,30	0,60	0,50	0,83	1,00	No
7.	$x3 \rightarrow x6$	0,50	0,80	0,63	0,78	1,04	No
8.	$x6 \rightarrow x3$	0,50	0,60	0,83	1,39	1,04	Yes
9.	$x4 \rightarrow x6$	0,40	0,80	0,50	0,63	1,00	No
10.	$x6 \rightarrow x4$	0,40	0,50	0,80	1,60	1,00	Yes
11.	$\{x1, x4\} \rightarrow x6$	0,30	0,80	0,38	0,47	1,25	No
12.	$x6 \rightarrow \{x1, x4\}$	0,30	0,30	1,00	3,33	1,25	Yes
13.	$\{x1, x6\} \rightarrow x4$	0,30	0,50	0,60	1,20	1,50	No
14.	$x4 \rightarrow \{x1, x6\}$	0,30	0,40	0,75	1,88	1,50	Yes
15.	$\{x4, x6\} \rightarrow x1$	0,30	0,50	0,60	1,20	1,50	No
16.	$x1 \rightarrow \{x4, x6\}$	0,30	0,40	0,75	1,88	1,50	Yes

Dengan demikian, diperoleh aturan asosiasi berdasarkan kombinasi set item data yang memenuhi *minimum Confidence* sebagai berikut.

Rule	Items	Confidence	Lift (Independen)	Lift	Ket
1	$x6 \rightarrow x1$	0,80	1,60	1,00	Kurang Erat
2	$x6 \rightarrow x3$	0,83	1,39	1,04	Erat
3	$x6 \rightarrow x4$	0,80	1,60	1,00	Kurang Erat
4	$x6 \rightarrow \{x1, x4\}$	1,00	3,33	1,25	Erat
5	$x4 \rightarrow \{x1, x6\}$	0,75	1,88	1,50	Erat
6	$x1 \rightarrow \{x4, x6\}$	0,75	1,88	1,50	Erat

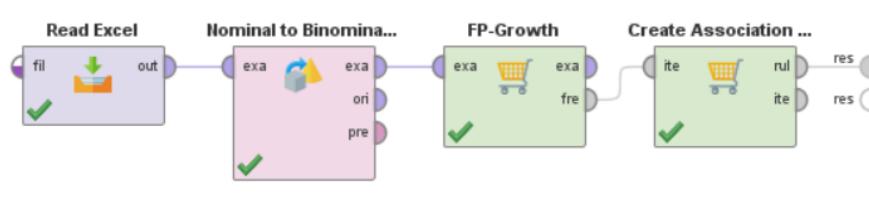
Contoh 9.10 FP-Growth: Asosiasi (Rapidminer)

Dataset : *dsTransaksiPenjualanObat* (terlampir)

Min Support : 0,1

Min Confidence : 0,25

Evaluation : *Confidence* dan *Lift*



Aturan asosiasi yang diperoleh adalah sebagai berikut.

No.	Premises	Conclusion	Support	Confidence	LaPlace	Gain	p-s	Lift	Conviction
4	Obat Ke-4 = Pot	Obat Ke-3 = Sagestam	0.041	0.370	0.937	-0.183	0.035	5.951	1.489
5	Obat Ke-3 = Pot	Obat Ke-2 = Sagestam	0.037	0.391	0.947	-0.154	0.030	5.239	1.520
6	Obat Ke-2 = Desoxymethasone	Obat Ke-3 = Pot	0.021	0.417	0.972	-0.079	0.016	4.366	1.551
7	Obat Ke-1 = Pot	Obat Ke-1 = Catherine	0.021	0.417	0.972	-0.079	0.016	4.782	1.565
8	Obat Ke-2 = Desoxymethasone	Obat Ke-3 = Sagestam	0.021	0.417	0.972	-0.079	0.018	6.694	1.608
9	Obat Ke-2 = Desoxymethasone	Obat Ke-4 = Pot, Obat Ke-3 = Sagestam	0.021	0.417	0.972	-0.079	0.019	10.042	1.643
10	Obat Ke-2 = Desoxymethasone	Obat Ke-4 = Pot	0.025	0.500	0.976	-0.075	0.019	4.463	1.778
11	Obat Ke-2 = Sagestam	Obat Ke-3 = Pot	0.037	0.500	0.965	-0.112	0.030	5.239	1.809
12	Obat Ke-5 = Pot	Obat Ke-4 = Sagestam	0.025	0.500	0.976	-0.075	0.023	17.214	1.942
13	Obat Ke-4 = Pot, Obat Ke-3 = Sagestam	Obat Ke-2 = Desoxymethasone	0.021	0.500	0.980	-0.062	0.019	10.042	1.900
14	Obat Ke-2 = Formyclo	Obat Ke-4 = Pot	0.021	0.556	0.984	-0.054	0.017	4.959	1.998
15	Obat Ke-3 = Sagestam	Obat Ke-4 = Pot	0.041	0.657	0.980	-0.083	0.035	5.951	2.664
16	Obat Ke-1 = Top Cort	Obat Ke-3 = Pot	0.021	0.714	0.992	-0.037	0.018	7.484	3.166
17	Obat Ke-4 = Catherine	Obat Ke-3 = Pot	0.021	0.833	0.996	-0.029	0.018	8.732	5.427
18	Obat Ke-4 = Pot, Obat Ke-2 = Desoxymethasone	Obat Ke-3 = Sagestam	0.021	0.833	0.996	-0.029	0.019	13.389	5.627
19	Obat Ke-4 = Sagestam	Obat Ke-5 = Pot	0.025	0.857	0.996	-0.033	0.023	17.214	6.651
20	Obat Ke-3 = Sagestam, Obat Ke-2 = Desoxymethasone	Obat Ke-4 = Pot	0.021	1	1	-0.021	0.018	8.926	∞

9.4 Soal Latihan C4.5, Linear Regression, & A-Priori

1. Unduh salah satu *dataset classification* pada *UCI Machine Learning Repository*, kemudian lakukan analisis klasifikasi menggunakan algoritma *C4.5*?
2. Unduh salah satu *dataset regression/estimation* pada *UCI Machine Learning Repository*, kemudian lakukan analisis regresi dan korelasi menggunakan algoritma *Linear Regression*?
3. Kumpulkan data transaksi dari salah satu toko di dekat lokasi tempat tinggal anda, kemudian lakukan analisis asosiasi menggunakan algoritma A-Priori?

Daftar Pustaka

- [1] Suyanto, Artificial Intelligence: Searching, Reasoning, Planning, dan Learning, Bandung: Informatika Bandung, 2011.
- [2] Suyanto, Machine Learning: Tingkat Dasar dan Lanjut, Bandung: Informatika Bandung, 2018.
- [3] T. M. Mitchell, Machine Learning, McGraw Hill, 1997.
- [4] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," in *Advances in Neural Information Processing Systems*, 305–313, 1989.
- [5] VisionLab, "Imagenet: Large Scale Visual Recognition Challange," 2017.
- [6] Nuance, "Dragon Speech Recognition Software," 2017.
- [7] V. Dhar, "Data Science and Prediction," *Communications of the ACM*, vol. 56, no. 12, p. 64–73, 2013.
- [8] northwestern.edu, "Masterss in Data Science," 2011. [Online]. Available: <https://sps.northwestern.edu/masters/data-science/index.php>. [Accessed 2019 8 24].
- [9] L. Maimon and O. Rokach, Data Mining and Knowledge Discovery Handbook 2nd Edition, New York: Springer, 2010.
- [10] D. T. Larose and C. D. Larose, Discovering Knowledge in Data: An Introduction to Data Mining 2nd ed, Wiley, 2014.
- [11] D. E. Knuth, The Art of Computer Programming (Vol 1), Addison-Wesley Company, 1973.
- [12] T. W. Parsons, Introduction to Algorithm in Pascal, Johns Wiley and Sons, 1995.
- [13] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithm (Third Edition), Massachusetts Institute of Technology, 2009.
- [14] Microsoft, "MSDN Liblary," 2010. [Online]. Available: <https://msdn.microsoft.com>. [Accessed 25 November 2017].
- [15] Wikipedia, "Struktur Data," 2016. [Online]. Available: https://id.wikipedia.org/wiki/Struktur_data. [Accessed 2016 5 2].

- [16] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, vol. 17, pp. 519-533, 2003.
- [17] S. Zhang, Z. Jin and X. Zhu, "Missing data imputation by utilizing information within incomplete instances," *Journal of Systems and Software*, vol. 83, no. 3, pp. 452-459, 2011.
- [18] P. H. Abreu, M. S. Santos, M. H. Abreu, B. Andrade and D. C. Silva, "Predicting Breast Cancer Recurrence Using Machine Learning Techniques: A Systematic Review," *ACM Computing Surveys*, vol. 49, no. 3, pp. 52:1-52:40, 2016.
- [19] S. Bashir, U. Qamar and F. H. Khan, "Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote based ensemble," *Quality and Quantity*, vol. 49, no. 5, pp. 2061-2076, 2015.
- [20] E. Prasetyo, Data Mining: Konsep dan Aplikasinya Menggunakan Matlab, Yogyakarta: Andi Offset, 2012.
- [21] S. Garcia, J. Luengo, J. A. Saez, V. Lopez and F. Herrera, "A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 734-750, 2013.
- [22] S. Kotsiantis and D. Kanellopoulos, "Discretization Techniques : A Recent Survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47-58, 2006.
- [23] J. W. Grzymala-Busse, "Discretization Based on Entropy and Multiple Scanning," *Entropy*, vol. 15, pp. 1486-1502, 2013.
- [24] Hawkins, Identification of Outliers, London: Chapman and Hall, 1980.
- [25] M. M. Suarez-Alvarez, D.-T. Pham, M. Y. Prostov and Y. I. Prostov, "Statistical approach to normalization of feature vectors and clustering of mixed datasets," in *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2012.
- [26] Rapidminer, "Welcome to RapidMiner Documentation," Rapidminer, 2019. [Online]. Available: <https://docs.rapidminer.com/>. [Accessed 4 11 2019].
- [27] J.-L. Bouchota, W. L. Trimble, G. Ditzler, Y. Lan, S. Essinger and G. Rosen, "Advances in Machine Learning for Processing and Comparison of Metagenomic Data," in *Computational Systems Biology (Second Edition)*, Science Direct, 2014.
- [28] R. T. Asmono, R. S. Wahono and A. Syukur, "Absolute Correlation Weighted Naïve Bayes for Software Defect Prediction," *Journal of Software Engineering*, vol. 1, no. 1, pp. 38-45, 2015.

- [29] R. J. Freund and W. J. Wilson, Statistical Methods (2nd ed.), Academic Press, 2003.
- [30] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, p. 389–422, 2002.
- [31] T. R. Golub, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, no. 5439, p. 531–537, 1999.
- [32] H. Zhang, "Learning Weighted Naive Bayes with Accurate Ranking," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004.
- [33] P. Pavlidis, J. Weston, J. Cai and W. N. Grundy, "Gene functional classification from heterogeneous data," in *Proceedings of the fifth annual international conference on Computational biology - RECOMB*, New York, 2001.
- [34] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, p. 906–914, 2000.
- [35] I. Fakhruzi, "An Artificial Neural Network with Bagging to Address Imbalance Datasets on Clinical Prediction," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018.
- [36] A. Ali, S. M. Shamsuddin and A. L. Ralescu, "Classification with class imbalance problem: A Review," *International Journal of Advances in Soft Computing and its Applications*, vol. 7, no. 3, pp. 176-204, 2015.
- [37] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463-484, 2012.
- [38] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1-39, 2010.
- [39] G. Bucknall and L. D. Tong, Data Structures and Algorithm: Annotated Reference with Examples, DotNetSlackers, 2008.
- [40] Wikipedia, "Cluster Analysis," Wikipedia, 25 10 2019. [Online]. Available: https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/Cluster_analysis&hl=id&sl=en&tl=id&client=srp. [Accessed 8 11 2019].

- [41] Abdurahman, Maman, Muhibin, Sambas, Somantri and Ating, Dasar-Dasar Metode Statistika Untuk Penelitian, Bandung: CV. Pustaka Setia, 2012.
- [42] Siregar and Syofian , Statistik Parametrik untuk Penelitian Kualitatif, Jakarta: Bumi Aksara, 2013.
- [43] T. Sutojo, E. Mulyanto and V. S, Kecerdasan Buatan, Yogyakarta: Andi Offset, 2011.
- [44] H. Xisheng, Z. Wang, C. Jin, Y. Zhen and X. Xue, "A Simplified Multi-Class Support Vector Machine with Reduced Dual Optimization," *Elsevier on Pattern Recognition Letters*, vol. 33, pp. 71-82, 2012.
- [45] A. I. S. Azis, V. Suhartono and H. Himawan, "Model Multi Class SVM Menggunakan Strategi 1V1 untuk Klasifikasi Wall Following Robot Navigation Data," *Jurnal Teknologi Informasi*, vol. 13, no. 2, pp. 170-187, 2017.
- [46] H. Chih-Wei and C.-J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Trans. Neural Netw*, pp. 415-425, 2002.
- [47] L. Jiang, C. Li, S. Wang and L. Zhang, "Deep feature weighting for naive Bayes and its application to text classification," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 26–39,, 2016.
- [48] J. M. Keller, M. R. Gray and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 4, pp. 580-585, 1985.
- [49] I. Abbas and A. I. S. Azis, "Integrasi Algoritma Singular Value Decomposition (SVD) dan Principal Component Analysis (PCA) untuk Pengurangan Dimensi pada Data Rekam Medis," *ILKOM*, 2014.
- [50] K. Crammer and Y. Singer, "On The Algorithmic Implementation of Multi-Class Kernel-Based Vector Machines," *Journal of Machine Learning Research*, vol. 2, pp. 265-292, 2001.

Glosarium

Algoritma, berasal dari kata *algorism* yang sering dihubungkan dengan *arithmetic*, sehingga lambat laun menjadi *algorithm* adalah serangkaian prosedur yang saling berinteraksi untuk mencapai suatu tujuan dalam memecahkan masalah tertentu. Algoritma dapat ditulis dalam bentuk **bahasa alamiah, pseudocode, flowchart, bagan drakon**, atau pun **bahasa pemrograman**. Dalam ilmu komputer dan matematika, algoritma dapat pula diistilahkan sebagai **metode komputasi**. Sedangkan dalam pemrograman komputer, algoritma dapat diistilahkan sebagai *method (function/procedure)*.

Bahasa pemrograman adalah suatu jenis aplikasi komputer (alat bantu) yang digunakan dalam **pemrograman komputer** yang merupakan kegiatan mengembangkan **program komputer** yang terdiri dari serangkaian **algoritma**.

Data adalah entitas/kumpulan/rekaman/catatan dari fakta, transaksi, atau objek tentang suatu kejadian yang belum membawa/memiliki arti, meskipun kemungkinan memiliki nilai di dalamnya. **Informasi** adalah data yang telah diolah sehingga memiliki nilai/makna/arti. **Pengetahuan** adalah gabungan dari informasi yang bertujuan untuk memberikan suatu informasi yang baru atau solusi pemecahan suatu masalah.

