

# **JARINGAN SARAF FUZZY UNTUK MENYELESAIKAN MASALAH KLASIFIKASI MULTIKELAS**

## **TUGAS AKHIR**

**Karya tulis sebagai salah satu syarat  
untuk memperoleh gelar Sarjana dari  
Institut Teknologi Bandung**

**Oleh**

**SYAHRUL ROFI**

**NIM: 10115077**

**(Program Studi Sarjana Matematika)**



**INSTITUT TEKNOLOGI BANDUNG**

**Juni 2019**

## **ABSTRAK**

# **JARINGAN SARAF FUZZY UNTUK MENYELESAIKAN MASALAH KLASIFIKASI MULTIKELAS**

Oleh

**Syahrul Rofi**

**NIM: 10115077**

Konstruksi program pembelajaran mesin yang dapat menyelesaikan masalah klasifikasi multikelas dipelajari pada tugas akhir ini. Jaringan saraf fuzzy, yaitu jaringan saraf tiruan yang proses *feed-forward*-nya berupa tahap-tahap dari sistem kontrol logika fuzzy yang dibangun dari data masukan dan keluaran yang tersedia, akan digunakan sebagai model pembelajaran mesin. Tahap-tahap tersebut adalah tahap fuzzifikasi, inferensi fuzzy, dan defuzzifikasi. Tahap inferensi fuzzy membutuhkan aturan fuzzy dan hasil dari tahap fuzzifikasi. Aturan fuzzy terdiri dari beberapa implikasi fuzzy yang saling berkaitan. Untuk mendapatkan model jaringan saraf fuzzy, dibutuhkan dua fase: identifikasi struktur dan identifikasi parameter. Fase identifikasi struktur menghasilkan struktur awal dari aturan fuzzy yang meliputi banyaknya implikasi pada aturan fuzzy dan nilai awal dari setiap parameter yang terlibat di dalam model jaringan saraf fuzzy. Fase identifikasi parameter melakukan perbaikan nilai-nilai dari parameter yang terlibat di dalam model. Untuk mendapatkan model jaringan saraf fuzzy yang paling optimal, diperlukan proses validasi silang secara berulang. Proses validasi silang menghasilkan rata-rata akurasi untuk suatu model. Model dengan proses validasi silang yang menghasilkan rata-rata akurasi terbesar merupakan model yang paling optimal. Untuk tugas akhir ini, dilakukan simulasi terhadap tiga jenis data, yaitu: data koordinat kartesius, data tanaman iris, dan data evaluasi mobil. Akan diterapkan tiga skema klasifikasi multikelas, yaitu: satu lawan semua, satu lawan satu, dan satu lawan orde yang lebih tinggi. Model jaringan saraf fuzzy yang diperoleh dengan skema klasifikasi satu lawan semua mempunyai akurasi lebih dari 95%. Model jaringan saraf fuzzy dengan skema klasifikasi satu lawan satu mempunyai akurasi setidaknya sebesar 94%. Model jaringan saraf fuzzy dengan skema klasifikasi satu lawan orde yang lebih tinggi mempunyai akurasi lebih dari 84%.

Kata kunci: pembelajaran mesin, jaringan saraf fuzzy, identifikasi struktur, identifikasi parameter, klasifikasi multikelas, validasi silang

## ABSTRACT

# FUZZY NEURAL NETWORK FOR SOLVING MULTI-CLASS CLASSIFICATION PROBLEM

by

**Syahrul Rofi**

**NIM: 10115077**

*A construction of machine learning programs that can solve multi-class classification problems is studied in this final project. Fuzzy neural network, i.e. an artificial neural network whose feed-forward process is the stages of fuzzy logic control system that is constructed based on available input and output data, will be used as machine learning models. The stages are fuzzification, fuzzy inference, and defuzzification. The stage of fuzzy inference requires fuzzy rules and the results of fuzzification stage. Fuzzy rules consist of several interrelated fuzzy implications. Two phases are required to get the fuzzy neural network model: structure identification and parameter identification. The structure identification phase produces the initial structure of fuzzy rules which includes number of implications for fuzzy rules and the initial values of each parameter involved in the fuzzy neural network model. The parameter identification phase refines the values of the parameters involved in the model. To get the most optimal fuzzy neural network model, a cross-validation process is needed repeatedly. The cross validation process produces an average accuracy for a model. The model with a cross validation process that produces the highest average accuracy is the most optimal model. For this final project, three types of data are simulated, i.e.: cartesian coordinate data, iris plant data, and car evaluation data. Three schemes of multi-class classification are applied: one against all, one against one, and one against higher order. The fuzzy neural network model obtained with a one against all classification scheme has an accuracy of more than 95%. The fuzzy neural network model obtained with a one against one classification scheme has an accuracy of at least 94%. The fuzzy neural network model obtained with a one against higher order scheme has an accuracy of more than 84%.*

*Keywords: machine learning, fuzzy neural network, structure identification, parameter identification, multi-class classification, cross validation*

# **JARINGAN SARAF FUZZY UNTUK MENYELESAIKAN MASALAH KLASIFIKASI MULTIKELAS**

Oleh

**Syahrul Rofi**

**NIM: 10115077**

**Program Studi Sarjana Matematika  
Institut Teknologi Bandung**

Menyetujui

Pembimbing

Tanggal 21 Juni 2019

---

(Dr. Agus Yodi Gunawan)

*Dipersembahkan untuk Mamah dan Bapa tersayang.*

## KATA PENGANTAR

*Bismillahirrahmanirrahim.*

*Ahmadu rabbillaha khaira maliki. Mushalliyan 'alannabiyyil musthafa. Wa alihil mustakmilinasy syarafa.*

Puji dan syukur penulis panjatkan kepada Allah SWT Yang Maha Merajai. Melalui qadha dan qadar-Nya, penulis dapat menyelesaikan tugas akhir dengan judul **Jaringan Saraf Fuzzy untuk Menyelesaikan Masalah Klasifikasi Multikelas** sebagai salah satu syarat kelulusan dari Program Studi Sarjana Matematika Institut Teknologi Bandung.

Tugas akhir ini membahas salah satu model pembelajaran mesin, yaitu jaringan saraf fuzzy. Pada praktiknya, proses pembangunan model jaringan saraf fuzzy ini menerapkan beberapa bidang ilmu matematika, di antaranya: konsep fuzzy, optimisasi, dan statistika. Penulis mengharapkan topik tugas akhir ini dapat dikembangkan lebih jauh lagi dan hasil penelitian yang diperoleh bisa lebih baik dari hasil penelitian pada tugas akhir ini.

Berbagai pihak telah membantu penulis dalam penyelesaian tugas akhir ini, baik secara langsung maupun tidak langsung. Penulis menyebut berbagai pihak ini sebagai *supporting system*. Penulis ingin memberikan apresiasi kepada *supporting system* tersebut dengan cara mengabadikan nama-nama mereka di dalam tugas akhir ini. *Supporting system* bagi penulis adalah sebagai berikut:

1. Kedua orang tua penulis, *Mamah* Ida Rosidah dan *Bapa* Abdul Rohim. Orang tua yang tidak terlalu banyak menuntut kepada anak-anaknya, hanya meminta anak-anaknya supaya tidak melupakan ibadah wajib. Penulis mengucapkan terima kasih kepada *Mamah* dan *Bapa* atas segalanya yang telah *Mamah* dan *Bapa* berikan: doa orang tua yang terus menerus, bantuan morel dan materiel, serta bimbingan dalam menjalani kehidupan.
2. Bapak Dr. Agus Yodi Gunawan sebagai dosen pembimbing Tugas Akhir penulis. Terima kasih telah menginisiasi pemahaman penulis mengenai konsep fuzzy dan telah menjadi tempat bertanya yang selalu ada jawabannya bagi penulis.
3. Bapak Dr. Spto Wahyu Indratno, M.Si. sebagai dosen penguji pada Seminar Tugas Akhir I penulis. Terima kasih telah membuka pemahaman penulis mengenai *big data* dan pembelajaran mesin.

4. Bapak Prof. Dr. Marcus Wono Setya Budhi dan Ibu Dr. Finny Oktariani, M.Si. sebagai dosen penguji pada Seminar Tugas Akhir II penulis. Terima kasih atas saran dan masukan yang telah Bapak dan Ibu berikan, sehingga isi tugas akhir ini bisa menjadi lebih baik dari sebelum seminar.
5. Bapak Warsoma Djohan, M.Si. sebagai dosen yang menerima penulis menjadi asistennya untuk matakuliah Pengantar Teknologi Informasi B. Terima kasih telah menyokong penulis dalam pengerjaan tugas akhir ini melalui fasilitas yang Bapak berikan.
6. Bapak Prof. Dr. M. Salman A.N., M.Si. sebagai dosen wali penulis. Terima kasih telah memberikan wejangan kepada penulis selama penulis belajar di Program Studi Sarjana Matematika ITB.
7. Simpang *Flat Earth*: Nyahmet, Rangga, Haris, Dancent, Firli, Febi, Andre, dan Arjun. Terima kasih telah bersama-sama dalam suka dan duka selama tiga tahun ini. Meminjam perkataan dari Nyahmet: semoga Tuhan mengampuni kita semua.
8. Teman-teman penulis yang cukup sering menghabiskan waktu bersama di lab: Adnan, Rizka, Aul, Manda, Clarissa, Fadhel, dan Raka. Terima kasih telah saling menemani dalam proses pengerjaan tugas akhir.
9. Semua teman-teman FRACTAL Matematika ITB 2015. Terima kasih telah bersama-sama menjalani perkuliahan di Matematika ITB.
10. HIMATIKA ITB, terutama Badan Pengurus HIMATIKA ITB 2018/2019, Tim Futsal HIMATIKA ITB, Pendiklat Sekolah Komandan Lapangan (Sekdan) HIMATIKA ITB 2017, dan teman-teman sesama peserta sekdan. Terima kasih telah memberikan ruang bagi penulis untuk aktualisasi diri dan menginisiasi penulis untuk terus berproses menjadi manusia seutuhnya.
11. Adik penulis, Mohammad Rizky Zakary. Terima kasih telah menemani penulis pada saat jenuh dari proses penulisan tugas akhir ini. Semoga kita bisa terus membahagiakan *Mamah* dan *Bapa, De*.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis akan sangat berterima kasih jika ada yang menyampaikan kritik dan saran mengenai tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi pembaca. Terima kasih

Bandung, Juni 2019

Penulis

# DAFTAR ISI

ABSTRAK .....	ii
<i>ABSTRACT</i> .....	iii
HALAMAN PENGESAHAN .....	iv
HALAMAN PERUNTUKAN .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR LAMPIRAN .....	ix
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
DAFTAR SINGKATAN DAN LAMBANG .....	xii
 Bab I    Pendahuluan .....	 1
Bab II   Jaringan Saraf Fuzzy .....	5
II.1   Himpunan Fuzzy .....	5
II.2   Operasi pada Himpunan Fuzzy .....	7
II.3   Logika Fuzzy .....	11
II.4   Aturan Fuzzy dan Inferensi Fuzzy .....	18
II.5   Sistem Kontrol Logika Fuzzy .....	20
II.6   Konstruksi Jaringan Saraf Fuzzy .....	31
Bab III   Skema Pembelajaran Mesin .....	47
III.1   Definisi dan Istilah dalam Pembelajaran Mesin .....	47
III.2   Pra Pemrosesan Data .....	49
III.3   Skema Klasifikasi Multikelas .....	56
III.4   Skema Pemilihan Model .....	59
Bab IV   Pengujian Model dan Skema .....	62
IV.1   Konstruksi Program .....	62
IV.2   Deskripsi dan Analisis Data .....	68
IV.3   Hasil Uji .....	72
Bab V    Kesimpulan dan Saran .....	89
V.1   Kesimpulan .....	89
V.2   Saran .....	89
DAFTAR PUSTAKA .....	91
LAMPIRAN .....	93



## DAFTAR LAMPIRAN

Lampiran A	Data Koordinat I . . . . .	94
Lampiran B	Data Koordinat II . . . . .	95
Lampiran C	Data Tanaman Iris . . . . .	96
Lampiran D	Data Evaluasi Mobil . . . . .	97

## DAFTAR GAMBAR

Gambar II.1	Fungsi keanggotaan untuk “ $x$ dekat dengan 3” . . . . .	7
Gambar II.2	Sistem kontrol logika fuzzy . . . . .	21
Gambar II.3	Jaringan saraf fuzzy . . . . .	33
Gambar II.4	Dua fase pada konstruksi jaringan saraf fuzzy . . . . .	34
Gambar II.5	Diagram alir identifikasi struktur . . . . .	42
Gambar II.6	Diagram alir identifikasi parameter . . . . .	46
Gambar IV.1	Tampilan awal program . . . . .	62
Gambar IV.2	Pilihan yang disediakan oleh program pada tahap pra pemrosesan data . . . . .	63
Gambar IV.3	Pemilihan skema klasifikasi multikelas, validasi silang, dan hiperparameter . . . . .	64
Gambar IV.4	Proses pembangunan dan pengujian model jaringan saraf fuzzy . . . . .	66
Gambar IV.5	Proses penyimpanan hasil uji . . . . .	67

## DAFTAR TABEL

Tabel II.1	Tabel <i>t-norm</i> dan <i>t-conorm</i> dasar . . . . .	9
Tabel II.2	Tabel kebenaran untuk implikasi pokok . . . . .	15
Tabel II.3	Operator implikasi fuzzy . . . . .	17
Tabel II.4	Contoh mekanisme inferensi Mamdani . . . . .	26
Tabel III.1	Contoh data latih yang belum diurutkan . . . . .	55
Tabel III.2	Hasil pengurutan fitur untuk masing-masing observasi . . .	55
Tabel III.3	Contoh data latih yang telah diurutkan . . . . .	55
Tabel IV.1	Statistika deskriptif untuk setiap fitur dari data koordinat I .	69
Tabel IV.2	Statistika deskriptif untuk setiap fitur dari data koordinat II	69
Tabel IV.3	Statistika deskriptif untuk setiap fitur dari data tanaman iris	70
Tabel IV.4	Kardinalitas setiap kategori pada setiap fitur dari data evaluasi mobil . . . . .	71
Tabel IV.5	Kardinalitas label kelas pada data koordinat I . . . . .	73
Tabel IV.6	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat I dengan skema OAA . . . . .	74
Tabel IV.7	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat I dengan skema OAO . . . . .	74
Tabel IV.8	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat I dengan skema OAHO . . . . .	75
Tabel IV.9	Kardinalitas label kelas pada data koordinat II . . . . .	76
Tabel IV.10	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat II dengan skema OAA . . . . .	77
Tabel IV.11	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat II dengan skema OAO . . . . .	79
Tabel IV.12	Hasil validasi silang <i>5-fold</i> data latih pada data koordinat II dengan skema OAHO . . . . .	80
Tabel IV.13	Kardinalitas label kelas pada data tanaman iris . . . . .	81
Tabel IV.14	Hasil validasi silang <i>5-fold</i> data latih pada data tanaman iris dengan skema OAA . . . . .	82
Tabel IV.15	Hasil validasi silang <i>5-fold</i> data latih pada data tanaman iris dengan skema OAO . . . . .	83
Tabel IV.16	Hasil validasi silang <i>5-fold</i> data latih pada data tanaman iris dengan skema OAHO . . . . .	84
Tabel IV.17	Kardinalitas label kelas pada data evaluasi mobil . . . . .	85
Tabel IV.18	Hasil validasi silang <i>5-fold</i> data latih pada data evaluasi mobil dengan skema OAA . . . . .	86
Tabel IV.19	Hasil validasi silang <i>5-fold</i> data latih pada data evaluasi mobil dengan skema OAO . . . . .	86
Tabel IV.20	Hasil validasi silang <i>5-fold</i> data latih pada data evaluasi mobil dengan skema OAHO . . . . .	87
Tabel IV.21	Ringkasan performa model jaringan saraf fuzzy . . . . .	88

## DAFTAR SINGKATAN DAN LAMBANG

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
GUI	<i>Graphical User Interface</i>	3
JSF	Jaringan Saraf Fuzzy	5
JST	Jaringan Saraf Tiruan	1
ML	<i>Machine Learning</i>	1
MSE	<i>Mean Square Error</i>	34
NaN	<i>Not a Number</i>	52
OAA	<i>One Against All</i>	57
OAHO	<i>One Against Higher Order</i>	58
OAQ	<i>One Against One</i>	57
PCA	<i>Principal Component Analysis</i>	53
SKLF	Sistem Kontrol Logika Fuzzy	20
SML	<i>Supervised Machine Learning</i>	1
SVD	<i>Singular Value Decomposition</i>	39
t-conorm	<i>Triangular co-norm</i>	8
t-norm	<i>Triangular norm</i>	8
TSK	Takagi-Sugeno-Kang (dalam aturan fuzzy dan mekanisme inferensi fuzzy)	18

LAMBANG	Nama	Pemakaian pertama kali pada halaman
$S$	Operator <i>t-conorm</i>	9
$T$	Operator <i>t-norm</i>	8
$\mathfrak{R}_i$	Implikasi ke- $i$ pada suatu aturan fuzzy	19
$\cap$	Operator irisan	7
$\cup$	Operator gabungan	7
$\eta$	<i>Learning rate</i>	45
$\wedge$	Operator konjungsi	13
$\vee$	Operator disjungsi	13
$\mathbb{N}$	Himpunan bilangan asli	19
$\mathbb{R}^+$	Himpunan bilangan real positif	10
$\mathbb{R}^n$	Ruang vektor bilangan real dengan dimensi $n$	43
$\mathbb{R}$	Himpunan bilangan real	7
$\mathbb{Z}$	Himpunan bilangan bulat	5
$\mu_A$	Fungsi keanggotaan dari himpunan fuzzy $A$	6
$\neg$	Operator negasi	12

LAMBANG	Nama	Pemakaian pertama kali pada halaman
$\rho$	Ambang batas minimal kemiripan data masukan	37
$\rightarrow$	Operator implikasi	14
round	Pembulatan bilangan real menjadi bilangan bulat terdekat	58
$\tau$	Pengali untuk ambang batas maksimal kemiripan data keluaran	37
trace	Penjumlahan semua entri pada diagonal utama dari suatu matriks persegi	39
$C$	Komplemen dari suatu himpunan	7
$s_0$	Inisialisasi untuk nilai simpangan baku, sebagai pengganti nilai simpangan baku yang bernilai nol atau tidak terdefinisi	36

## Bab I Pendahuluan

Pembelajaran mesin atau *machine learning* (ML), sebagaimana didefinisikan oleh Mohri dkk. (2018) sebagai sarana yang memungkinkan komputer untuk memanfaatkan informasi masa lalu dengan tujuan meningkatkan kinerja atau membuat prediksi yang akurat, telah menjadi sarana yang tepat dan menjanjikan di era revolusi industri 4.0. Wuest dkk. (2016) telah berpendapat dari perspektif manufaktur mengenai alasan ML menjadi sarana yang ampuh di era ini. Tantangan utama di era revolusi industri 4.0 adalah terjadinya peningkatan kompleksitas, kedinamisan, dan ketersediaan data. ML dapat mengatasi tantangan-tantangan tersebut karena sarana ini dapat mengatasi masalah dan dataset yang berdimensi tinggi dengan upaya yang masuk akal. ML juga dapat memperoleh pola dari data yang ada dan mendapatkan perkiraan perilaku masa depan (Alpaydn, 2010). Informasi baru ini dapat mendukung pemilik proses dalam pengambilan keputusan atau digunakan untuk meningkatkan sistem secara otomatis (Wuest dkk., 2016).

Salah satu masalah yang dapat ditangani oleh algoritma ML adalah masalah klasifikasi multikelas. Masalah klasifikasi multikelas ini termasuk tugas dari SML (*supervised machine learning*). SML merupakan skenario ML yang dilakukan terhadap data yang setiap observasinya memiliki label tertentu (Mohri dkk., 2018). Pada masalah klasifikasi multikelas, terdapat lebih dari dua label yang berbeda dari semua sampel data, dan setiap labelnya berupa kategori, bukan bilangan real. Data yang hanya memiliki dua label kategori yang berbeda merupakan masalah klasifikasi biner. Model ML yang digunakan untuk menyelesaikan masalah klasifikasi biner di antaranya adalah model pengklasifikasi bayes, model regresi logistik, *k-nearest neighbours*, *support vector machine*, dan lain-lain (Rogers dan Girolami, 2017). Model-model ini dapat diperluas untuk menyelesaikan masalah klasifikasi multikelas dengan cara menggunakan skema pengkodean tertentu terhadap setiap labelnya. Skema pengkodean yang biasa digunakan pada klasifikasi multikelas yaitu skema satu lawan semua, skema satu lawan satu, dan skema satu lawan orde yang lebih tinggi (Ou dan Murphey, 2007).

Model Jaringan Saraf Tiruan (JST), sebagaimana didefinisikan oleh Kriesel (2007)

dan Graupe (2013) sebagai jaringan komputasi yang terdiri dari koneksi yang memiliki nilai bobot antar-neuron dengan tujuan untuk mensimulasikan proses pengambilan keputusan dalam neuron dari sistem saraf biologis, merupakan model pembelajaran mesin yang sangat efisien untuk menangani data yang berukuran sangat besar. Model JST dapat digunakan untuk menyelesaikan masalah klasifikasi multikelas seperti yang telah dilakukan oleh Kraipeerapun dkk. (2006) dan Ou dan Murphey (2007). Model JST sangat sederhana secara komputasi dan algoritma, memiliki fitur pengorganisasian tersendiri yang memungkinkannya menangani berbagai masalah, dan memiliki tingkat keparalelan yang tinggi (Graupe, 2013). JST bekerja dengan cara melakukan transformasi terhadap informasi yang ada secara berlapis-lapis karena JST memiliki beberapa lapisan tersembunyi. Akibatnya, JST dapat mengurangi dan menghapus kontaminasi pada data mentah, sehingga JST berguna dalam pengurangan gangguan (Deng dkk., 2017). Schmidhuber (2015) telah merangkum bahwa model ini juga berhasil mengantarkan penggunaannya memenangkan berbagai kompetisi. JST dengan garis tunda internal pada kekacauan getaran intensitas dari laser NH<sub>3</sub> memenangkan kompetisi *Santa Fe time-series*. JST Bayes berdasarkan ansambel jaringan saraf memenangkan *NIPS 2003 Feature Selection Challenge*. Sebuah sistem yang memuat pertumbuhan JST telah memenangkan kontes *CASP 2012* pada prediksi peta kontak protein.

Sebagaimana telah disebutkan sebelumnya bahwa di era revolusi industri 4.0 jumlah ketersediaan data meningkat secara signifikan. Tetapi, data besar ini mungkin memuat sejumlah gangguan yang tinggi dan ketidakpastian yang tidak dapat diprediksi (Deng dkk., 2017). Jaringan saraf tiruan, sebagaimana telah dijelaskan di paragraf sebelumnya, mampu mengurangi gangguan. Untuk memasukkan unsur ketidakpastian, dibutuhkan model ML yang menggunakan konsep logika fuzzy. Hal ini dikarenakan nilai kebenaran dari suatu pernyataan logika fuzzy bersifat tidak pasti antara benar dan salah (Zadeh, 1965). Lebih jauh lagi, logika fuzzy dapat mengatasi ketidakpastian pada data mentah dengan cara membangun aturan fuzzy secara fleksibel (Deng dkk., 2017).

Dengan demikian, dibutuhkan model ML yang dapat mengatasi kelemahan dari data besar. Model ML yang dibutuhkan ini adalah gabungan antara JST dengan konsep logika fuzzy yang disebut dengan model jaringan saraf fuzzy. Model jaringan saraf fuzzy ini telah digunakan oleh Lee dan Ouyang (2003) dan Yeh dkk. (2012) pada SML yang bukan masalah klasifikasi. Masalah klasifikasi multikelas juga dapat

diselesaikan menggunakan model jaringan saraf fuzzy seperti yang telah dilakukan oleh Ghongade dan Ghatol (2007) dan Rangkuti dkk. (2018).

Berdasarkan latar belakang di atas, tujuan tugas akhir ini adalah sebagai berikut.

- Merancang program yang dapat mengimplementasikan model dan skema ML. Program ini terdiri dari tiga skema klasifikasi multikelas. Setiap skema klasifikasi menggunakan model ML berupa jaringan saraf fuzzy. Program ini dibuat untuk menyelesaikan masalah klasifikasi multikelas.
- Menguji model dan skema ML dengan cara membandingkan akurasi dari tiga skema klasifikasi multikelas untuk setiap data yang menjadi objek pengujian.

Metode yang digunakan dalam penyusunan tugas akhir ini adalah metode deskriptif melalui studi literatur. Studi literatur yang digunakan yaitu dengan cara mempelajari jurnal dan buku. Selain itu, dilakukan juga proses pembangunan model jaringan saraf fuzzy beserta skema ML tertentu berdasarkan studi literatur. Kemudian model jaringan fuzzy dan skemanya ini diimplementasikan ke dalam program. Penulis merancang program dengan bentuk *Graphical User Interface* (GUI) supaya memudahkan dalam proses pembangunan model jaringan saraf fuzzy serta pengujian model dan skemanya. Program ini dirancang menggunakan bahasa pemrograman python versi 3.6.4 dengan bantuan beberapa *package*, di antaranya: *numpy*, *pandas*, *scikit-learn*, dan *tkinter*.

Dengan adanya program yang dirancang melalui tugas akhir ini, maka telah terbentuk program baru yang nantinya dapat digunakan untuk menyelesaikan masalah klasifikasi multikelas dengan model jaringan saraf fuzzy pada data lain di dunia nyata. Jika diperoleh akurasi yang memuaskan, maka akan menambah keyakinan dalam menggunakan program ini untuk membangun dan menguji model jaringan saraf fuzzy dari data lain. Program yang dibuat ini berbentuk GUI, sehingga program ini dapat digunakan oleh siapapun tanpa harus melakukan instalasi python terlebih dahulu.

Supaya data dapat digunakan untuk membangun serta menguji model dan skema ML yang akan dibangun, maka data tersebut harus memenuhi syarat-syarat berikut ini.

- Data harus terdiri dari bagian masukan dan bagian keluaran. Bagian masukan dapat memuat beberapa kolom yang setiap entrinya dapat berupa bilangan real ataupun kategori. Bagian keluaran hanya memuat satu kolom yang setiap



entrinya berupa kategori, bukan bilangan real. Banyaknya kategori yang berbeda pada seluruh bagian keluaran harus lebih dari dua. Hanya terdapat satu kategori yang termuat pada setiap observasi dalam data keluaran.

- Data tidak memuat entri dengan nilai yang kosong.

Semua data yang akan digunakan ini diasumsikan memiliki kolom-kolom pada bagian masukan yang saling bebas. Selain itu, diasumsikan juga setiap sampel pada data keluaran bergantung kepada semua data masukan dari sampel tersebut. Data yang digunakan juga diasumsikan tidak memiliki pencilan.

Tugas akhir ini terdiri dari 5 bab. Bab I memaparkan latar belakang, tujuan penelitian, metode penelitian, manfaat penelitian, batasan masalah, dan asumsi yang digunakan.

Pada Bab II, penulis mengelaborasi model ML yang digunakan, yaitu jaringan saraf fuzzy. Pembahasannya dimulai dari konsep dasar fuzzy.