Struktur data adalah wadah (cara penyimpanan, penyusunan, dan pengaturan yang efisien) dari data yang diolah/dimanipulasi oleh **algoritma**.

Artificial Intelligence (Kecerdasan Buatan) merupakan salah satu disiplin ilmu yang dapat membuat program komputer secara otomatis semakin cerdas. Metode-metodenya diistilahkan **Soft Computing** yang pada prinsipnya terdiri dari pendekatan pencarian (*searching*), optimasi (*optimization*), penalaran (*reasoning*), dan pembelajaran (*learning*). **Machine Learning** merupakan salah satu bagian dari *Soft Computing*, terkait dengan pendekatan *learning*. Sedangkan **Fuzzy Logic** merupakan salah satu metode *Soft Computing*, terkait dengan pendekatan *reasoning*. **Machine Learning** adalah metode-metode komputasi (algoritma-algoritma) yang dapat melakukan pembelajaran terhadap data/obyek untuk menggali informasi atau menemukan pengetahuan atau aturan sehingga dapat mendukung pengambilan keputusan. Sedangkan **Fuzzy Logic** adalah metode komputasi (algoritma) yang dapat melakukan penalaran samar terhadap data/obyek melalui aturan-aturan (kepakanan) yang ditetapkan kepadanya untuk mengambil suatu keputusan. **Machine Learning** memecahkan suatu masalah menggunakan pendekatan pembelajaran (*learning*), sedangkan **Fuzzy Logic** memecahkan suatu masalah menggunakan pendekatan penalaran (*reasoning*).

Secara umum **Data Science** adalah penggalian atau mengekstrak atau analisis data untuk menemukan/menghasilkan informasi atau pengetahuan atau aturan yang tepat/akurat/benar. Oleh karena itu, metode-metode **Soft Computing** seperti **Machine Learning** dan **Fuzzy Logic** digunakan dalam *Data Science*.

Dalam pemrograman komputer dan *Data Science*, **type** adalah gambaran dari obyek yang dapat menampung/memiliki berbagai *variable, method (function/procedure)*,

dan *event*. Umumnya, *type* dari suatu data (**tipe data**) terbagi menjadi **numerik (interval dan ratio)** dan **kategorikal/diskrit (binominal, nominal, dan ordinal)**.

Dalam pemrograman komputer dan *data science*, **variable (field)** atau **property** atau **parameter** atau **attribute** atau **feature**) adalah tempat untuk menyimpan atau menampung nilai dari suatu data, sehingga dapat mendefinisikan dirinya sebagai suatu *type/object*, dapat menjadi anggota dari suatu *type/object*, dapat bernilai suatu *type/object*, dan dapat membentuk suatu vektor, namun tidak dapat memiliki *method* di dalamnya.

Dalam pemrograman komputer dan *Data Science*, **method (function/procedure)** adalah suatu aksi atau tindakan yang dimiliki suatu *type/object* untuk menyelesaikan masalah tertentu yang dijalankan/dilaksanakan oleh *event*. Suatu *function* dapat mendefinisikan dirinya sebagai suatu *type/object*, sehingga memiliki nilai/output. Sedangkan *procedure* tidak dapat mendefinisikan dirinya sebagai suatu *type/object*. Suatu *function* dan *procedure* dapat pula memiliki parameter input. Suatu *function* dan *procedure* dapat mengimplementasikan *variable* dan *type/object* di dalamnya. Dengan demikian, *method (function/procedure)* dapat dikatakan sebagai suatu atau serangkaian **algoritma**.

Dalam *Machine Learning* dan *Data Science*, **unsupervised learning** adalah proses pembelajaran tanpa guru (data yang dipelajari tidak memiliki output), sedangkan **supervised learning** adalah proses pembelajaran dengan/berdasarkan guru (data yang dipelajari memiliki/berdasarkan outputnya). Pekerjaan/masalah **clustering** (klasterisasi) dan **association** (asosiasi) termasuk dalam *unsupervised learning*, sedangkan **classification** (klasifikasi) dan **regression/estimation** (regresi/estimasi) termasuk dalam *supervised learning*.

Dalam *Machine Learning* dan *Data Science*, **clustering** adalah pengelompokan data tanpa output yang dapat digunakan untuk informasi/pengetahuan, pengenalan pola, prediksi, identifikasi, pendukung keputusan, dll.

Dalam *Machine Learning* dan *Data Science*, **association** adalah hubungan antara set item data (bukan hubungan antar variabel/atribut, sehingga hanya mengolah satu variabel/atribut saja) yang dapat digunakan untuk aturan asosiasi, informasi/pengetahuan, prediksi, pendukung keputusan, *market basket analysis*, dll.

Dalam *Machine Learning* dan *Data Science*, **classification** adalah pengelompokan data dengan/berdasarkan outputnya (label *class*) yang dapat digunakan untuk prediksi, informasi/pengetahuan, pengenalan pola, identifikasi, pendukung keputusan, dll.

Dalam *Machine Learning* dan *Data Science*, **regression/estimation** adalah estimasi data dengan/berdasarkan outputnya yang bertipe numerik yang dapat digunakan untuk prediksi, informasi/pengetahuan, pengenalan pola, identifikasi, pendukung keputusan, dll.

Dalam *Data Science* atau *Data Mining* atau *Big Data*, **pra pengolahan data** adalah proses pengolahan data sebelum data diolah/dianalisis dalam proses/tujuan utama pengolahan data tersebut yang umumnya terdiri dari *missing value replacement*, *data type transformation*, *aggregation*, *smoothing noisy data*, *dimensionality*

reduction, feature selection, attribute weighting, feature extraction, unbalanced class reduction, dan data validation.

Dalam pra pengolahan data, **missing value replacement** bertujuan untuk mengganti nilai data yang tidak dikenali sehingga dapat mereduksi bias dan meningkatkan kinerja metode komputasi yang digunakan dalam proses utama pengolahan data. Pendekatan yang umum digunakan untuk *missing value replacement* yaitu imputasi **mean/mode**.

Dalam pra pengolahan data, **data type transformation** bertujuan untuk merubah/transformasi tipe suatu data sehingga data dapat diolah oleh metode komputasi yang digunakan dalam proses utama pengolahan data. Secara garis besar, terdiri dari transformasi tipe data kategorikal/diskrit ke numerik (**encoding**) dan numerik ke kategorikal/diskrit (**discretization**). Pendekatan yang umum digunakan untuk *data discretization* yaitu **Binning** dan **Entropy**.

Dalam pra pengolahan data, **aggregation** bertujuan untuk menyatukan beberapa data atau variabel/atribut melalui pengkombinasian beberapa nilai pada data menjadi suatu nilai tunggal sehingga dapat mereduksi dimensi data yang dengan itu dapat mereduksi kompleksitas komputasi dan meningkatkan kinerja metode komputasi yang digunakan dalam proses utama pengolahan data. Pendekatan yang umum digunakan untuk *aggregation* yaitu **sum, min, max, mean**, dll.

Dalam pra pengolahan data, **smoothing noisy data** bertujuan untuk mereduksi *outliers* atau data yang *noise* sehingga dapat mereduksi bias dan meningkatkan kinerja metode komputasi yang digunakan dalam proses utama pengolahan data. *Smoothing noisy data* dapat dilakukan dengan cara **outliers detection/removed** (membuang *outliers*) atau dengan cara **data normalization** (normalisasi data). Metode-metode *Machine Learning*, seperti *k-Nearest Neighbor* dapat digunakan untuk *outliers detection*. Sedangkan metode yang umum digunakan untuk *data normalization* yaitu **Min-Max Normalization, Z-Score, Decimal Scaling, Sigmoidal**, dan **Softmax**.

Dalam pra pengolahan data, **dimensionality reduction** bertujuan untuk mereduksi dimensi data (mengurangi variabel/atribut yang kurang relevan) sehingga dapat mereduksi kompleksitas komputasi dan meningkatkan kinerja metode komputasi yang digunakan dalam proses utama pengolahan data. Metode yang umum digunakan untuk *dimensionality reduction* yaitu **Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Independent Component Analysis (ICA), Generalized Hebbian Algorithm (GHA), dan Self Organizing Map (SOM)**.

Dalam pra pengolahan data, **feature selection** memiliki tujuan yang sama dengan *dimensionality reduction*, namun *feature selection* memilih atribut-atribut yang relevan berdasarkan kinerja dari metode komputasi yang digunakan atas kekuatan relevansi atribut-atribut, misalnya melalui pembobotan terhadap setiap atribut. Metode yang umum digunakan untuk *feature selection* yaitu **Forward Selection, Backward Elimination, Bruto Force, Weight-Guided**, dan *Evolutionary (Genetic Algorithm)*.

Dalam pra pengolahan data, **attribute weighting** mirip dengan *feature selection* karena keduanya melakukan pembobotan terhadap atribut-atribut. Bedanya,

attribute weighting tidak bertujuan untuk mereduksi dimensi data. *Attribute weighting* memberikan bobot setiap atribut agar atribut tidak dianggap bersifat independen oleh metode komputasi yang digunakan dalam proses utama pengolahan data (setiap atribut memiliki pengaruh yang berbeda-beda). Dengan demikian, metode-metode yang dapat digunakan untuk *feature selection*, dapat pula digunakan untuk *attribute weighting*.

Dalam pra pengolahan data, yang membedakan *feature extraction* dengan *dimensionality reduction*, *feature selection*, dan *attribute weighting* adalah bahwa *feature extraction* dilakukan pada data yang belum terstruktur, seperti data *image*, *video*, *text*, *voice*, dll. *Feature extraction* bertujuan untuk melakukan transformasi data dari data yang tidak terstruktur menjadi terstruktur sehingga dapat diolah oleh metode komputasi yang digunakan dalam proses utama pengolahan data. Pendekatan yang digunakan tergantung pada jenis data yang digunakan, misalnya jika data *image/video* maka pendekatannya adalah *image processing*, data teks dengan pendekatan *text processing*, dst.

Dalam pra pengolahan data, *unbalanced class reduction* bertujuan untuk mereduksi ketidakseimbangan label *class* sehingga dapat meningkatkan kinerja metode komputasi yang digunakan dalam proses utama pengolahan data. Metode *ensemble* yang umum digunakan untuk *unbalanced class reduction* yaitu *AdaBoost*, *Bagging*, *Stacking*, dan *Weighted Vote*.

Evaluasi model adalah pengujian/evaluasi terhadap model dari metode komputasi yang digunakan dalam mengolah/menganalisis data untuk mengukur kinerja metode tersebut. Metode yang digunakan dalam evaluasi model tergantung dari pekerjaan analisis data tersebut. Pendekatan *Confusion Matrix* biasanya digunakan untuk mengukur kinerja *accuracy*, *precision*, *recall* (*specificity* dan *sensitivity*), dan *F-Measure* dari model klasifikasi. Pengukuran kompleksitas algoritma terhadap waktu dan ruang biasanya digunakan untuk mengukur kinerja kompleksitas dari metode komputasi yang digunakan. Pendekatan *Mean Square Error* (MSE), *Root Mean Squared Error* (RMSE), *Standard Error Estimation* (SEE), *Percentage Error* (PE), dan *Mean Absolute Percentage Error* (MAPE) biasanya digunakan untuk mengukur kinerja estimasi *error* dari model regresi/estimasi. Pendekatan **evaluasi internal**, **evaluasi eksternal**, dan **evaluasi aplikasi** biasanya digunakan untuk mengukur kinerja dari model klasterisasi. Pendekatan *Lift Ratio* biasanya digunakan untuk mengukur kinerja kekuatan hubungan asosiasi antara set item data dalam suatu aturan asosiasi. Pendekatan **uji koefisien korelasi** secara **parsial** maupun **simultan**, uji korelasi regresi *T-Test*, uji korelasi regresi *F-Test* biasanya digunakan untuk mengukur kekuatan hubungan antar variabel.

Dalam proses *Fuzzy Logic*, *Fuzzification* adalah proses untuk merubah nilai tegas (*crisp*) inputan menjadi nilai *Fuzzy* (derajat keanggotaan) menggunakan suatu *Membership Function* (Fungsi Keanggotaan) yang secara umum terdiri dari *Triangular Membership Function* (*trimf*), *S & Z Shaped Membership Function* (*smf* dan *zmf*), *PI Membership Function* (*pimf*), *Trapezoidal Membership Function* (*trapmf*), *Sigmoidal Membership Function* (*sigmf*), *Gaussian Membership Function* (*gaussmf*), dan *Generalized Bell-Shaped Membership Function* (*gbellmf*).

Dalam proses *Fuzzy Logic*, **Knowledge Base** adalah kumpulan aturan-aturan (*rules*) dalam bentuk pernyataan IF ... THEN

Dalam proses *Fuzzy Logic*, **Machine Inference** adalah proses untuk merubah inputan menjadi output-output. *Machine Inference* menggunakan fungsi implikasi **Max-Min** atau **Dot-Product** untuk memperoleh α -predikat dari *rule* ke-*i* (ai) dan output *rule* ke-*i* (yi).

Dalam proses *Fuzzy Logic*, **Defuzzification** adalah proses merubah output-output yang diperoleh pada *Machine Inference* menjadi satu nilai *crisp* output. Terdapat dua pendekatan/metode yang biasanya digunakan pada proses *Defuzzification*, yaitu **Average** dan **Centroid**.

Artificial Neural Network (ANN) merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasifikasi dan regresi/estimasi. Algoritma ANN yang umum digunakan adalah **Backpropagation**. Sedangkan **Adaptive Neuro Fuzzy Inference System (ANFIS)** merupakan pengembangan dari metode *Fuzzy Logic* dan ANN. Penerapan ANN pada *Fuzzy Logic* membuat algoritma *Fuzzy Logic* mampu melakukan proses pembelajaran (*learning*), sehingga ANFIS mampu melakukan penalaran (*reasoning*) sekaligus pembelajaran (*learning*).

Support Vector Machine (SVM) merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasifikasi dan regresi/estimasi. Pada masalah klasifikasi, SVM pada prinsipnya hanya dapat menangani masalah *binary classification*, sehingga disebut **Binary SVM**. Namun pendekatan seperti **One Versus One (1V1)** dan **One Versus Rest (1VR)** dapat digunakan pada SVM agar dapat menangani masalah *multi classification*. Salah satu algoritma **Multi Class SVM** yang umum digunakan adalah **LibSVM**. Pengembangan SVM yang lainnya untuk dapat menangani masalah *multi classification* adalah **Fuzzy SVM (FSVM)**, yang mana pendekatan *Membership Function* dari *Fuzzy Logic* diterapkan pada *class* yang tidak dapat diklasifikasikan oleh *Binary SVM*.

K-Means merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasterisasi. **K-Means** menggunakan pendekatan perhitungan jarak (*dissimilarity*) dalam mengelompokkan data. Metode *distance measure* (perhitungan jarak) yang umum digunakan, yaitu **Euclidean** untuk memperoleh jarak terdekat antar dua data dan **Manhattan (Cityblock)** untuk memperoleh jarak terjauh antar dua data. Salah satu pengembangan dari **K-Means** adalah **Fuzzy C-Means (FCM)**, yang mana *Membership Function* dari *Fuzzy Logic* digunakan untuk menentukan nilai/derajat keanggotaan suatu data tidak lagi secara tegas, melainkan dalam interval 0 hingga 1.

Naïve Bayes (NB) merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasifikasi. NB menggunakan pendekatan **probabilistic** dalam mengklasifikasikan data. Agar NB dapat menangani data bertipe numerik, maka distribusi *Gaussian* (distribusi normal) dapat diterapkan untuk menentukan probabilitas numerik dari atribut numerik (**Gaussian Naïve Bayes**) atau menggunakan pendekatan *Kernel* (**Kernel Naïve Bayes**). Salah satu kelemahan NB adalah menganggap bahwa setiap atribut bersifat independen (setiap atribut dianggap sama pentingnya), padahal asumsi ini tidak selalu tepat. Salah satu

cara untuk mengatasi kelemahan NB tersebut adalah dengan pendekatan *attribute weighting*. Salah satu metode *attribute weighting* yang dapat diterapkan pada NB adalah **Absolute Correlation Coefficient** (ACC). Pengembangan NB tersebut dinamakan **Absolute Correlation Weighted Naïve Bayes** (AC W-NB).

k-Nearest Neighbor (k-NN) merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasifikasi. *k-NN* menggunakan pendekatan perhitungan jarak (*dissimilarity*) dalam mengklasifikasikan data. Metode *distance measure* (perhitungan jarak) yang umum digunakan, yaitu **Euclidean** untuk memperoleh jarak terdekat antar dua data dan **Manhattan** (**Cityblock**) untuk memperoleh jarak terjauh antar dua data. Salah satu kelemahan k-NN pada proses penentuan output menggunakan pendekatan **Majority Vote** adalah menganggap bahwa setiap data tetangga memiliki derajat yang sama, asumsi ini dapat menyebabkan k-NN sensitif terhadap data yang *noise* jika nilai *k* (jumlah data tetangga) terlalu kecil atau kesalahan prediksi akibat distorsi data jika nilai *k* terlalu besar. Salah satu cara untuk mengatasi kelemahan k-NN tersebut adalah dengan pendekatan *weighting* (pembobotan) pada data tetangga (*distance weighted* atau *neighborhood weighted*). Beberapa pengembangan k-NN terkait masalah tersebut adalah **Weighted k-NN**, **Fuzzy k-NN** (FkNN), dan **Fuzzy k-NN in every class** (FkNNC). Pada FkNN, pendekatan derajat keanggotaan *Fuzzy* digunakan untuk menentukan derajat keanggotaan setiap *label class*. FkNN mendefinisikan derajat keanggotaan setiap *label class* berdasarkan nilai jarak setiap data tetangga, kemudian menetapkan *label class* yang memiliki probabilitas tertinggi sebagai hasil klasifikasi (output). Sedangkan pada FkNNC, sedikit modifikasi dilakukan pada FkNN dengan memberikan sejumlah *k* tetangga pada setiap *label class* (setiap *label class* memiliki jumlah tetangga yang sama sebanyak *k*). Dengan demikian, cara kerja FkNNC sebenarnya berbeda dengan k-NN, *Weighted k-NN*, bahkan FkNN karena setiap *label class* pada FkNNC memiliki tetangga terdekat sebanyak *k*, yang mana tetangga yang paling dekat untuk suatu *label class* adalah data latih milik *label class* tersebut yang jaraknya paling dekat dengan data uji/prediksi, bukan data latih yang paling dekat jaraknya dengan data uji seperti pada standar k-NN dan lainnya. Dengan kata lain, *label class* pada FkNNC sangat berpengaruh.

C4.5 merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah klasifikasi. C4.5 merupakan pengembangan dari **Decision Tree** agar dapat menangani data bertipe numerik dengan cara melakukan diskretisasi data. C4.5 menggunakan pendekatan **Entropy**, **Information Gain**, **Gain**, dan **Gain Ratio** dalam membentuk pohon keputusan (aturan) untuk mengklasifikasikan data.

Linear Regression merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah regresi/estimasi. *Linear Regression* menggunakan pendekatan matriks determinan jika terdapat 3 atau lebih variabel bebas yang diolah.

A-Priori merupakan salah satu metode *Machine Learning* yang dapat digunakan untuk menangani masalah asosiasi. *A-priori* menggunakan pendekatan **Support**, **Confidence**, dan **Lift Ratio** dalam membentuk aturan asosiasi dari dua set item data.

Daftar Indeks

1

1V1, xii, 115, 116, 117, 119, 120, 184, 189
1VR, 115, 116, 189

A

AC W-NB, xii, 140, 141, 151, 152, 190
ACC, 37, 140, 142, 151, 190
accuracy, 28, 33, 36, 40, 41, 50, 188
Adaboost, 39
aggregation, 19, 22, 186, 187
AI, 1, 4, 62, 74
algorithm complexity, 48
algoritma, ii, 2, 4, 9, 10, 11, 12, 15, 19, 20, 23, 32, 34, 37, 48, 49, 53, 54, 55, 63, 74, 76, 78, 84, 88, 90, 91, 102, 106, 120, 122, 130, 132, 134, 135, 140, 143, 151, 152, 154, 156, 166, 175, 180, 185, 186, 188, 189
ANFIS, vii, xii, 102, 103, 104, 106, 120, 189
ANN, ii, v, vii, xii, 2, 6, 28, 32, 33, 35, 36, 42, 89, 90, 94, 97, 100, 101, 102, 106, 107, 108, 114, 115, 120, 166, 189
anomaly detection, 23, 24, 26, 27
Anova, 7, 8
A-Priori, ii, vi, 2, 6, 153, 175, 176, 177, 178, 180, 190
asosiasi, 6, 7, 8, 23, 56, 57, 175, 176, 177, 178, 179, 180, 186, 188, 190
attribute weighting, 29, 36, 37, 38, 140, 143, 187, 188, 190

B

Backpropagation, v, xii, 89, 90, 91, 93, 94, 97, 100, 102, 189
Backward Elimination, 33, 35, 36, 187
Bagging, vii, xi, 39, 40, 46, 183, 188
Bayes Theorem, 134
Binning, xi, xiii, 20, 21, 156, 187
bivariate, 7
Bruto Force, 33, 36, 187

C

C4.5, ii, vi, xiii, 2, 6, 8, 20, 153, 154, 155, 156, 157, 158, 163, 164, 180, 190
Canonical Correlation Analysis, 8
case substitution, 18
centroid, 53, 78, 80, 82, 122, 123, 124, 125, 126, 127, 128, 130
Chebyshev, 123, 150
Cityblock, 24, 26, 123, 128, 189, 190
class, v, 4, 14, 18, 20, 23, 25, 26, 37, 39, 40, 41, 43, 45, 46, 50, 54, 56, 93, 100, 101, 106, 108, 109, 113, 115, 116, 117, 119, 120, 133, 134, 135, 136, 137, 138, 140, 141, 142, 143, 144, 146, 147, 148, 149, 150, 151, 156, 160, 162, 163, 183, 186, 187, 188, 189, 190

cold deck, 18
Confidence, 9, 56, 176, 177, 178, 179, 190
Confusion Matrix, ix, xi, 26, 28, 32, 35, 40, 48, 50, 54, 55, 60, 100, 113, 117, 118, 138, 144, 164, 188
Conjoint Analysis, 7

D

data discretization, 19, 20, 187
data normalization, 23, 27, 28, 46, 187
data science, 4, 5, 14, 15, 16, 186
data transformation, 19
data type transformation, 19, 46, 186, 187
data validation, 43, 46, 187
Davies-Bouldin Index, 53
Decimal Scaling, 27, 187
Defuzzification, v, 61, 63, 75, 76, 77, 78, 80, 84, 85, 103, 189
Dice Index, 54, 55
dimensionality reduction, 29, 30, 31, 32, 33, 36, 38, 187, 188
dissimilarity, 53, 55, 122, 123, 143, 144, 151, 189, 190
distance weighting, 146, 147
dsigmf, vii, 64, 70
Dunn Index, 53, 54

E

encoding, 8, 18, 19, 46, 139, 187
ensemble, 39, 40, 41, 182, 188
Entropy, xi, xiii, 20, 21, 22, 155, 156, 157, 158, 159, 160, 161, 162, 163, 182, 187, 190
estimasi, 6, 7, 8, 18, 23, 30, 33, 48, 51, 52, 59, 90, 93, 100, 108, 143, 166, 167, 169, 174, 186, 188, 189, 190
Euclidean, 24, 123, 124, 130, 144, 146, 148, 149, 150, 151, 189, 190
event, 14, 15, 186
Evolutionary, 33, 35, 36, 187

F

FCM, ii, v, xii, 2, 6, 121, 129, 130, 132, 189
feature extraction, 19, 29, 30, 31, 38, 187, 188
feature selection, 19, 29, 30, 33, 34, 35, 36, 37, 38, 46, 140, 143, 187, 188
FkNN, xiii, 147, 148, 149, 150, 151, 152, 190
FkNNC, xiii, 149, 150, 151, 152, 190
F-Measure, 50, 54, 55, 188
For, 12, 49
Forward Selection, 33, 34, 35, 36, 46, 187
Fowlkes – Mallows Index, 54, 55
FP-Growth, 175
F-Score, 53
FSVM, 119, 189

F-Test, 59, 169, 174, 188
function, 13, 14, 15, 16, 25, 34, 46, 82, 100, 104, 113, 138, 144, 164, 185, 186
Fuzzification, iv, 61, 63, 64, 76, 78, 84, 102, 188
Fuzzy, i, ii, iii, iv, v, vii, xii, 1, 2, 6, 61, 62, 63, 64, 74, 75, 76, 77, 78, 79, 80, 81, 84, 85, 87, 88, 89, 102, 119, 120, 121, 129, 130, 131, 133, 147, 149, 151, 152, 185, 188, 189, 190
Fuzzy Logic Mamdani, 63, 75, 78, 80, 81, 88
Fuzzy Logic Sugeno, 63, 75, 84, 85, 102
Fuzzy Logic Tsukamoto, 63, 75, 76, 77, 78, 84, 88

G

Gain, 20, 21, 22, 154, 155, 157, 158, 159, 160, 161, 162, 163, 190
Gain Ratio, 155, 159, 190
Gaussian, iv, v, xii, 8, 20, 23, 40, 71, 72, 133, 136, 138, 140, 141, 188, 189
gaussmf, vii, 64, 71, 72, 81, 83, 103, 188
gbellmf, vii, 64, 72, 73, 188
GHA, 30, 32, 187
global optimal, 107
Guilford Empirical Rules, 58

H

hidden layer, 28, 32, 35, 91, 94, 95, 96, 97
Holdout, vii, 17, 43, 117
hot deck, 18
hyperplane, 106, 107, 108, 109, 110, 112, 119

I

ICA, 30, 31, 187
ID3, 154
If – Then, 11, 12
image processing, 38, 188
Information Gain, 20, 21, 155, 157, 190

J

Jaccard Index, 54, 55

K

Kernel, xii, 19, 106, 107, 111, 113, 114, 115, 117, 118, 136, 138, 184, 189
Kernel Naïve Bayes, 189
KFACWNB-NN, v, 151
K-Fold Cross Validation, vii, xi, 17, 43, 44, 45, 101, 113, 114, 138, 145, 165
klasifikasi, 6, 7, 8, 18, 23, 30, 33, 37, 41, 42, 43, 48, 50, 54, 55, 90, 100, 106, 107, 108, 109, 110, 111, 115, 116, 120, 134, 135, 136, 138, 142, 143, 144, 146, 147, 150, 151, 152, 154, 155, 180, 186, 188, 189, 190
klasterisasi, 6, 7, 8, 23, 53, 54, 55, 56, 122, 128, 132, 186, 188, 189
K-Means, ii, v, xii, 2, 53, 121, 122, 123, 124, 128, 129, 130, 132, 189

k-NN, ii, v, xi, xii, xiii, 2, 6, 18, 24, 26, 46, 51, 133, 143, 144, 146, 147, 148, 149, 150, 151, 152, 190
Knowledge Base, iv, 61, 63, 73, 76, 78, 84, 102, 189
koefisien determinasi, 58
koefisien korelai parsial, 57
koefisien korelasi, 57, 58, 59, 60, 167, 169, 174, 188
koefisien korelasi simultan, 58
komputasi, i, 4, 9, 22, 29, 30, 32, 33, 34, 43, 48, 56, 106, 107, 115, 175, 185, 187, 188
komputer, 1, 4, 9, 10, 11, 14, 185, 186

L

lazy learning, 135, 143
Leave One Out, 43
LibSVM, xii, 118, 119, 120, 189
Lift, xi, 56, 57, 176, 178, 179, 188, 190
Linear Regression, ii, vi, xiii, 2, 153, 166, 167, 168, 169, 170, 172, 174, 180, 190
local optimal, 107

M

Machine Inference, v, xii, 61, 63, 74, 75, 76, 78, 84, 103, 189
machine learning, i, ii, 1, 2, 4, 5, 6, 7, 8, 16, 19, 23, 24, 37, 39, 41, 43, 48, 51, 52, 53, 54, 90, 102, 106, 108, 119, 122, 135, 166, 175
majority vote, 144, 146, 151
Manhattan, 26, 123, 124, 144, 149, 150, 151, 189, 190
MAPE, 52, 188
Matlab, ii, xi, xii, xiii, 2, 14, 16, 24, 25, 45, 46, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 76, 78, 80, 81, 84, 85, 88, 100, 103, 113, 117, 128, 130, 132, 138, 144, 164, 182
mean/mode, 18, 46, 187
Membership Function, iv, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 76, 78, 79, 81, 84, 87, 102, 103, 104, 119, 120, 129, 188, 189
method, 13, 14, 15, 20, 156, 185, 186
mf2mf, vii, 64, 73
Minkowski, 123
Min-Max Normalization, 27, 28, 41, 46, 187
missing value, 18, 46, 186, 187
MSE, 52, 93, 97, 104, 113, 167, 188
multivariate, 7

N

Naïve Bayes, ii, v, xii, 2, 8, 20, 37, 40, 41, 133, 134, 135, 136, 138, 140, 143, 151, 152, 182, 189
Nearest, v, 18, 24, 133, 143, 144, 149, 151, 187, 190
neighborhood weighting, 146
neuron, 90, 94, 95, 96, 100
NIIA, 18

NMI, 54, 56
noisy data, 23, 27, 28, 46, 143, 147, 151, 186, 187

O

object, 14, 16, 186
outliers, 23, 25, 26, 46, 143, 187

P

PCA, xi, 30, 31, 32, 33, 46, 184, 187
PE, 52, 188
pemrograman, 9, 10, 11, 14, 15, 185, 186
pimf, vii, 64, 67, 81, 83, 188
PNSR, 52, 60
precision, 28, 33, 36, 40, 50, 188
probabilitas, i, 1, 23, 34, 40, 51, 62, 134, 136, 137, 138, 140, 141, 142, 143, 147, 149, 150, 151, 189, 190
procedure, 13, 14, 15, 185, 186
psigmf, 64, 70
PSO, 36
Purity, 54