Pada Bab III, penulis membahas skema ML yang digunakan untuk menyelesaikan masalah klasifikasi multikelas. Skema ML yang dibahas meliputi pra pemrosesan data, skema klasifikasi multikelas, dan skema pemilihan model.

Pada Bab IV, penulis menampilkan dan membahas hasil pengujian model dan skema ML terhadap empat data.

Pada Bab V, akan diuraikan rangkuman hasil penelitian secara keseluruhan dan ditutup dengan saran terkait penelitian ini.

## Bab II Jaringan Saraf Fuzzy

Jaringan saraf fuzzy (JSF) dapat dipandang sebagai jaringan saraf tiruan (JST) yang lapisan tersembunyi dan lapisan keluarannya merupakan hasil dari langkah-langkah dalam sistem kontrol logika fuzzy. Keluaran dari sistem kontrol logika fuzzy berupa tindakan kontrol. Sistem kontrol logika fuzzy akan dijelaskan pada Subbab II.5. Sama seperti JST, JSF juga menghasilkan suatu model. Tetapi, model yang dihasilkan berupa sistem kontrol logika fuzzy tertentu sedemikian sehingga dapat meminimalkan galat antara tindakan kontrol dan keluaran yang diinginkan. Untuk meminimalkan galat ini, dibutuhkan suatu konstruksi JSF yang tepat. Konstruksi ini akan melalui tahapan-tahapan tertentu yang akan dijelaskan pada Subbab II.6. Sebelum membahas sistem kontrol logika fuzzy dan konstruksi JSF, bab ini akan membahas konsep dasar fuzzy terlebih dahulu. Konsep dasar ini meliputi himpunan fuzzy beserta operasinya yang akan dijelaskan pada Subbab II.1 dan II.2. Selanjutnya, akan dijelaskan logika fuzzy pada Subbab II.3 yang meliputi negasi, konjungsi, disjungsi, dan implikasi. Pembahasan pada tiga subbab pertama akan menggunakan pendekatan konsep dasar himpunan klasik dan logika klasik. Subbab II.4 akan membahas aturan fuzzy dan inferensi fuzzy yang keduanya memiliki peranan yang sangat penting dalam sistem kontrol logika fuzzy.

### II.1 Himpunan Fuzzy

Perbedaan antara himpunan klasik dan himpunan fuzzy terletak pada tegas atau tidaknya sifat keanggotaan. Pada himpunan klasik, keanggotaannya bersifat tegas. Sedangkan keanggotaan dari himpunan fuzzy bersifat tidak tegas atau kabur.

Misalkan himpunan klasik  $C$  didefinisikan oleh  $C = \{3k : k \in \mathbb{Z}\}$  atau himpunan yang terdiri dari bilangan bulat yang merupakan kelipatan dari 3. Maka 6 adalah anggota himpunan klasik  $C$  dan 7 bukan anggota himpunan klasik  $C$ . Nilai kebenaran dari keanggotaan suatu himpunan klasik hanya dapat direpresentasikan oleh nilai benar atau salah. Pernyataan “6 adalah anggota himpunan klasik  $C$ ” bernilai benar. Pernyataan “7 adalah anggota himpunan klasik  $C$ ” bernilai salah.

Berdasarkan nilai kebenaran ini, dapat dibangun fungsi indikator dari himpunan klasik  $C$  yang mendefinisikan apakah suatu anggota  $\mathbb{Z}$  adalah anggota himpunan  $C$  atau bukan. Misalkan  $f_C$  adalah fungsi indikator dari himpunan klasik  $C$ . Maka  $f_C$  didefinisikan oleh

$$f_C(x) = \begin{cases} 1, & \text{jika } x \in C \\ 0, & \text{jika } x \notin C \end{cases}$$

untuk setiap  $x \in \mathbb{Z}$ .

Keanggotaan himpunan fuzzy bergantung pada suatu fungsi keanggotaannya. Misalkan himpunan fuzzy  $A$  terdiri dari bilangan real yang dekat dengan 3. Maka 3, 5, dan 10 adalah anggota himpunan fuzzy  $A$ , tetapi dengan nilai fungsi keanggotaan yang berbeda-beda. Bilangan real yang lebih dekat dengan 3 akan memiliki nilai fungsi keanggotaan yang lebih besar. Bilangan real 100 juga anggota himpunan fuzzy  $A$ , tetapi dengan nilai fungsi keanggotaan yang cukup kecil. Jika nilai pernyataan yang benar dan salah berturut-turut diwakili oleh 1 dan 0, maka nilai kebenaran dari keanggotaan suatu himpunan fuzzy dapat direpresentasikan oleh bilangan real dari 0 sampai dengan 1. Pernyataan “3 adalah anggota himpunan fuzzy  $A$ ” memiliki nilai kebenaran yang lebih besar dari pernyataan “2,5 adalah anggota himpunan fuzzy  $A$ ”.

Himpunan fuzzy telah diperkenalkan oleh Zadeh (1965). Pendefinisian himpunan fuzzy ditulis secara formal oleh Fullér (2000).

**Definisi 1** Misalkan  $X$  adalah himpunan takkosong. Sebuah himpunan fuzzy  $A$  di  $X$  dikarakterisasi oleh fungsi keanggotaannya, yaitu

$$\mu_A : X \rightarrow [0, 1]$$

dan  $\mu_A(x)$  diinterpretasikan sebagai derajat keanggotaan dari elemen  $x$  di himpunan fuzzy  $A$  untuk setiap  $x \in X$ .

Berdasarkan Definisi 1, himpunan fuzzy  $A$  di  $X$  dapat ditulis sebagai

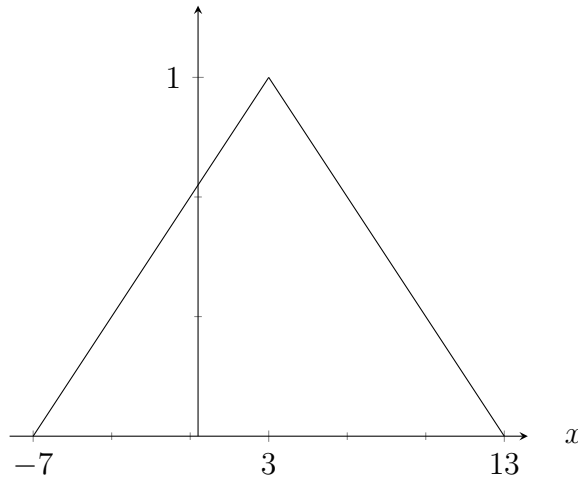
$$A = \{(x, \mu_A(x)) : x \in X\}.$$

Contoh II.1 bertujuan supaya konsep himpunan fuzzy dan fungsi keanggotaanya di bilangan real dapat lebih mudah dipahami.

**Contoh II.1** Misalkan himpunan fuzzy  $A$  di  $\mathbb{R}$  terdiri dari bilangan real  $x$  yang dekat dengan 3 dan didefinisikan oleh  $A = \{(x, \mu_A(x)) : x \in \mathbb{R}\}$  dengan

$$\mu_A(x) = \begin{cases} 1 + \frac{x-3}{10}, & -7 \leq x \leq 3 \\ 1 - \frac{x-3}{10}, & 3 < x \leq 13 \\ 0, & x \text{ lainnya.} \end{cases}$$

Grafik dari fungsi keanggotaan  $\mu_A$  ini dapat dilihat pada Gambar II.1. Berdasarkan grafik tersebut, jelas bahwa 3 memiliki derajat keanggotaan yang paling besar, yaitu  $\mu_A(3) = 1$ .



Gambar II.1: Fungsi keanggotaan untuk “ $x$  dekat dengan 3”

## II.2 Operasi pada Himpunan Fuzzy

Pada himpunan klasik, terdapat operasi irisan, gabungan, dan komplemen. Misalkan  $C$  dan  $D$  adalah himpunan klasik di  $X$ . Irisan dari  $C$  dan  $D$  ( $C \cap D$ ) menyatakan himpunan yang setiap anggotanya adalah anggota dari himpunan  $C$  sekaligus anggota dari himpunan  $D$ . Setiap anggota dari gabungan himpunan  $C$  dan  $D$  ( $C \cup D$ ) adalah anggota dari himpunan  $C$  atau anggota dari himpunan  $D$ . Komplemen dari himpunan  $D$  ( $D^C$ ) terdiri dari anggota himpunan semesta  $X$ ,

tetapi bukan anggota himpunan  $D$ . Jika dibangun fungsi keanggotaan dari  $C \cap D$ ,  $C \cup D$ , dan  $D^C$ , maka untuk setiap  $x \in X$  berlaku

$$\begin{aligned} f_{C \cap D}(x) &= \begin{cases} 1, & \text{jika } f_C(x) = 1 \text{ dan } f_D(x) = 1 \\ 0, & x \text{ lainnya} \end{cases} \\ f_{C \cup D}(x) &= \begin{cases} 1, & \text{jika } f_C(x) = 1 \text{ atau } f_D(x) = 1 \\ 0, & x \text{ lainnya} \end{cases} \\ f_{D^C}(x) &= \begin{cases} 1, & \text{jika } f_D(x) = 0 \\ 0, & x \text{ lainnya} \end{cases} \end{aligned}$$

Berdasarkan ketentuan di atas, maka fungsi keanggotaan  $f_{C \cap D}$ ,  $f_{C \cup D}$ , dan  $f_{D^C}$  juga dapat ditulis dengan cara sebagai berikut:

$$f_{C \cap D}(x) = \min\{f_C(x), f_D(x)\} = f_C(x)f_D(x) \quad (2.1)$$

$$f_{C \cup D}(x) = \max\{f_C(x), f_D(x)\} = f_C(x) + f_D(x) - f_C(x)f_D(x) \quad (2.2)$$

$$f_{D^C}(x) = 1 - f_D(x) \quad (2.3)$$

Himpunan fuzzy mempunyai operasi komplemen, irisan dan gabungan juga. Karena keanggotaan himpunan fuzzy bersifat tidak tegas dan bergantung kepada fungsi keanggotaannya, maka setiap himpunan fuzzy yang dihasilkan dari pengoperasian himpunan-himpunan fuzzy mempunyai fungsi keanggotaannya sendiri. Definisi fungsi keanggotaan dari komplemen suatu himpunan fuzzy sesuai dengan Persamaan (2.3). Fungsi keanggotaan dari irisan dan gabungan himpunan-himpunan fuzzy berturut-turut didefinisikan oleh *t-norm* dan *t-conorm*. Dalam teori himpunan fuzzy, *t-norm* dan *t-conorm* berturut-turut digunakan untuk memodelkan nilai kebenaran dari konjungsi dan disjungsi pada suatu pernyataan (Fullér, 2000).

**Definisi 2** (*t-norm*) Pemetaan  $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$  disebut *t-norm* jika dan hanya jika memenuhi sifat-sifat berikut ini:

- $T(x, y) = T(y, x) \quad \forall x, y \in [0, 1]$  (*simetri*)
- $T(x, T(y, z)) = T(T(x, y), z) \quad \forall x, y, z \in [0, 1]$  (*asosiatif*)
- $T(x, y) \leq T(x', y') \quad \forall x, y, x', y' \in [0, 1]$  dengan  $x \leq x'$  dan  $y \leq y'$  (*monoton*)
- $T(x, 1) = x \quad \forall x \in [0, 1]$  (*satu sebagai elemen identitas*)

**Definisi 3** (*t-conorm*) Pemetaan  $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$  disebut *t-conorm* jika dan hanya jika memenuhi sifat-sifat berikut ini:

- $S(x, y) = S(y, x) \quad \forall x, y \in [0, 1]$  (*simetri*)
- $S(x, S(y, z)) = S(S(x, y), z) \quad \forall x, y, z \in [0, 1]$  (*asosiatif*)
- $S(x, y) \leq S(x', y') \quad \forall x, y, x', y' \in [0, 1]$  dengan  $x \leq x'$  dan  $y \leq y'$  (*monoton*)
- $S(x, 0) = x \quad \forall x \in [0, 1]$  (*nol sebagai elemen identitas*)

Misalkan  $T$  adalah *t-norm*. Misalkan *t-conorm*  $S$  memenuhi persamaan  $S(a, b) = 1 - T(1 - a, 1 - b)$  untuk setiap  $a, b \in [0, 1]$ . Maka dapat dikatakan bahwa  $S$  diturunkan dari  $T$  (Fullér, 2000).

Berdasarkan Definisi 2 dan 3, maka muncul beberapa pasangan operator *t-norm* dan *t-conorm* dasar. Pada Tabel II.1 ditampilkan beberapa pasangan *t-norm* dan *t-conorm* dasar yang cukup sering digunakan.

Tabel II.1: Tabel *t-norm* dan *t-conorm* dasar (Fullér, 2000)

	<i>T-norm</i> ( $T(a, b)$ )	<i>T-conorm</i> ( $S(a, b)$ )
Minimum Maksimum	$\min\{a, b\}$	$\max\{a, b\}$
Łukasiewicz	$\max\{a + b - 1, 0\}$	$\min\{a + b, 1\}$
Probabilistik	$ab$	$a + b - ab$

Perhatikan bahwa *t-norm* minimum dan probabilistik sesuai dengan Persamaan (2.1), serta *t-conorm* maksimum dan probabilistik sesuai dengan Persamaan (2.2). Karena operasi irisan berkaitan dengan konjungsi dan *t-norm* analog dengan Persamaan (2.1) yang merupakan fungsi keanggotaan dari irisan himpunan klasik, maka *t-norm* dapat dijadikan sebagai derajat kebenaran suatu konjungsi dari pernyataan-pernyataan logika yang bersifat fuzzy, yaitu pernyataan yang nilai kebenarannya berada pada selang  $[0, 1]$ . Dengan menggunakan argumen yang serupa dan bersesuaian, *t-conorm* juga dapat dijadikan sebagai derajat kebenaran suatu disjungsi dari pernyataan-pernyataan logika yang bersifat fuzzy. Dengan demikian, *t-norm* dan *t-conorm* sangat tepat digunakan untuk mendefinisikan fungsi keanggotaan dari irisan dan gabungan himpunan fuzzy.

**Definisi 4** (Irisan, gabungan dan komplemen pada himpunan fuzzy) Misalkan  $A$  dan  $B$  adalah himpunan fuzzy di  $X$  dengan fungsi keanggotaan  $\mu_A$  dan  $\mu_B$

berturut-turut. Maka fungsi keanggotaan dari  $A \cap B$ ,  $A \cup B$ , dan  $A^C$  berturut-turut didefinisikan oleh

$$\begin{aligned}\mu_{A \cap B}(x) &= T(\mu_A(x), \mu_B(x)) \quad \forall x \in X \\ \mu_{A \cup B}(x) &= S(\mu_A(x), \mu_B(x)) \quad \forall x \in X \\ \mu_{A^C}(x) &= 1 - \mu_A(x) \quad \forall x \in X\end{aligned}$$

dengan  $T$  adalah  $t$ -norm dan  $S$  adalah  $t$ -conorm yang diturunkan dari  $T$ .

Contoh II.2 dapat memberikan gambaran mengenai operasi irisan, gabungan, dan komplemen pada himpunan fuzzy.

**Contoh II.2** Misalkan diberikan himpunan fuzzy  $M$  dan  $B$  di  $\mathbb{R}^+$ . Himpunan fuzzy  $M$  dan  $B$  ini menyatakan volume dari suatu gas ideal. Himpunan fuzzy  $M$  menyatakan volume “sedang”, sedangkan himpunan fuzzy  $B$  menyatakan volume “besar”. Misalkan  $\mu_M(v) = \exp \left[ - \left( \frac{v-20}{5} \right)^2 \right]$  dan  $\mu_B(v) = \frac{1}{1 + \exp(23-v)}$  untuk setiap  $v \in \mathbb{R}^+$  adalah fungsi keanggotaan dari himpunan fuzzy  $M$  dan  $B$  berturut-turut.

- Jika operasi  $t$ -norm yang digunakan adalah  $t$ -norm probabilistik, maka diperoleh  $M \cap B = \left\{ (v, \mu_{M \cap B}(v)) : v \in \mathbb{R}^+ \right\}$  dengan

$$\begin{aligned}\mu_{M \cap B}(v) &= T(\mu_M(v), \mu_B(v)) = \mu_M(v)\mu_B(v) \\ &= \frac{\exp \left[ - \left( \frac{v-20}{5} \right)^2 \right]}{1 + \exp(23-v)}.\end{aligned}$$

Akibatnya, operasi  $t$ -conorm yang mendefinisikan fungsi keanggotaan dari gabungan himpunan fuzzy  $M$  dan  $B$  haruslah menggunakan  $t$ -conorm probabilistik juga. Maka diperoleh  $M \cup B = \left\{ (v, \mu_{M \cup B}(v)) : v \in \mathbb{R}^+ \right\}$  dengan

$$\begin{aligned}\mu_{M \cup B}(v) &= S(\mu_M(v), \mu_B(v)) = \mu_M(v) + \mu_B(v) - \mu_M(v)\mu_B(v) \\ &= \frac{1 + \exp \left[ 23 - v - \left( \frac{v-20}{5} \right)^2 \right]}{1 + \exp(23-v)}.\end{aligned}$$

- Jika operasi t-norm yang digunakan adalah operasi minimum, maka diperoleh  $M \cap B = \left\{ (v, \mu_{M \cap B}(v)) : v \in \mathbb{R}^+ \right\}$  dengan

$$\begin{aligned} \mu_{M \cap B}(v) &= T(\mu_M(v), \mu_B(v)) = \min \{ \mu_M(v), \mu_B(v) \} \\ &= \begin{cases} \mu_B(v) = \frac{1}{1 + \exp(23 - v)}, & 0 < v \leq 23,4758 \\ \mu_M(v) = \exp \left[ - \left( \frac{v - 20}{5} \right)^2 \right], & v > 23,4758. \end{cases} \end{aligned}$$

Akibatnya, operasi t-conorm yang mendefinisikan fungsi keanggotaan dari gabungan himpunan fuzzy  $M$  dan  $B$  haruslah menggunakan operasi maksimum. Maka diperoleh  $M \cup B = \left\{ (v, \mu_{M \cup B}(v)) : v \in \mathbb{R}^+ \right\}$  dengan

$$\begin{aligned} \mu_{M \cup B}(v) &= S(\mu_M(v), \mu_B(v)) = \max \{ \mu_M(v), \mu_B(v) \} \\ &= \begin{cases} \mu_M(v) = \exp \left[ - \left( \frac{v - 20}{5} \right)^2 \right], & 0 < v \leq 23,4758 \\ \mu_B(v) = \frac{1}{1 + \exp(23 - v)}, & v > 23,4758. \end{cases} \end{aligned}$$

- Misalkan himpunan fuzzy  $B^C$  di  $\mathbb{R}^+$  menyatakan volume yang “tidak besar” dari suatu gas ideal. Maka fungsi keanggotaan  $\mu_{B^C}$  adalah

$$\begin{aligned} \mu_{B^C}(v) &= 1 - \mu_B(v) = 1 - \frac{1}{1 + \exp(23 - v)} \\ &= \frac{1}{1 + \exp(v - 23)}. \end{aligned}$$

### II.3 Logika Fuzzy

Nilai kebenaran suatu pernyataan pada logika klasik dapat dinyatakan dengan 1 untuk nilai “benar” dan 0 untuk nilai “salah”. Misalkan  $p$  dan  $q$  adalah pernyataan logika klasik dengan

$$p : “x \text{ adalah anggota himpunan } C” \Leftrightarrow “x \in C” \quad (2.4)$$

$$q : “y \text{ adalah anggota himpunan } D” \Leftrightarrow “y \in D” \quad (2.5)$$

Himpunan  $C$  dan himpunan  $D$  di sini adalah himpunan klasik. Misalkan  $\tau(\cdot)$  menyatakan derajat kebenaran suatu pernyataan logika klasik. Maka  $\tau(p) = 1$  dan



$\tau(q) = 1$  ketika  $x \in C$  dan  $y \in D$ . Jika  $x \notin C$  atau pernyataan  $p$  bernilai salah maka  $\tau(p) = 0$ , dan jika  $y \notin D$  maka  $\tau(q) = 0$ .

Berdasarkan pernyataan logika klasik  $p$  yang didefinisikan pada (2.4), maka negasi dari pernyataan  $p$  adalah

$$\neg p : "x \text{ bukan anggota himpunan } C" \Leftrightarrow "x \notin C"$$

Ketika pernyataan  $\neg p$  bernilai benar atau " $x \notin C$ ", maka  $\tau(\neg p) = 1$ . Jika  $x \notin C$ , maka pernyataan  $p$  bernilai salah, sehingga  $\tau(p) = 0$ . Ketika pernyataan  $\neg p$  bernilai salah atau " $x \in C$ ", maka  $\tau(\neg p) = 0$ . Jika  $x \in C$ , maka pernyataan  $p$  bernilai benar, sehingga  $\tau(p) = 1$ . Dengan demikian, derajat kebenaran dari pernyataan  $\neg p$  dapat dinyatakan dengan

$$\tau(\neg p) = 1 - \tau(p) \quad (2.6)$$

Nilai kebenaran suatu pernyataan pada logika fuzzy dapat dinyatakan dengan bilangan real pada selang  $[0, 1]$ . Misalkan  $A$  adalah himpunan fuzzy di  $X$  dan  $B$  adalah himpunan fuzzy di  $Y$ . Misalkan fungsi keanggotaan dari himpunan fuzzy  $A$  dan  $B$  berturut-turut adalah  $\mu_A$  dan  $\mu_B$ . Misalkan pernyataan logika fuzzy  $r$  dan  $s$  didefinisikan sebagai

$$\begin{aligned} r : "x \text{ adalah anggota dari himpunan fuzzy } A" \\ \Leftrightarrow x \text{ anggota } A \end{aligned} \quad (2.7)$$

$$\begin{aligned} s : "y \text{ adalah anggota dari himpunan fuzzy } B" \\ \Leftrightarrow y \text{ anggota } B \end{aligned} \quad (2.8)$$

Maka dapat diinterpretasikan bahwa  $r$  dan  $s$  adalah pernyataan logika fuzzy dengan derajat kebenaran  $\mu_A(x)$  dan  $\mu_B(y)$  berturut-turut. Dengan kata lain,  $\tau(r) = \mu_A(x)$  dan  $\tau(s) = \mu_B(y)$ . Pada pernyataan logika fuzzy, terdapat istilah-istilah penting, di antaranya:

- Variabel linguistik, yaitu variabel yang nilainya berupa himpunan fuzzy atau variabel yang nilainya didefinisikan dalam istilah linguistik. Pada dua pernyataan logika fuzzy di atas,  $x$  dan  $y$  merupakan variabel linguistik.
- Nilai linguistik, yaitu himpunan fuzzy atau istilah linguistik yang mendeskripsikan variabel linguistik. Nilai linguistik dideskripsikan oleh fungsi keanggotaan

himpunan fuzzy. Pada pernyataan  $r$  dan  $s$ ,  $A$  dan  $B$  adalah nilai linguistik dari variabel  $x$  dan  $y$  berturut-turut.

- Himpunan semesta, yaitu domain dari nilai linguistik. Pada pernyataan  $r$ , himpunan  $X$  adalah himpunan semesta.

Misalkan  $\neg r$  adalah negasi dari pernyataan logika fuzzy  $r$ . Maka  $\neg r$  dapat diinterpretasikan sebagai “ $x$  adalah anggota dari himpunan fuzzy  $A^C$ ”. Berdasarkan derajat kebenaran dari negasi suatu pernyataan logika klasik yang dinyatakan pada (2.6) dan definisi dari komplemen suatu himpunan fuzzy pada Definisi 4, maka derajat kebenaran dari pernyataan  $\neg r$  dapat dinyatakan dengan

$$\tau(\neg r) = 1 - \tau(r) = 1 - \mu_A(x) = \mu_{A^C}(x)$$

Derajat kebenaran dari konjungsi, disjungsi, dan implikasi fuzzy dapat ditentukan dengan menggunakan konsep perluasan dari konjungsi, disjungsi, dan implikasi pada logika klasik. Konjungsi dan disjungsi fuzzy akan dibahas pada Subbab II.3.1. Implikasi fuzzy akan dibahas pada Subbab II.3.2.

### II.3.1 Konjungsi dan Disjungsi Fuzzy

Konjungsi yang melibatkan minimal dua pernyataan logika klasik disebut dengan konjungsi pokok. Misalkan  $p \wedge q$  adalah konjungsi pokok dengan pernyataan  $p$  dan  $q$  berturut-turut didefinisikan pada (2.4) dan (2.5). Konjungsi pokok  $p \wedge q$  bernilai benar jika dan hanya jika pernyataan  $p$  dan  $q$  keduanya bernilai benar. Dengan demikian, derajat kebenaran  $p \wedge q$  ditentukan oleh

$$\tau(p \wedge q) = \begin{cases} 1, & \text{jika } \tau(p) = \tau(q) = 1 \\ 0, & \text{lainnya} \end{cases}$$

Berdasarkan ketentuan di atas, derajat kebenaran  $p \wedge q$  juga dapat dinyatakan oleh

$$\tau(p \wedge q) = \min\{\tau(p), \tau(q)\} = \tau(p)\tau(q) = \max\{\tau(p) + \tau(q) - 1, 0\} \quad (2.9)$$

Disjungsi yang melibatkan minimal dua pernyataan logika klasik disebut dengan disjungsi pokok. Misalkan  $p \vee q$  adalah disjungsi pokok dengan pernyataan  $p$  dan  $q$  berturut-turut didefinisikan pada (2.4) dan (2.5). Disjungsi pokok  $p \vee q$  bernilai benar jika dan hanya jika salah satu dari pernyataan  $p$  dan  $q$  bernilai benar. Dengan

demikian, derajat kebenaran  $p \vee q$  ditentukan oleh

$$\tau(p \vee q) = \begin{cases} 1, & \text{jika } \tau(p) = 1 \text{ atau } \tau(q) = 1 \\ 0, & \text{lainnya} \end{cases}$$

Berdasarkan ketentuan di atas, derjat kebenaran  $p \vee q$  juga dapat dinyatakan oleh

$$\begin{aligned} \tau(p \vee q) &= \text{maks}\{\tau(p), \tau(q)\} = \tau(p) + \tau(q) - \tau(p)\tau(q) \\ &= \min\{\tau(p) + \tau(q), 1\} \end{aligned} \quad (2.10)$$

Konjungsi yang melibatkan minimal dua pernyataan logika fuzzy disebut dengan konjungsi fuzzy. Disjungsi yang melibatkan minimal dua pernyataan logika fuzzy juga disebut dengan disjungsi fuzzy. Misalkan  $r$  dan  $s$  adalah dua pernyataan logika fuzzy yang didefinisikan pada (2.7) dan (2.8) berturut-turut. Misalkan  $r \wedge s$  dan  $r \vee s$  berturut-turut adalah konjungsi dan disjungsi fuzzy. Maka derajat kebenaran dari  $r \wedge s$  dan  $r \vee s$  dapat ditulis

$$\begin{aligned} \tau(r \wedge s) &= \mu_{A \wedge B}(x, y) \\ \tau(r \vee s) &= \mu_{A \vee B}(x, y) \end{aligned}$$

Bentuk  $A \wedge B$  dan  $A \vee B$  di sini dapat dipandang sebagai himpunan fuzzy di  $X \times Y$ . Himpunan fuzzy  $A \wedge B$  dan  $A \vee B$  berturut-turut memiliki fungsi keanggotaan  $\mu_{A \wedge B}$  dan  $\mu_{A \vee B}$  yang keduanya memetakan  $X \times Y$  terhadap selang  $[0, 1]$ . Berdasarkan Persamaan (2.9) dan (2.10), maka perluasan dari konjungsi dan disjungsi pokok ke konjungsi dan disjungsi fuzzy adalah

$$\begin{aligned} \mu_{A \wedge B}(x, y) &= T(\mu_A(x), \mu_B(y)) \\ \mu_{A \vee B}(x, y) &= S(\mu_A(x), \mu_B(y)) \end{aligned}$$

dengan  $T$  adalah  $t$ -norm dan  $S$  adalah  $t$ -conorm yang diturunkan dari  $T$ .

### II.3.2 Implikasi Fuzzy

Implikasi yang melibatkan minimal dua pernyataan logika klasik disebut dengan implikasi pokok. Misalkan  $p \rightarrow q$  adalah implikasi pokok dengan  $p$  dan  $q =$  berturut-turut didefinisikan pada (2.4) dan (2.5). Pernyataan  $p$  disebut sebagai bagian pendahulu dari implikasi pokok  $p \rightarrow q$  dan pernyataan  $q$  disebut sebagai

bagian konsekuensi dari  $p \rightarrow q$ . Interpretasi dari implikasi pokok  $p \rightarrow q$  adalah bahwa derajat kebenaran dari  $p \rightarrow q$  mengkuantifikasi sejauh mana  $q$  setidaknya memiliki derajat kebenaran yang sama dengan  $p$  (Fullér, 2000), yaitu

$$\tau(p \rightarrow q) = \begin{cases} 1, & \text{jika } \tau(p) \leq \tau(q) \\ 0, & \text{lainnya} \end{cases} \quad (2.11)$$

Berdasarkan ketentuan derajat kebenaran ini, diperoleh Tabel II.2.

Tabel II.2: Tabel kebenaran untuk implikasi pokok

$\tau(p)$	$\tau(q)$	$\tau(p) \rightarrow \tau(q)$
1	1	1
0	1	1
0	0	1
1	0	0

Implikasi yang melibatkan minimal dua pernyataan logika fuzzy disebut dengan implikasi fuzzy. Ketentuan derajat kebenaran implikasi pokok pada Persamaan (2.11) dapat diperluas menjadi ketentuan untuk derajat kebenaran dari implikasi fuzzy. Misalkan pernyataan  $r$  dan  $s$  yang didefinisikan pada (2.7) dan (2.8) adalah dua pernyataan logika fuzzy. Misalkan pula  $r \rightarrow s$  adalah implikasi fuzzy dan dapat dinyatakan dengan

$$r \rightarrow s : \text{Jika } x \text{ adalah } A, \text{ maka } y \text{ adalah } B$$

Pada implikasi fuzzy ini, terdapat istilah-istilah yang berasosiasi dengan istilah dalam pernyataan logika fuzzy dan implikasi pokok:

- $x$  disebut sebagai variabel linguistik bagian pendahulu dengan nilai linguistik  $A$ ,
- $y$  disebut sebagai variabel linguistik bagian konsekuensi dengan nilai linguistik  $B$ .

Derajat kebenaran dari  $r \rightarrow s$  dapat ditulis  $\tau(r \rightarrow s) = \mu_{A \rightarrow B}(x, y)$ . Bentuk  $A \rightarrow B$  di sini dapat dipandang sebagai himpunan fuzzy di  $X \times Y$ . Himpunan fuzzy  $A \rightarrow B$  memiliki fungsi keanggotaan  $\mu_{A \rightarrow B}$  yang merupakan pemetaan dari  $X \times Y$  terhadap selang  $[0, 1]$ . Berdasarkan Persamaan (2.11), salah satu perluasan

dari implikasi pokok ke implikasi fuzzy yang memungkinkan adalah

$$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, & \text{jika } \mu_A(x) \leq \mu_B(y) \\ 0, & \text{lainnya} \end{cases}$$

Operator implikasi fuzzy ini disebut dengan operator *Standard Strict*.

Namun, operator implikasi fuzzy ini tidak tepat untuk diaplikasikan di dunia nyata. Sebagai contoh, misalkan  $\mu_A(x_1) = 0,7$  dan  $\mu_B(y_1) = 0,7$  untuk suatu  $x_1 \in X$  dan  $y_1 \in Y$ . Maka berdasarkan definisi dari operator *Standard Strict*, mengakibatkan

$$\mu_{A \rightarrow B}(x_1, y_1) = 1$$

Misalkan  $\mu_B(y_2) = 0,699$  untuk suatu  $y_2 \in Y$ . Dapat diinterpretasikan bahwa derajat keanggotaan  $y_2$  di  $B$  adalah derajat keanggotaan  $y_1$  di  $B$  yang tidak dibulatkan ke dalam satu angka di belakang tanda koma. Selanjutnya, diperoleh

$$\mu_{A \rightarrow B}(x_1, y_2) = 0$$

Contoh ini menunjukkan bahwa perubahan kecil pada derajat keanggotaan dari bagian konsekuensi dapat menyebabkan penyimpangan besar pada nilai kebenaran dari implikasi fuzzy. Dengan kata lain, operator ini sangat sensitif terhadap pembulatan.

Implikasi pokok  $p \rightarrow q$  ekuivalen dengan  $\neg p \vee q$ . Oleh karena itu, derajat kebenaran dari  $p \rightarrow q$  dapat dinyatakan oleh

$$\tau(p \rightarrow q) = \tau(\neg p \vee q) = \max\{\tau(\neg p), \tau(q)\} = \max\{1 - \tau(p), \tau(q)\}$$

Dengan menggunakan konsep perluasan, fungsi keanggotaan  $\mu_{A \rightarrow B}$  juga dapat dinyatakan oleh

$$\mu_{A \rightarrow B}(x, y) = \max\{1 - \mu_A(x), \mu_B(y)\}, \forall x \in X, y \in Y$$

Operator ini disebut operator implikasi *Kleene-Dienes*.

Perhatikan bahwa derajat kebenaran implikasi pokok  $p \rightarrow q$  dapat dinyatakan juga oleh

$$\tau(p \rightarrow q) = \sup \{z \in \{0, 1\} : \min\{\tau(p), z\} \leq \tau(q)\}$$

Dengan menggunakan konsep perluasan, fungsi keanggotaan  $\mu_{A \rightarrow B}$  juga dapat dinyatakan oleh

$$\begin{aligned}\mu_{A \rightarrow B}(x, y) &= \sup \{z \in [0, 1] : \min\{\mu_A(x), z\} \leq \mu_B(y)\} \\ &= \begin{cases} 1, & \text{jika } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \text{lainnya} \end{cases}\end{aligned}$$

untuk setiap  $x \in X$  dan  $y \in Y$ . Operator ini disebut operator implikasi *Gödel*.

Dalam aplikasi praktis, untuk memodelkan hubungan sebab-akibat antara variabel-variabel yang bersifat fuzzy sering digunakan operator implikasi *Mamdani*. Operator ini secara sederhana mengambil nilai minimum dari derajat kebenaran bagian pendahulu dan derajat kebenaran bagian konsekuensi pada suatu implikasi fuzzy. Dengan kata lain, operator implikasi *Mamdani* dapat dinyatakan oleh

$$\mu_{A \rightarrow B}(x, y) = \min\{\mu_A(x), \mu_B(y)\}, \forall x \in X, y \in Y$$

Misalkan  $\mu_A(x_3) = 0$  dan  $\mu_B(y_3) = 0$  untuk suatu  $x_3 \in X$  dan  $y_3 \in Y$ . Dengan menggunakan operator implikasi *Mamdani*, diperoleh  $\mu_{A \rightarrow B}(x_3, y_3) = 0$ . Dengan demikian, operator ini bukanlah perluasan yang benar dari implikasi pokok. Namun, dalam sistem yang berbasis pengetahuan, kita biasanya tidak tertarik dengan aturan fuzzy (kumpulan dari implikasi fuzzy) yang bagian-bagian pendahulunya bernilai salah.

Tabel II.3: Operator implikasi fuzzy

Nama	Definisi
Zadeh	$\mu_{A \rightarrow B}(x, y) = \max\{1 - \mu_A(x), \min\{\mu_A(x), \mu_B(y)\}\}$
Łukasiewicz	$\mu_{A \rightarrow B}(x, y) = \min\{1, 1 - \mu_A(x) + \mu_B(y)\}$
Kleene-Dienes	$\mu_{A \rightarrow B}(x, y) = \max\{1 - \mu_A(x), \mu_B(y)\}$
Mamdani	$\mu_{A \rightarrow B}(x, y) = \min\{\mu_A(x), \mu_B(y)\}$
Larsen	$\mu_{A \rightarrow B}(x, y) = \mu_A(x)\mu_B(y)$
<i>Standard Strict</i>	$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, & \text{jika } \mu_A(x) \leq \mu_B(y) \\ 0, & \text{lainnya} \end{cases}$
Gödel	$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, & \text{jika } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y), & \text{lainnya} \end{cases}$
Gaines	$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1, & \text{jika } \mu_A(x) \leq \mu_B(y) \\ \frac{\mu_B(y)}{\mu_A(x)}, & \text{lainnya} \end{cases}$

Terdapat tiga jenis operator implikasi fuzzy, yaitu:

- Implikasi- $S$ , didefinisikan oleh

$$\mu_{A \rightarrow B}(x, y) = S(\mu_{A^c}(x), \mu_B(y)) = S(1 - \mu_A(x), \mu_B(y))$$

dengan  $S$  adalah  $t$ -conorm. Contoh dari implikasi- $S$  ini adalah implikasi *Lukasiewicz* dan *Kleene-Dienes*.

- Implikasi- $R$ , didefinisikan oleh

$$\mu_{A \rightarrow B}(x, y) = \sup \{z \in [0, 1] : T(\mu_A(x), z) \leq \mu_B(y)\}$$

dengan  $T$  adalah  $t$ -norm. Contoh dari implikasi- $R$  adalah implikasi *Gödel* dan *Gaines*.

- Implikasi  $t$ -norm. Misalkan  $T$  adalah  $t$ -norm. Maka implikasi ini didefinisikan oleh

$$\mu_{A \rightarrow B}(x, y) = T(\mu_A(x), \mu_B(y))$$

Walaupun implikasi ini tidak memenuhi sifat-sifat dari implikasi pokok, implikasi ini sering digunakan sebagai model dalam berbagai aplikasi logika fuzzy. Contoh dari implikasi  $t$ -norm adalah implikasi *Mamdani* dan *Larsen*.

Pendefinisian dari semua contoh operator yang disebutkan di atas dapat dilihat pada Tabel II.3.

## II.4 Aturan Fuzzy dan Inferensi Fuzzy

Aturan fuzzy terdiri dari kumpulan implikasi-implikasi fuzzy yang saling berkaitan. Bagian pendahulu dan bagian konsekuensi pada setiap implikasi dalam aturan fuzzy merupakan pernyataan logika fuzzy. Aturan fuzzy memiliki dua struktur umum yang berbeda, yaitu:

- aturan fuzzy Zadeh yang digagas oleh Zadeh (1973), dan
- aturan fuzzy Takagi-Sugeno-Kang (TSK) yang digagas oleh Takagi dan Sugeno (1985) dan Sugeno dan Kang (1988).

Perbedaan dari dua struktur ini terletak pada bagian konsekuensi. Pada aturan fuzzy Zadeh, setiap pernyataan yang berada di dalam bagian konsekuensi merupakan pernyataan logika fuzzy. Pada aturan fuzzy TSK, bagian konsekuensi dari setiap implikasinya berupa fungsi dari variabel-variabel linguistik pada bagian pendahulu.

Misalkan  $A_{1,j}, A_{2,j}, \dots, A_{m,j}$  adalah himpunan fuzzy di  $X_j$  dengan  $j = 1, 2, \dots, n$  dan  $x_1, x_2, \dots, x_n$  adalah variabel linguistik bagian pendahulu. Misalkan juga diberikan aturan fuzzy Zadeh dengan implikasi fuzzy sebanyak  $m$ . Variabel linguistik pada bagian konsekuensi hanya ada satu, yaitu  $y$ , dengan nilai linguistik  $B_1, B_2, \dots, B_m$  pada himpunan semesta  $Y$ . Maka aturan fuzzy ini akan berbentuk

$$\begin{aligned} \mathfrak{R}_1 : & \text{ Jika } x_1 \text{ anggota } A_{1,1}, \dots, x_n \text{ anggota } A_{1,n}, \text{ maka } y \text{ anggota } B_1 \\ \mathfrak{R}_2 : & \text{ Jika } x_1 \text{ anggota } A_{2,1}, \dots, x_n \text{ anggota } A_{2,n}, \text{ maka } y \text{ anggota } B_2 \\ & \vdots \\ \mathfrak{R}_m : & \text{ Jika } x_1 \text{ anggota } A_{m,1}, \dots, x_n \text{ anggota } A_{m,n}, \text{ maka } y \text{ anggota } B_m \end{aligned}$$

dengan  $\mathfrak{R}_i$  adalah aturan fuzzy ke- $i$ .

Misalkan diberikan aturan fuzzy TSK yang terdiri dari sebanyak  $m$  implikasi fuzzy. Misalkan pula fungsi pada bagian konsekuensi berupa fungsi yang linier terhadap variabel linguistik bagian pendahulu. Maka aturan fuzzy ini akan berbentuk

$$\begin{aligned} \mathfrak{R}_1 : & \text{ Jika } x_1 \text{ anggota } A_{1,1}, \dots, x_n \text{ anggota } A_{1,n}, \text{ maka } y_1 = \mathbf{b}_1 \cdot \tilde{\mathbf{x}} \\ \mathfrak{R}_2 : & \text{ Jika } x_1 \text{ anggota } A_{2,1}, \dots, x_n \text{ anggota } A_{2,n}, \text{ maka } y_2 = \mathbf{b}_2 \cdot \tilde{\mathbf{x}} \\ & \vdots \\ \mathfrak{R}_m : & \text{ Jika } x_1 \text{ anggota } A_{m,1}, \dots, x_n \text{ anggota } A_{m,n}, \text{ maka } y_m = \mathbf{b}_m \cdot \tilde{\mathbf{x}} \end{aligned}$$

dengan  $\mathbf{b}_i = (b_{i,0}, b_{i,1}, b_{i,2}, \dots, b_{i,n})$  untuk setiap  $i = 1, 2, \dots, m$  dan  $\tilde{\mathbf{x}} = (1, x_1, x_2, \dots, x_n)$ . Pada aturan fuzzy ini, bagian konsekuensi merupakan hasil dari operasi  $\mathbf{b}_i \cdot \tilde{\mathbf{x}}$ . Maka variabel  $y$  tidak terlihat seperti variabel linguistik karena tidak terlihat jelas sebagai anggota dari himpunan fuzzy. Tetapi,  $\mathbf{b}_i \cdot \tilde{\mathbf{x}}$  dapat dimodelkan sebagai himpunan fuzzy  $B_i$ ,  $i = 1, 2, \dots, m$ . Dengan demikian, menurut Mendel (2017) aturan fuzzy ini dapat dibuat menyerupai aturan fuzzy Zadeh dengan fungsi keanggotaan himpunan fuzzy  $B_i$  adalah

$$\mu_{B_i}(y) = \begin{cases} 1, & \text{jika } y = \mathbf{b}_i \cdot \tilde{\mathbf{x}} \\ 0, & \text{lainnya} \end{cases} \quad i = 1, 2, \dots, m \quad (2.12)$$

Himpunan fuzzy  $B_i$  ini disebut himpunan fuzzy diskrit. Hal ini dikarenakan

$$|\{y \in Y : \mu_{B_i}(y) > 0\}| = 1 \in \mathbb{N}.$$



Dengan kata lain, banyaknya anggota  $Y$  yang derajat keanggotaannya positif di himpunan fuzzy  $B_i$  adalah berhingga.

Inferensi klasik dikenal sebagai proses penarikan kesimpulan dari serangkaian fakta berdasarkan kumpulan aturan tertentu (Nedjah dan de Macedo Mourelle, 2005). Salah satu jenis inferensi klasik adalah *modus ponens*. Misalkan  $p$  dan  $q$  adalah pernyataan logika klasik yang didefinisikan pada (2.4) dan (2.5) berturut-turut. Misalkan diberikan

Aturan :  $p \rightarrow q$

Fakta :  $p$

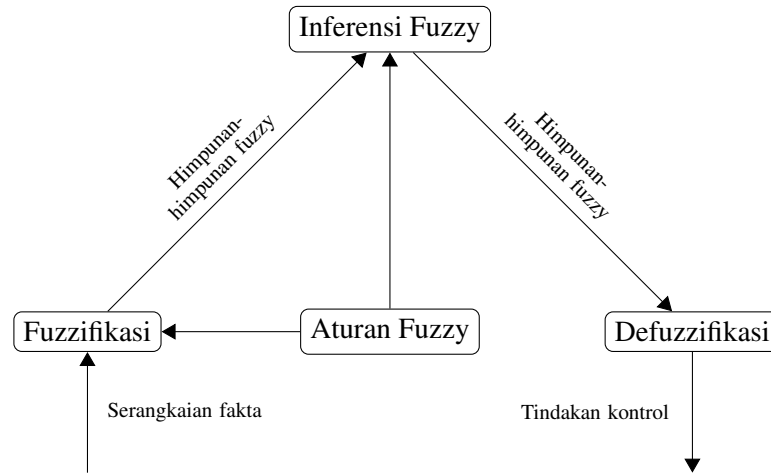
Berdasarkan *modus ponens*, diperoleh kesimpulan  $q$ .

Inferensi fuzzy adalah proses penarikan kesimpulan dari serangkaian fakta berdasarkan aturan fuzzy tertentu. Namun, serangkaian faktanya merupakan pernyataan-pernyataan logika fuzzy. Proses inferensi fuzzy juga menerapkan prinsip *modus ponens* yang sama (Nedjah dan de Macedo Mourelle, 2005). Mekanisme inferensi fuzzy akan dijelaskan pada Subbab II.5.

## II.5 Sistem Kontrol Logika Fuzzy

Sistem kontrol logika fuzzy (SKLF) adalah sistem kontrol yang berfungsi untuk mentransformasi serangkaian fakta menjadi suatu kesimpulan berupa tindakan kontrol berdasarkan aturan fuzzy tertentu melalui proses fuzzifikasi, inferensi fuzzy, dan defuzzifikasi. Bagian pendahulu pada setiap implikasi dalam aturan fuzzy dipandang sebagai suatu kondisi pada domain aplikasi sistem kontrol. Bagian konsekuensi pada setiap implikasi dalam aturan fuzzy dipandang sebagai tindakan kontrol untuk sistem yang dikendalikan. Oleh karena itu, pada SKLF, variabel linguistik bagian pendahulu disebut dengan variabel kondisi dan variabel linguistik bagian konsekuensi disebut variabel kontrol. Pada dasarnya, SKLF menyediakan cara yang mudah untuk mengekspresikan kebijakan kontrol dan domain pengetahuan (Fullér, 2000).

Bentuk umum dari SKLF diberikan pada Gambar II.2. SKLF terdiri dari empat bagian utama: fuzzifikasi, aturan fuzzy, inferensi fuzzy, dan defuzzifikasi. Pada tahap fuzzifikasi, serangkaian fakta yang bersifat nonfuzzy ditransformasi menjadi



Gambar II.2: Sistem kontrol logika fuzzy

bilangan real pada selang  $[0, 1]$  menggunakan fungsi keanggotaan nilai linguistik dari variabel kondisi. Selanjutnya, dilakukan inferensi fuzzy terhadap hasil fuzzifikasi berdasarkan aturan fuzzy. Inferensi fuzzy ini akan menghasilkan himpunan fuzzy yang berpadanan dengan nilai linguistik dari variabel kontrol. Misalkan terdapat sebanyak  $k$  variabel kontrol. Maka ada sebanyak  $k$  nilai linguistik berupa himpunan fuzzy yang terlibat pada bagian konsekuensi dari setiap implikasinya. Akibatnya, himpunan fuzzy yang dihasilkan dari tahap ini ada sebanyak  $k$ . Inferensi fuzzy yang dilakukan pada tahap ini akan dijelaskan pada Bagian II.5.2. Pada akhirnya, dilakukan defuzzifikasi terhadap himpunan fuzzy yang dihasilkan dari tahap inferensi. Defuzzifikasi ini akan menghasilkan sebanyak  $k$  tindakan kontrol. Metode-metode defuzzifikasi akan dijelaskan pada Bagian II.5.1.

### II.5.1 Metode Defuzzifikasi

Defuzzifikasi adalah proses untuk memilih anggota yang mewakili himpunan fuzzy yang dihasilkan dari proses inferensi fuzzy (Fullér, 2000). Defuzzifikasi akan menghasilkan keluaran yang bersifat nonfuzzy. Pada SKLF, hasil defuzzifikasi dipandang sebagai tindakan kontrol. Misalkan himpunan fuzzy  $B$  di  $Y$  dengan fungsi keanggotaan  $\mu_B$  adalah hasil dari suatu proses inferensi fuzzy. Misalkan defuzzifier adalah operator defuzzifikasi. Maka diperoleh

$$y_0 = \text{defuzzifier}(\mu_B)$$

dengan  $y_0$  adalah tindakan kontrol dari sistem kontrol logika fuzzy dan  $y_0 \in Y$ .

Operator defuzzifikasi yang sering digunakan adalah sebagai berikut:

- **Center-of-Area/Gravity.** Hasil defuzzifikasi dari himpunan fuzzy  $B$  di  $Y$  didefinisikan sebagai *centroid* dari fungsi keanggotaan  $\mu_B$ , yaitu:

$$y_0 = \frac{\int_Y y \mu_B(y) dy}{\int_Y \mu_B(y) dy}$$

Jika  $B$  adalah himpunan fuzzy diskrit, maka

$$y_0 = \frac{\sum_{y \in Y} y \mu_B(y)}{\sum_{y \in Y} \mu_B(y)}$$

- **First-of-Maxima.** Hasil defuzzifikasi dari himpunan fuzzy  $B$  di  $Y$  adalah elemen dengan nilai terkecil yang memaksimalkan  $\mu_B$ , yaitu:

$$y_0 = \min\{y \in Y : y = \arg \max_{w \in Y} \mu_B(w)\}$$

- **Middle-of-Maxima.** Hasil defuzzifikasi dari himpunan fuzzy  $B$  di  $Y$  adalah rata-rata dari semua nilai nonfuzzy yang memaksimalkan  $\mu_B$ , yaitu:

$$y_0 = \frac{\int_G y dy}{\int_G dy}$$

dengan  $G$  menyatakan himpunan bagian dari  $Y$  yang elemen-elemennya memaksimalkan  $\mu_B$ , yaitu  $G = \{y \in Y : \mu_B(y) = \max_w \mu_B(w)\}$ . Jika  $G$  diskrit, maka

$$y_0 = \frac{1}{|G|} \sum_{y \in G} y$$

- **Max-Criterion.** Hasil defuzzifikasi dari himpunan fuzzy  $B$  di  $Y$  adalah sembarang elemen dari himpunan yang terdiri dari nilai nonfuzzy yang memaksimalkan  $\mu_B$ , yaitu:

$$y_0 \in \{y \in Y : y = \arg \max_{w \in Y} \mu_B(w)\}$$

## II.5.2 Mekanisme Inferensi

Berdasarkan struktur aturan fuzzy yang telah dijelaskan pada Subbab II.4, maka ada dua mekanisme inferensi yang lazim digunakan dalam SKLF. Dua mekanisme inferensi ini adalah: mekanisme inferensi Mamdani dan mekanisme inferensi Takagi-Sugeno-Kang (TSK).

### Mekanisme Inferensi Mamdani

Inferensi Mamdani merupakan inferensi fuzzy yang dilakukan terhadap aturan fuzzy Zadeh. Operator implikasi fuzzy yang digunakan pada inferensi ini adalah operator mamdani, yaitu operator minimum. Penghubung antar implikasi pada aturan fuzzy diinterpretasikan sebagai “atau”, sehingga operator yang digunakan untuk menghubungkan semua implikasinya adalah operator maksimum.

Misalkan diberikan SKLF berikut ini

$\mathfrak{R}_1$ :	Jika	$x_1$ anggota $A_{1,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{1,n}$ ,	maka	$y$ anggota $B_1$
$\mathfrak{R}_2$ :	Jika	$x_1$ anggota $A_{2,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{2,n}$ ,	maka	$y$ anggota $B_2$
				$\vdots$		
$\mathfrak{R}_m$ :	Jika	$x_1$ anggota $A_{m,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{m,n}$ ,	maka	$y$ anggota $B_m$
Fakta :		$x_1 = a_1$ ,	$\dots$ ,	$x_n = a_n$		
Konklusi :						$y = y_0$

dengan  $A_{1,j}, A_{2,j}, \dots, A_{m,j}$  adalah himpunan fuzzy di  $X_j$  untuk setiap  $j = 1, 2, \dots, n$ , dan  $B_1, B_2, \dots, B_m$  adalah himpunan fuzzy di  $Y$ . Variabel linguistik bagian pendahulu di sini adalah  $x_1, x_2, \dots, x_n$ . Pada sistem kontrol ini hanya terdapat satu variabel linguistik bagian konsekuensi, yaitu  $y$ . Tugas utama dari SKLF ini adalah untuk memperoleh tindakan kontrol tertentu, yaitu nilai dari  $y_0$ . Langkah-langkah untuk menyelesaikan sistem kontrol ini dengan menggunakan inferensi mamdani adalah sebagai berikut

1. Lakukan fuzzifikasi terhadap serangkaian fakta berdasarkan nilai linguistik dari variabel kondisi. Kemudian operasikan hasil fuzzifikasi tersebut dengan operasi konjungsi fuzzy ( $t$ -norm). Misalkan hasil operasi ini dinyatakan dengan  $\alpha_i$  untuk  $i = 1, 2, \dots, m$ . Maka diperoleh

$$\alpha_i = T(\mu_{A_{i,1}}(a_1), \mu_{A_{i,2}}(a_2), \dots, \mu_{A_{i,n}}(a_n)), \quad i = 1, 2, \dots, m$$

- Operasikan bagian pendahulu dan bagian konsekuensi pada setiap implikasi menggunakan operator mamdani. Maka diperoleh sebanyak  $m$  himpunan fuzzy baru di  $Y$ , yaitu himpunan fuzzy  $B'_1, B'_2, \dots, B'_m$  dengan fungsi keanggotaan

$$\mu_{B'_i}(y) = \min \{ \alpha_i, \mu_{B_i}(y) \}, \quad y \in Y \quad i = 1, 2, \dots, m$$

- Selanjutnya, dilakukan operasi  $t$ -conorm untuk menggabungkan sebanyak  $m$  himpunan fuzzy dari setiap implikasi. Maka diperoleh himpunan fuzzy baru di  $Y$ , yaitu himpunan fuzzy  $B$  dengan fungsi keanggotaan

$$\mu_B(y) = S \left( \mu_{B'_1}(y), \mu_{B'_2}(y), \dots, \mu_{B'_m}(y) \right), \quad y \in Y$$

- Langkah terakhir, lakukan defuzzifikasi terhadap himpunan fuzzy  $B$  untuk mendapatkan nilai dari tindakan kontrol  $y_0$ , sehingga diperoleh

$$y_0 = defuzzifier(\mu_B)$$

Contoh II.3 berikut ini memberikan gambaran tambahan mengenai mekanisme inferensi Mamdani.