Q

Quadratic Programming, 115

R

Rand Index, 54, 55
Rapidminer, ii, xi, xii, xiii, 2, 14, 28, 32, 34, 35, 46, 76, 102, 114, 118, 132, 174, 179, 182
reasoning, i, ii, 1, 2, 62, 74, 102, 185, 189
recall, 28, 33, 36, 40, 50, 188
record, 13, 39
regresi, 6, 7, 8, 30, 33, 48, 51, 59, 90, 106, 120, 166, 169, 170, 174, 180, 186, 188, 189, 190
Repeat – Until, 12
RMSE, 52, 60, 102, 103, 104, 105, 114, 115, 167, 174, 188

S

searching, i, 1, 62, 185
SEE, 52, 60, 188
sensitivity, 50, 188
sigmf, vii, 64, 69, 70, 188
Sigmoidal, iv, 27, 69, 70, 187, 188
Silhouette Coefficients, 53
similarity, 53, 55
smf, vii, 64, 65, 66, 67, 81, 83, 87, 188
soft computing, i, 1, 4
Softmax, 27, 46, 187

SOM, 30, 32, 187
specificity, 50, 188
Split Info, 21, 22, 157
SRM, 106, 108
Stacking, vii, 39, 188
standard deviation, 27, 37, 71, 136
structure, 14
struktur data, 13, 48, 143
supervised learning, 6, 7, 30, 33, 106, 182, 186
Support, v, 9, 41, 56, 89, 175, 176, 177, 178, 179, 183, 184, 189, 190
SVD, xi, 30, 31, 32, 33, 46, 184, 187
SVM, ii, v, vii, xii, 2, 6, 37, 42, 89, 106, 107, 108, 109, 110, 111, 113, 114, 115, 116, 117, 119, 120, 166, 184, 189
Switch Case, 12

T

text processing, 38, 188
TIK, 1
trapmf, vii, 64, 68, 81, 82, 87, 103, 104, 106, 188
trimf, 64, 68, 73, 76, 77, 78, 79, 80, 81, 82, 84, 85, 86, 87, 103, 188
T-Test, 7, 59, 60, 169, 174, 188
type, 14, 19, 34, 46, 185, 186, 187

U

unbalanced class, 39, 54, 119, 149, 188
univariate, 7
unsupervised learning, 6, 7, 30, 32, 122, 186

V

variable, 13, 14, 15, 16, 185, 186

W

Weighted k-NN, 147, 148, 149, 151, 152, 190
Weighted Vote, vii, xi, 39, 41, 42, 188
Weight-Guided, 33, 36, 187
While, 12
WNB, 140

Z

zmf, vii, 64, 66, 67, 81, 83, 87, 188
Z-Score, xi, 27, 28, 32, 33, 35, 36, 40, 46, 187
Z-Test, 7

Lampiran

Lampiran 1. Dataset: Heart Disease – Cleveland

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	Y1	Y2
1	63	1	1	145	233	1	2	150	0	2,3	3	0	6	0	0
2	67	1	4	160	286	0	2	108	1	1,5	2	3	3	2	1
3	67	1	4	120	229	0	2	129	1	2,6	2	2	7	1	1
4	37	1	3	130	250	0	0	187	0	3,5	3	0	3	0	0
5	41	0	2	130	204	0	2	172	0	1,4	1	0	3	0	0
6	56	1	2	120	236	0	0	178	0	0,8	1	0	3	0	0
7	62	0	4	140	268	0	2	160	0	3,6	3	2	3	3	1
8	57	0	4	120	354	0	0	163	1	0,6	1	0	3	0	0
9	63	1	4	130	254	0	2	147	0	1,4	2	1	7	2	1
10	53	1	4	140	203	1	2	155	1	3,1	3	0	7	1	1
11	57	1	4	140	192	0	0	148	0	0,4	2	0	6	0	0
12	56	0	2	140	294	0	2	153	0	1,3	2	0	3	0	0
13	56	1	3	130	256	1	2	142	1	0,6	2	1	6	2	1
14	44	1	2	120	263	0	0	173	0	0	1	0	7	0	0
15	52	1	3	172	199	1	0	162	0	0,5	1	0	7	0	0
16	57	1	3	150	168	0	0	174	0	1,6	1	0	3	0	0
17	48	1	2	110	229	0	0	168	0	1	3	0	7	1	1
18	54	1	4	140	239	0	0	160	0	1,2	1	0	3	0	0
19	48	0	3	130	275	0	0	139	0	0,2	1	0	3	0	0
20	49	1	2	130	266	0	0	171	0	0,6	1	0	3	0	0
21	64	1	1	110	211	0	2	144	1	1,8	2	0	3	0	0
22	58	0	1	150	283	1	2	162	0	1	1	0	3	0	0
23	58	1	2	120	284	0	2	160	0	1,8	2	0	3	1	1
24	58	1	3	132	224	0	2	173	0	3,2	1	2	7	3	1
25	60	1	4	130	206	0	2	132	1	2,4	2	2	7	4	1
26	50	0	3	120	219	0	0	158	0	1,6	2	0	3	0	0
27	58	0	3	120	340	0	0	172	0	0	1	0	3	0	0
28	66	0	1	150	226	0	0	114	0	2,6	3	0	3	0	0
29	43	1	4	150	247	0	0	171	0	1,5	1	0	3	0	0
30	40	1	4	110	167	0	2	114	1	2	2	0	7	3	1
31	69	0	1	140	239	0	0	151	0	1,8	1	2	3	0	0
32	60	1	4	117	230	1	0	160	1	1,4	1	2	7	2	1
33	64	1	3	140	335	0	0	158	0	0	1	0	3	1	1
34	59	1	4	135	234	0	0	161	0	0,5	2	0	7	0	0
35	44	1	3	130	233	0	0	179	1	0,4	1	0	3	0	0
36	42	1	4	140	226	0	0	178	0	0	1	0	3	0	0
37	43	1	4	120	177	0	2	120	1	2,5	2	0	7	3	1
38	57	1	4	150	276	0	2	112	1	0,6	2	1	6	1	1
39	55	1	4	132	353	0	0	132	1	1,2	2	1	7	3	1
40	61	1	3	150	243	1	0	137	1	1	2	0	3	0	0
41	65	0	4	150	225	0	2	114	0	1	2	3	7	4	1
42	40	1	1	140	199	0	0	178	1	1,4	1	0	7	0	0
43	71	0	2	160	302	0	0	162	0	0,4	1	2	3	0	0
44	59	1	3	150	212	1	0	157	0	1,6	1	0	3	0	0
45	61	0	4	130	330	0	2	169	0	0	1	0	3	1	1
46	58	1	3	112	230	0	2	165	0	2,5	2	1	7	4	1
47	51	1	3	110	175	0	0	123	0	0,6	1	0	3	0	0
48	50	1	4	150	243	0	2	128	0	2,6	2	0	7	4	1
49	65	0	3	140	417	1	2	157	0	0,8	1	1	3	0	0
50	53	1	3	130	197	1	2	152	0	1,2	3	0	3	0	0
51	41	0	2	105	198	0	0	168	0	0	1	1	3	0	0

52	65	1	4	120	177	0	0	140	0	0,4	1	0	7	0	0
53	44	1	4	112	290	0	2	153	0	0	1	1	3	2	1
54	44	1	2	130	219	0	2	188	0	0	1	0	3	0	0
55	60	1	4	130	253	0	0	144	1	1,4	1	1	7	1	1
56	54	1	4	124	266	0	2	109	1	2,2	2	1	7	1	1
57	50	1	3	140	233	0	0	163	0	0,6	2	1	7	1	1
58	41	1	4	110	172	0	2	158	0	0	1	0	7	1	1
59	54	1	3	125	273	0	2	152	0	0,5	3	1	3	0	0
60	51	1	1	125	213	0	2	125	1	1,4	1	1	3	0	0
61	51	0	4	130	305	0	0	142	1	1,2	2	0	7	2	1
62	46	0	3	142	177	0	2	160	1	1,4	3	0	3	0	0
63	58	1	4	128	216	0	2	131	1	2,2	2	3	7	1	1
64	54	0	3	135	304	1	0	170	0	0	1	0	3	0	0
65	54	1	4	120	188	0	0	113	0	1,4	2	1	7	2	1
66	60	1	4	145	282	0	2	142	1	2,8	2	2	7	2	1
67	60	1	3	140	185	0	2	155	0	3	2	0	3	1	1
68	54	1	3	150	232	0	2	165	0	1,6	1	0	7	0	0
69	59	1	4	170	326	0	2	140	1	3,4	3	0	7	2	1
70	46	1	3	150	231	0	0	147	0	3,6	2	0	3	1	1
71	65	0	3	155	269	0	0	148	0	0,8	1	0	3	0	0
72	67	1	4	125	254	1	0	163	0	0,2	2	2	7	3	1
73	62	1	4	120	267	0	0	99	1	1,8	2	2	7	1	1
74	65	1	4	110	248	0	2	158	0	0,6	1	2	6	1	1
75	44	1	4	110	197	0	2	177	0	0	1	1	3	1	1
76	65	0	3	160	360	0	2	151	0	0,8	1	0	3	0	0
77	60	1	4	125	258	0	2	141	1	2,8	2	1	7	1	1
78	51	0	3	140	308	0	2	142	0	1,5	1	1	3	0	0
79	48	1	2	130	245	0	2	180	0	0,2	2	0	3	0	0
80	58	1	4	150	270	0	2	111	1	0,8	1	0	7	3	1
81	45	1	4	104	208	0	2	148	1	3	2	0	3	0	0
82	53	0	4	130	264	0	2	143	0	0,4	2	0	3	0	0
83	39	1	3	140	321	0	2	182	0	0	1	0	3	0	0
84	68	1	3	180	274	1	2	150	1	1,6	2	0	7	3	1
85	52	1	2	120	325	0	0	172	0	0,2	1	0	3	0	0
86	44	1	3	140	235	0	2	180	0	0	1	0	3	0	0
87	47	1	3	138	257	0	2	156	0	0	1	0	3	0	0
88	53	0	3	128	216	0	2	115	0	0	1	0	3	0	0
89	53	0	4	138	234	0	2	160	0	0	1	0	3	0	0
90	51	0	3	130	256	0	2	149	0	0,5	1	0	3	0	0
91	66	1	4	120	302	0	2	151	0	0,4	2	0	3	0	0
92	62	0	4	160	164	0	2	145	0	6,2	3	3	7	3	1
93	62	1	3	130	231	0	0	146	0	1,8	2	3	7	0	0
94	44	0	3	108	141	0	0	175	0	0,6	2	0	3	0	0
95	63	0	3	135	252	0	2	172	0	0	1	0	3	0	0
96	52	1	4	128	255	0	0	161	1	0	1	1	7	1	1
97	59	1	4	110	239	0	2	142	1	1,2	2	1	7	2	1
98	60	0	4	150	258	0	2	157	0	2,6	2	2	7	3	1
99	52	1	2	134	201	0	0	158	0	0,8	1	1	3	0	0
100	48	1	4	122	222	0	2	186	0	0	1	0	3	0	0
101	45	1	4	115	260	0	2	185	0	0	1	0	3	0	0
102	34	1	1	118	182	0	2	174	0	0	1	0	3	0	0
103	57	0	4	128	303	0	2	159	0	0	1	1	3	0	0
104	71	0	3	110	265	1	2	130	0	0	1	1	3	0	0
105	49	1	3	120	188	0	0	139	0	2	2	3	7	3	1
106	54	1	2	108	309	0	0	156	0	0	1	0	7	0	0
107	59	1	4	140	177	0	0	162	1	0	1	1	7	2	1
108	57	1	3	128	229	0	2	150	0	0,4	2	1	7	1	1

109	61	1	4	120	260	0	0	140	1	3,6	2	1	7	2	1
110	39	1	4	118	219	0	0	140	0	1,2	2	0	7	3	1
111	61	0	4	145	307	0	2	146	1	1	2	0	7	1	1
112	56	1	4	125	249	1	2	144	1	1,2	2	1	3	1	1
113	52	1	1	118	186	0	2	190	0	0	2	0	6	0	0
114	43	0	4	132	341	1	2	136	1	3	2	0	7	2	1
115	62	0	3	130	263	0	0	97	0	1,2	2	1	7	2	1
116	41	1	2	135	203	0	0	132	0	0	2	0	6	0	0
117	58	1	3	140	211	1	2	165	0	0	1	0	3	0	0
118	35	0	4	138	183	0	0	182	0	1,4	1	0	3	0	0
119	63	1	4	130	330	1	2	132	1	1,8	1	3	7	3	1
120	65	1	4	135	254	0	2	127	0	2,8	2	1	7	2	1
121	48	1	4	130	256	1	2	150	1	0	1	2	7	3	1
122	63	0	4	150	407	0	2	154	0	4	2	3	7	4	1
123	51	1	3	100	222	0	0	143	1	1,2	2	0	3	0	0
124	55	1	4	140	217	0	0	111	1	5,6	3	0	7	3	1
125	65	1	1	138	282	1	2	174	0	1,4	2	1	3	1	1
126	45	0	2	130	234	0	2	175	0	0,6	2	0	3	0	0
127	56	0	4	200	288	1	2	133	1	4	3	2	7	3	1
128	54	1	4	110	239	0	0	126	1	2,8	2	1	7	3	1
129	44	1	2	120	220	0	0	170	0	0	1	0	3	0	0
130	62	0	4	124	209	0	0	163	0	0	1	0	3	0	0
131	54	1	3	120	258	0	2	147	0	0,4	2	0	7	0	0
132	51	1	3	94	227	0	0	154	1	0	1	1	7	0	0
133	29	1	2	130	204	0	2	202	0	0	1	0	3	0	0
134	51	1	4	140	261	0	2	186	1	0	1	0	3	0	0
135	43	0	3	122	213	0	0	165	0	0,2	2	0	3	0	0
136	55	0	2	135	250	0	2	161	0	1,4	2	0	3	0	0
137	70	1	4	145	174	0	0	125	1	2,6	3	0	7	4	1
138	62	1	2	120	281	0	2	103	0	1,4	2	1	7	3	1
139	35	1	4	120	198	0	0	130	1	1,6	2	0	7	1	1
140	51	1	3	125	245	1	2	166	0	2,4	2	0	3	0	0
141	59	1	2	140	221	0	0	164	1	0	1	0	3	0	0
142	59	1	1	170	288	0	2	159	0	0,2	2	0	7	1	1
143	52	1	2	128	205	1	0	184	0	0	1	0	3	0	0
144	64	1	3	125	309	0	0	131	1	1,8	2	0	7	1	1
145	58	1	3	105	240	0	2	154	1	0,6	2	0	7	0	0
146	47	1	3	108	243	0	0	152	0	0	1	0	3	1	1
147	57	1	4	165	289	1	2	124	0	1	2	3	7	4	1
148	41	1	3	112	250	0	0	179	0	0	1	0	3	0	0
149	45	1	2	128	308	0	2	170	0	0	1	0	3	0	0
150	60	0	3	102	318	0	0	160	0	0	1	1	3	0	0
151	52	1	1	152	298	1	0	178	0	1,2	2	0	7	0	0
152	42	0	4	102	265	0	2	122	0	0,6	2	0	3	0	0
153	67	0	3	115	564	0	2	160	0	1,6	2	0	7	0	0
154	55	1	4	160	289	0	2	145	1	0,8	2	1	7	4	1
155	64	1	4	120	246	0	2	96	1	2,2	3	1	3	3	1
156	70	1	4	130	322	0	2	109	0	2,4	2	3	3	1	1
157	51	1	4	140	299	0	0	173	1	1,6	1	0	7	1	1
158	58	1	4	125	300	0	2	171	0	0	1	2	7	1	1
159	60	1	4	140	293	0	2	170	0	1,2	2	2	7	2	1
160	68	1	3	118	277	0	0	151	0	1	1	1	7	0	0
161	46	1	2	101	197	1	0	156	0	0	1	0	7	0	0
162	77	1	4	125	304	0	2	162	1	0	1	3	3	4	1
163	54	0	3	110	214	0	0	158	0	1,6	2	0	3	0	0
164	58	0	4	100	248	0	2	122	0	1	2	0	3	0	0
165	48	1	3	124	255	1	0	175	0	0	1	2	3	0	0