**Contoh II.3** Misalkan diberikan sistem gas ideal yang terdiri dari *volume* dengan satuan *liter*, *suhu* dengan satuan *Kelvin*, dan *tekanan* dengan satuan *atm*. Sistem gas ideal ini dapat dipandang sebagai SKLF dengan *volume* dan *suhu* sebagai variabel kondisi dan *tekanan* sebagai variabel kontrolnya. Misalkan

- Variabel kondisi *volume* memiliki tiga nilai linguistik: besar, standar, dan kecil. Fungsi keanggotaan dari tiga nilai linguistik ini didefinisikan sebagai berikut

$$\begin{aligned} \mu_{besar}(V) &= \frac{1}{1 + \exp \left( \frac{32 - V}{5} \right)} \\ \mu_{standar}(V) &= \exp \left[ -\frac{1}{2} \left( \frac{V - 18,5}{2} \right)^2 \right] \\ \mu_{kecil}(V) &= \frac{1}{1 + \exp \left( \frac{V - 13}{2} \right)} \end{aligned}$$

dengan  $V$  menyatakan variabel kondisi *volume*.

- Variabel kondisi *suhu* memiliki tiga nilai linguistik: panas, normal, dan dingin

dengan fungsi keanggotaan

$$\begin{aligned}\mu_{panas}(T) &= \frac{1}{1 + \exp\left(\frac{323 - T}{4}\right)} \\ \mu_{normal}(T) &= \exp\left[-\frac{1}{2}\left(\frac{T - 299,5}{4}\right)^2\right] \\ \mu_{dingin}(T) &= \frac{1}{1 + \exp\left(\frac{T - 276}{4}\right)}\end{aligned}$$

dengan  $T$  menyatakan variabel kondisi *suhu*.

- Variabel kontrol *tekanan* memiliki tiga nilai linguistik: tinggi, sedang, dan rendah dengan fungsi keanggotaan

$$\begin{aligned}\mu_{tinggi}(P) &= \frac{1}{1 + \exp\left(\frac{1,7 - P}{0,18}\right)} \\ \mu_{sedang}(P) &= \exp\left[-\frac{1}{2}\left(\frac{P - 1,2}{0,12}\right)^2\right] \\ \mu_{rendah}(P) &= \frac{1}{1 + \exp\left(\frac{P - 0,86}{0,09}\right)}\end{aligned}$$

dengan  $P$  menyatakan variabel kontrol *tekanan*.

Sembilan nilai linguistik di atas berada di dalam himpunan semesta  $\mathbb{R}^+$ . Misalkan SKLF tersebut adalah sebagai berikut

$\mathfrak{R}_1$  : Jika *volume* besar, *suhu* panas, maka *tekanan* sedang

$\mathfrak{R}_2$  : Jika *volume* kecil, *suhu* normal, maka *tekanan* tinggi

$\mathfrak{R}_3$  : Jika *volume* standar, *suhu* dingin, maka *tekanan* rendah

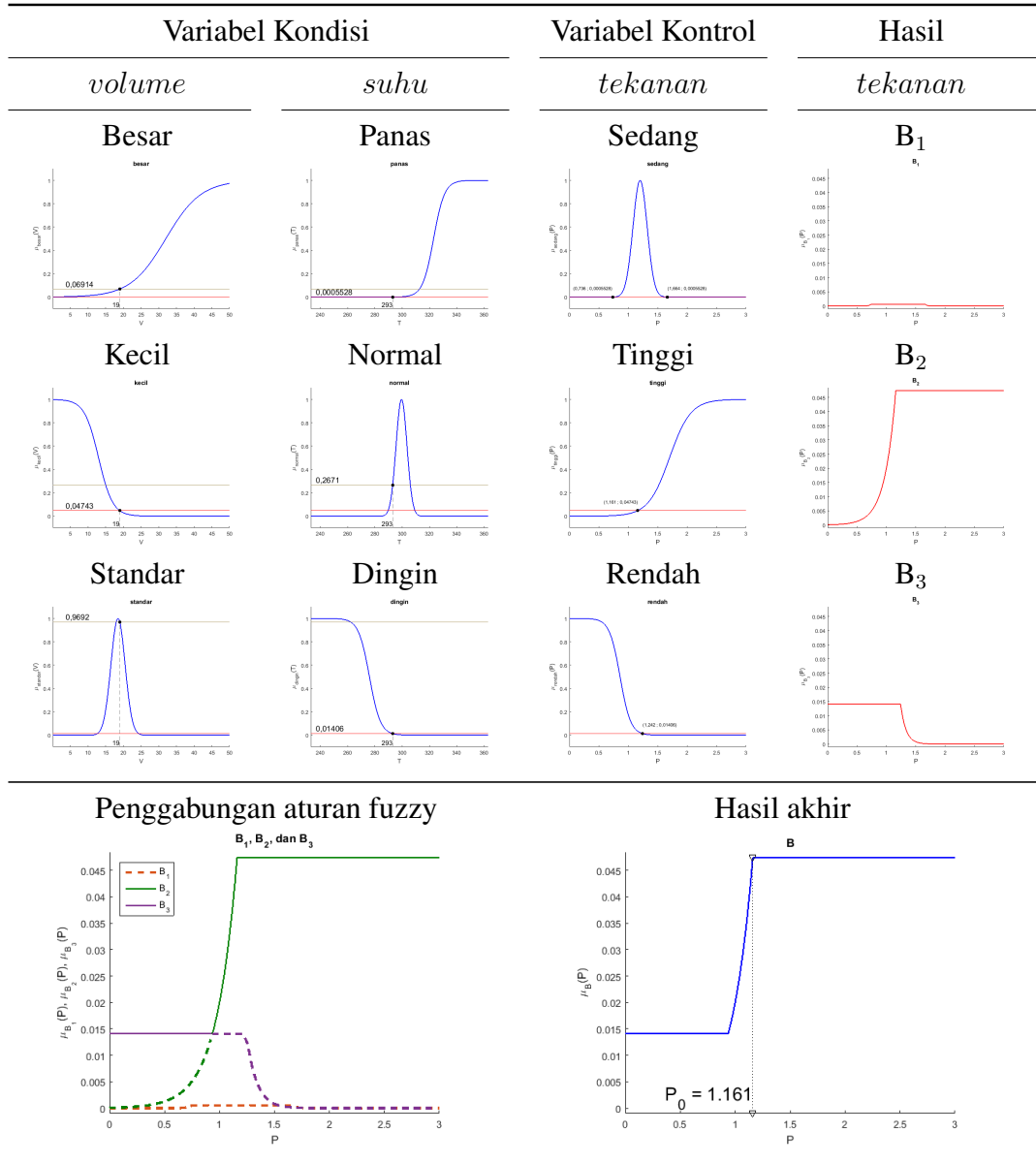
Fakta : *volume* = 19, *suhu* = 293

---

Konklusi : *tekanan* =  $P_0$

Misalkan digunakan operator minimum untuk *t-norm* dan operator maksimum untuk *t-conorm*. Inferensi mamdani untuk menyelesaikan SKLF ini dapat dilihat pada Tabel II.4. Lakukan fuzzifikasi terhadap serangkaian fakta dan operasi *t-norm*

Tabel II.4: Contoh mekanisme inferensi Mamdani



bagian pendahulu untuk setiap implikasi, sehingga diperoleh

$$\begin{aligned}
 \mu_{\text{besar}}(19) &= 0,0691, & \mu_{\text{panas}}(293) &= 0,0006, & \text{maka } \alpha_1 &= 0,0006, \\
 \mu_{\text{kecil}}(19) &= 0,0474, & \mu_{\text{normal}}(293) &= 0,2671, & \text{maka } \alpha_2 &= 0,0474, \\
 \mu_{\text{standar}}(19) &= 0,9692, & \mu_{\text{dingin}}(293) &= 0,0141, & \text{maka } \alpha_3 &= 0,0141
 \end{aligned}$$

Selanjutnya operasikan bagian pendahulu di atas dengan nilai linguistik dari variabel kontrol dengan menggunakan operator implikasi Mamdani. Akibatnya, diperoleh tiga himpunan fuzzy baru di  $\mathbb{R}^+$ , yaitu himpunan fuzzy  $B_1$ ,  $B_2$ , dan  $B_3$ .

Untuk setiap  $P \in \mathbb{R}^+$ , fungsi keanggotaan himpunan fuzzy  $B_1$ ,  $B_2$ , dan  $B_3$  adalah

$$\begin{aligned}\mu_{B_1}(P) &= \min\{\alpha_1, \mu_{sedang}(P)\} = \min\{0,0006, \mu_{sedang}(P)\} \\ &= \begin{cases} 0,0006, & 0,736 \leq P \leq 1,664 \\ \exp\left[-\frac{1}{2}\left(\frac{P-1,2}{0,12}\right)^2\right], & \text{lainnya} \end{cases} \\ \mu_{B_2}(P) &= \min\{\alpha_2, \mu_{tinggi}(P)\} = \min\{0,0474, \mu_{tinggi}(P)\} \\ &= \begin{cases} \frac{1}{1 + \exp\left(\frac{1,7-P}{0,18}\right)}, & P < 1,161 \\ 0,0474, & \text{lainnya} \end{cases} \\ \mu_{B_3}(P) &= \min\{\alpha_3, \mu_{rendah}(P)\} = \min\{0,0141, \mu_{rendah}(P)\} \\ &= \begin{cases} \frac{1}{1 + \exp\left(\frac{P-0,86}{0,09}\right)}, & P \geq 1,242 \\ 0,0141, & \text{lainnya} \end{cases}\end{aligned}$$

Gabungkan himpunan fuzzy  $B_1$ ,  $B_2$ , dan  $B_3$  menggunakan operasi *t-conorm*. Maka diperoleh himpunan fuzzy  $B$  di  $\mathbb{R}^+$  dengan fungsi keanggotaan

$$\begin{aligned}\mu_B(P) &= \max\{\mu_{B_1}(P), \mu_{B_2}(P), \mu_{B_3}(P)\} \\ &= \begin{cases} 0,0141, & P < 1,242 \\ \frac{1}{1 + \exp\left(\frac{1,7-P}{0,18}\right)}, & 1,242 \leq P < 1,161 \\ 0,0474, & P \geq 1,161 \end{cases}\end{aligned}$$

Misalkan operator defuzzifikasi yang digunakan adalah *first-of-maxima*. Maka diperoleh

$$\begin{aligned}P_0 &= \min\left\{P \in \mathbb{R}^+ : \mu_B(P) = \max_w \mu_B(w) = 0,0474\right\} \\ &= \min\{P \in \mathbb{R}^+ : P \geq 1,161\} \\ &= 1,161\end{aligned}$$

Dengan demikian, tindakan kontrol dari SKLF ini adalah  $P_0 = 1,161$ . Artinya,



ketika volume suatu gas ideal sebesar 19 liter dan suhunya bernilai 293 Kelvin, maka berdasarkan aturan fuzzy ini dan mekanisme inferensi mamdani, diperoleh tekanan sebesar 1,161 atm.

### Mekanisme Inferensi Takagi-Sugeno-Kang

Inferensi TSK merupakan inferensi fuzzy yang dilakukan terhadap aturan fuzzy TSK. Pada dasarnya, mekanisme inferensi TSK sama seperti mekanisme inferensi Mamdani. Operator implikasi menggunakan operator Mamdani, dan menggunakan “atau” sebagai penghubung antar implikasi. Tetapi, inferensi TSK memiliki kasus khusus berupa nilai linguistik variabel kontrol yang diskrit dan menyerupai pendefinisian himpunan klasik, sebagaimana telah diformulasikan pada Persamaan (2.12). Selain itu, metode defuzzifikasi yang digunakan pada SKLF ini hanya metode *center-of-area* karena himpunan fuzzy yang dihasilkan dari inferensi ini merupakan himpunan fuzzy diskrit.

Misalkan diberikan himpunan fuzzy  $A_{1,j}, A_{2,j}, \dots, A_{m,j}$  di  $X_j$  dengan  $j = 1, 2, \dots, n$ ,  $\mathbf{b}_i = (b_{i,0}, b_{i,1}, \dots, b_{i,n})$  dengan  $i = 1, 2, \dots, m$ , dan  $\tilde{\mathbf{x}} = (1, x_1, x_2, \dots, x_n)$ . Asumsikan  $\mathbf{b}_i \neq \mathbf{b}_k$  untuk setiap  $i \neq k$ . Kemudian diberikan SKLF berikut ini

$\mathfrak{R}_1$ :	Jika	$x_1$ anggota $A_{1,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{1,n}$ ,	maka	$y = \mathbf{b}_1 \cdot \tilde{\mathbf{x}}$
$\mathfrak{R}_2$ :	Jika	$x_1$ anggota $A_{2,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{2,n}$ ,	maka	$y = \mathbf{b}_2 \cdot \tilde{\mathbf{x}}$
						$\vdots$
$\mathfrak{R}_m$ :	Jika	$x_1$ anggota $A_{m,1}$ ,	$\dots$ ,	$x_n$ anggota $A_{m,n}$ ,	maka	$y = \mathbf{b}_m \cdot \tilde{\mathbf{x}}$
Fakta :		$x_1 = a_1$ ,	$\dots$ ,	$x_n = a_n$		
Konklusi :						$y = y_0$

Pandang bagian konsekuensi dari setiap implikasi di atas sebagai “ $y$  adalah  $B_i$ ” untuk setiap  $i = 1, 2, \dots, m$  dengan fungsi keanggotaan  $\mu_{B_i}$  sama seperti fungsi keanggotaan pada (2.12). Karena terdapat serangkaian fakta  $x_1 = a_1, x_2 = a_2, \dots, x_n = a_n$ , maka fungsi keanggotaan  $\mu_{B_i}$  menjadi

$$\mu_{B_i} = \begin{cases} 1, & \text{jika } y = \mathbf{b}_i \cdot \tilde{\mathbf{a}}, \quad \tilde{\mathbf{a}} = (1, a_1, a_2, \dots, a_n) \\ 0, & \text{lainnya} \end{cases}$$

Langkah-langkah untuk mendapatkan nilai  $y_0$  sebagai tindakan kontrol dari SKLF

ini sama seperti langkah-langkah pada inferensi Mamdani. Perbedaannya terletak pada kasus khusus yang ada di variabel kontrol. Rincian langkah-langkahnya adalah sebagai berikut

1. Lakukan fuzzifikasi terhadap serangkaian fakta berdasarkan nilai linguistik dari variabel kondisi. Kemudian operasikan hasil fuzzifikasi tersebut dengan operasi konjungsi fuzzy (*t-norm*). Misalkan hasil operasi ini dinyatakan dengan  $\alpha_i$  untuk  $i = 1, 2, \dots, m$ . Maka diperoleh

$$\alpha_i = T(\mu_{A_{i,1}}(a_1), \mu_{A_{i,2}}(a_2), \dots, \mu_{A_{i,n}}(a_n)), \quad i = 1, 2, \dots, m$$

2. Operasikan bagian pendahulu dan bagian konsekuensi pada setiap implikasi menggunakan operator mamdani. Maka diperoleh sebanyak  $m$  himpunan fuzzy baru di  $Y$ , yaitu himpunan fuzzy  $B'_1, B'_2, \dots, B'_m$  dengan fungsi keanggotaan

$$\begin{aligned} \mu_{B'_i}(y) &= \min \{ \alpha_i, \mu_{B_i}(y) \}, \quad y \in Y \quad i = 1, 2, \dots, m \\ &= \begin{cases} \min \{ \alpha_i, 1 \} = \alpha_i, & y = \mathbf{b}_i \cdot \tilde{\mathbf{a}} \\ \min \{ \alpha_i, 0 \} = 0, & y \text{ lainnya} \end{cases} \end{aligned}$$

3. Selanjutnya, dilakukan operasi *t-conorm* untuk menggabungkan sebanyak  $m$  himpunan fuzzy dari setiap implikasi. Maka diperoleh himpunan fuzzy baru di  $Y$ , yaitu himpunan fuzzy  $B$  dengan fungsi keanggotaan

$$\mu_B(y) = S(\mu_{B'_1}(y), \mu_{B'_2}(y), \dots, \mu_{B'_m}(y)), \quad y \in Y$$

Karena untuk setiap  $i = 1, 2, \dots, m$   $\mu_{B'_i}(y) = \alpha_i$  hanya dipenuhi oleh  $y = \mathbf{b}_i \cdot \tilde{\mathbf{a}}$  dan  $\mu_{B'_i}(y) = 0$  untuk  $y$  yang lain, maka

$$\mu_B(y) = \begin{cases} S(\alpha_1, 0, \dots, 0) = \alpha_1, & y = \mathbf{b}_1 \cdot \tilde{\mathbf{a}} \\ \vdots & \vdots \\ S(0, 0, \dots, \alpha_i, \dots, 0, 0) = \alpha_i, & y = \mathbf{b}_i \cdot \tilde{\mathbf{a}} \\ \vdots & \vdots \\ S(0, 0, \dots, 0, \alpha_m) = \alpha_m, & y = \mathbf{b}_m \cdot \tilde{\mathbf{a}} \\ 0, & y \text{ lainnya} \end{cases}$$

Dengan demikian,  $B$  merupakan himpunan fuzzy diskrit.

4. Langkah terakhir, lakukan defuzzifikasi menggunakan metode *center-of-area*

terhadap himpunan fuzzy  $B$  untuk mendapatkan nilai dari tindakan kontrol  $y_0$ , sehingga diperoleh

$$y_0 = \frac{\sum_{y \in Y} \mu_B(y)y}{\sum_{y \in Y} \mu_B(y)}$$

Misalkan  $y_i = b_i \cdot \tilde{a}$  untuk setiap  $i = 1, 2, \dots, m$ . Maka  $\mu_B(y) > 0$  hanya berlaku untuk  $y \in \{y_1, y_2, \dots, y_m\}$ . Dengan demikian, diperoleh

$$y_0 = \frac{\sum_{i=1}^m \mu_B(y_i)y_i}{\sum_{i=1}^m \mu_B(y_i)} = \frac{\sum_{i=1}^m \alpha_i y_i}{\sum_{i=1}^m \alpha_i}$$

Contoh II.4 berikut ini memberikan gambaran tambahan mengenai mekanisme inferensi TSK.

**Contoh II.4** Misalkan diberikan sistem gas ideal yang terdiri dari volume ( $V$ ) dengan satuan *liter*, suhu ( $T$ ) dengan satuan *Kelvin*, dan tekanan ( $P$ ) dengan satuan *atm*. Sistem gas ideal ini dapat dipandang sebagai SKLF dengan  $V$  dan  $T$  sebagai variabel kondisi dan  $P$  adalah variabel kontrolnya. Misalkan

- Variabel kondisi  $V$  memiliki tiga nilai linguistik: besar, standar, dan kecil. Fungsi keanggotaan dari tiga nilai linguistik ini didefinisikan pada Contoh II.3.
- Variabel kondisi  $T$  memiliki tiga nilai linguistik: panas, normal, dan dingin dengan fungsi keanggotaan didefinisikan pada Contoh II.3.

Misalkan SKLF tersebut adalah sebagai berikut

$$\mathfrak{R}_1 : \text{Jika } V \text{ besar, } T \text{ panas, maka } P = 1,181 - 0,045V + 0,003T$$

$$\mathfrak{R}_2 : \text{Jika } V \text{ kecil, } T \text{ normal, maka } P = 2,3312 + 0,0711V - 0,003T$$

$$\mathfrak{R}_3 : \text{Jika } V \text{ standar, } T \text{ dingin, maka } P = 21,71 - 0,05V - 0,07T$$

$$\text{Fakta : } V = 19, \quad T = 293$$

---


$$\text{Konklusi : } P = P_0$$

Misalkan digunakan operator minimum untuk  $t$ -norm. Lakukan fuzzifikasi terhadap serangkaian fakta dan operasi  $t$ -norm bagian pendahulu untuk setiap implikasi, sehingga diperoleh

$$\begin{aligned}\mu_{besar}(19) &= 0,0691, & \mu_{panas}(293) &= 0,0006, & \text{maka } \alpha_1 &= 0,0006, \\ \mu_{kecil}(19) &= 0,0474, & \mu_{normal}(293) &= 0,2671, & \text{maka } \alpha_2 &= 0,0474, \\ \mu_{standar}(19) &= 0,9692, & \mu_{dingin}(293) &= 0,0141, & \text{maka } \alpha_3 &= 0,0141\end{aligned}$$

Selanjutnya, tentukan nilai  $P_1$ ,  $P_2$ , dan  $P_3$  berdasarkan serangkaian fakta dan variabel kontrol  $P$  pada setiap konsekuensi.

$$\begin{aligned}P_1 &= 1,181 - 0,045(19) + 0,003(293) &= 1,205 \\ P_2 &= 2,3312 + 0,0711(19) - 0,003(293) &= 2,8031 \\ P_3 &= 21,71 - 0,05(19) - 0,07(293) &= 0,25\end{aligned}$$

Dengan demikian, diperoleh tindakan kontrol  $P_0$ , yaitu

$$P_0 = \frac{\sum_{i=1}^3 \alpha_i P_i}{\sum_{i=1}^3 \alpha_i} = \frac{0,0006(1,205) + 0,0474(2,8031) + 0,0141(0,25)}{0,0006 + 0,0474 + 0,0141} = 2,2079$$

## II.6 Konstruksi Jaringan Saraf Fuzzy

Aturan fuzzy yang terdapat pada SKLF yang dibahas pada Subbab II.5 berasal dari pengetahuan para ahli. Pada subbab ini akan dipelajari cara membangun aturan fuzzy yang bukan berasal dari pengetahuan para ahli, melainkan berdasarkan data masukan dan keluaran yang diberikan. Aturan fuzzy yang akan dibangun hanya terbatas pada aturan fuzzy TSK. Fungsi keanggotaan dari nilai linguistik variabel kondisi hanya menggunakan fungsi Gauss, yaitu

$$g(x; m, s) = \exp \left[ -\frac{1}{2} \left( \frac{x - m}{s} \right)^2 \right] \quad (2.13)$$

dengan  $m$  dan  $s$  merupakan parameter tertentu yang akan dicari.

Misalkan diberikan variabel kondisi  $x_1, x_2, \dots, x_n$ . Untuk setiap variabel kondisi  $x_j$  memiliki nilai linguistik berupa himpunan fuzzy  $A_{1,j}, A_{2,j}, \dots, A_{r,j}$  di  $X_j$  dengan

$j = 1, 2, \dots, n$ . Setiap himpunan fuzzy  $A_{i,j}$  memiliki fungsi keanggotaan berupa fungsi Gauss, yaitu

$$\mu_{A_{i,j}}(x_j) = g(x_j; m_{i,j}, s_{i,j})$$

Untuk menyingkat penulisan notasi, derajat keanggotaan  $x_j$  di himpunan fuzzy  $A_{i,j}$  dinyatakan sebagai  $G_{i,j}$ , dengan kata lain  $G_{i,j} = \mu_{A_{i,j}}(x_j)$ . Fungsi  $g$  didefinisikan pada Persamaan (2.13). Selanjutnya, diberikan variabel kontrol  $y_1, y_2, \dots, y_p$ . Variabel-variabel kontrol ini dapat ditulis ke dalam bentuk vektor menjadi

$$\mathbf{y} = (y_1, y_2, \dots, y_p)$$

Untuk setiap implikasi ke- $i$ , entri di dalam vektor variabel kontrol ini ditentukan oleh  $\mathbf{y}^T = \mathbf{B}_i \tilde{\mathbf{x}}^T$  dengan  $i = 1, 2, \dots, r$  dan

$$\mathbf{B}_i = \begin{bmatrix} b_{i,1,0} & b_{i,1,1} & b_{i,1,2} & \dots & b_{i,1,n} \\ b_{i,2,0} & b_{i,2,1} & b_{i,2,2} & \dots & b_{i,2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{i,p,0} & b_{i,p,1} & b_{i,p,2} & \dots & b_{i,p,n} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{i,1} \\ \mathbf{b}_{i,2} \\ \vdots \\ \mathbf{b}_{i,p} \end{bmatrix} \quad (2.14)$$

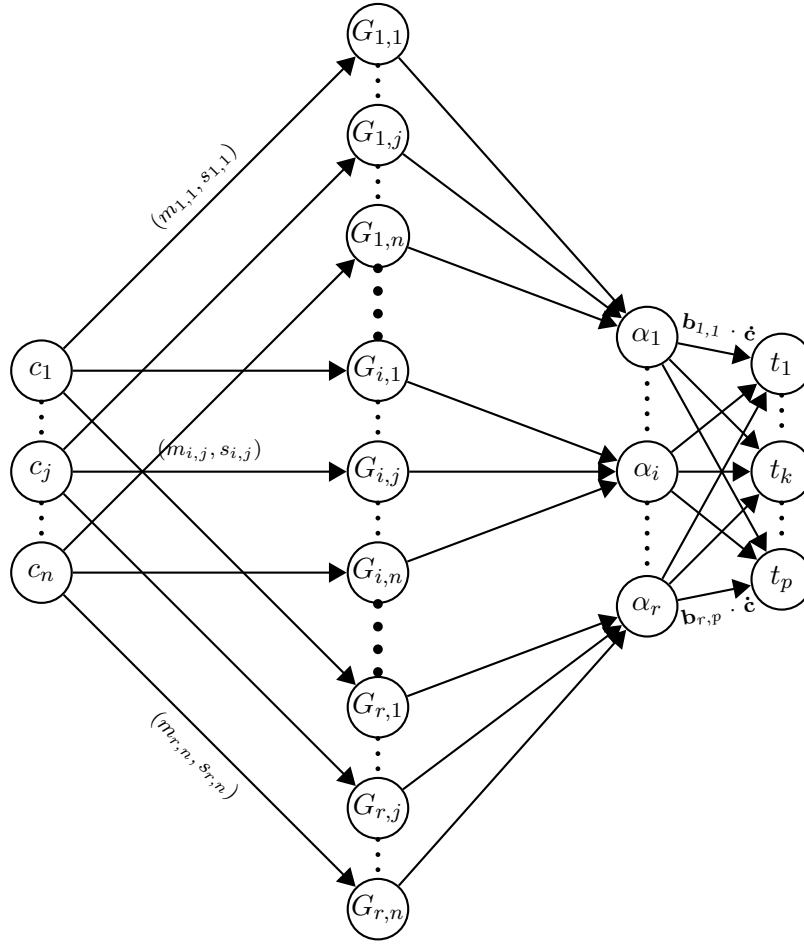
$$\tilde{\mathbf{x}} = (1, x_1, x_2, \dots, x_n)$$

Kemudian diberikan serangkaian fakta  $x_1 = c_1, x_2 = c_2, \dots, x_n = c_n$  dan konklusi yang berupa tindakan kontrol  $\mathbf{y} = \mathbf{d} = (d_1, d_2, \dots, d_p)$ . Setiap fakta  $c_j$  dengan  $j = 1, 2, \dots, n$  berasal dari data masukan pada suatu observasi dan setiap tindakan kontrol  $d_k$  dengan  $k = 1, 2, \dots, p$  berasal dari data keluaran pada observasi tersebut. Berdasarkan premis-premis ini, diperoleh SKLF berikut

$\mathfrak{R}_1$ :	Jika $x_1$ anggota $A_{1,1}, \dots, x_n$ anggota $A_{1,n}$ ,	maka $\mathbf{y}^T = \mathbf{B}_1 \tilde{\mathbf{x}}^T$
$\mathfrak{R}_2$ :	Jika $x_1$ anggota $A_{2,1}, \dots, x_n$ anggota $A_{2,n}$ ,	maka $\mathbf{y}^T = \mathbf{B}_2 \tilde{\mathbf{x}}^T$
	$\vdots$	
$\mathfrak{R}_r$ :	Jika $x_1$ anggota $A_{r,1}, \dots, x_n$ anggota $A_{r,n}$ ,	maka $\mathbf{y}^T = \mathbf{B}_r \tilde{\mathbf{x}}^T$
Fakta :	$x_1 = c_1, \dots, x_n = c_n$	
Konklusi :		$\mathbf{y} = \mathbf{d}$

(2.15)

SKLF pada (2.15) dapat diilustrasikan sebagai suatu jaringan saraf yang dapat dilihat pada Gambar II.3. Jaringan saraf ini disebut dengan jaringan saraf fuzzy (JSF). Hal ini dikarenakan jaringan tersebut dibangun dari aturan fuzzy dan penentuan nilai dari neuron-neuronnya berdasarkan tahapan-tahapan pada SKLF.



Gambar II.3: Jaringan saraf fuzzy

JSF untuk kasus ini terdiri dari lapisan masukan, dua lapisan tersembunyi, dan lapisan keluaran. Neuron pada lapisan masukan diperoleh dari serangkaian fakta pada SKLF (2.15). Neuron pada lapisan tersembunyi pertama adalah hasil dari fuzzifikasi dari neuron pada lapisan masukan. Neuron pada lapisan tersembunyi kedua merupakan hasil operasi  $t$ -norm dari blok-blok neuron pada lapisan tersembunyi pertama. Penentuan blok ini berdasarkan aturan fuzzy pada SKLF (2.15). Pada jaringan saraf fuzzy, operator  $t$ -norm yang digunakan adalah operator probabilitas (perkalian). Maka nilai dari neuron-neuron pada lapisan tersembunyi kedua adalah

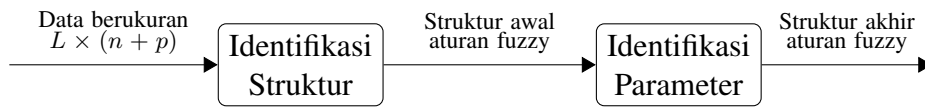
$$\begin{aligned}\alpha_i &= \prod_{j=1}^n G_{i,j} = \prod_{j=1}^n g(c_j; m_{i,j}, s_{i,j}) = \prod_{j=1}^n \exp \left[ -\frac{1}{2} \left( \frac{c_j - m_{i,j}}{s_{i,j}} \right)^2 \right] \\ &= \exp \left[ -\frac{1}{2} \sum_{j=1}^n \left( \frac{c_j - m_{i,j}}{s_{i,j}} \right)^2 \right], \quad i = 1, 2, \dots, r\end{aligned}$$

Neuron pada lapisan keluaran merupakan tindakan kontrol aktual yang dinyatakan dengan  $\mathbf{t} = (t_1, t_2, \dots, t_p)$ , yaitu

$$\mathbf{t}^T = \frac{\sum_{i=1}^r \alpha_i \mathbf{B}_i \tilde{\mathbf{c}}^T}{\sum_{i=1}^r \alpha_i}$$

dengan  $\tilde{\mathbf{c}} = (1, c_1, c_2, \dots, c_n)$  dan matriks  $\mathbf{B}_i$  didefinisikan pada (2.14). Tindakan kontrol aktual  $\mathbf{t}$  tidak sama dengan tindakan kontrol yang diinginkan pada SKLF 2.15 (d). Tetapi,  $\mathbf{t}$  yang diperoleh harus sangat dekat dengan  $\mathbf{d}$ .

Berdasarkan penjelasan di atas, entri pada vektor  $\mathbf{t}$  ditentukan dari proses *feedforward* pada JSF. Proses *feedforward* ini analog dengan seluruh tahapan pada SKLF, yaitu: fuzzifikasi, inferensi, dan defuzzifikasi. Nilai neuron-neuron pada lapisan tersembunyi pertama bergantung kepada nilai dari parameter  $m_{i,j}$  dan  $s_{i,j}$  dengan  $i = 1, 2, \dots, r$  dan  $j = 1, 2, \dots, n$ . Nilai neuron-neuron pada lapisan tersembunyi kedua tidak dipengaruhi oleh parameter apapun karena hanya dilakukan operasi *t-norm*. Nilai  $\mathbf{t}$  sangat dipengaruhi oleh parameter  $b_{i,k,0}$  dan  $b_{i,k,j}$  dengan  $i = 1, 2, \dots, r$ ,  $k = 1, 2, \dots, p$  dan  $j = 1, 2, \dots, n$ . Dengan demikian, target utama dari konstruksi JSF adalah menentukan nilai dari parameter-parameter  $m_{i,j}$ ,  $s_{i,j}$ ,  $b_{i,k,0}$ , dan  $b_{i,k,j}$  sedemikian sehingga dapat meminimalkan rata-rata galat kuadrat atau *mean square error* (MSE) antara tindakan kontrol aktual  $(t_1, t_2, \dots, t_p)$  dan tindakan kontrol yang diinginkan  $(d_1, d_2, \dots, d_p)$ .



Gambar II.4: Dua fase pada konstruksi jaringan saraf fuzzy

Misalkan diberikan data dengan observasi sebanyak  $L$ . Misalkan setiap observasi memiliki sebanyak  $n$  masukan dan  $p$  keluaran, sehingga setiap observasi dapat dipandang sebagai vektor masukan dengan dimensi  $n$  dan vektor keluaran dengan dimensi  $p$ . Maka dapat dikatakan bahwa data ini memiliki ukuran sebesar  $L \times (n + p)$ . Menurut Lee dan Ouyang (2003) dan Yeh dkk. (2012), konstruksi JSF berdasarkan data ini harus melalui dua fase, yaitu: identifikasi struktur dan identifikasi parameter. Fase identifikasi struktur dilakukan untuk memperoleh struktur

awal dari aturan fuzzy. Struktur awal ini meliputi banyaknya implikasi pada aturan fuzzy dan inisialisasi nilai dari setiap parameter. Fase identifikasi parameter dilakukan untuk memperbaiki nilai dari parameter-parameter  $m_{i,j}$ ,  $s_{i,j}$ ,  $b_{i,k,0}$ , dan  $b_{i,k,j}$  sedemikian sehingga target utama dari konstruksi JSF tercapai. Dua fase ini telah diringkas di dalam Gambar II.4.

### II.6.1 Identifikasi Struktur

Pada fase ini, terdapat dua tahap utama. Pertama, pengelompokan data ke dalam kluster menggunakan algoritma *self-constructing clustering*. Tahap pertama ini akan menghasilkan kluster-kluster dari data yang diberikan. Setiap Kluster ke- $i$  memiliki parameter  $\mathbf{m}_i = (m_{i,1}, m_{i,2}, \dots, m_{i,n})$  dan  $\mathbf{s}_i = (s_{i,1}, s_{i,2}, \dots, s_{i,n})$ . Parameter  $\mathbf{m}_i$  dan  $\mathbf{s}_i$ , berturut-turut, adalah vektor rata-rata dan vektor simpangan baku dari vektor data masukan yang ada di kluster ke- $i$ . Nantinya, setiap parameter  $m_{i,j}$  dan  $s_{i,j}$  yang diperoleh dari tahap ini menjadi nilai awal dari parameter untuk fungsi Gauss yang bersesuaian. Oleh karena itu, kaitannya dengan JSF dan SKLF, setiap kluster dapat dipandang sebagai satu implikasi pada aturan fuzzy. Selanjutnya, tahap kedua adalah menentukan nilai dari parameter pada variabel kontrol, yaitu entri dari setiap matriks  $\mathbf{B}_i$ , menggunakan metode dekomposisi nilai singular.

#### *Self-constructing Clustering*

Asumsikan data yang akan diolah berukuran  $L \times (n + p)$ . Setiap observasi ke- $l$  dalam data ini memiliki pola  $[\mathbf{c}^{(l)}, \mathbf{d}^{(l)}]$ , dengan

$$\mathbf{c}^{(l)} = (c_1^{(l)}, c_2^{(l)}, \dots, c_n^{(l)})$$

menyatakan  $n$  nilai data masukan pada observasi ke- $l$  dan

$$\mathbf{d}^{(l)} = (d_1^{(l)}, d_2^{(l)}, \dots, d_p^{(l)})$$

menyatakan  $p$  nilai data keluaran pada observasi ke- $l$ .

Sebagaimana telah dijelaskan sebelumnya, algoritma *self-constructing clustering* mengelompokkan masing-masing observasi dari data ke dalam kluster. Pengelompokan ini berdasarkan uji kemiripan data masukan dan uji kemiripan data keluaran. Setiap kluster memuat beberapa observasi tertentu dari data. Setiap kluster dikarakterisasi oleh hasil operasi *t-norm* dari fungsi keanggotaan nilai linguistik dari



variabel kondisi. Karena fungsi keanggotaan yang digunakan adalah fungsi Gauss dan operator  $t$ -norm yang digunakan adalah perkalian, maka setiap kluster dikarakterisasi oleh hasil kali dari sebanyak  $n$  fungsi Gauss. Selain itu, setiap kluster juga memiliki vektor ketinggian, yaitu vektor rata-rata dari data keluaran yang ada di dalam kluster tersebut (Yeh dkk., 2012).

Dalam *self-constructing clustering*, setiap observasi dari data diperiksa satu per satu. Pada awalnya, observasi pertama dimasukkan ke dalam kluster pertama. Untuk setiap observasi berikutnya, diuji kemiripan data masukan dan kemiripan data keluaran antara observasi tersebut dengan kluster-kluster yang telah ada. Pengujian kemiripan ini akan memutuskan apakah observasi tersebut dimasukkan ke dalam kluster tertentu yang telah ada atau observasi tersebut menjadi anggota pertama pada kluster yang baru. Setelah semua observasi diperiksa, akan diperoleh banyaknya kluster yang terbentuk. Rincian dari *self-constructing clustering* akan dijelaskan di bawah ini.

Misalkan  $Q_1, Q_2, \dots, Q_r$  adalah kluster-kluster yang telah terbentuk. Setiap kluster  $Q_i$  memiliki

- vektor rata-rata dari data masukan, yaitu  $\mathbf{m}_i = (m_{i,1}, m_{i,2}, \dots, m_{i,n})$ ,
- vektor simpangan baku dari data masukan, yaitu  $\mathbf{s}_i = (s_{i,1}, s_{i,2}, \dots, s_{i,n})$ , dan
- vektor ketinggian, yaitu  $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \dots, h_{i,p})$ .

Misalkan  $|Q_i|$  adalah banyaknya observasi yang ada di dalam kluster  $Q_i$ . Pada proses awal dari *self-constructing clustering*,  $r = 1$  dan observasi pertama berada di dalam kluster  $Q_1$ . Akibatnya, pada proses awal ini, diperoleh  $\mathbf{m}_1 = (c_1^{(1)}, c_2^{(1)}, \dots, c_n^{(1)})$  dan  $\mathbf{h}_1 = (d_1^{(1)}, d_2^{(1)}, \dots, d_p^{(1)})$ . Simpangan baku tidak terdefinisi pada data yang hanya memuat satu observasi, maka pada proses awal ini, vektor  $\mathbf{s}_1$  yang berdimensi  $n$  didefinisikan oleh  $\mathbf{s}_1 = \mathbf{s}_0 = (\underbrace{s_0, s_0, \dots, s_0}_n)$  dengan  $s_0$  adalah suatu bilangan real positif. Untuk observasi ke- $l$ , yaitu  $[\mathbf{c}^{(l)}, \mathbf{d}^{(l)}]$  dengan  $l = 2, 3, \dots, L$ , akan dihitung kemiripan antara observasi ke- $l$  tersebut dengan kluster  $Q_i$ . Kemiripan data masukan antara observasi ke- $l$  dengan kluster  $Q_i$  untuk  $i = 1, 2, \dots, r$  adalah sebagai berikut

$$\alpha_i(\mathbf{c}^{(l)}) = \prod_{j=1}^n g(c_j^{(l)}; m_{i,j}, s_{i,j}) = \prod_{j=1}^n \exp \left[ -\frac{1}{2} \left( \frac{c_j^{(l)} - m_{i,j}}{s_{i,j}} \right)^2 \right]$$

$$\Longleftrightarrow \alpha_i(\mathbf{c}^{(l)}) = \exp \left[ -\frac{1}{2} \sum_{j=1}^n \left( \frac{c_j^{(l)} - m_{i,j}}{s_{i,j}} \right)^2 \right] \quad (2.16)$$

Observasi ke- $l$  dikatakan mirip dengan klaster  $Q_i$  jika memenuhi

$$\alpha_i(\mathbf{c}^{(l)}) \geq \rho \quad (2.17)$$

dengan  $\rho$  adalah suatu bilangan real pada selang  $[0, 1]$ . Munculnya kriteria ini dikarenakan  $\alpha_i(\mathbf{c}^{(l)}) \approx 1$  ketika vektor  $\mathbf{c}^{(l)}$  sangat dekat dengan vektor  $\mathbf{m}_i$  dan  $\alpha_i(\mathbf{c}^{(l)}) \approx 0$  ketika vektor  $\mathbf{c}^{(l)}$  sangat jauh dari vektor  $\mathbf{m}_i$ . Nilai  $\rho$  disebut sebagai ambang batas minimal kemiripan data masukan. Selanjutnya, dihitung kemiripan data keluaran antara observasi ke- $l$  dengan klaster  $Q_i$  sebagai berikut

$$e_i(\mathbf{d}^{(l)}) = \|\mathbf{d}^{(l)} - \mathbf{h}_i\| \quad (2.18)$$

Observasi ke- $l$  dikatakan mirip dengan klaster  $Q_i$  jika memenuhi

$$e_i(\mathbf{d}^{(l)}) \leq \tau u \quad (2.19)$$

dengan  $\tau \in [0, 1]$  dan  $u = \max_{l, l' \in \{1, 2, \dots, L\}} \|\mathbf{d}^{(l)} - \mathbf{d}^{(l')}\|$ . Munculnya kriteria ini dikarenakan nilai dari  $e_i(\mathbf{d}^{(l)})$  akan menuju 0 jika vektor  $\mathbf{d}^{(l)}$  sangat dekat dengan vektor  $\mathbf{h}_i$  dan nilai dari  $e_i(\mathbf{d}^{(l)})$  akan menuju  $u$  jika vektor  $\mathbf{d}^{(l)}$  sangat jauh dari vektor  $\mathbf{h}_i$ . Nilai  $\tau u$  disebut sebagai ambang batas maksimal kemiripan data keluaran.

Setelah menghitung kemiripan data masukan dan kemiripan data keluaran antara sampel ke- $l$  dengan klaster yang telah ada, akan muncul dua kasus yang mungkin terjadi. Kasus pertama, observasi ke- $l$  tidak mirip dengan semua klaster yang telah ada. Dengan kata lain, tidak ada  $i \in \{1, 2, \dots, r\}$  yang memenuhi Pertidaksamaan (2.17) dan (2.19) untuk observasi ke- $l$ . Untuk kasus ini,  $r$  bertambah satu, sehingga terbentuk klaster baru. Proses pembentukan klaster baru ini seperti pada proses pembentukan klaster pertama, yaitu

$$\mathbf{m}_r = \mathbf{c}^{(l)}, \mathbf{s}_r = \mathbf{s}_0, \mathbf{h}_r = \mathbf{d}^{(l)} \text{ dengan } r = r + 1 \quad (2.20)$$

Kasus kedua, observasi ke- $l$  mirip dengan beberapa klaster yang telah ada. Dengan kata lain, terdapat  $i \in \{1, 2, \dots, r\}$  yang memenuhi Pertidaksamaan (2.17) dan (2.19) untuk observasi ke- $l$ . Misalkan observasi ke- $l$  mirip dengan klaster

$Q_{i_1}, Q_{i_2}, \dots, Q_{i_f}$  dan

$$v = \arg \max_{i=i_1, i_2, \dots, i_f} \alpha_i(\mathbf{c}^{(l)}) \quad (2.21)$$

Maka dalam kasus ini, dapat diasumsikan bahwa klaster yang paling dekat dengan observasi ke- $l$  adalah klaster  $Q_v$ . Akibatnya, observasi ke- $l$  menjadi anggota baru dari klaster  $Q_v$ , sehingga entri dari vektor  $\mathbf{m}_v$ ,  $\mathbf{s}_v$ , dan  $\mathbf{h}_v$  berubah. Perubahan entri dari vektor-vektor ini mengikuti urutan pendefinisian di bawah ini

$$\gamma = \frac{(|Q_v| - 1) \left(s_{v,j}^{(\text{lama})}\right)^2 + |Q_v| \left(m_{v,j}^{(\text{lama})}\right)^2 + \left(c_j^{(l)}\right)^2}{|Q_v|} \quad (2.22)$$

$$m_{v,j}^{(\text{baru})} = \frac{|Q_v| m_{v,j}^{(\text{lama})} + c_j^{(l)}}{|Q_v| + 1} \quad (2.23)$$

$$\omega = \frac{(|Q_v| + 1) \left(m_{v,j}^{(\text{baru})}\right)^2}{|Q_v|} \quad (2.24)$$

$$s_{v,j}^{(\text{baru})} = \sqrt{\gamma - \omega} \quad (2.25)$$

$$h_{v,k}^{(\text{baru})} = \frac{|Q_v| h_{v,k}^{(\text{lama})} + d_k^{(l)}}{|Q_v| + 1} \quad (2.26)$$

$$|Q_v| = |Q_v| + 1 \quad (2.27)$$

untuk  $j = 1, 2, \dots, n$  dan  $k = 1, 2, \dots, p$ . Jika terdapat  $j$  sedemikian sehingga  $s_{v,j}^{(\text{baru})} = 0$ , maka  $s_{v,j}^{(\text{baru})}$  tersebut dimodifikasi menjadi  $s_{v,j}^{(\text{baru})} = s_0$ . Hal ini dilakukan karena fungsi Gauss tidak akan terdefinisi ketika  $s_{v,j}^{(\text{baru})} = 0$ .

Misalkan diperoleh sebanyak  $r$  klaster setelah semua observasi data diperiksa. Maka diperoleh aturan fuzzy dengan implikasi sebanyak  $r$ , yaitu  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_r$ . Setiap implikasi  $\mathfrak{R}_i$  memiliki bentuk yang sama dengan implikasi pada (2.15) dengan fungsi keanggotaan dari  $A_{i,j}$ ,  $j = 1, 2, \dots, n$  adalah

$$\mu_{A_{i,j}}(x_j) = g(x_j; m_{i,j}, s_{i,j})$$

dengan  $m_{i,j}$  dan  $s_{i,j}$ , berturut-turut, adalah rata-rata dan simpangan baku dari data masukan ke- $j$  yang ada di dalam klaster  $Q_i$ . Fungsi  $g$  telah didefinisikan pada (2.13).

## Dekomposisi Nilai Singular

Dekomposisi nilai singular atau *singular value decomposition* (SVD) digunakan untuk menentukan parameter pada variabel kontrol dari setiap implikasi  $\mathfrak{R}_i$ , yaitu matriks  $\mathbf{B}_i$  yang telah didefinisikan pada Persamaan (2.14). Penentuan entri dari matriks  $\mathbf{B}_i$  ini berdasarkan pada observasi-observasi data masukan dan keluaran yang ada di dalam kluster  $Q_i$ . Misalkan  $L_i = |Q_i|$  menyatakan banyaknya observasi yang merupakan anggota dari kluster  $Q_i$ . Misalkan juga observasi yang berada di dalam kluster  $Q_i$  ini adalah  $[\mathbf{c}^{(i_1)}, \mathbf{d}^{(i_1)}], [\mathbf{c}^{(i_2)}, \mathbf{d}^{(i_2)}], \dots, [\mathbf{c}^{(i_{L_i})}, \mathbf{d}^{(i_{L_i})}]$ . Maka matriks  $\mathbf{B}_i$  merupakan solusi kuadrat terkecil dari  $\|\mathbf{D} - \tilde{\mathbf{C}}\mathbf{B}_i\|$ . Notasi  $\|\mathbf{X}\|$  untuk suatu matriks  $\mathbf{X}$  menyatakan *norm* dari matriks  $\mathbf{X}$  yang didefinisikan oleh  $\|\mathbf{X}\| = \sqrt{\text{trace}(\mathbf{X}^T \mathbf{X})}$ . Jadi, target utama dari SVD adalah mencari solusi optimal dari masalah optimisasi berikut ini

$$\min_{\mathbf{B}_i} \|\mathbf{D} - \tilde{\mathbf{C}}\mathbf{B}_i\| \quad (2.28)$$

dengan

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}^{(i_1)} \\ \mathbf{d}^{(i_2)} \\ \vdots \\ \mathbf{d}^{(i_{L_i})} \end{bmatrix} = \begin{bmatrix} d_1^{(i_1)} & d_2^{(i_1)} & \dots & d_p^{(i_1)} \\ d_1^{(i_2)} & d_2^{(i_2)} & \dots & d_p^{(i_2)} \\ \vdots & \vdots & & \vdots \\ d_1^{(i_{L_i})} & d_2^{(i_{L_i})} & \dots & d_p^{(i_{L_i})} \end{bmatrix} \quad (2.29)$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} \tilde{\mathbf{c}}^{(i_1)} \\ \tilde{\mathbf{c}}^{(i_2)} \\ \vdots \\ \tilde{\mathbf{c}}^{(i_{L_i})} \end{bmatrix} = \begin{bmatrix} 1 & c_1^{(i_1)} & c_2^{(i_1)} & \dots & c_n^{(i_1)} \\ 1 & c_1^{(i_2)} & c_2^{(i_2)} & \dots & c_n^{(i_2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & c_1^{(i_{L_i})} & c_2^{(i_{L_i})} & \dots & c_n^{(i_{L_i})} \end{bmatrix} \quad (2.30)$$

dan matriks  $\mathbf{B}_i$  didefinisikan pada (2.14). Perhatikan bahwa masalah optimisasi pada (2.28) merupakan masalah optimisasi pada regresi linier multivariat. Akibatnya, masalah optimisasi tersebut dapat diselesaikan dengan cara mensubstitusi  $\mathbf{B}_i = (\tilde{\mathbf{C}}^T \tilde{\mathbf{C}})^{-1} \tilde{\mathbf{C}}^T \mathbf{D}$  asalkan matriks  $\tilde{\mathbf{C}}^T \tilde{\mathbf{C}}$  bukan matriks singular (determinannya bernilai nol). Kondisi ini terpenuhi jika  $L_i > (n + 1) + 1 = n + 2$  dan setiap kolom dari matriks  $\tilde{\mathbf{C}}$  bebas linier (Rencher, 2002). Tetapi, hasil dari *self-constructing clustering* pada tahap sebelumnya tidak menjamin pertidaksamaan  $L_i > n + 2$  akan terpenuhi untuk setiap  $i \in \{1, 2, \dots, r\}$ . Dengan demikian, metode ini tidak dapat digunakan dalam fase identifikasi struktur.

Lee dan Ouyang (2003) menyelesaikan masalah optimisasi pada (2.28) dengan melakukan SVD terhadap matriks  $\tilde{\mathbf{C}}$  pada (2.30). Berdasarkan teorema yang telah dibuktikan oleh Golub dan Loan (2013), matriks  $\tilde{\mathbf{C}}$  dapat didekomposisi menjadi

$$\tilde{\mathbf{C}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (2.31)$$

dengan  $\mathbf{U}$  dan  $\mathbf{V}$  adalah matriks ortonormal yang berukuran  $L_i \times L_i$  dan  $(n+1) \times (n+1)$  berturut-turut, dan  $\mathbf{\Sigma}$  adalah matriks yang berukuran  $L_i \times (n+1)$ . Misalkan setiap entri dari  $\mathbf{\Sigma}$  dinyatakan oleh  $\sigma_{u,w}$ . Maka

$$\sigma_{u,w} = \begin{cases} \sqrt{\lambda_w}, & \text{jika } u = w \leq q = \text{rank}(\tilde{\mathbf{C}}) \\ 0, & u, w \text{ lainnya} \end{cases}$$

dengan  $\lambda_w$  adalah nilai eigen positif dari matriks  $\tilde{\mathbf{C}}^T \tilde{\mathbf{C}}$  atau  $\tilde{\mathbf{C}} \tilde{\mathbf{C}}^T$  dan  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_q$ . Vektor eigen dari  $\tilde{\mathbf{C}}^T \tilde{\mathbf{C}}$  yang bersesuaian membentuk kolom-kolom dari matriks  $\mathbf{U}$ , dan vektor eigen dari  $\tilde{\mathbf{C}} \tilde{\mathbf{C}}^T$  yang bersesuaian membentuk kolom-kolom dari matriks  $\mathbf{V}$ .

Dengan mensubstitusi Persamaan (2.31) ke dalam masalah optimisasi pada (2.28) akan diperoleh

$$\min_{\mathbf{B}_i} \|\mathbf{D} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \mathbf{B}_i\|. \quad (2.32)$$

Karena  $\mathbf{U}$  adalah matriks ortonormal, maka  $\mathbf{U}^{-1} = \mathbf{U}^T$  dan  $\|\mathbf{U}^T\| = 1$ . Akibatnya, masalah optimisasi pada (2.32) menjadi

$$\min_{\mathbf{B}_i} \|\mathbf{U}^T \mathbf{D} - \mathbf{\Sigma} \mathbf{V}^T \mathbf{B}_i\|. \quad (2.33)$$

Selanjutnya, tinjau hubungan antara  $q = \text{rank}(\tilde{\mathbf{C}})$  dengan banyaknya baris pada matriks  $\tilde{\mathbf{C}}$ , yaitu  $L_i$ . Maka terdapat dua kasus yang mungkin terjadi, yaitu  $q < L_i$  atau  $q = L_i$ .