166	57	1	4	132	207	0	0	168	1	0	1	0	7	0	0
167	52	1	3	138	223	0	0	169	0	0	1	0	3	0	0
168	54	0	2	132	288	1	2	159	1	0	1	1	3	0	0
169	35	1	4	126	282	0	2	156	1	0	1	0	7	1	1
170	45	0	2	112	160	0	0	138	0	0	2	0	3	0	0
171	70	1	3	160	269	0	0	112	1	2,9	2	1	7	3	1
172	53	1	4	142	226	0	2	111	1	0	1	0	7	0	0
173	59	0	4	174	249	0	0	143	1	0	2	0	3	1	1
174	62	0	4	140	394	0	2	157	0	1,2	2	0	3	0	0
175	64	1	4	145	212	0	2	132	0	2	2	2	6	4	1
176	57	1	4	152	274	0	0	88	1	1,2	2	1	7	1	1
177	52	1	4	108	233	1	0	147	0	0,1	1	3	7	0	0
178	56	1	4	132	184	0	2	105	1	2,1	2	1	6	1	1
179	43	1	3	130	315	0	0	162	0	1,9	1	1	3	0	0
180	53	1	3	130	246	1	2	173	0	0	1	3	3	0	0
181	48	1	4	124	274	0	2	166	0	0,5	2	0	7	3	1
182	56	0	4	134	409	0	2	150	1	1,9	2	2	7	2	1
183	42	1	1	148	244	0	2	178	0	0,8	1	2	3	0	0
184	59	1	1	178	270	0	2	145	0	4,2	3	0	7	0	0
185	60	0	4	158	305	0	2	161	0	0	1	0	3	1	1
186	63	0	2	140	195	0	0	179	0	0	1	2	3	0	0
187	42	1	3	120	240	1	0	194	0	0,8	3	0	7	0	0
188	66	1	2	160	246	0	0	120	1	0	2	3	6	2	1
189	54	1	2	192	283	0	2	195	0	0	1	1	7	1	1
190	69	1	3	140	254	0	2	146	0	2	2	3	7	2	1
191	50	1	3	129	196	0	0	163	0	0	1	0	3	0	0
192	51	1	4	140	298	0	0	122	1	4,2	2	3	7	3	1
193	43	1	4	132	247	1	2	143	1	0,1	2	0	7	1	1
194	62	0	4	138	294	1	0	106	0	1,9	2	3	3	2	1
195	68	0	3	120	211	0	2	115	0	1,5	2	0	3	0	0
196	67	1	4	100	299	0	2	125	1	0,9	2	2	3	3	1
197	69	1	1	160	234	1	2	131	0	0,1	2	1	3	0	0
198	45	0	4	138	236	0	2	152	1	0,2	2	0	3	0	0
199	50	0	2	120	244	0	0	162	0	1,1	1	0	3	0	0
200	59	1	1	160	273	0	2	125	0	0	1	0	3	1	1
201	50	0	4	110	254	0	2	159	0	0	1	0	3	0	0
202	64	0	4	180	325	0	0	154	1	0	1	0	3	0	0
203	57	1	3	150	126	1	0	173	0	0,2	1	1	7	0	0
204	64	0	3	140	313	0	0	133	0	0,2	1	0	7	0	0
205	43	1	4	110	211	0	0	161	0	0	1	0	7	0	0
206	45	1	4	142	309	0	2	147	1	0	2	3	7	3	1
207	58	1	4	128	259	0	2	130	1	3	2	2	7	3	1
208	50	1	4	144	200	0	2	126	1	0,9	2	0	7	3	1
209	55	1	2	130	262	0	0	155	0	0	1	0	3	0	0
210	62	0	4	150	244	0	0	154	1	1,4	2	0	3	1	1
211	37	0	3	120	215	0	0	170	0	0	1	0	3	0	0
212	38	1	1	120	231	0	0	182	1	3,8	2	0	7	4	1
213	41	1	3	130	214	0	2	168	0	2	2	0	3	0	0
214	66	0	4	178	228	1	0	165	1	1	2	2	7	3	1
215	52	1	4	112	230	0	0	160	0	0	1	1	3	1	1
216	56	1	1	120	193	0	2	162	0	1,9	2	0	7	0	0
217	46	0	2	105	204	0	0	172	0	0	1	0	3	0	0
218	46	0	4	138	243	0	2	152	1	0	2	0	3	0	0
219	64	0	4	130	303	0	0	122	0	2	2	2	3	0	0
220	59	1	4	138	271	0	2	182	0	0	1	0	3	0	0
221	41	0	3	112	268	0	2	172	1	0	1	0	3	0	0
222	54	0	3	108	267	0	2	167	0	0	1	0	3	0	0

223	39	0	3	94	199	0	0	179	0	0	1	0	3	0	0
224	53	1	4	123	282	0	0	95	1	2	2	2	7	3	1
225	63	0	4	108	269	0	0	169	1	1,8	2	2	3	1	1
226	34	0	2	118	210	0	0	192	0	0,7	1	0	3	0	0
227	47	1	4	112	204	0	0	143	0	0,1	1	0	3	0	0
228	67	0	3	152	277	0	0	172	0	0	1	1	3	0	0
229	54	1	4	110	206	0	2	108	1	0	2	1	3	3	1
230	66	1	4	112	212	0	2	132	1	0,1	1	1	3	2	1
231	52	0	3	136	196	0	2	169	0	0,1	2	0	3	0	0
232	55	0	4	180	327	0	1	117	1	3,4	2	0	3	2	1
233	49	1	3	118	149	0	2	126	0	0,8	1	1	3	1	1
234	74	0	2	120	269	0	2	121	1	0,2	1	1	3	0	0
235	54	0	3	160	201	0	0	163	0	0	1	1	3	0	0
236	54	1	4	122	286	0	2	116	1	3,2	2	2	3	3	1
237	56	1	4	130	283	1	2	103	1	1,6	3	0	7	2	1
238	46	1	4	120	249	0	2	144	0	0,8	1	0	7	1	1
239	49	0	2	134	271	0	0	162	0	0	2	0	3	0	0
240	42	1	2	120	295	0	0	162	0	0	1	0	3	0	0
241	41	1	2	110	235	0	0	153	0	0	1	0	3	0	0
242	41	0	2	126	306	0	0	163	0	0	1	0	3	0	0
243	49	0	4	130	269	0	0	163	0	0	1	0	3	0	0
244	61	1	1	134	234	0	0	145	0	2,6	2	2	3	2	1
245	60	0	3	120	178	1	0	96	0	0	1	0	3	0	0
246	67	1	4	120	237	0	0	71	0	1	2	0	3	2	1
247	58	1	4	100	234	0	0	156	0	0,1	1	1	7	2	1
248	47	1	4	110	275	0	2	118	1	1	2	1	3	1	1
249	52	1	4	125	212	0	0	168	0	1	1	2	7	3	1
250	62	1	2	128	208	1	2	140	0	0	1	0	3	0	0
251	57	1	4	110	201	0	0	126	1	1,5	2	0	6	0	0
252	58	1	4	146	218	0	0	105	0	2	2	1	7	1	1
253	64	1	4	128	263	0	0	105	1	0,2	2	1	7	0	0
254	51	0	3	120	295	0	2	157	0	0,6	1	0	3	0	0
255	43	1	4	115	303	0	0	181	0	1,2	2	0	3	0	0
256	42	0	3	120	209	0	0	173	0	0	2	0	3	0	0
257	67	0	4	106	223	0	0	142	0	0,3	1	2	3	0	0
258	76	0	3	140	197	0	1	116	0	1,1	2	0	3	0	0
259	70	1	2	156	245	0	2	143	0	0	1	0	3	0	0
260	57	1	2	124	261	0	0	141	0	0,3	1	0	7	1	1
261	44	0	3	118	242	0	0	149	0	0,3	2	1	3	0	0
262	58	0	2	136	319	1	2	152	0	0	1	2	3	3	1
263	60	0	1	150	240	0	0	171	0	0,9	1	0	3	0	0
264	44	1	3	120	226	0	0	169	0	0	1	0	3	0	0
265	61	1	4	138	166	0	2	125	1	3,6	2	1	3	4	1
266	42	1	4	136	315	0	0	125	1	1,8	2	0	6	2	1
267	52	1	4	128	204	1	0	156	1	1	2	0	3	2	1
268	59	1	3	126	218	1	0	134	0	2,2	2	1	6	2	1
269	40	1	4	152	223	0	0	181	0	0	1	0	7	1	1
270	42	1	3	130	180	0	0	150	0	0	1	0	3	0	0
271	61	1	4	140	207	0	2	138	1	1,9	1	1	7	1	1
272	66	1	4	160	228	0	2	138	0	2,3	1	0	6	0	0
273	46	1	4	140	311	0	0	120	1	1,8	2	2	7	2	1
274	71	0	4	112	149	0	0	125	0	1,6	2	0	3	0	0
275	59	1	1	134	204	0	0	162	0	0,8	1	2	3	1	1
276	64	1	1	170	227	0	2	155	0	0,6	2	0	7	0	0
277	66	0	3	146	278	0	2	152	0	0	2	1	3	0	0
278	39	0	3	138	220	0	0	152	0	0	2	0	3	0	0
279	57	1	2	154	232	0	2	164	0	0	1	1	3	1	1

280	58	0	4	130	197	0	0	131	0	0,6	2	0	3	0	0
281	57	1	4	110	335	0	0	143	1	3	2	1	7	2	1
282	47	1	3	130	253	0	0	179	0	0	1	0	3	0	0
283	55	0	4	128	205	0	1	130	1	2	2	1	7	3	1
284	35	1	2	122	192	0	0	174	0	0	1	0	3	0	0
285	61	1	4	148	203	0	0	161	0	0	1	1	7	2	1
286	58	1	4	114	318	0	1	140	0	4,4	3	3	6	4	1
287	58	0	4	170	225	1	2	146	1	2,8	2	2	6	2	1
288	58	1	2	125	220	0	0	144	0	0,4	2	0	7	0	0
289	56	1	2	130	221	0	2	163	0	0	1	0	7	0	0
290	56	1	2	120	240	0	0	169	0	0	3	0	3	0	0
291	67	1	3	152	212	0	2	150	0	0,8	2	0	7	1	1
292	55	0	2	132	342	0	0	166	0	1,2	1	0	3	0	0
293	44	1	4	120	169	0	0	144	1	2,8	3	0	6	2	1
294	63	1	4	140	187	0	2	144	1	4	1	2	7	2	1
295	63	0	4	124	197	0	0	136	1	0	2	0	3	1	1
296	41	1	2	120	157	0	0	182	0	0	1	0	3	0	0
297	59	1	4	164	176	1	2	90	0	1	2	2	6	3	1
298	57	0	4	140	241	0	0	123	1	0,2	2	0	7	1	1
299	45	1	1	110	264	0	0	132	0	1,2	2	0	7	1	1
300	68	1	4	144	193	1	0	141	0	3,4	2	2	7	2	1
301	57	1	4	130	131	0	0	115	1	1,2	2	1	7	3	1
302	57	0	2	130	236	0	2	174	0	0	2	1	3	1	1
303	38	1	3	138	175	0	0	173	0	0	1	0	3	0	0

VARIABLE	VALUE	TYPE	
X1	Umur	0	
X2	Jenis Kelamin	Wanita (0), Pria (1)	
X3	Jenis Sakit Dada	Tipikal Angina (1), Atipikal Angina (2), Non Angina (3), Asimptomatic (4)	
X4	Tekanan Darah	0 (mm Hg)	
X5	Kolesterol	0 (mg/dl)	
X6	Kadar Gula	< 120 mg/dl (0), >= 120 mg/dl (1)	
X7	Rekam Jantung	Normal (0), Kelainan ST-T (1), Hypertrofi Ventrikel Kiri & Estes (2)	
X8	Tekanan Jantung	0	
X9	Nyeri Dada	No (0), Yes (1)	
X10	Oldpeak	0,0	
X11	Slope	Naik (1), Datar (2), Turun (3)	
X12	Denyut Jantung	0, 1, 2, 3	
X13	preThal	Normal (3), Cacat Tetap (6), Cacat Reversibel (7)	
Class	Diagnosis	0 = Sehat	
		1 = Resiko Rendah	
		2 = Resiko Sedang	
		3 = Resiko Tinggi	
	Diagnosis 2	4 = Resiko Sangat Tinggi	
		0 = Sehat	
		1 = Sakit	
Missing Value (Replace to: Nominal & Ordinal = Mode, Numeric = Mean)		6	
Instances		303	