- Untuk  $q < L_i$ , matriks  $\mathbf{\Sigma}$  dan  $\mathbf{U}$  dapat dipartisi menjadi

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}' \\ \mathbf{0} \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \mathbf{U}' & \mathbf{U}'' \end{bmatrix}$$

dengan  $\mathbf{\Sigma}'$  berukuran  $q \times (n+1)$ ,  $\mathbf{0}$  berukuran  $(L_i - q) \times (n+1)$ ,  $\mathbf{U}'$  berukuran

$L_i \times q$  dan  $\mathbf{U}''$  berukuran  $L_i \times (L_i - q)$ . Maka masalah optimisasi pada (2.33) menjadi

$$\min_{\mathbf{B}_i} \left\| \begin{bmatrix} \mathbf{U}'^T \\ \mathbf{U}''^T \end{bmatrix} \mathbf{D} - \begin{bmatrix} \mathbf{\Sigma}' \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T \mathbf{B}_i \right\| = \min_{\mathbf{B}_i} \left\| \begin{bmatrix} \mathbf{U}'^T \mathbf{D} - \mathbf{\Sigma}' \mathbf{V}^T \mathbf{B}_i \\ \mathbf{U}''^T \mathbf{D} \end{bmatrix} \right\|.$$

Dengan demikian, solusi dari masalah optimisasi pada (2.28) adalah  $\hat{\mathbf{B}}_i$  sedemikian sehingga  $\mathbf{U}'^T \mathbf{D} - \mathbf{\Sigma}' \mathbf{V}^T \hat{\mathbf{B}}_i = \mathbf{0}$ , yaitu

$$\hat{\mathbf{B}}_i = \mathbf{V} \left( \mathbf{\Sigma}'^T \mathbf{\Sigma}' \right)^{-1} \mathbf{\Sigma}'^T \mathbf{U}'^T \mathbf{D} = \mathbf{V} \mathbf{Z} \mathbf{U}'^T \mathbf{D} \quad (2.34)$$

dengan matriks  $\mathbf{Z} = \left( \mathbf{\Sigma}'^T \mathbf{\Sigma}' \right)^{-1} \mathbf{\Sigma}'^T$  berukuran  $(n+1) \times q$  dan entri  $z_{w,u}$  untuk  $\mathbf{Z}$  adalah

$$z_{w,u} = \begin{cases} \frac{1}{\sqrt{\lambda_w}}, & \text{jika } w = u \\ 0, & w, u \text{ lainnya} \end{cases} \quad (2.35)$$

untuk  $w = 1, 2, \dots, n+1$ , dan  $u = 1, 2, \dots, q$

- Untuk  $q = L_i$ , solusi dari masalah optimisasi pada (2.28) adalah  $\hat{\mathbf{B}}_i$  sedemikian sehingga  $\mathbf{U}^T \mathbf{D} - \mathbf{\Sigma} \mathbf{V}^T \hat{\mathbf{B}}_i = \mathbf{0}$ , yaitu

$$\hat{\mathbf{B}}_i = \mathbf{V} \left( \mathbf{\Sigma}^T \mathbf{\Sigma} \right)^{-1} \mathbf{\Sigma}^T \mathbf{U}^T \mathbf{D} = \mathbf{V} \mathbf{Z} \mathbf{U}^T \mathbf{D} \quad (2.36)$$

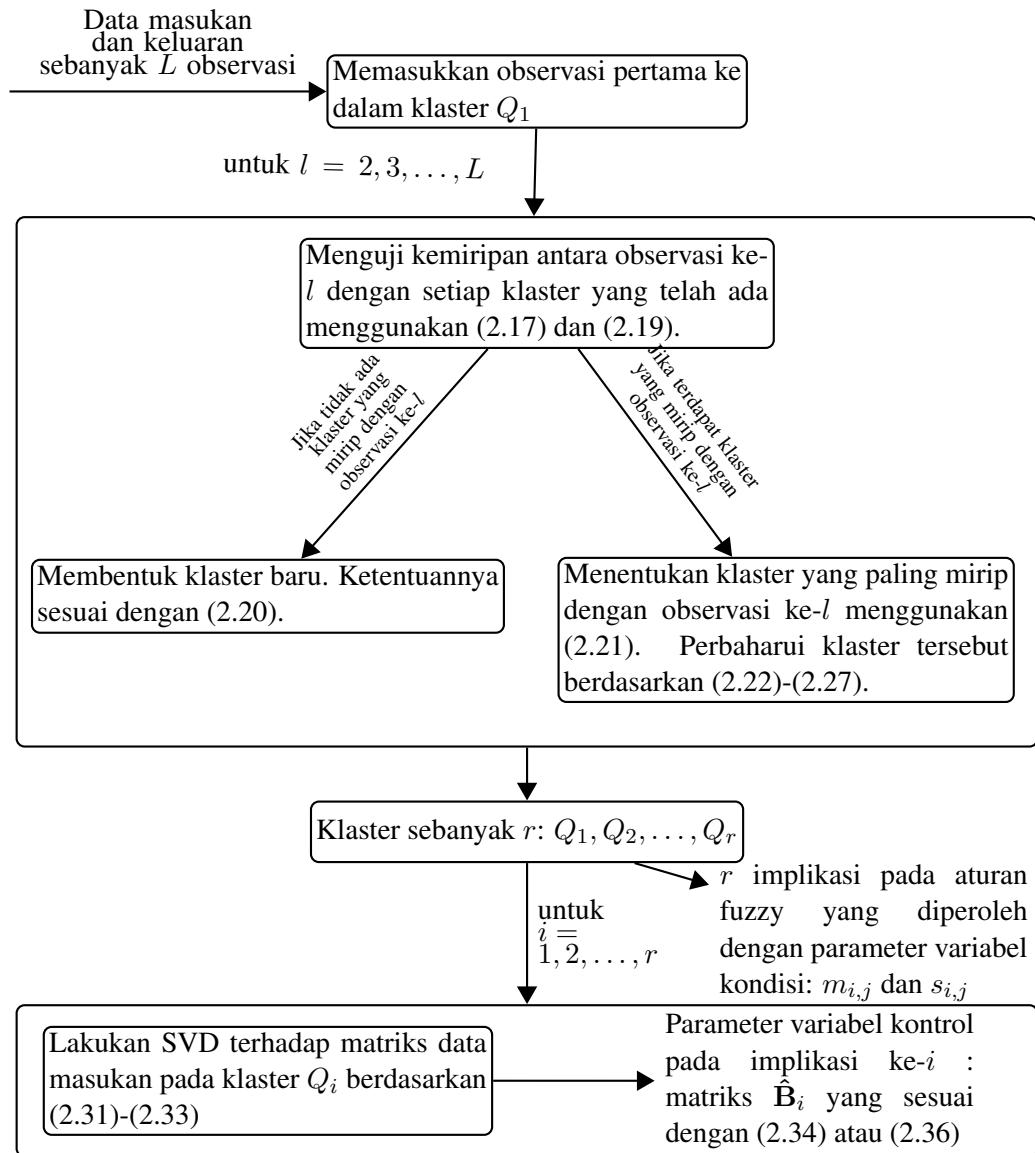
dengan matriks  $\mathbf{Z} = \left( \mathbf{\Sigma}^T \mathbf{\Sigma} \right)^{-1} \mathbf{\Sigma}^T$  berukuran  $(n+1) \times q$  dan entri  $z_{w,u}$  untuk  $\mathbf{Z}$  didefinisikan pada (2.35).

Setelah melalui tahap *self-constructing clustering* dan SVD, maka fase identifikasi struktur menghasilkan struktur awal dari aturan fuzzy dengan implikasi sebanyak  $r$ . Setiap implikasi  $\mathfrak{R}_i, i = 1, 2, \dots, r$  dari aturan fuzzy ini, memiliki bentuk

$$\mathfrak{R}_i : \text{Jika } x_1 \text{ adalah } A_{i,1}, \dots, x_n \text{ adalah } A_{i,n}, \text{ maka } y = \mathbf{B}_i \tilde{\mathbf{x}}$$

dengan  $\mu_{A_{i,j}}(x_j) = g(x_j; m_{i,j}, s_{i,j})$  untuk  $j = 1, 2, \dots, n$ . Nilai awal untuk semua parameter  $m_{i,j}$  dan  $s_{i,j}$  diperoleh dari tahap *self-constructing clustering*. Nilai awal untuk entri dari semua matriks  $\mathbf{B}_i$  didapatkan melalui metode SVD.

Fase identifikasi struktur telah diringkas pada diagram alir dalam Gambar II.5. Kompleksitas waktu untuk fase identifikasi struktur tidak terlalu besar. Hal ini dikarenakan tidak ada observasi yang digunakan secara berulang pada setiap tahap.



Gambar II.5: Diagram alir identifikasi struktur

Akibatnya, kompleksitas waktu untuk fase ini hanya bergantung kepada banyaknya observasi yang diberikan. Dengan demikian, waktu yang dibutuhkan komputer untuk mengeksekusi fase identifikasi struktur bisa sangat singkat.

## II.6.2 Identifikasi Parameter

Pada fase identifikasi parameter, nilai dari setiap parameter diperbaiki sedemikian sehingga dapat meminimalkan MSE antara tindakan kontrol aktual dan tindakan kontrol yang diinginkan. Lee dan Ouyang (2003) menggunakan metode *gradient descent* untuk memperbaiki nilai dari semua parameter  $m_{i,j}$  dan  $s_{i,j}$ ,  $i = 1, 2, \dots, r$  dan  $j = 1, 2, \dots, n$ . Sementara itu, Yeh dkk. (2012) menggunakan metode *particle*

*swarm optimization* untuk memperbaiki nilai dari semua parameter  $m_{i,j}$  dan  $s_{i,j}$ . Untuk memperbaiki nilai dari entri matriks  $B_i$ , mereka menggunakan metode SVD. Tetapi, mereka menggunakan prinsip yang sama, yaitu: ketika memperbaiki nilai dari parameter pada variabel kondisi (parameter  $m_{i,j}$  dan  $s_{i,j}$ ), parameter pada variabel kontrol (matriks  $B_i$ ) diperlakukan sebagai konstanta, begitu juga sebaliknya.

Dalam tugas akhir ini, penulis akan menggunakan metode *gradient descent* untuk memperbaiki nilai dari semua parameter, baik parameter pada variabel kondisi, maupun parameter pada variabel kontrol. Hal ini dikarenakan metode *gradient descent* lebih mudah untuk diterapkan. Prinsip perbaikan parameter mengikuti prinsip yang digunakan oleh Lee dan Ouyang (2003) dan Yeh dkk. (2012). Urutan parameter yang akan diperbaiki mengikuti prosedur *backpropagation* pada JST seperti yang telah dijelaskan oleh Kriesel (2007). Jadi, untuk setiap iterasi, parameter yang akan diperbaiki terlebih dahulu adalah parameter pada variabel kontrol, yaitu setiap entri dari semua matriks  $B_i$ . Selanjutnya, pada iterasi tersebut dilakukan perbaikan untuk parameter pada variabel kondisi.

Identifikasi parameter merupakan fase terakhir dalam konstruksi JSF. Akibatnya, target utama dari konstruksi JSF menjadi target utama dari identifikasi parameter. Jadi, target utama dari identifikasi parameter adalah meminimalkan MSE antara tindakan kontrol aktual dan tindakan kontrol yang diinginkan.

Misalkan diberikan data yang berukuran  $L \times (n + p)$ . Setiap observasi ke- $l$ ,  $l = 1, 2, \dots, L$ , memiliki pola  $[c^{(l)}, d^{(l)}]$  dengan  $c^{(l)} \in \mathbb{R}^n$  menyatakan vektor data masukan pada observasi ke- $l$  dan  $d^{(l)} \in \mathbb{R}^p$  menyatakan vektor data keluaran pada observasi ke- $l$ . Selanjutnya, fase identifikasi struktur menghasilkan sebanyak  $r$  implikasi untuk aturan fuzzy yang merepresentasikan data tersebut. Dengan demikian, secara matematis target utama dari identifikasi parameter adalah meminimalkan fungsi  $\mathcal{L}$  berikut ini

$$\mathcal{L}(\mathbf{M}, \mathbf{S}, \mathbf{B}) = \frac{1}{Lp} \sum_{l=1}^L \sum_{k=1}^p \left( d_k^{(l)} - t_k^{(l)} \right)^2. \quad (2.37)$$

Pada Persamaan (2.37),  $\mathbf{M}$  dan  $\mathbf{S}$  menyatakan matriks yang berukuran  $r \times n$  dan entri dari kedua matriks tersebut, berturut-turut, adalah  $m_{i,j}$  dan  $s_{i,j}$  dengan  $i = 1, 2, \dots, r$  dan  $j = 1, 2, \dots, n$ . Entri  $m_{i,j}$  dan  $s_{i,j}$  tersebut menyatakan parameter untuk implikasi ke- $i$  pada variabel kondisi ke- $j$ . Sementara itu,  $\mathbf{B}$



adalah kumpulan dari  $r$  matriks yang terdiri dari matriks  $\mathbf{B}_i, i = 1, 2, \dots, r$  dan masing-masing berukuran  $p \times (n + 1)$ . Misalkan vektor  $\mathbf{b}_{i,k}$  menyatakan baris ke- $k$  dari matriks  $\mathbf{B}_i$ . Maka vektor  $\mathbf{b}_{i,k}$  merupakan vektor di  $\mathbf{R}^{n+1}$  yang menyatakan parameter untuk implikasi ke- $i$  pada variabel kontrol ke- $k$ . Terakhir,  $d_k^{(l)}$  menyatakan data keluaran ke- $k$  untuk observasi ke- $l$  dan  $t_k^{(l)}$  menyatakan tindakan kontrol aktual ke- $k$  untuk observasi ke- $l$  yang dihitung melalui proses *feedforward* pada JSF. Rincian dari penentuan nilai  $t_k^{(l)}$  adalah sebagai berikut.

$$t_k^{(l)} = \frac{\sum_{i=1}^r \alpha_i^{(l)} \mathbf{b}_{i,k} \cdot (\dot{\mathbf{c}}^{(l)})}{\sum_{i=1}^r \alpha_i^{(l)}} = \frac{\sum_{i=1}^r \left[ \alpha_i^{(l)} \left( b_{i,k,0} + \sum_{j=1}^n b_{i,k,j} c_j^{(l)} \right) \right]}{\sum_{i=1}^r \alpha_i^{(l)}} \quad (2.38)$$

dengan

$$\alpha_i^{(l)} = \exp \left[ -\frac{1}{2} \sum_{j=1}^n \left( \frac{c_j^{(l)} - m_{i,j}}{s_{i,j}} \right)^2 \right], \quad i = 1, 2, \dots, r \quad (2.39)$$

Parameter  $m_{i,j}$  dan  $s_{i,j}$  yang akan diperbaiki ada sebanyak  $2rn$ . Banyaknya parameter yang merupakan entri dari setiap matriks  $\mathbf{B}_i$  adalah sebanyak  $rp(n + 1)$ . Misalkan  $K$  menyatakan banyaknya parameter yang akan diperbaiki. Maka  $K = 2rn + rp(n + 1) = r(n(p + 2) + p)$ .

Langkah pertama fase identifikasi parameter menggunakan metode *gradient descent* adalah menurunkan fungsi  $\mathcal{L}$  terhadap masing-masing dari  $K$  parameter secara parsial. Selanjutnya, setiap parameter diperbaiki dengan cara mengurangi hasil turunan parsial yang bersesuaian dan telah dikalikan dengan suatu bobot terhadap nilai sebelumnya dari parameter tersebut. Berdasarkan aturan rantai dalam turunan, serta (2.37), (2.38), dan (2.39), turunan parsial dari fungsi  $\mathcal{L}$  terhadap setiap parameter adalah

$$\frac{\partial \mathcal{L}}{\partial b_{i,k,0}} = \sum_{l=1}^L \frac{\partial \mathcal{L}}{\partial t_k^{(l)}} \frac{\partial t_k^{(l)}}{\partial b_{i,k,0}} = \frac{2}{L} \sum_{l=1}^L \frac{\alpha_i^{(l)} (t_k^{(l)} - d_k^{(l)})}{\sum_{v=1}^r \alpha_v^{(l)}} \quad (2.40)$$

$$\frac{\partial \mathcal{L}}{\partial b_{i,k,j}} = \sum_{l=1}^L \frac{\partial \mathcal{L}}{\partial t_k^{(l)}} \frac{\partial t_k^{(l)}}{\partial b_{i,k,j}} = \frac{2}{L} \sum_{l=1}^L \frac{\alpha_i^{(l)} c_j^{(l)} (t_k^{(l)} - d_k^{(l)})}{\sum_{v=1}^r \alpha_v^{(l)}} \quad (2.41)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial m_{i,j}} &= \sum_{l=1}^L \sum_{k=1}^p \frac{\partial \mathcal{L}}{\partial t_k^{(l)}} \frac{\partial t_k^{(l)}}{\partial \alpha_i^{(l)}} \frac{\partial \alpha_i^{(l)}}{\partial m_{i,j}} \\ &= \frac{2}{Lp} \sum_{l=1}^L \left\{ \frac{\alpha_i^{(l)} (c_j^{(l)} - m_{i,j})}{s_{i,j}^2 \sum_{v=1}^r \alpha_v^{(l)}} \sum_{k=1}^p \left[ (t_k^{(l)} - d_k^{(l)}) (\mathbf{b}_{i,k} \cdot \tilde{\mathbf{c}}^{(l)} - t_k^{(l)}) \right] \right\} \end{aligned} \quad (2.42)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial s_{i,j}} &= \sum_{l=1}^L \sum_{k=1}^p \frac{\partial \mathcal{L}}{\partial t_k^{(l)}} \frac{\partial t_k^{(l)}}{\partial \alpha_i^{(l)}} \frac{\partial \alpha_i^{(l)}}{\partial s_{i,j}} \\ &= \frac{2}{Lp} \sum_{l=1}^L \left\{ \frac{\alpha_i^{(l)} (c_j^{(l)} - m_{i,j})}{s_{i,j}^3 \sum_{v=1}^r \alpha_v^{(l)}} \sum_{k=1}^p \left[ (t_k^{(l)} - d_k^{(l)}) (\mathbf{b}_{i,k} \cdot \tilde{\mathbf{c}}^{(l)} - t_k^{(l)}) \right] \right\} \end{aligned} \quad (2.43)$$

Perbaikan nilai untuk setiap parameter adalah

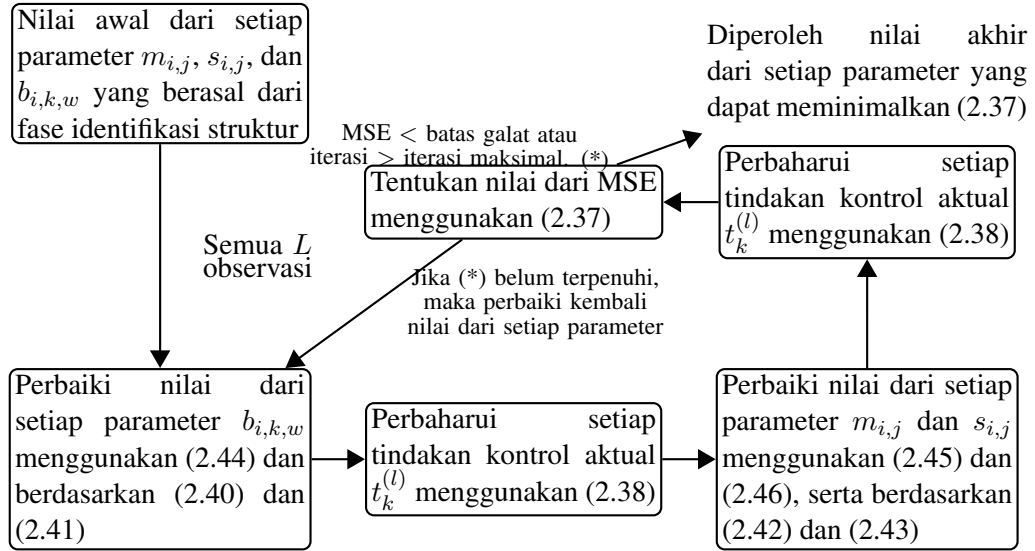
$$b_{i,k,w}^{(\text{baru})} = b_{i,k,w}^{(\text{lama})} - \eta \frac{\partial \mathcal{L}}{\partial b_{i,k,w}} \quad (2.44)$$

$$m_{i,j}^{(\text{baru})} = m_{i,j}^{(\text{lama})} - \eta \frac{\partial \mathcal{L}}{\partial m_{i,j}} \quad (2.45)$$

$$s_{i,j}^{(\text{baru})} = s_{i,j}^{(\text{lama})} - \eta \frac{\partial \mathcal{L}}{\partial s_{i,j}} \quad (2.46)$$

untuk  $i = 1, 2, \dots, r$ ,  $k = 1, 2, \dots, p$ ,  $w = 0, 1, \dots, n$ , dan  $j = 1, 2, \dots, n$ . Notasi  $\eta$  pada (2.44)-(2.46) merupakan suatu konstanta yang menyatakan *learning rate*. Nilai *learning rate* ini dipilih pada kisaran nilai  $0,01 \leq \eta \leq 0,9$  (Kriesel, 2007).

Langkah-langkah dalam fase identifikasi parameter telah diringkas pada diagram alir dalam Gambar II.6. Rinciannya adalah sebagai berikut. Setelah dilakukan perbaikan parameter  $b_{i,k,w}$  menggunakan (2.44), nilai-nilai yang baru dari parameter  $b_{i,k,w}$  digunakan untuk memperbaharui setiap nilai dari  $t_k^{(l)}$  berdasarkan (2.38). Selanjutnya, parameter  $b_{i,k,w}$  dan nilai  $t_k^{(l)}$  yang baru ini digunakan untuk



Gambar II.6: Diagram alir identifikasi parameter

menghitung  $\frac{\partial \mathcal{L}}{\partial m_{i,j}}$  dan  $\frac{\partial \mathcal{L}}{\partial s_{i,j}}$  pada (2.42) dan (2.43). Kemudian, dilakukan pembaharuan parameter  $m_{i,j}$  dan  $s_{i,j}$  menggunakan (2.45) dan (2.46) berturut-turut. Setelah itu, setiap nilai dari  $t_k^{(l)}$  diperbaharui kembali menggunakan semua parameter yang telah diperbaharui. Terakhir, periksa MSE antara  $t_k^{(l)}$  dan  $d_k^{(l)}$  yang nilainya dihitung menggunakan (2.37). Jika MSE masih lebih dari batas galat yang diinginkan, maka fase identifikasi parameter belum selesai, sehingga harus memperbaiki kembali nilai-nilai dari setiap parameter. Jika MSE kurang dari atau sama dengan batas galat yang diinginkan, maka proses identifikasi parameter berhenti, sehingga diperoleh JSF yang optimal dan sesuai dengan data ini. Tetapi, berapapun nilai dari MSE, proses identifikasi parameter tetap berhenti jika banyaknya iterasi telah melebihi banyaknya iterasi maksimal.

## Bab III Skema Pembelajaran Mesin

Pada bagian ini, pembelajaran mesin atau ML (*machine learning*) dipandang sebagai algoritma yang digunakan untuk membuat prediksi yang akurat berdasarkan informasi yang diberikan. Algoritma ML terdiri dari model dan skema. Model ML yang digunakan dalam tugas akhir ini adalah model jaringan saraf fuzzy (JSF) yang telah dijelaskan pada Bab II, terutama pada Subbab II.6. Skema ML merupakan alur dan strategi ML yang digunakan untuk membangun model ML.

### III.1 Definisi dan Istilah dalam Pembelajaran Mesin

Sebelum membahas skema dalam pembelajaran mesin, akan dipaparkan terlebih dahulu definisi dan istilah yang sering digunakan dalam pembelajaran mesin. Hal ini bertujuan supaya pembahasan selanjutnya menjadi lebih efisien. Berikut ini definisi dan istilah terkait ML yang telah diringkas oleh Mohri dkk. (2018) dengan beberapa perubahan seperlunya.

- **Observasi**, yaitu suatu objek yang mempunyai beberapa atribut dengan nilai tertentu. Observasi-observasi yang mempunyai atribut yang sama akan membentuk suatu data. Sebagai contoh, misalkan terdapat objek titik di dalam ruang dimensi tiga. Misalkan objek tersebut memiliki atribut-atribut yang berupa koordinat pada sumbu  $x$ ,  $y$ , dan  $z$ , serta nomor oktan. Maka satu objek titik ini disebut observasi. Jika terdapat sekumpulan objek titik yang memiliki atribut-atribut yang disebutkan sebelumnya, maka sekumpulan objek titik ini membentuk data titik di dalam ruang dimensi tiga.
- **Fitur**, yaitu atribut-atribut yang saling bebas atau tidak saling mempengaruhi dari suatu observasi pada data. Fitur sering direpresentasikan sebagai vektor. Pada bab sebelumnya, fitur sering disebut sebagai data masukan. Pada contoh sebelumnya, fitur dari data tersebut adalah koordinat pada sumbu  $x$ ,  $y$ , dan  $z$ .
- **Label**, yaitu atribut pada suatu observasi pada data yang nilainya bergantung kepada fitur. Label dapat berupa bilangan real atau kategori. Pada masalah klasifikasi, label berupa kategori. Di dalam bab sebelumnya, label sering disebut

sebagai data keluaran. Pada contoh sebelumnya, label dari data tersebut adalah nomor oktan karena nomor oktan sangat bergantung kepada nilai dari fitur  $x$ ,  $y$ , dan  $z$ .

- **Hiperparameter**, yaitu parameter bebas yang tidak dihasilkan dari model ML, tetapi digunakan sebagai *input* untuk membangun model ML. Berdasarkan penjelasan pada Subbab II.6, hiperparameter pada model JSF adalah nilai dari  $\rho$ ,  $\tau$ ,  $s_0$ , dan  $\eta$ . Tiga nilai pertama digunakan pada fase identifikasi struktur. Nilai  $\eta$  digunakan pada fase identifikasi parameter.
- **Data latih**, yaitu kumpulan observasi pada data yang digunakan untuk menghasilkan suatu model ML. Pada contoh sebelumnya (klasifikasi nomor oktan), setiap observasi pada data latih harus mempunyai nomor oktan. Algoritma ML akan mempelajari setiap observasi pada data latih ini sedemikian sehingga dapat menghasilkan model ML yang keluarannya dapat mendekati atau bahkan menyamai nomor oktan untuk setiap observasi yang bersesuaian. Data latih juga dibagi menjadi dua bagian, yaitu:
  - **Learning set**, yaitu kumpulan observasi pada data latih yang digunakan untuk menghasilkan suatu model ML berdasarkan hiperparameter dengan nilai-nilai tertentu.
  - **Data validasi**, yaitu kumpulan observasi pada data latih yang digunakan untuk mengevaluasi performa model ML yang dihasilkan berdasarkan hiperparameter dengan nilai-nilai tertentu.
- **Data uji**, yaitu kumpulan observasi pada data yang bukan bagian dari data latih dan digunakan untuk menguji performa dari model ML yang dihasilkan oleh data latih. Pada contoh klasifikasi nilai oktan, model ML akan memprediksi nomor oktan untuk setiap sampel pada data latih. Kemudian hasil prediksi ini dibandingkan dengan nomor oktan yang sebenarnya. Dengan demikian, label pada data uji hanya digunakan untuk mengevaluasi model ML yang dihasilkan.
- **Loss function**, yaitu suatu fungsi yang mengukur perbedaan antara label yang dihasilkan oleh model ML dan label yang terdapat pada data. Pada model JSF dalam tugas akhir ini, *loss function*-nya berupa MSE yang didefinisikan pada (2.37).
- **Himpunan hipotesis**, yaitu himpunan fungsi yang memetakan fitur-fitur pada data terhadap label. Pada model JSF, fungsi termasuk ke dalam himpunan hipotesis adalah serangkaian proses *feedforward* pada JSF atau serangkaian proses pada Sistem Kontrol Logika Fuzzy (SKLF).