Lampiran 2. Dataset: Tinggi Berat Badan

No.	Tinggi	Berat	Y	No.	Tinggi	Berat	Y	No.	Tinggi	Berat	Y
1	210	63	Outlier	168	155	80	Not Outlier	335	189	76	Not Outlier
2	220	58	Outlier	169	81	36	Not Outlier	336	154	86	Not Outlier
3	230	46	Outlier	170	98	69	Not Outlier	337	159	68	Not Outlier
4	240	35	Outlier	171	156	46	Not Outlier	338	177	87	Not Outlier
5	250	65	Outlier	172	98	42	Not Outlier	339	116	66	Not Outlier
6	45	51	Outlier	173	137	58	Not Outlier	340	52	36	Not Outlier
7	40	80	Outlier	174	150	53	Not Outlier	341	81	25	Not Outlier
8	35	59	Outlier	175	75	34	Not Outlier	342	93	24	Not Outlier
9	30	62	Outlier	176	132	56	Not Outlier	343	145	89	Not Outlier
10	25	75	Outlier	177	156	72	Not Outlier	344	71	46	Not Outlier
11	71	95	Outlier	178	116	48	Not Outlier	345	103	15	Not Outlier
12	91	96	Outlier	179	185	71	Not Outlier	346	88	47	Not Outlier
13	66	97	Outlier	180	53	35	Not Outlier	347	142	73	Not Outlier
14	148	98	Outlier	181	74	27	Not Outlier	348	57	29	Not Outlier
15	175	99	Outlier	182	192	66	Not Outlier	349	66	15	Not Outlier
16	51	10	Outlier	183	142	81	Not Outlier	350	166	82	Not Outlier
17	130	9	Outlier	184	189	71	Not Outlier	351	100	17	Not Outlier
18	178	8	Outlier	185	177	81	Not Outlier	352	184	89	Not Outlier
19	101	7	Outlier	186	97	33	Not Outlier	353	67	42	Not Outlier
20	95	6	Outlier	187	113	56	Not Outlier	354	185	82	Not Outlier
21	26	5	Outlier	188	153	56	Not Outlier	355	173	68	Not Outlier
22	32	6	Outlier	189	79	27	Not Outlier	356	186	75	Not Outlier
23	41	7	Outlier	190	156	79	Not Outlier	357	107	47	Not Outlier
24	27	8	Outlier	191	71	47	Not Outlier	358	128	53	Not Outlier
25	29	9	Outlier	192	152	71	Not Outlier	359	155	57	Not Outlier
26	249	98	Outlier	193	57	20	Not Outlier	360	159	66	Not Outlier
27	237	99	Outlier	194	149	51	Not Outlier	361	157	89	Not Outlier
28	222	96	Outlier	195	117	48	Not Outlier	362	109	19	Not Outlier
29	219	97	Outlier	196	140	88	Not Outlier	363	153	77	Not Outlier
30	217	95	Outlier	197	195	83	Not Outlier	364	99	44	Not Outlier
31	125	56	Not Outlier	198	175	77	Not Outlier	365	107	53	Not Outlier
32	76	31	Not Outlier	199	141	56	Not Outlier	366	114	53	Not Outlier
33	185	65	Not Outlier	200	101	63	Not Outlier	367	147	55	Not Outlier
34	167	73	Not Outlier	201	103	45	Not Outlier	368	94	51	Not Outlier
35	149	78	Not Outlier	202	188	66	Not Outlier	369	160	56	Not Outlier
36	137	51	Not Outlier	203	147	66	Not Outlier	370	153	75	Not Outlier
37	96	45	Not Outlier	204	177	79	Not Outlier	371	137	55	Not Outlier
38	137	67	Not Outlier	205	144	58	Not Outlier	372	146	78	Not Outlier
39	101	16	Not Outlier	206	108	66	Not Outlier	373	132	56	Not Outlier
40	171	77	Not Outlier	207	55	44	Not Outlier	374	156	69	Not Outlier
41	191	87	Not Outlier	208	147	63	Not Outlier	375	98	18	Not Outlier
42	110	42	Not Outlier	209	109	62	Not Outlier	376	181	70	Not Outlier
43	99	61	Not Outlier	210	59	31	Not Outlier	377	57	19	Not Outlier
44	79	41	Not Outlier	211	97	33	Not Outlier	378	149	61	Not Outlier
45	162	83	Not Outlier	212	61	33	Not Outlier	379	158	82	Not Outlier
46	83	17	Not Outlier	213	64	39	Not Outlier	380	165	88	Not Outlier
47	74	16	Not Outlier	214	62	37	Not Outlier	381	137	70	Not Outlier
48	78	18	Not Outlier	215	128	46	Not Outlier	382	149	72	Not Outlier
49	159	66	Not Outlier	216	151	49	Not Outlier	383	138	57	Not Outlier
50	97	47	Not Outlier	217	167	85	Not Outlier	384	103	17	Not Outlier
51	130	57	Not Outlier	218	142	81	Not Outlier	385	153	83	Not Outlier
52	98	48	Not Outlier	219	133	66	Not Outlier	386	150	73	Not Outlier
53	186	83	Not Outlier	220	115	60	Not Outlier	387	186	69	Not Outlier
54	103	45	Not Outlier	221	154	67	Not Outlier	388	121	54	Not Outlier

55	55	41	Not Outlier	222	63	28	Not Outlier	389	84	38	Not Outlier
56	192	77	Not Outlier	223	137	52	Not Outlier	390	83	50	Not Outlier
57	102	38	Not Outlier	224	108	45	Not Outlier	391	83	16	Not Outlier
58	59	35	Not Outlier	225	141	67	Not Outlier	392	58	39	Not Outlier
59	176	86	Not Outlier	226	147	67	Not Outlier	393	93	20	Not Outlier
60	69	27	Not Outlier	227	81	15	Not Outlier	394	133	46	Not Outlier
61	115	69	Not Outlier	228	181	87	Not Outlier	395	163	70	Not Outlier
62	189	73	Not Outlier	229	139	58	Not Outlier	396	124	70	Not Outlier
63	122	69	Not Outlier	230	190	74	Not Outlier	397	109	63	Not Outlier
64	181	88	Not Outlier	231	143	54	Not Outlier	398	160	88	Not Outlier
65	110	28	Not Outlier	232	80	15	Not Outlier	399	101	40	Not Outlier
66	80	38	Not Outlier	233	192	82	Not Outlier	400	102	20	Not Outlier
67	55	44	Not Outlier	234	121	66	Not Outlier	401	102	25	Not Outlier
68	126	66	Not Outlier	235	83	30	Not Outlier	402	104	61	Not Outlier
69	98	61	Not Outlier	236	152	47	Not Outlier	403	58	26	Not Outlier
70	80	29	Not Outlier	237	178	65	Not Outlier	404	108	29	Not Outlier
71	100	15	Not Outlier	238	171	81	Not Outlier	405	107	36	Not Outlier
72	56	43	Not Outlier	239	181	76	Not Outlier	406	157	68	Not Outlier
73	69	35	Not Outlier	240	149	69	Not Outlier	407	139	59	Not Outlier
74	110	65	Not Outlier	241	59	29	Not Outlier	408	144	87	Not Outlier
75	160	67	Not Outlier	242	114	55	Not Outlier	409	102	43	Not Outlier
76	140	70	Not Outlier	243	96	23	Not Outlier	410	154	66	Not Outlier
77	84	20	Not Outlier	244	133	59	Not Outlier	411	159	80	Not Outlier
78	139	62	Not Outlier	245	146	58	Not Outlier	412	106	48	Not Outlier
79	121	70	Not Outlier	246	96	46	Not Outlier	413	119	54	Not Outlier
80	164	74	Not Outlier	247	179	78	Not Outlier	414	172	78	Not Outlier
81	167	86	Not Outlier	248	152	75	Not Outlier	415	106	38	Not Outlier
82	118	64	Not Outlier	249	147	85	Not Outlier	416	187	77	Not Outlier
83	124	61	Not Outlier	250	74	24	Not Outlier	417	94	33	Not Outlier
84	122	50	Not Outlier	251	146	83	Not Outlier	418	74	17	Not Outlier
85	91	49	Not Outlier	252	166	84	Not Outlier	419	172	71	Not Outlier
86	56	19	Not Outlier	253	58	35	Not Outlier	420	192	69	Not Outlier
87	104	60	Not Outlier	254	67	22	Not Outlier	421	148	90	Not Outlier
88	104	48	Not Outlier	255	119	46	Not Outlier	422	147	65	Not Outlier
89	156	51	Not Outlier	256	144	56	Not Outlier	423	128	53	Not Outlier
90	52	23	Not Outlier	257	105	64	Not Outlier	424	153	61	Not Outlier
91	56	18	Not Outlier	258	140	82	Not Outlier	425	158	58	Not Outlier
92	141	67	Not Outlier	259	51	17	Not Outlier	426	123	60	Not Outlier
93	55	18	Not Outlier	260	126	67	Not Outlier	427	95	55	Not Outlier
94	136	56	Not Outlier	261	154	67	Not Outlier	428	117	55	Not Outlier
95	109	18	Not Outlier	262	113	48	Not Outlier	429	107	68	Not Outlier
96	179	86	Not Outlier	263	174	78	Not Outlier	430	191	69	Not Outlier
97	77	36	Not Outlier	264	157	67	Not Outlier	431	140	70	Not Outlier
98	61	17	Not Outlier	265	130	57	Not Outlier	432	133	59	Not Outlier
99	85	41	Not Outlier	266	185	83	Not Outlier	433	103	43	Not Outlier
100	108	49	Not Outlier	267	167	76	Not Outlier	434	150	77	Not Outlier
101	111	60	Not Outlier	268	179	67	Not Outlier	435	132	64	Not Outlier
102	90	45	Not Outlier	269	183	83	Not Outlier	436	95	24	Not Outlier
103	116	47	Not Outlier	270	139	53	Not Outlier	437	144	77	Not Outlier
104	73	25	Not Outlier	271	63	39	Not Outlier	438	136	53	Not Outlier
105	163	81	Not Outlier	272	180	68	Not Outlier	439	166	78	Not Outlier
106	93	18	Not Outlier	273	74	19	Not Outlier	440	74	23	Not Outlier
107	103	19	Not Outlier	274	153	51	Not Outlier	441	167	72	Not Outlier
108	187	87	Not Outlier	275	120	57	Not Outlier	442	119	65	Not Outlier
109	108	26	Not Outlier	276	92	50	Not Outlier	443	87	31	Not Outlier
110	76	25	Not Outlier	277	154	74	Not Outlier	444	140	85	Not Outlier
111	171	68	Not Outlier	278	94	22	Not Outlier	445	79	31	Not Outlier

112	154	71	Not Outlier	279	147	80	Not Outlier	446	146	90	Not Outlier
113	157	90	Not Outlier	280	194	72	Not Outlier	447	146	80	Not Outlier
114	127	46	Not Outlier	281	103	57	Not Outlier	448	101	50	Not Outlier
115	171	81	Not Outlier	282	127	55	Not Outlier	449	56	32	Not Outlier
116	144	70	Not Outlier	283	105	40	Not Outlier	450	169	80	Not Outlier
117	126	53	Not Outlier	284	73	36	Not Outlier	451	109	50	Not Outlier
118	143	80	Not Outlier	285	152	77	Not Outlier	452	100	48	Not Outlier
119	167	89	Not Outlier	286	82	27	Not Outlier	453	101	38	Not Outlier
120	148	78	Not Outlier	287	129	52	Not Outlier	454	92	52	Not Outlier
121	108	47	Not Outlier	288	90	56	Not Outlier	455	57	16	Not Outlier
122	96	19	Not Outlier	289	141	82	Not Outlier	456	50	29	Not Outlier
123	166	89	Not Outlier	290	189	77	Not Outlier	457	108	62	Not Outlier
124	68	38	Not Outlier	291	174	71	Not Outlier	458	113	67	Not Outlier
125	184	88	Not Outlier	292	110	49	Not Outlier	459	130	62	Not Outlier
126	105	22	Not Outlier	293	137	66	Not Outlier	460	186	87	Not Outlier
127	141	55	Not Outlier	294	102	49	Not Outlier	461	93	60	Not Outlier
128	143	87	Not Outlier	295	146	65	Not Outlier	462	100	37	Not Outlier
129	127	49	Not Outlier	296	152	88	Not Outlier	463	84	26	Not Outlier
130	151	83	Not Outlier	297	110	57	Not Outlier	464	100	40	Not Outlier
131	189	88	Not Outlier	298	102	67	Not Outlier	465	192	82	Not Outlier
132	178	87	Not Outlier	299	55	46	Not Outlier	466	104	33	Not Outlier
133	73	28	Not Outlier	300	142	77	Not Outlier	467	179	70	Not Outlier
134	169	89	Not Outlier	301	87	30	Not Outlier	468	187	78	Not Outlier
135	62	23	Not Outlier	302	153	57	Not Outlier	469	122	50	Not Outlier
136	140	47	Not Outlier	303	176	88	Not Outlier	470	150	72	Not Outlier
137	181	84	Not Outlier	304	160	88	Not Outlier	471	180	87	Not Outlier
138	183	78	Not Outlier	305	99	40	Not Outlier	472	106	23	Not Outlier
139	57	33	Not Outlier	306	150	57	Not Outlier	473	103	17	Not Outlier
140	137	48	Not Outlier	307	131	59	Not Outlier	474	184	71	Not Outlier
141	82	32	Not Outlier	308	141	51	Not Outlier	475	93	64	Not Outlier
142	118	68	Not Outlier	309	189	78	Not Outlier	476	109	28	Not Outlier
143	180	66	Not Outlier	310	85	19	Not Outlier	477	89	38	Not Outlier
144	92	45	Not Outlier	311	187	86	Not Outlier	478	70	21	Not Outlier
145	149	84	Not Outlier	312	186	77	Not Outlier	479	117	70	Not Outlier
146	102	68	Not Outlier	313	51	37	Not Outlier	480	94	58	Not Outlier
147	159	74	Not Outlier	314	103	34	Not Outlier	481	143	78	Not Outlier
148	140	62	Not Outlier	315	113	53	Not Outlier	482	70	47	Not Outlier
149	112	64	Not Outlier	316	103	64	Not Outlier	483	150	53	Not Outlier
150	110	58	Not Outlier	317	99	21	Not Outlier	484	87	50	Not Outlier
151	82	50	Not Outlier	318	146	67	Not Outlier	485	190	70	Not Outlier
152	108	23	Not Outlier	319	51	18	Not Outlier	486	108	54	Not Outlier
153	155	68	Not Outlier	320	168	74	Not Outlier	487	89	46	Not Outlier
154	142	68	Not Outlier	321	89	46	Not Outlier	488	149	58	Not Outlier
155	58	33	Not Outlier	322	121	56	Not Outlier	489	97	42	Not Outlier
156	144	73	Not Outlier	323	145	64	Not Outlier	490	188	66	Not Outlier
157	106	45	Not Outlier	324	171	74	Not Outlier	491	154	64	Not Outlier
158	143	47	Not Outlier	325	104	19	Not Outlier	492	164	71	Not Outlier
159	110	24	Not Outlier	326	51	48	Not Outlier	493	155	50	Not Outlier
160	177	71	Not Outlier	327	117	46	Not Outlier	494	124	57	Not Outlier
161	78	44	Not Outlier	328	64	50	Not Outlier	495	91	54	Not Outlier
162	152	61	Not Outlier	329	105	49	Not Outlier	496	146	48	Not Outlier
163	158	74	Not Outlier	330	88	21	Not Outlier	497	123	66	Not Outlier
164	104	70	Not Outlier	331	66	16	Not Outlier	498	54	46	Not Outlier
165	66	15	Not Outlier	332	57	35	Not Outlier	499	94	43	Not Outlier
166	87	24	Not Outlier	333	63	32	Not Outlier	500	153	78	Not Outlier
167	120	45	Not Outlier	334	191	67	Not Outlier				