- **Parameter**, yaitu nilai-nilai yang terlibat secara langsung pada suatu fungsi yang berada di dalam himpunan hipotesis. Parameter dengan nilai-nilai tertentu dapat meminimalkan *loss function*. Parameter juga dapat merepresentasikan suatu model ML yang dihasilkan. Berdasarkan pembahasan pada Subbab II.6, parameter-parameter dalam model JSF adalah matriks  $M$ , matriks  $S$ , dan kumpulan matriks  $B_i, i = 1, 2, \dots, r$ .

Sebelum mengeksekusi model JSF, data yang akan digunakan harus melalui tahap pra pemrosesan terlebih dahulu. Hal ini dikarenakan setiap data belum tentu mempunyai entri nilai yang cocok untuk model JSF. Sebagai contoh, jika suatu entri pada data mempunyai nilai yang sangat besar, maka dapat menghasilkan nilai yang takhingga ketika memasukkannya ke dalam fungsi Gauss. Hal-hal yang perlu dilakukan pada tahap pra pemrosesan data akan dijelaskan pada Subbab III.2. Selain itu, model JSF memerlukan nilai tertentu dari hiperparameter. Nilai yang berbeda pada hiperparameter belum tentu menghasilkan model JSF dengan parameter yang sama. Oleh karena itu, diperlukan skema untuk memilih nilai tertentu dari hiperparameter ini. Skema tersebut akan dijelaskan pada Subbab III.4.

Pada masalah klasifikasi multikelas, label berupa kategori. Sedangkan, model JSF hanya dapat memproses bilangan real. Oleh karena itu, diperlukan metode untuk mengonversi label ke dalam bilangan real. Setiap metode yang digunakan akan menghasilkan struktur JSF yang berbeda-beda. Akibatnya, setiap metode harus menggunakan skema klasifikasi yang berbeda-beda. Skema klasifikasi ini akan dipaparkan pada Subbab III.3.

## III.2 Pra Pemrosesan Data

Pra pemrosesan data memiliki dampak yang signifikan dalam generalisasi performa dari algoritma SML termasuk algoritma untuk menyelesaikan masalah klasifikasi multikelas (Kotsiantis dkk., 2006; Alexandropoulos dkk., 2019). Kotsiantis dkk. (2006) dan Alexandropoulos dkk. (2019) telah membahas deteksi pencilan, pengukuran gangguan pada data, penanganan entri data dengan nilai yang kosong, pemilihan fitur, dan ekstraksi fitur sebagai proses-proses yang dilakukan pada tahap pra pemrosesan data. Potdar dkk. (2017) secara khusus telah membahas perbandingan antara metode-metode pengkodean fitur yang bersifat kategori. Pemisahan data latih dan data uji dengan metode semi acak telah dibahas oleh Liu dan Cocca (2017). Sementara itu, khusus untuk model JSF, Yeh dkk. (2012)

telah menyarankan untuk mengurutkan data latih dengan ketentuan tertentu sebelum memasuki fase identifikasi struktur.

Pada tugas akhir ini, teknik pra pemrosesan data yang akan dibahas hanya meliputi pengkodean fitur yang bersifat kategori, pemisahan data latih dan data uji, normalisasi fitur, ekstraksi fitur, dan pengurutan observasi pada data latih. Hal ini dikarenakan hanya teknik-teknik pra pemrosesan data tersebut yang akan diterapkan terhadap data yang diuji. Pada praktiknya, sebagian besar teknik pra pemrosesan ini dibantu oleh fungsi-fungsi yang disediakan di dalam *package scikit-learn* python. Hanya algoritma pengurutan sampel pada data latih yang dirancang sendiri oleh penulis.

### III.2.1 Pengkodean Fitur yang Bersifat Kategori

Model JSF hanya dapat mengolah observasi-observasi pada data yang nilai fiturnya berupa bilangan real. Oleh karena itu, jika ada fitur yang nilai-nilainya berupa kategori, maka harus dikonversi terlebih dahulu ke dalam bilangan real. Konversi kategori menjadi bilangan real ini disebut dengan pengkodean. Terdapat berbagai metode pengkodean, di antaranya adalah *one hot encoding*, *ordinal encoding*, *sum encoding*, *helmert encoding*, dll (Potdar dkk., 2017). Pada tugas akhir ini, akan digunakan metode *one hot encoding*, karena metode ini sangat mudah untuk diterapkan terhadap data yang akan diuji.

Metode *one hot encoding* merupakan metode yang paling sering digunakan dalam pengkodean (Potdar dkk., 2017). Secara intuitif, metode pengkodean ini sangat mudah untuk dipahami. Metode *one hot encoding* mengonversi suatu fitur kategori menjadi vektor berdimensi tertentu dengan entri-entrinya berupa bilangan biner 0 atau 1. Misalkan pada suatu fitur terdapat  $d$  kategori yang berbeda. Dengan metode *one hot encoding*, fitur ini dikonversi menjadi vektor bilangan biner dengan dimensi  $d$ . Misalkan suatu observasi pada fitur ini memiliki nilai 'kategori ke- $k$ '. Maka setelah dilakukan pengkodean, fitur untuk sampel tersebut akan menjadi vektor berdimensi  $d$  dengan nilai 1 untuk entri ke- $k$  dan nilai 0 untuk entri vektor lainnya.

Pada tugas akhir ini, terdapat catatan khusus untuk fitur yang hanya memiliki dua kategori berbeda, yakni fitur ini tidak akan dikonversi menjadi vektor berdimensi 2. Untuk fitur yang seperti ini, salah satu kategorinya akan dikonversi menjadi 0 dan kategori lain dikonversi menjadi 1. Hal ini dilakukan karena akan ada dua

vektor yang saling bergantung secara linier jika tetap dikonversi menjadi vektor dua dimensi. Fitur yang saling bergantung ini akan menggagalkan asumsi yang telah disebutkan pada Bab I.

Untuk menerapkan metode *one hot encoding* pada seluruh observasi, penulis menggunakan fasilitas '*LabelEncoder()*' dan '*OneHotEncoder()*'. Dua fasilitas ini telah disediakan oleh *package scikit-learn* di dalam python. Fasilitas '*LabelEncoder()*' digunakan untuk mengonversi kategori yang mungkin ber-type *string* menjadi numerik. Fasilitas '*OneHotEncoder()*' digunakan untuk menerapkan metode *one hot encoding*.

### III.2.2 Pemisahan Data Latih dan Data Uji

Pada tugas akhir ini, setiap observasi-observasi pada data akan dipisahkan terlebih dahulu sebelum dijadikan sebagai objek pengujian model JSF. Observasi-observasi tersebut dipisahkan ke dalam dua kelompok, yaitu kelompok data latih dan kelompok data uji. Pada proses pengujian, perbandingan banyaknya observasi antara data latih dan data uji biasanya adalah 7:3 (Liu dan Cocea, 2017). Pada tugas akhir ini, akan digunakan perbandingan 8:2. Observasi-observasi dipisahkan sedemikian sehingga banyaknya label yang berbeda pada data latih sama dengan banyaknya label yang berbeda pada data uji. Dengan kata lain, tidak ada satu pun kategori label yang tidak termuat di dalam data latih dan data uji.

Proses pemisahan data ini akan menggunakan metode deterministik, yakni observasi-observasi yang ada di dalam data latih pada suatu proses pemisahan data sama dengan observasi-observasi data latih pada proses pemisahan data yang dilakukan di waktu yang berbeda. Dengan kata lain, hasil dari pemisahan data tidak akan berbeda meskipun dilakukan secara berulang. Untuk memenuhi ketentuan-ketentuan yang disebutkan di atas, penulis memanfaatkan fasilitas '*train\_test\_split*' yang tersedia di dalam *package scikit-learn* python dengan menetapkan nilai *RandomState* sebagai suatu bilangan bulat positif yang tetap.

### III.2.3 Penskalaan Nilai untuk Setiap Fitur

Fitur-fitur yang berupa bilangan real belum tentu memiliki satuan yang sama. Sebagai contoh, pada data gas ideal terdapat fitur suhu dan volume. Satuan dari dua fitur ini jelas berbeda. Meskipun terdapat data yang satuan dari setiap fiturnya



sama, tetapi tidak akan menjamin bahwa nilai-nilai pada setiap fitur akan cocok dengan fungsi pada himpunan hipotesis. Contohnya, data klasifikasi oktan yang pernah dibahas di awal bab ini. Fitur-fitur pada data ini memiliki satuan yang sama, yaitu satuan pada besaran posisi, seperti sentimeter. Tetapi, jika nilai pada fitur-fitur ini terlalu besar atau terlalu kecil, maka dapat menghasilkan nilai takhingga ketika dimasukkan ke dalam fungsi yang ada di dalam himpunan hipotesis. Himpunan hipotesis untuk model JSF memuat suatu fungsi yang di dalamnya terdapat fungsi Gauss. Ketika nilai yang dimasukkan ke dalam fungsi Gauss sangat besar, seperti 3.000, maka kemungkinan besar hasil evaluasi fungsi Gauss terhadap nilai tersebut akan menuju takhingga. Dalam komputasi, nilai yang menuju takhingga tidak akan dianggap sebagai suatu bilangan atau NaN (*Not a Number*). Jika proses pembangunan model JSF dilanjutkan dan masih memuat nilai NaN, maka pembangunan model JSF akan gagal.

Untuk mengatasi dua hal yang disebutkan di atas, maka dibutuhkan proses penskalaan ke bawah (*down scaling*) untuk setiap fitur pada data. Proses penskalaan ini disebut dengan normalisasi fitur. Normalisasi fitur akan membuat nilai-nilai pada fitur bertransformasi menjadi nilai yang dapat diterima oleh berbagai jenis fungsi terutama jenis fungsi eksponensial. Terdapat dua jenis normalisasi fitur yang sering digunakan dalam pra pemrosesan data (Kotsiantis dkk., 2006). Dua jenis ini adalah sebagai berikut.

- Normalisasi minimal-maksimal. Normalisasi fitur ini menggunakan nilai minimal dan maksimal dari setiap fitur pada data latih sebagai acuan transformasi data. Normalisasi ini akan mengakibatkan nilai pada setiap fitur berada di dalam selang yang sama, misalkan selang  $[a, b]$ . Dengan demikian, penulisan normalisasi minimal-maksimal secara matematis adalah

$$(x_j^{(l)})^{(\text{baru})} = \frac{(x_j^{(l)})^{(\text{lama})} - \min_{(l')} (x_j^{(l')})^{(\text{lama})}}{\max_{(l')} (x_j^{(l')})^{(\text{lama})} - \min_{(l')} (x_j^{(l')})^{(\text{lama})}} (b - a) + a \quad (3.1)$$

dengan  $(x_j^{(l)})^{(\text{lama})}$  dan  $(x_j^{(l)})^{(\text{baru})}$  berturut-turut menyatakan nilai lama dan nilai baru dari fitur ke- $j$  pada observasi ke- $l$  dari data latih. Sebagian besar ilmuwan data dan peneliti yang menggunakan normalisasi minimal-maksimal sering mentransformasi fitur-fitur pada data latih menjadi bilangan real pada selang  $[0, 1]$ .

- Normalisasi standar. Normalisasi fitur ini menggunakan rata-rata dan simpangan

baku dari setiap fitur pada data latih sebagai acuan transformasi data. Normalisasi standar bertujuan untuk menyamakan rata-rata dan simpangan baku dari semua fitur yang ada. Rata-rata untuk setiap fitur bernilai 0 dan simpangan bakunya bernilai 1. Dengan demikian, penulisan normalisasi standar secara matematis adalah sebagai berikut.

$$(x_j^{(l)})^{(\text{baru})} = \frac{(x_j^{(l)})^{(\text{lama})} - \bar{x}_j^{(\text{lama})}}{s_{x_j}^{(\text{lama})}} \quad (3.2)$$

Perlu diperhatikan bahwa nilai minimal dan maksimal pada Persamaan (3.1) untuk setiap fitur ke- $j$  hanya berdasarkan pada data latih. Begitu juga dengan rata-rata dan simpangan baku pada (3.2). Normalisasi fitur pada data uji berdasarkan nilai-nilai yang digunakan pada normalisasi fitur data latih. Selain itu, dua jenis normalisasi ini tidak dapat digunakan beriringan. Hal ini dikarenakan tujuan dari dua jenis normalisasi ini sangat berbeda.

#### III.2.4 Analisis Komponen Utama

Berdasarkan penjelasan pada Bab I, salah satu tantangan utama di era revolusi industri 4.0 adalah peningkatan kompleksitas data. Peningkatan kompleksitas ini termasuk peningkatan banyaknya fitur pada data yang disebut dengan dimensi dari data. Supaya proses membangun model JSF berjalan dengan efektif meskipun data memiliki dimensi yang sangat besar, maka diperlukan proses ekstraksi fitur. Salah satu metode untuk mengekstrak fitur adalah analisis komponen utama atau PCA (principal component analysis). PCA bekerja dengan cara memproyeksi setiap observasi terhadap beberapa vektor. PCA bertujuan untuk mengurangi dimensi data (Rogers dan Girolami, 2017). Setiap vektor hasil proyeksi dari PCA berupa kombinasi linier dari fitur-fitur pada data asal.

Misalkan suatu data terdiri dari  $L$  observasi, yaitu  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(L)}$ . Setiap observasi ke- $l$  memiliki  $n$  fitur, yaitu  $\mathbf{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \dots, x_n^{(l)})$ . Maka PCA akan memproyeksi setiap  $\mathbf{x}^{(l)}$  menjadi suatu komponen utama  $z_j^{(l)}$  pada observasi ke- $l$ , yaitu

$$z_j^{(l)} = \mathbf{w}_j \cdot \mathbf{x}^{(l)}$$

dengan vektor  $\mathbf{w}_j$  merupakan vektor yang dinormalkan dan berupa vektor eigen dari matriks kovariansi data tersebut. Nilai eigen yang bersesuaian dengan  $\mathbf{w}_j$  menjadi variansi dari komponen utama  $z_j^{(l)}$ . Matriks kovariansi tidak hanya memiliki satu

pasang nilai dan vektor eigen. Vektor eigen yang dipilih pertama untuk memproyeksikan setiap observasi adalah vektor yang nilai eigennya paling besar. Vektor eigen ini, sebut saja vektor  $\mathbf{w}_1$ , akan menghasilkan komponen utama pertama pada observasi ke- $l$ , yaitu  $z_1^{(l)} = \mathbf{x}^{(l)} \cdot \mathbf{w}_1$ . Selanjutnya, komponen utama kedua merupakan hasil proyeksi dari vektor eigen dengan nilai eigen terbesar kedua, dan seterusnya. Dengan demikian, komponen utama  $z_1$  memiliki variansi terbesar dan komponen terakhir ( $z_n$ ) memiliki variansi terkecil.

Misalkan suatu data memiliki  $n$  fitur. Banyaknya fitur ini dapat dipandang sebagai dimensi dari data. Data yang berdimensi  $n$  ini akan diproyeksi terhadap dimensi  $q < n$  dengan menggunakan PCA. Maka PCA telah berperan dalam proses pengurangan dimensi pada data tersebut. Konsekuensinya, total variansi pada hasil proyeksi tidak terserap sepenuhnya dari data asli. Pengurangan dimensi data ini akan sangat berguna ketika data digambarkan pada grafik dua dimensi. Jika diperlukan normalisasi data dan fitur-fitur pada data akan diekstrak menggunakan PCA, maka normalisasi yang harus dilakukan adalah normalisasi standar (Rogers dan Girolami, 2017).

### III.2.5 Pengurutan Observasi pada Data Latih

Pada fase identifikasi struktur di dalam proses pembangunan model JSF, urutan observasi dari data latih sangat berpengaruh (Yeh dkk., 2012). Metode pengurutan observasi yang digunakan oleh Yeh dkk. (2012) adalah metode heuristik. Nilai maksimal dari semua fitur pada observasi pertama harus lebih besar dari nilai maksimal semua fitur pada observasi-observasi yang lain. Begitu juga dengan nilai maksimal dari observasi kedua, harus lebih besar dari nilai maksimal semua fitur pada observasi-observasi setelahnya. Dengan demikian, observasi-observasi pada data latih diurutkan berdasarkan nilai maksimal dari fitur-fiturnya. Jika terdapat nilai fitur maksimal yang sama pada dua observasi atau lebih, maka pengurutan observasi-observasi ini berdasarkan nilai terbesar kedua dari fitur-fiturnya, dan seterusnya. Contoh III.1 mendeskripsikan proses pengurutan observasi pada data latih dengan rinci.

**Contoh III.1** (Contoh pengurutan observasi pada data latih) Misalkan diberikan data latih pada Tabel III.1. Data latih ini akan digunakan untuk membangun suatu model JSF.

Tabel III.1: Contoh data latih yang belum diurutkan

Indeks	Fitur			Label
	$x$	$y$	$z$	Oktan
1	1	2	1	Oktan I
2	2	-3	4	Oktan IV
3	4	2	-1	Oktan V
4	1	-1	1	Oktan IV
5	-2	3	-3	Oktan VI
6	1	3	2	Oktan I
7	-4	2	-1	Oktan VI
8	1	3	-4	Oktan V

Sebelum mengurutkan data latih tersebut, urutkan terlebih dahulu nilai terbesar sampai nilai terkecil dari fitur-fitur pada setiap observasinya. Hasil dari pengurutan fitur untuk setiap observasi ini diberikan pada Tabel III.2. Nilai fitur terbesar sampai terkecil berturut-turut dinyatakan pada kolom “Maks 1”, “Maks 2”, dan “Maks 3”.

Tabel III.2: Hasil pengurutan fitur untuk masing-masing observasi

Indeks	Fitur			Pengurutan fitur			Label
	$x$	$y$	$z$	Maks 1	Maks 2	Maks 3	Oktan
1	1	2	1	2	1	1	Oktan I
2	2	-3	4	4	2	-3	Oktan IV
3	4	2	-1	4	2	-1	Oktan V
4	1	-1	1	1	1	-1	Oktan IV
5	-2	3	-3	3	-2	-3	Oktan VI
6	1	3	2	3	2	1	Oktan I
7	-4	2	-1	2	-1	-4	Oktan VI
8	1	3	-4	3	1	-4	Oktan V

Selanjutnya, observasi-observasi pada data latih di dalam Tabel III.2 ini fitur-fiturnya diurutkan secara menurun berdasarkan urutan sebagai berikut: “Maks 1”, “Maks 2”, dan “Maks 3”. Hasil proses pengurutan untuk contoh data latih pada Tabel III.1 diberikan pada Tabel III.3.

Tabel III.3: Contoh data latih yang telah diurutkan

No. urut	Indeks	Fitur			Label
		$x$	$y$	$z$	Oktan
1	3	4	2	-1	Oktan V
2	2	2	-3	4	Oktan IV
3	6	1	3	2	Oktan I
4	8	1	3	-4	Oktan V
5	5	-2	3	-3	Oktan VI
6	1	1	2	1	Oktan I
7	7	-4	2	-1	Oktan VI
8	4	1	-1	1	Oktan IV

Urutan observasi dari data latih pada Tabel III.3 ini akan digunakan untuk membangun model JSF yang dimulai dari fase identifikasi struktur.

### III.3 Skema Klasifikasi Multikelas

Sebagaimana telah disebutkan sebelumnya bahwa label pada data yang termasuk masalah klasifikasi multikelas berupa kategori. Supaya dapat digunakan untuk membangun model JSF, label yang berbentuk kategori ini harus dikonversi menjadi bilangan real. Pada klasifikasi biner, label cukup dikonversi menjadi bilangan 0 untuk suatu kategori dan bilangan 1 untuk kategori yang lain. Klasifikasi multikelas juga akan mengonversi label menjadi sejumlah digit biner 0 dan 1. Strategi pengonversian label ini disebut dengan pengkodean label.

Pengkodean label dapat dilakukan dengan berbagai metode. Setiap metode akan mengakibatkan skema klasifikasi yang berbeda-beda dari metode lainnya. Oleh karena itu, pengkodean label ini merupakan tahap yang paling dasar dari skema klasifikasi multikelas. Setiap skema klasifikasi multikelas memiliki metode yang berbeda-beda dalam proses pembangunan model JSF. Selain itu, proses prediksi label pada data uji dan data validasi juga berbeda untuk setiap skema klasifikasinya. Pada tugas akhir ini, akan digunakan tiga skema klasifikasi multikelas yang dijelaskan oleh Ou dan Murphey (2007), yaitu skema satu lawan semua, skema satu lawan satu, dan skema satu lawan orde yang lebih tinggi.

Masalah klasifikasi multikelas dengan banyaknya  $p$  kelas yang berbeda dideskripsikan secara formal sebagai berikut. Misalkan diberikan data berukuran  $L \times (n+1)$ , yaitu data yang terdiri dari  $L$  observasi dan setiap observasinya memiliki  $n$  fitur. Selanjutnya, diberikan *learning set* yang merupakan sub himpunan dari data asal. Setiap observasi  $\mathbf{x}^{(l)}$  di dalam *learning set* berasosiasi dengan suatu label kelas  $\mathcal{C}$ ,  $\mathcal{C} \in \mathfrak{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$  dengan  $\mathcal{C}_h \neq \mathcal{C}_k$  untuk setiap  $h \neq k$  dan  $p > 2$ . Misalkan  $F$  adalah fungsi yang ada di dalam himpunan hipotesis. Maka  $F$  memetakan  $\mathbf{x}^{(l)}$  terhadap himpunan  $\mathfrak{C}$ , atau  $F(\mathbf{x}^{(l)}) \in \mathfrak{C}$ .

Pada skema satu lawan semua, setiap kelas  $\mathcal{C}_k$  ditandingkan melawan kelas-kelas yang lain. Model JSF yang dibangun menggunakan skema ini hanya berupa model JSF tunggal dan terdiri dari  $p$  neuron pada lapisan keluaran. Pada skema satu lawan satu, setiap kelas  $\mathcal{C}_k$  ditandingkan dengan kelas  $\mathcal{C}_h$ ,  $k \neq h$ . Dengan kata lain, skema satu lawan satu menandingkan setiap kelasnya satu per satu. Model JSF yang dibangun dengan skema ini berupa model JSF berganda dengan jumlah jaringan yang terbentuk adalah sebanyak  $\binom{p}{2} = \frac{p(p-1)}{2}$ . Pada skema satu lawan orde yang lebih tinggi, menandingkan setiap kelas dengan kelas-kelas lain yang kardinal-

itasnya lebih besar. Skema ini akan menghasilkan model JSF berganda yang terdiri dari  $p - 1$  jaringan.

### III.3.1 Skema Satu Lawan Semua

Sebagaimana telah disebutkan sebelumnya, skema satu lawan semua atau *one against all* (OAA) menandingkan setiap kelas melawan semua kelas yang lain. Maka pengkodean label pada skema OAA adalah sebagai berikut. Setiap label kelas  $C_k$  dikonversi menjadi vektor  $\mathbf{d} = (d_1, d_2, \dots, d_p)$  dengan

$$d_j = \begin{cases} 1, & \text{jika } j = k \\ 0, & j \text{ lainnya} \end{cases}$$

Dengan demikian, pengkodean label ini sama dengan pengkodean fitur dengan metode *one hot encoding*.

Model yang dihasilkan menggunakan skema OAA berupa model JSF tunggal dengan  $p$  neuron pada lapisan keluarannya, seperti yang telah digambarkan pada Gambar II.3. *Feedforward* pada JSF akan menghasilkan sebanyak  $p$  neuron keluaran, yaitu  $\mathbf{t} = (t_1, t_2, \dots, t_p)$ . Dengan kata lain, model JSF ini tidak langsung menghasilkan suatu label kelas. Ketika data validasi dan data uji dimasukkan ke dalam model JSF, maka keluarannya harus berupa label kelas untuk keperluan prediksi, bukan vektor  $\mathbf{t}$ . Dengan demikian, diperlukan suatu fungsi keputusan  $v$  yang memetakan  $\mathbf{t}$  menjadi suatu label kelas. Untuk skema OAA, fungsi keputusan  $v$  memetakan vektor  $\mathbf{t}$  terhadap label kelas yang indeksinya memiliki entri terbesar pada vektor  $\mathbf{t}$ . Jadi, fungsi keputusan  $v$  untuk skema OAA adalah

$$\begin{aligned} v(\mathbf{t}) &= C_k, \text{ dengan} \\ k &= \arg \max_{h=1,2,\dots,p} (t_h) \end{aligned} \tag{3.3}$$

### III.3.2 Skema Satu Lawan Satu

Skema satu lawan satu atau *one against one* (OAO) menandingkan setiap kelas melawan satu kelas yang lain. Model JSF yang terbentuk adalah model JSF berganda yang terdiri dari  $N$  jaringan,  $N = \frac{p(p-1)}{2}$ . Setiap jaringan hanya mempunyai satu neuron pada lapisan keluaran. Misalkan kelas  $C_h$  ditandingkan melawan kelas  $C_k$ ,  $1 \leq h < k \leq p$ . Maka akan terbentuk satu JSF dari total  $N$

jaringan pada model JSF, sebut saja JSF  $\{h, k\}$ . Observasi-observasi dari *learning set* yang digunakan untuk membentuk JSF  $\{h, k\}$  hanyalah observasi yang memiliki label kelas  $\mathcal{C}_h$  atau  $\mathcal{C}_k$ . Misalkan nilai neuron yang diinginkan pada lapisan keluaran dari JSF  $\{h, k\}$  dinyatakan dengan  $d_{h,k}$ . Maka  $d_{h,k} = 1$  untuk observasi yang memiliki label  $\mathcal{C}_k$  dan  $d_{h,k} = 0$  untuk observasi yang memiliki label  $\mathcal{C}_h$ .

Misalkan suatu model JSF dengan skema OAO telah diperoleh. Ketika data uji atau data validasi dimasukkan ke dalam model tersebut, maka setiap observasinya akan menghasilkan  $N$  nilai keluaran. Misalkan keluaran dari JSF  $\{h, k\}$  dinyatakan dengan  $t_{h,k}$  untuk setiap  $h = 1, 2, \dots, p-1$  dan  $k = h+1, h+2, \dots, p$ . Fungsi keputusan  $v$  memetakan vektor  $\mathbf{t} = (t_{1,2}, t_{1,3}, \dots, t_{p-1,p})$  terhadap label kelas yang indeksinya mempunyai suara terbanyak pada neuron keluaran dari setiap JSF. Maka fungsi keputusan  $v$  pada skema OAO untuk menentukan label kelas berdasarkan  $N$  nilai keluaran ini adalah

$$v(\mathbf{t}) = \mathcal{C}_k, \text{ dengan} \\ k = \arg \max_{h=1,2,\dots,p} \left\{ \left( \sum_{q=1}^{h-1} \text{round}(t_{q,h}) + \sum_{q=h+1}^p [1 - \text{round}(t_{h,q})] \right) \right\} \quad (3.4)$$

### III.3.3 Skema Satu Lawan Orde yang Lebih Tinggi

Skema satu lawan orde yang lebih tinggi atau *one against higher order* (OAHO) menandingkan setiap kelas dengan kelas-kelas lain yang kardinalitasnya lebih besar. Oleh karena itu, sebelum memberi indeks terhadap setiap kelas, harus dilakukan penghitungan jumlah observasi untuk masing-masing kelas dalam *learning set*. Jumlah observasi untuk kelas  $\mathcal{C}_k$  disebut dengan kardinalitas kelas  $\mathcal{C}_k$  yang dinyatakan dengan  $|\mathcal{C}_k|$ . Setelah menghitung  $|\mathcal{C}_k|$  untuk setiap  $k = 1, 2, \dots, p$ , lakukan pengurutan dari yang terkecil. Dengan demikian, diperoleh  $\mathfrak{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$  dengan  $|\mathcal{C}_1| \leq |\mathcal{C}_2| \leq \dots \leq |\mathcal{C}_p|$ . Selanjutnya, dilakukan pengkodean untuk setiap label kelas.