Lampiran 3. Dataset: Traffic Light

No.	AT	TT	GJ	WT	No.	AT	TT	GJ	WT	No.	AT	TT	GJ	WT
1	0	4	3	80	78	14	4	2	70	155	7	2	1	35
2	1	4	2	90	79	15	2	3	25	156	8	7	3	54
3	2	4	1	40	80	16	4	3	40	157	9	2	3	62
4	3	4	1	45	81	17	4	1	25	158	10	2	1	73
5	4	2	2	85	82	18	4	1	25	159	11	2	3	30
6	5	2	1	25	83	19	2	3	55	160	12	4	1	67
7	6	2	3	70	84	20	7	3	80	161	13	4	1	53
8	7	2	1	30	85	0	4	2	35	162	14	2	1	61
9	8	7	2	80	86	1	2	1	65	163	15	4	2	58
10	9	7	3	90	87	2	7	2	55	164	16	2	2	74
11	10	10	3	40	88	3	2	3	95	165	17	7	2	46
12	11	7	3	40	89	4	2	2	45	166	18	2	3	37
13	12	2	2	90	90	5	4	2	122	167	19	2	2	57
14	13	7	3	90	91	6	4	3	55	168	20	4	1	64
15	14	7	2	55	92	7	4	3	85	169	0	2	3	65
16	15	4	1	45	93	8	2	1	60	170	1	7	2	42
17	16	2	3	98	94	9	2	2	70	171	2	7	2	57
18	17	2	2	85	95	10	7	2	123	172	3	7	1	76
19	18	4	2	65	96	11	2	1	35	173	4	2	1	35
20	19	10	3	45	97	12	2	1	50	174	5	2	2	33
21	20	4	1	70	98	13	4	3	50	175	6	2	3	77
22	0	2	1	70	99	14	2	3	55	176	7	2	2	40
23	1	7	1	50	100	15	2	2	90	177	8	4	3	79
24	2	2	3	95	101	16	4	3	25	178	9	2	1	55
25	3	7	3	70	102	17	7	1	35	179	10	2	3	52
26	4	4	2	75	103	18	2	3	90	180	11	2	3	56
27	5	10	1	25	104	19	2	3	60	181	12	4	2	52
28	6	2	1	85	105	20	4	3	40	182	13	4	3	35
29	7	2	3	60	106	0	4	3	34	183	14	4	2	61
30	8	2	3	123	107	1	4	2	74	184	15	2	3	41
31	9	4	1	80	108	2	4	1	64	185	16	4	3	40
32	10	4	3	35	109	3	4	1	78	186	17	4	1	30
33	11	4	3	90	110	4	2	2	79	187	18	4	1	77
34	12	4	3	80	111	5	2	1	56	188	19	2	3	36
35	13	2	1	75	112	6	2	3	61	189	20	7	3	60
36	14	10	2	60	113	7	2	1	80	190	0	4	2	71
37	15	7	3	35	114	8	7	2	80	191	1	2	1	73
38	16	2	1	35	115	9	7	3	34	192	2	7	2	71
39	17	2	1	95	116	10	10	3	56	193	3	2	3	33
40	18	7	3	25	117	11	7	3	75	194	4	2	2	51
41	19	7	3	95	118	12	2	2	77	195	5	4	2	33
42	20	4	1	112	119	13	7	3	74	196	6	4	3	64
43	0	2	1	70	120	14	7	2	34	197	7	4	3	30
44	1	10	3	80	121	15	4	1	71	198	8	2	1	36
45	2	4	1	85	122	16	2	3	73	199	9	2	2	70
46	3	4	2	55	123	17	2	2	49	200	10	7	2	56
47	4	4	1	45	124	18	4	2	54	201	11	2	1	80
48	5	2	2	45	125	19	10	3	40	202	12	2	1	30
49	6	4	2	50	126	20	4	1	62	203	13	4	3	66
50	7	2	1	90	127	0	2	1	33	204	14	2	3	64
51	8	7	3	50	128	1	7	1	42	205	15	2	2	39
52	9	2	3	80	129	2	2	3	78	206	16	4	3	69
53	10	2	1	75	130	3	7	3	34	207	17	7	1	41
54	11	2	3	90	131	4	4	2	40	208	18	2	3	64

55	12	4	1	90	132	5	10	1	52	209	19	2	3	45
56	13	4	1	55	133	6	2	1	38	210	20	4	3	50
57	14	2	1	30	134	7	2	3	36	211	0	4	3	79
58	15	4	2	127	135	8	2	3	46	212	1	4	2	96
59	16	2	2	75	136	9	4	1	79	213	2	4	1	83
60	17	7	2	60	137	10	4	3	53	214	3	4	1	49
61	18	2	3	85	138	11	4	3	57	215	4	2	2	117
62	19	2	2	75	139	12	4	3	77	216	5	2	1	107
63	20	4	1	85	140	13	2	1	55	217	6	2	3	54
64	0	2	3	70	141	14	10	2	63	218	7	2	1	61
65	1	7	2	108	142	15	7	3	44	219	8	7	2	84
66	2	7	2	55	143	16	2	1	60	220	9	7	3	92
67	3	7	1	45	144	17	2	1	60	221	10	10	3	92
68	4	2	1	45	145	18	7	3	63	222	11	7	3	100
69	5	2	2	70	146	19	7	3	38	223	12	2	2	75
70	6	2	3	90	147	20	4	1	32	224	13	7	3	77
71	7	2	2	105	148	0	2	1	32	225	14	7	2	110
72	8	4	3	40	149	1	10	3	46	226	15	4	1	90
73	9	2	1	65	150	2	4	1	64	227	16	2	3	98
74	10	2	3	50	151	3	4	2	80	228	17	2	2	44
75	11	2	3	95	152	4	4	1	70	229	18	4	2	47
76	12	4	2	95	153	5	2	2	74	230	19	10	3	106
77	13	4	3	40	154	6	4	2	34	231	20	4	1	55

Lampiran 4. Dataset: Pangan (Time Series)

Tahun	Bulan	Luas lahan (Ha)	Jml Produksi (Ton)	Jlm Produktifitas	Harga (Rp/Kg)
2001	1	36.610	81.720	22,32	2.150
2001	2	36.845	121.786	22,05	2.040
2001	3	36.630	96.982	24,10	2.008
2001	4	38.252	122.449	26,15	2.058
2001	5	37.957	98.245	25,20	2.021
2001	6	37.096	119.735	22,25	2.057
2001	7	39.682	102.794	25,30	2.007
2001	8	38.944	87.590	26,35	2.030
2001	9	36.026	103.839	25,40	2.027
2001	10	38.301	106.878	27,45	2.013
2001	11	38.808	94.486	24,50	2.037
2001	12	36.163	125.003	23,55	2.015
2002	1	45.718	130.251	28,49	2.050
2002	2	47.532	172.280	28,05	2.083
2002	3	45.500	167.231	30,10	2.063
2002	4	50.192	157.260	31,15	2.041
2002	5	45.614	134.892	28,20	2.091
2002	6	50.068	149.333	28,25	2.018
2002	7	52.695	153.154	29,30	2.021
2002	8	54.209	174.020	30,35	2.007
2002	9	46.861	132.420	29,40	2.073
2002	10	49.253	154.079	30,45	2.014
2002	11	47.933	180.944	31,50	2.064
2002	12	45.920	154.195	31,55	2.099
2003	1	58.716	183.998	31,34	2.155
2003	2	58.014	222.681	32,05	2.094
2003	3	59.824	224.208	34,10	2.055
2003	4	63.507	249.611	31,15	2.007
2003	5	65.782	208.290	31,20	2.090
2003	6	66.839	241.322	31,25	2.017

2003	7	59.649	220.389	33,30	2.039
2003	8	58.784	232.258	31,35	2.042
2003	9	61.503	246.533	34,40	2.011
2003	10	66.136	222.962	32,45	2.096
2003	11	69.772	239.960	33,50	2.069
2003	12	60.426	200.161	31,55	2.077
2004	1	72.529	251.214	34,64	2.003
2004	2	99.615	282.757	37,05	2.039
2004	3	86.889	291.353	34,10	2.092
2004	4	93.747	379.767	34,15	2.049
2004	5	85.943	324.292	34,20	2.089
2004	6	90.563	353.426	36,25	2.054
2004	7	91.957	287.751	36,30	2.002
2004	8	94.533	375.151	34,35	2.023
2004	9	75.911	362.417	37,40	2.082
2004	10	84.538	268.427	37,45	2.027
2004	11	98.721	367.087	34,50	2.042
2004	12	94.089	278.662	36,55	2.088
2005	1	107.752	400.046	37,13	2.100
2005	2	108.932	404.418	38,05	2.197
2005	3	107.794	408.891	38,10	2.162
2005	4	108.182	402.116	38,15	2.187
2005	5	108.177	406.228	38,20	2.108
2005	6	108.518	411.587	38,25	2.115
2005	7	108.474	414.311	38,30	2.131
2005	8	108.592	400.307	38,35	2.165
2005	9	108.551	412.879	38,40	2.130
2005	10	108.838	407.521	38,45	2.157
2005	11	108.819	410.346	38,50	2.181
2005	12	107.049	409.079	38,55	2.200
2006	1	109.792	416.222	37,91	2.220
2006	2	112.278	460.645	44,05	2.325
2006	3	118.532	517.046	38,10	2.490
2006	4	109.320	462.861	46,15	2.515
2006	5	116.946	469.732	39,20	2.385
2006	6	118.979	489.930	45,25	2.538
2006	7	112.061	431.325	39,30	2.498
2006	8	118.372	511.053	42,35	2.473
2006	9	109.118	554.378	37,40	2.567
2006	10	117.082	521.139	39,45	2.532
2006	11	118.185	538.847	47,50	2.311
2006	12	117.919	491.712	41,55	2.465
2007	1	119.027	572.785	48,12	2.604
2007	2	119.858	594.422	49,05	2.889
2007	3	132.060	588.286	49,10	3.061
2007	4	138.944	619.612	49,15	2.800
2007	5	127.856	588.414	49,20	2.778
2007	6	135.435	715.561	49,25	3.010
2007	7	123.196	613.740	49,30	2.774
2007	8	137.756	728.408	49,35	2.924
2007	9	130.985	641.500	49,40	2.665
2007	10	149.400	733.000	49,45	2.641
2007	11	144.094	664.750	49,50	2.857
2007	12	142.095	641.222	49,55	2.920
2008	1	156.436	753.598	48,17	3.123
2008	2	145.273	676.884	45,05	3.392
2008	3	137.020	583.163	47,10	3.246

2008	4	144.078	678.483	47,15	3.191
2008	5	135.492	663.603	48,20	3.325
2008	6	130.099	609.807	46,25	3.219
2008	7	145.009	571.802	48,30	3.260
2008	8	128.268	579.329	47,35	3.390
2008	9	122.056	745.155	48,40	3.203
2008	10	137.918	569.900	47,45	3.122
2008	11	132.337	589.567	45,50	3.321
2008	12	135.563	656.843	46,55	3.408
2009	1	124.798	569.110	45,60	3.590
2009	2	125.007	659.539	46,05	3.570
2009	3	127.106	627.577	46,10	3.578
2009	4	137.996	594.775	46,15	3.575
2009	5	124.447	630.459	45,20	3.601
2009	6	124.775	640.796	45,25	3.672
2009	7	129.524	664.426	46,30	3.630
2009	8	130.149	607.455	46,35	3.537
2009	9	135.187	579.610	47,40	3.555
2009	10	129.687	591.797	45,45	3.544
2009	11	137.390	622.734	47,50	3.695
2009	12	132.358	611.102	45,55	3.616
2010	1	143.833	679.167	47,22	3.732
2010	2	136.709	670.203	46,05	3.700
2010	3	139.701	610.106	44,10	3.786
2010	4	135.378	628.446	47,15	3.756
2010	5	137.900	638.220	47,20	3.729
2010	6	142.734	647.153	45,25	3.729
2010	7	141.171	665.360	47,30	3.723
2010	8	135.661	659.893	45,35	3.770
2010	9	140.672	621.076	47,40	3.770
2010	10	138.085	635.757	44,45	3.737
2010	11	138.145	674.488	44,50	3.780
2010	12	137.942	665.634	46,55	3.765
2011	1	135.754	605.782	44,62	3.800
2011	2	134.840	621.276	47,05	3.820
2011	3	134.600	616.496	47,10	3.851
2011	4	134.762	634.760	45,15	4.112
2011	5	134.818	619.203	47,20	4.019
2011	6	134.687	605.060	46,25	4.043
2011	7	134.278	638.186	45,30	4.104
2011	8	134.998	632.345	45,35	4.064
2011	9	134.665	635.036	45,40	3.911
2011	10	134.851	622.630	45,45	4.189
2011	11	134.988	619.445	44,50	3.817
2011	12	134.182	623.034	45,55	3.849
2012	1	135.543	644.754	47,57	4.220
2012	2	137.375	662.523	47,05	4.849
2012	3	137.776	664.101	47,10	5.457
2012	4	135.131	651.823	47,15	5.304
2012	5	139.988	647.786	46,20	5.439
2012	6	135.826	652.966	47,25	4.871
2012	7	137.535	653.851	47,30	5.206
2012	8	139.760	652.495	46,35	4.672
2012	9	137.131	653.141	47,40	5.380
2012	10	136.558	651.990	47,45	4.733
2012	11	135.571	650.099	46,50	4.743
2012	12	137.511	651.176	47,55	4.949

2013	1	140.423	669.094	47,65	5.742
2013	2	144.074	696.360	48,05	5.610
2013	3	147.939	681.533	48,10	5.621
2013	4	142.939	709.034	47,15	5.693
2013	5	141.642	714.473	48,20	5.677
2013	6	147.563	692.868	48,25	5.655
2013	7	141.488	714.075	48,30	5.643
2013	8	142.376	679.321	47,35	5.657
2013	9	142.314	692.522	47,40	5.680
2013	10	144.190	713.751	48,45	5.698
2013	11	141.209	708.627	48,50	5.656
2013	12	141.776	685.091	47,55	5.650
2014	1	148.816	719.780	48,37	5.755
2014	2	140.025	687.797	48,05	5.755
2014	3	129.993	644.499	48,10	5.718
2014	4	145.443	684.139	49,15	5.711
2014	5	142.693	664.807	48,20	5.711
2014	6	147.724	706.586	49,25	5.712
2014	7	131.346	704.938	48,30	5.712
2014	8	142.198	718.255	48,35	5.749
2014	9	140.272	680.347	48,40	5.708
2014	10	146.462	678.649	48,45	5.767
2014	11	144.629	691.317	48,50	5.793
2014	12	146.367	689.515	49,55	5.744
2015	1	129.131	643.512	49,83	5.804
2015	2	129.540	665.676	48,05	5.811
2015	3	129.817	652.369	48,10	5.898
2015	4	129.321	656.566	48,15	5.894
2015	5	129.882	644.683	48,20	5.833
2015	6	129.150	655.320	49,25	5.809
2015	7	129.259	666.526	48,30	5.806
2015	8	129.708	667.435	48,35	5.893
2015	9	129.111	663.616	48,40	5.845
2015	10	129.321	649.460	49,45	5.848
2015	11	129.815	652.496	48,50	5.841
2015	12	129.924	649.327	49,55	5.900

Lampiran 5. Dataset: Transaksi Penjualan Obat

No.	Obat					
1	Masker					
2	Masker					
3	Masker					
4	Cefixim	Neurosanbe	Sagestam			
5	Cefadroxil	Ritez	Sagestam	Cerini		
6	Dexocart	Sagestam	Cerini			
7	Catherine	Desox	Pot			
8	Metocl	Antasida				
9	Cefadroxil	Mefenamat	Ranitidine	Nacl		
10	Cefixim	Ranitidine	Ambroxol			
11	Catherine	Desox	Sagestam	Pot		
12	Desox	Top Cort	Catherine	Pot	Lemodex	Sotatic
13	Catherine	Top Cort	Desox	Pot	Sotatic	
14	Desox	Sotatic	Catherine	Pot	Betameth	
15	Sagestam					
16	Amlodipine	Dulcolax	Ketoconazole			
17	Asering	Infus	Ranitidine	Domperidon	Antasida	Oralit

18	Infus	Nacl	Neurosanbe	Ranitidine	Ceftriaxon	Curcuma	Codein
19	Domperidon	Antasida					
20	Nacl						
21	Tetagam	Lidocain	Nacl	Ciprofloxacin	Mefenamat		
22	Amlodipine						
23	Cefixim	Metronidazole	Mefenamat				
24	Noroid	Desox	Pot	Nacl			
25	Desoxi	Pot	Cerini				
26	Top Cort	Sagestam	Pot	Catherine			
27	Top Cort	Sagestam	Pot	Catherine			
28	Pot	Desox	Microlax	Catherine	Hydroc		
29	Ketoconazole	Hydroc	Pot	Myconazole			
30	Desox	Pot	Catherine				
31	Colergis	Cinolon					
32	Funtas	Fargoxin					
33	Codein	Mefenamat	Cefixim				
34	Paracetamol	Cefadroxil					
35	Fenofibrate						
36	Mefinal						
37	Dopamed	Mefenamat					
38	Mefinal	Ibuprofen					
39	Amlodipine						
40	Nacl	Lidocain	Cefadroxil	Mefenamat			
41	Dopamed	Mefenamat	Mefinal	Ibuprofen			
42	Amlodipine						
43	Nacl	Lidokain	Cefadroxil	Mefenamat			
44	Mefinal						
45	Dexam						
46	Ambroxol	Oralit	Catherine				
47	Dexam	Sagestam	Pot	Catherine	Cefadroxil		
48	Clobesan						
49	Sodermix	Top Cort	Catherine				
50	Sagestam	Cinolon	Catherine	Cefadroxil			
51	Dexam						
52	Ambroxol	Oralit	Catherine				
53	Dexocart	Sagestam	Pot	Catherine	Cefadroxil		
54	ProTb3Kid						
55	Cerini	Scabimite	Sagestam				
56	Cerini	Desoxi	Pot				
57	Catherine	Metformin	Desox	Vaseline	Hydroc	Pot	
58	Catherine	Scabimite	Hydroc	Sagestam			
59	ProTb3Kid						
60	Cerini	Scabimite	Sagestam				
61	Cerini	Dexoxi	Vaseline	Pot			
62	Cetirizine	Metronidazole	Desox	Vaseline	Hydroc	Vaseline	Pot
63	Catherine	Ketoconazole	Pot				
64	Catherine	Scabimite	Hydroc	Sagestam	Pot		
65	Ritez	Ketoconazole	Hydroc	Sagestam	Pot		
66	Prolac	Desox	Sagestam	Pot			
67	Ritez	Griseofulvin	Lactacyd				
68	Loratadine	Ketoconazole	Formyco	Pibaksin Oint			
69	Cerini	Metronidazole	Desox	Vaseline	Pot		
70	Top Cort	Sagestam	Pot				
71	Celfera	Domperidon	Apazole	Pirofel			
72	Catherine	Dipasolon					
73	Top Cort	Vaseline	Pot				
74	Masker						

75	Catherine	Ketoconazole	Top Cort	Pot			
76	Masker						
77	Cerini	Dipasolon					
78	Masker						
79	Probiokid						
80	Masker						
81	Catherine	Formyco	Formyco	Ketoconazole	Pot		
82	Cerini						
83	Cerini						
84	Nacl	Cefadroxil	Ciprofloxacin	Metronidazole	Kidmin		
85	Masker						
86	Antasida						
87	Masker						
88	Cetirizine	Formyco	Formyco	Ketoconazole	Pot		
89	Probiokid						
90	Neurosanbe	Dipasolon					
91	Sagestam						
92	Cefadroxil	Sagestam					
93	Catherine	LFC	Formyco	Ketoconazole	Pot		
94	Probiokid	Neurosanbe					
95	Masker						
96	Paracetamol						
97	Captopril						
98	Vitamin C						
99	Sodermix						
100	Paracetamol						
101	Infus	Nacl	Ranitidine				
102	Mefenamat						
103	Ranitidine						
104	Masker						
105	Masker						
106	Betamol	Antalgin					
107	Ciprofloxacin						
108	Masker						
109	Nerofa	Prednison					
110	Masker						
111	Mefenamat						
112	Ringer L	Infus	Cerini	Protocin	Kateter	Mannitol	Urine Bag
113	Diatabs						
114	Tremenza						
115	Ranitidine	Domperidon					
116	Domperidon	Amoxicillin					
117	Codein	Formyco	Cetinal	Pot	Sulfur		
118	Sagestam	Desox	Pot	Cetinal	Sulfur		
119	Cefadroxil	Cinolon	Colergis	Sagestam	Pot		
120	Cefadroxil	Top Cort	Sagestam	Pot	Cerini		
121	Sannol	Mefenamat	Ciprofloxacin				
122	Ketoconazole	Formyco	Hydroc	Pot	Cetirizine		
123	Desox	Cefadroxil	Cerini	Desox	Sagestam		
124	Mefenamat	Norelut	Ranitidine	Cefixim			
125	Desox	Cerni	Dipasolon				
126	Top Cort	Desox	Vaseline	Pot	Cerini		
127	Cinolon	Sagestam	Cetinal	Pot			
128	Cerini	Desox	Sagestam	Pot			
129	Methyl P	Ranitidine	Desox				
130	Betamol						
131	Mefenamat	Ranitidine	Neurosanbe				

132	Sagestam						
133	Cinolon	Sagestam	Pot				
134	Formyco	Microlax	Pot				
135	Cetinal	Desox	Sagestam	Pot	Cefadroxil		
136	Mefurosan	Hydroc					
137	Prolic	Sagestam	Cinolon	Pot			
138	Codein	Cefadroxil					
139	Cerini	Myconazole	Pot	Catherine			
140	Sagestam	Mefurosan	Hydroc	Pot			
141	Salbutamol	Ambroxol	Cetirizine	Surplus	Ventolin		
142	Mefenamat	Meloxicam	Omeprazole				
143	Azytromicin	Cetinal	Desox	Sagestam	Pot		
144	Cetirizine	Dextrom	Sagestam	Pot			
145	Cetirizine	Sagestam	Desox	Pot			
146	Sagestam	Cinolon	Pot	Colergis	Cefotaxime		
147	Desox	Sagestam	Pot				
148	Cefotaxime	Sagestam	Cerini	Desox	Pot		
149	Cerini	Top Cort	Desox	Pot			
150	Cerini	Desox	Pot				
151	Aminophyllin	Noroid	Mefenamat				
152	Cefotaxime	Cefadroxil	Decoxin	Sagestam	Pot		
153	Mefurosan	Sagestam	Pot				
154	Cerini	Desox	Pot				
155	Microlax						
156	Cefotaxime	Dipasolon	Ranitidine				
157	Ranitidine						
158	Amoxan						
159	Asering	Infus	Zink	Paracetamol	Ondansentron		
160	Tetagam	Ciprofloxacin	Mefenamat	Nacl			
161	Transfusi	Kateter	Umbilikalis	Suction	Neo - K	Dispo	
162	Dexam						
163	Dexam	Vitamin C	Tonotan				
164	Aminofusin	Ranitidine	Mefenamat	Neropin			
165	Cetadop	Cefadroxil	Gentamicin	Pot			
166	Microlax	Top Cort	Sagestam	Cerini			
167	Top Cort	Desox	Sagestam	Pot	Sulfur		
168	Top Cort	Sagestam	Pot	Catarlent			
169	Hydroc	Funtas	Pot				
170	Sanmol	Cefadroxil	Amoxicillin				
171	Cetirizine	Formyco	Ketorolac	Pot			
172	Codein						
173	Cetirizine	Formyco	Hydroc	Pot			
174	Catherine	Formyco	Ketorolac	Pot			
175	Traneksamat	Ulsafat	Ondansentron	Allopurinol			
176	Catherine	Formyco	Formyco	Ketorolac	Pot		
177	Ciprofloxacin	Codein	Nerofa				
178	Catherine	Scabimite	Formyco				
179	Cefadroxil	Mefenamat	Nerofa				
180	Spasmolit						
181	Masker						
182	Betamol						
183	Masker						
184	Levopar						
185	Acylovir	Acyclovir	Paracetamol	Amoxicillin			
186	Erytromycin	Vitamin C					
187	Paracetamol	Cefixim	Meloxicam				
188	Masker						

189	Masker							
190	Cerini	Sagestam	Scabimite					
191	Cetirizine							
192	Cetirizine							
193	Cetirizine							
194	Cetirizine	Sagestam						
195	Loratasine	Ketokonazole						
196	Amoxilin	Vitamin C						
197	Dexocart							
198	Catherine							
199	Vitamin C	Culk						
200	Catherine							
201	Ketokonazole	Pot						
202	Catherine							
203	D 3	Lidocain						
204	Amoxilin	Mefenamat						
205	Tetagam	Cefadroxil	Mefenamat	Neurodex				
206	Cateter	Aqua	D 10					
207	Valesco	Yosenob						
208	Catherine	Cefadroxil	Desoxi	Sagestam	Pot			
209	Catherine	Hydroc	Sagestam	Pot				
210	Loratadine	Hydroc	Sagestam	Pot				
211	Catherine	Desoxi	Sagestam	Pot				
212	Dexam	Catherine						
213	Antasida	Spasmolit	Pcr Tab	Omeprazole				
214	Amoxicillin	Sanmol						
215	Amoxicillin	As Mef Tab	Lidocain	D 3				
216	Cefadroxil	Traneksamat						
217	Cefadroxil	Aqua	D 10	Sanmol				
218	Sanmol	Cefadroxil						
219	Vitamin C							
220	Mefenamat	Dexam	Ranitidine					
221	Sanmol	Rhinof Tab						
222	Cefadroxil	Mefenamat	Methyl P					
223	Amoxicillin	Mefenamat						
224	Hydroc	Ambroxol						
225	Nacl	Lidocain	Tetagam	Ciprofloxacin	Mefenamat			
226	Mefenamat	Amoxicillin	Nacl	Cromic				
227	Antasida	Domperidon						
228	Nacl	Mefenamat	Ciprofloxacin					
229	Infus	Ranitidine	Biocombin	Metocl				
230	Nacl	Mefenamat	Ciprofloxacin					
231	Abct 20	Infus	Ranitidine	Metocl	Biocombin			
232	Catherine	Top Cort	Pot					
233	Mefenamat	Cefadroxil	Metronidazole					
234	Formyco	Formyco	Ketoconazole					
235	Spasmolit	Pct	Nerofa					
236	Cefadroxil	Mefenamat						
237	Ampicillin	Amoxicillin						
238	Ambroxol	Paracetamol	Cefadroxil					
239	Salbutamol	Ambroxol	Paracetamol	Cefadroxil				
240	Ampicillin	Amoxicillin						
241	Cefadroxil	Sagestam	Ketoconazole					

Biografi Penulis

Budy Santoso

Sejak Agustus 2009 mengabdi sebagai dosen di Program Studi Teknik Informatika Fakultas Ilmu Komputer UNISAN Gorontalo. Lulus S1 dari Teknik Informatika STMIK Dipanegara Makassar pada tahun 2009. Selanjutnya mendapat gelar *Master of Engineering* dari Departemen Teknik Elektro dan Teknologi Informasi (DTETI) UGM Yogyakarta pada tahun 2016. Penulis juga aktif melakukan penelitian di bidang *Machine Learning* dan *Internet of Thing*. Selain sebagai dosen, penulis juga aktif mengerjakan proyek-proyek di bidang *web programming*. Para pembaca dapat menghubungi, berkomunikasi, berdiskusi, atau bertanya mengenai isi buku ini melalui budinho.jr@gmail.com.



Azminuddin I. S. Azis

Lahir di Teminabuan, 14 Oktober 1979 dari rahim ibunya yang bernama Miriati Samir yang didampingi oleh ayahnya yang bernama Abd. Azis Na'ga. Menikah dengan Iin Novianty Rais Rauf pada tahun 2014, lalu dikaruniai seorang anak laki-laki bernama Mushlih Azis. Menyelesaikan pendidikan S1 jurusan Sistem Informasi di STMIK Dipanegara Makassar pada tahun 2008. Menyelesaikan pendidikan S2 jurusan Teknik Informatika di UDINUS Semarang pada tahun 2013. Sejak tahun 2010 hingga sekarang aktif sebagai dosen untuk mata kuliah *Machine Learning*, *Soft Computing*, *Data Mining*, dan *Object Oriented Programming* di Fakultas Ilmu Komputer UNISAN Gorontalo. Selain aktif sebagai dosen, menulis buku dan meneliti di bidang *computer science* merupakan aktifitas lainnya. Dua bukunya yang memperoleh insentif buku ajar dari Kementerian Riset, Teknologi, dan Pendidikan Tinggi yaitu “Fundamental Pemrograman” dan “Karakteristik Penelitian Ilmu Komputer.” Freelancer di bidang *Software Engineering* merupakan profesi lainnya, yang mana hingga saat ini sudah puluhan proyek pengembangan *software* untuk instansi-instansi pemerintah maupun perusahaan-perusahaan swasta di Makassar, Gorontalo, dan Mamuju yang telah diselesaikan. Para pembaca dapat menghubungi, berkomunikasi, berdiskusi, atau bertanya mengenai isi buku ini melalui azdi.azminuddinazis@gmail.com atau youtube/fb/ig: Azminuddin Azis.



Zohrahayaty

Lahir di Enrekang, 12 November 1977. Menyelesaikan pendidikan S1 jurusan Teknik Informatika di STMIK Handayani Makassar pada tahun 2001. Menyelesaikan pendidikan S2 jurusan Teknik Informatika di UDINUS Semarang pada tahun 2011. Saat ini sebagai Dekan Fakultas Ilmu Komputer UNISAN Gorontalo dan aktif mengajar mata kuliah Metodologi Penelitian Ilmu Komputer, Interaksi Manusia dan Komputer, serta Sistem Informasi. Selain aktif sebagai dosen, menulis buku dan meneliti di bidang *computer science* merupakan aktifitas lainnya. Salah satu bukunya yang memperoleh insentif buku ajar dari Kementerian Riset, Teknologi, dan Pendidikan Tinggi yaitu “Karakteristik Penelitian Ilmu Komputer.” Para pembaca dapat menghubungi, berkomunikasi, berdiskusi, atau bertanya mengenai isi buku ini melalui zohrahayaty123@gmail.com.