Pengkodean untuk setiap label kelas mengikuti mekanisme berikut ini. Pertandingkan kelas  $\mathcal{C}_1$  melawan  $\mathfrak{C}_1^+ = \{\mathcal{C}_2, \mathcal{C}_3, \dots, \mathcal{C}_p\}$ . Akibatnya, terbentuk JSF  $\{\mathcal{C}_1, \mathfrak{C}_1^+\}$  dengan *learning set* yang dinyatakan sebagai  $\mathfrak{C}_1$  dan terdiri dari semua observasi pada *learning set* asal. Misalkan nilai neuron yang diinginkan pada lapisan keluaran dari JSF  $\{\mathcal{C}_1, \mathfrak{C}_1^+\}$  dinyatakan dengan  $d_{1,1^+}$ . Maka  $d_{1,1^+} = 1$  untuk observasi pada  $\mathfrak{C}_1$  yang label kelasnya berupa  $\mathcal{C}_1$  dan  $d_{1,1^+} = 0$  untuk observasi

pada  $\mathfrak{C}_1^+ \subset \mathfrak{C}_1$ . Selanjutnya, untuk setiap  $k = 2, 3, \dots, p-1$  pertandingkan kelas  $\mathcal{C}_k$  melawan  $\mathfrak{C}_k^+ = \{\mathcal{C}_{k+1}, \mathcal{C}_{k+2}, \dots, \mathcal{C}_p\}$ . Akibatnya, terbentuk JSF  $\{\mathcal{C}_k, \mathfrak{C}_k^+\}$  dengan *learning set* yang dinyatakan sebagai  $\mathfrak{C}_k$  dan terdiri dari observasi-observasi pada *learning set* asal yang label kelasnya adalah  $\mathcal{C}_k, \mathcal{C}_{k+1}, \dots, \mathcal{C}_p$ . Misalkan nilai neuron yang diinginkan pada lapisan keluaran dari JSF  $\{\mathcal{C}_k, \mathfrak{C}_k^+\}$  dinyatakan dengan  $d_{k,k^+}$ . Maka  $d_{k,k^+} = 1$  untuk observasi pada  $\mathfrak{C}_k$  yang label kelasnya berupa  $\mathcal{C}_k$  dan  $d_{k,k^+} = 0$  untuk observasi pada  $\mathfrak{C}_k^+ \subset \mathfrak{C}_k$ . Dengan demikian, terbentuk JSF berganda yang terdiri dari  $p-1$  jaringan.

Misalkan telah diperoleh suatu model JSF dengan skema OAHO. Ketika data uji dan data validasi dimasukkan ke dalam model tersebut, maka setiap observasinya akan menghasilkan sebanyak  $p-1$  nilai keluaran. Misalkan keluaran dari JSF  $\{\mathcal{C}_k, \mathfrak{C}_k^+\}$  untuk setiap  $k = 1, 2, \dots, p-1$  dinyatakan dengan  $t_k$ . Untuk setiap observasi  $\mathbf{x}^{(l)}$  pada data uji atau data validasi, vektor  $\mathbf{t} = (t_1, t_2, \dots, t_{p-1})$  dikonversi menjadi label kelas dengan langkah-langkah sesuai Algoritma III.1.

---

**Algoritma III.1: Konversi vektor  $\mathbf{t}$  menjadi label kelas dalam skema OAHO**

---

```

1  input:  $p$  //banyaknya kelas label yang berbeda
2           $\mathbf{t}$  //keluaran dari model JSF, berdimensi  $p-1$ 
3  output:  $\mathcal{C}_k$ 
4  mulai
5       $k = 1$ 
6      selagi  $k < p$ 
7          jika  $\text{round}(t_k) = 1$ 
8              menghasilkan  $\mathcal{C}_k$ 
9              berhenti
10         kalau_tidak
11              $k = k + 1$ 
12         berakhir
13         menghasilkan  $\mathcal{C}_p$ 
14 berakhir

```

---

### III.4 Skema Pemilihan Model

Model ML yang akan dibangun pada tugas akhir ini hanya model JSF. Meskipun hanya satu jenis model ML yang akan dibangun, banyaknya fungsi pada himpunan hipotesis yang diperoleh bisa lebih dari satu. Fungsi-fungsi tersebut dikarakterisasi oleh sejumlah parameter. Parameter yang dihasilkan ini akan sangat bergantung kepada penentuan nilai-nilai dari hiperparameter. Oleh karena itu, untuk mendapatkan model JSF yang optimal dibutuhkan alur dan strategi dalam penentuan nilai-nilai dari hiperparameter. Selanjutnya, alur dan strategi yang dibutuhkan ini akan



disebut dengan skema pemilihan model.

Dalam skema pemilihan model, Mohri dkk. (2018) mengusulkan untuk menggunakan validasi silang  $K$ -fold terhadap data latih. Misalkan diberikan data latih  $\mathfrak{T}$  yang terdiri dari  $L$  observasi. Setiap observasi terdiri dari  $n$  fitur dan satu label kelas. Misalkan banyaknya label kelas yang berbeda pada data latih  $\mathfrak{T}$  ada sebanyak  $p$ . Selanjutnya, data latih  $\mathfrak{T}$  dipartisi ke dalam  $K$  sub data latih, yaitu  $\mathfrak{T}_1, \mathfrak{T}_2, \dots, \mathfrak{T}_K$ . Proses partisi ini menggunakan metode bertingkat (*stratified*). Dengan kata lain, proporsi label kelas  $\mathcal{C}_k$  untuk setiap  $k = 1, 2, \dots, p$  di dalam data latih  $\mathfrak{T}$  hampir sama dengan proporsi  $\mathcal{C}_k$  di dalam sub data latih  $\mathfrak{T}_\iota$  untuk setiap  $\iota = 1, 2, \dots, K$ . Selain itu, kardinalitas antar sub data latih harus saling berdekatan.

Misalkan  $\theta = (\rho, \tau, s_0, \eta)$  menyatakan vektor hiperparameter untuk model JSF. Pilih nilai untuk setiap entri dari vektor  $\theta$ . Kemudian lakukan perulangan sebanyak  $K$  kali. Untuk setiap perulangan ke- $\iota$ , tetapkan sub data latih  $\mathfrak{T}_\iota$  sebagai data validasi, sedangkan sub data latih lainnya digabungkan dan dijadikan *learning set*. Untuk setiap perulangan, hitung akurasi model JSF yang telah diterapkan ke dalam data validasi. Misalkan  $a_\iota$  menyatakan akurasi model JSF yang diperoleh pada perulangan ke- $\iota$ ,  $|\mathfrak{T}_\iota|$  menyatakan banyaknya observasi di dalam data validasi pada perulangan ke- $\iota$ ,  $\mathcal{C}^{(l)}$  dan  $\hat{\mathcal{C}}^{(l)}$  berturut-turut menyatakan label yang sebenarnya dan label hasil model JSF untuk observasi ke- $l$  pada data validasi. Maka  $a_\iota$  adalah

$$a_\iota = \frac{1}{|\mathfrak{T}_\iota|} \sum_{l=1}^{|\mathfrak{T}_\iota|} 1_{\mathcal{C}^{(l)}=\hat{\mathcal{C}}^{(l)}} \quad (3.5)$$

dengan

$$1_{\mathcal{C}^{(l)}=\hat{\mathcal{C}}^{(l)}} = \begin{cases} 1, & \text{jika } \mathcal{C}^{(l)} = \hat{\mathcal{C}}^{(l)} \\ 0, & \text{lainnya} \end{cases}$$

Setelah semua sub data latih digunakan sebagai data validasi, hitung rata-rata dan simpangan baku dari akurasi model JSF. Misalkan  $\bar{a}(\theta)$  dan  $s_a(\theta)$  berturut-turut menyatakan rata-rata dan simpangan baku dari  $a_\iota$ ,  $\iota = 1, 2, \dots, K$  untuk suatu vektor hiperparameter  $\theta$ . Maka

$$\bar{a}(\theta) = \frac{1}{L} \sum_{\iota=1}^L a_\iota \quad (3.6)$$

dan

$$s_a(\boldsymbol{\theta}) = \sqrt{\frac{1}{L-1} \sum_{\iota=1}^L [a_{\iota} - \bar{a}(\boldsymbol{\theta})]^2} \quad (3.7)$$

Dengan demikian, model JSF dengan vektor hiperparamter  $\boldsymbol{\theta}$  dapat dikatakan memiliki akurasi sebesar  $\bar{a}(\boldsymbol{\theta}) \pm s_a(\boldsymbol{\theta})$ .

Selanjutnya, pilih nilai yang lain untuk entri dari vektor hiperparameter  $\boldsymbol{\theta}$ . Lakukan kembali langkah di atas. Pemilihan nilai untuk entri dari vektor hiperparameter  $\boldsymbol{\theta}$  dapat dilakukan berkali-kali sedemikian sehingga mendapatkan akurasi model JSF ( $\bar{a}(\boldsymbol{\theta}) \pm s_a(\boldsymbol{\theta})$ ) yang dianggap paling optimal.

Misalkan diperoleh vektor hiperparameter  $\boldsymbol{\theta}_0$  dengan akurasi model JSF dianggap paling optimal. Maka hiperparameter  $\boldsymbol{\theta}_0$  adalah hiperparameter yang akan dipilih untuk membangun model JSF berdasarkan data latih keseluruhan. Selanjutnya, model JSF yang diperoleh ini diterapkan terhadap data uji untuk mendapatkan nilai akurasi model JSF berdasarkan data uji.

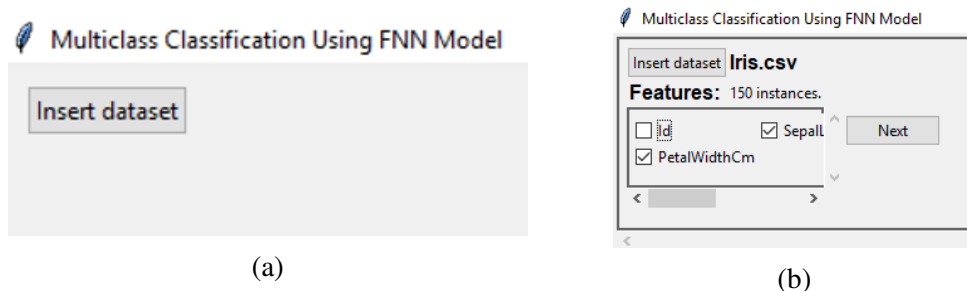
## Bab IV Pengujian Model dan Skema

Model dan skema pembelajaran mesin atau *machine learning* (ML) yang telah dibahas pada Bab II dan III akan diimplementasikan ke dalam program. Program ini akan dirancang sedemikian sehingga hasil rancangannya berbentuk GUI. Konstruksi programnya akan dijelaskan pada Subbab IV.1. Program ini dirancang untuk membangun model jaringan saraf fuzzy (JSF) dengan tiga skema klasifikasi multikelas. Program ini juga ditujukan untuk menguji model JSF yang telah diperoleh.

Data yang digunakan untuk membangun dan menguji model JSF telah memenuhi syarat-syarat yang disebutkan pada Bab I. Terdapat empat data yang digunakan. Empat data ini akan dijelaskan lebih lanjut pada Subbab IV.2. Selanjutnya, akan dibahas hasil pengujian model JSF dan skema klasifikasi multikelas terhadap masing-masing dari empat data ini. Hasil dari pengujian terhadap data ini dibahas pada Subbab IV.3.

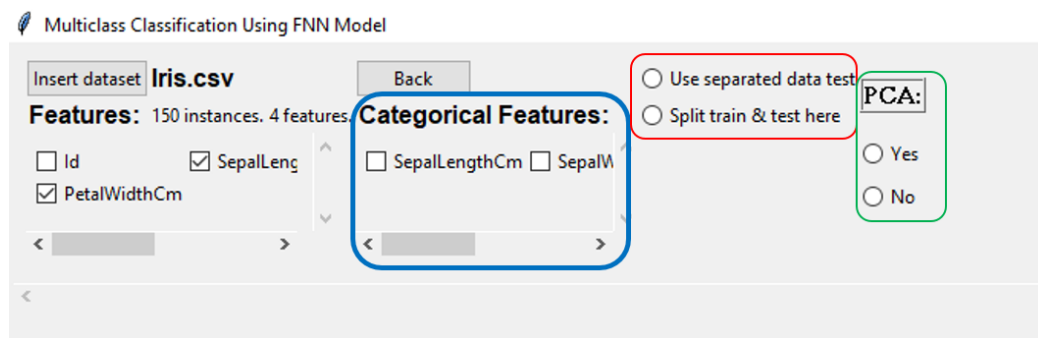
### IV.1 Konstruksi Program

Tampilan awal dari program yang telah dirancang oleh penulis dapat dilihat pada Gambar IV.1. Untuk membangun model JSF menggunakan program ini, pengguna harus memasukkan data terlebih dahulu menggunakan tombol ‘Insert dataset’ (Gambar IV.1a). Selanjutnya, pengguna diminta untuk menentukan fitur-fitur yang akan digunakan (Gambar IV.1b).

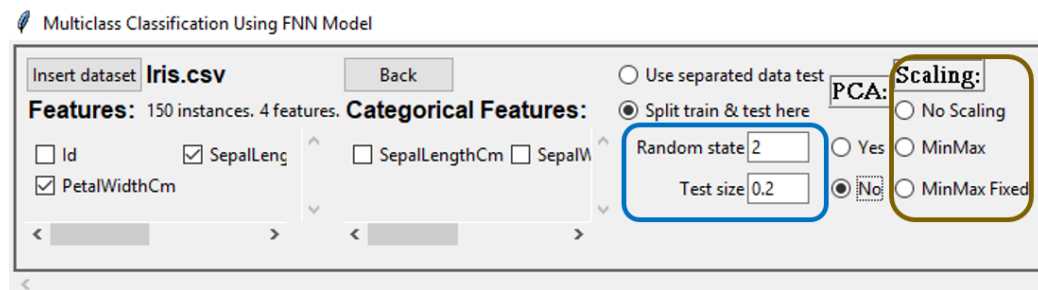


Gambar IV.1: Tampilan awal program

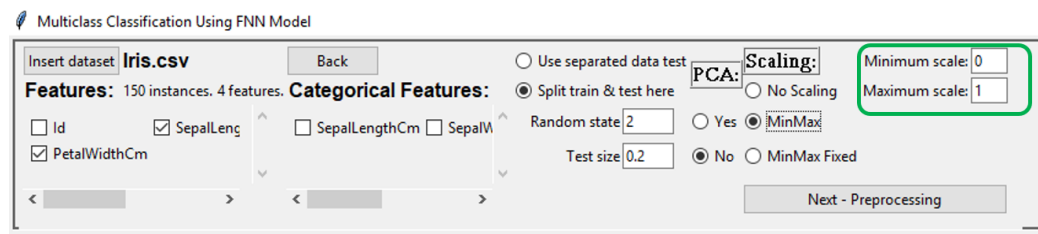
Program ini menyediakan beberapa pilihan untuk tahap pra pemrosesan data. Pada Gambar IV.2a, kotak berwarna biru digunakan untuk menentukan fitur-fitur yang berupa kategori. Di samping itu, pengguna memiliki pilihan terkait data uji yang digunakan. Pilihan ini disediakan pada kotak merah (Gambar IV.2a). Jika data uji yang digunakan terpisah dari data yang telah dimasukkan, maka nantinya pengguna harus memasukkan data uji. Jika data uji ingin dipisahkan menggunakan program ini yang dibantu oleh *package sklearn* python, maka pengguna harus memasukkan ukuran data uji dan *random state* seperti pada kotak biru di dalam Gambar IV.2b. Untuk setiap data, dua nilai yang ada di kotak biru ini dibuat tetap sebagaimana yang tercantum di dalam gambar.



(a)



(b)

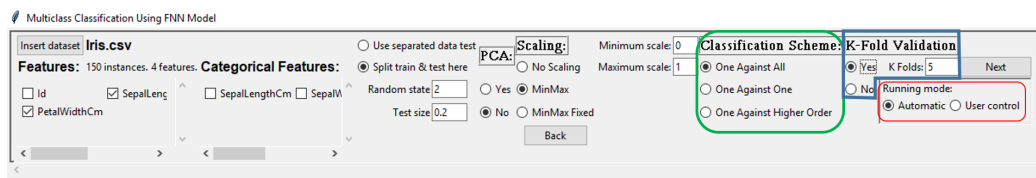


(c)

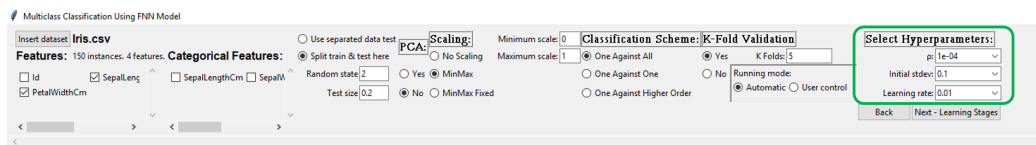
Gambar IV.2: Pilihan yang disediakan oleh program pada tahap pra pemrosesan data

Pilihan untuk menggunakan PCA atau tidak menggunakannya juga disediakan oleh program ini. Jika ingin menggunakan PCA, maka program secara otomatis akan

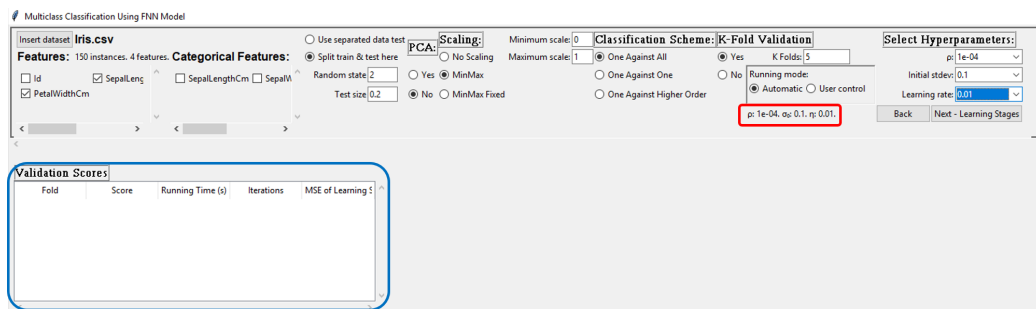
menggunakan normalisasi standar untuk penskalaan dan pengguna akan diminta untuk memasukkan banyaknya komponen utama yang diinginkan. Jika tidak, maka pengguna akan memilih jenis penskalaan nilai pada data latih sesuai dengan yang tertera di dalam kotak berwarna emas pada Gambar IV.2b. Pilihan ‘*MinMax Fixed*’ digunakan ketika fitur-fitur pada data memiliki skala nilai yang sama, diketahui nilai minimum dan maksimumnya, serta nilai-nilai dari fitur berpeluang tidak dapat diterima oleh model JSF. Pilihan ‘*MinMax Fixed*’ dan ‘*MinMax*’ sama-sama akan meminta nilai skala minimal dan maksimal yang diinginkan (Gambar IV.2c). Selanjutnya, dengan menekan tombol ‘**Next - Preprocessing**’, maka program akan melakukan pra pemrosesan data sesuai dengan pilihan pengguna yang telah dibahas sebelumnya.



(a)



(b)



(c)

Gambar IV.3: Pemilihan skema klasifikasi multikelas, validasi silang, dan hiperparameter

Gambar IV.3 menunjukkan bahwa program ini dapat menyediakan berbagai pilihan untuk menentukan skema ML. Di dalam kotak hijau pada Gambar IV.3a pengguna dapat memilih skema klasifikasi multikelas yang akan digunakan. Pilihan *running mode* di dalam kotak merah pada Gambar IV.3a menyediakan pilihan mode untuk menjalankan skema pembangunan model JSF. Jika pengguna ingin menen-

tukan nilai dari batas galat dan iterasi maksimal secara manual, serta ingin melanjutkan *running* ke sub data latih berikutnya dengan cara manual, maka gunakan pilihan ‘*user control*’. Jika pengguna menginginkan program secara otomatis dapat membangun dan menguji model JSF dari awal sampai dengan selesai, maka gunakan pilihan ‘*automatic*’. Ketika memilih pilihan ini, maka nilai dari batas galat dan iterasi maksimal, berturut-turut, ditetapkan sebesar  $10^{-4}$  dan 1.000. Sebagian besar data dan skema klasifikasi multikelas menggunakan pilihan ‘*automatic*’ untuk membangun model JSF berdasarkan data tersebut.

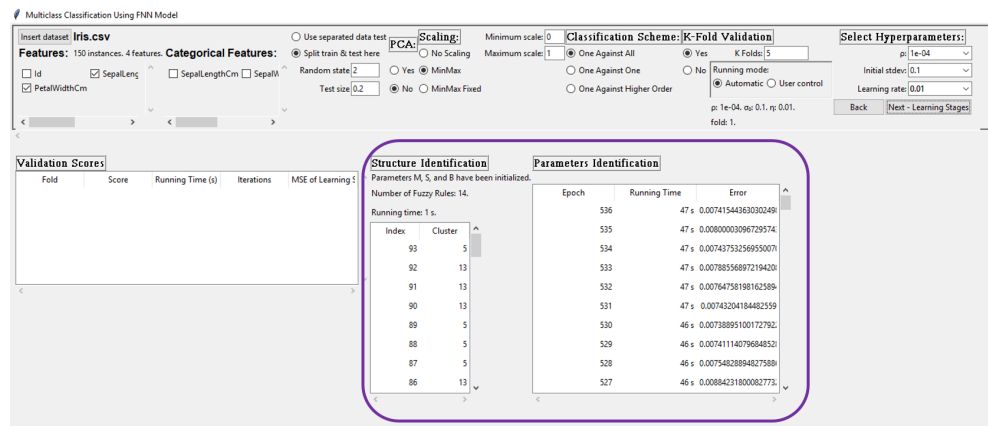
Pilihan antara akan menggunakan validasi silang atau tidak menggunakannya terdapat di dalam kotak biru pada Gambar IV.3a. Lebih jauh lagi, jika pengguna memilih untuk menggunakan validasi silang, program ini akan menyediakan fasilitas berupa banyaknya *fold* (sub data latih) yang diinginkan. Pada skema pembangunan model JSF dan pengujian setiap data di dalam tugas akhir ini, pilihan “tidak menggunakan validasi silang” digunakan ketika hiperparameter ‘terbaik’ telah diperoleh untuk suatu skema klasifikasi multikelas.

Bagian yang cukup penting dalam skema pembangunan model ML, termasuk model JSF, adalah penentuan nilai dari hiperparameter. Penentuan nilai-nilai hiperparameter disediakan di dalam kotak hijau pada Gambar IV.3b. Pada tugas akhir ini, hiperparameter yang akan berubah-ubah adalah  $\rho$ ,  $s_0$ , dan  $\eta$ . Nilai  $\tau$  dijadikan tetap.

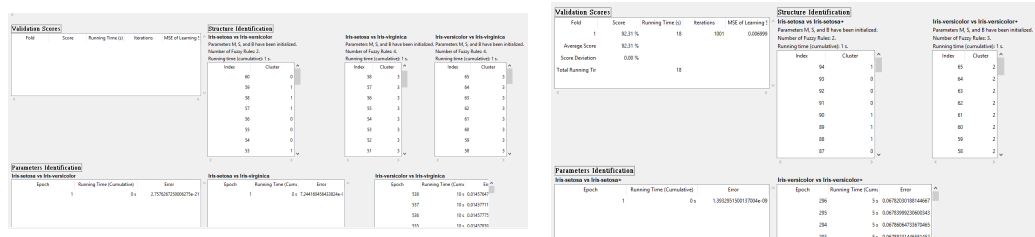
Berikut ini adalah penjelasan mengenai alasan dari nilai  $\tau$  yang dijadikan tetap. Berdasarkan penjelasan pada Subbab III.3, setiap skema klasifikasi multikelas mengakibatkan model JSF hanya bernilai 0 atau 1 pada semua neuron di dalam lapisan keluaran. Dalam skema OAA, nilai neuron-neuron pada lapisan keluaran dari observasi ke- $l$  dinyatakan dengan  $\mathbf{d}^{(l)} = (d_1^{(l)}, d_2^{(l)}, \dots, d_p^{(l)})$  dengan  $p$  menyatakan banyaknya label kelas yang berbeda pada data. Pada skema OAA, jika observasi ke- $l$  memiliki label kelas  $\mathcal{C}_k$ , maka entri ke- $k$  dari vektor  $\mathbf{d}^{(l)}$  bernilai 1 dan entri yang tersisa bernilai 0. Akibatnya, nilai dari  $u$  pada (2.19) selalu sama dengan  $\sqrt{2}$ . Dalam skema OAO dan OAHO, nilai neuron pada lapisan keluaran dari setiap jaringannya hanya di antara 0 atau 1. Akibatnya, nilai dari  $u$  pada (2.19) selalu sama dengan 1. Dengan demikian, untuk setiap  $\tau < 1$  pada Pertidaksamaan (2.19) akan menghasilkan keputusan yang sama terkait kemiripan data keluaran dengan klaster yang telah terbentuk dalam fase identifikasi struktur. Jadi, nilai hiperparameter  $\tau$  dapat dijadikan tetap. Pada model JSF yang dibangun melalui tugas akhir ini, ditetapkan  $\tau = 0,3$ .

Nilai hiperparameter  $\rho$  dipilih dari himpunan  $\{10^{-4}, 10^{-8}, 10^{-12}, 10^{-20}, 10^{-25}, 10^{-50}, 10^{-75}, 10^{-125}, 10^{-200}, 10^{-275}, 10^{-350}\}$ . Nilai  $\rho$  yang sangat kecil digunakan ketika data berdimensi cukup besar atau nilai simpangan baku dari fitur-fiturnya cukup besar. Nilai hiperparameter  $s_0$  dipilih dari himpunan  $\{0,1, 0,2, 0,25, 0,3, 0,4, 0,5, 0,6\}$  yang setiap anggotanya dikalikan dengan selisih antara nilai maksimum dan minimum dari keseluruhan nilai fitur. Nilai hiperparameter  $\eta$  dipilih dari himpunan  $\{0,01, 0,05, 0,1\}$ .

Setelah setiap hiperparameter dipilih, maka program akan menampilkan nilai-nilai hiperparameter yang akan digunakan, sebagaimana tertera di dalam kotak merah pada Gambar IV.3c. Selain itu, akan ditampilkan juga tabel skor akurasi untuk validasi silang pada kotak biru. Tabel ini akan terisi ketika satu iterasi dari proses validasi silang telah selesai. Selanjutnya, gunakan tombol ‘Next - Learning Stages’ untuk memasuki tahap utama dari pembelajaran mesin ini.



(a) Skema OOA



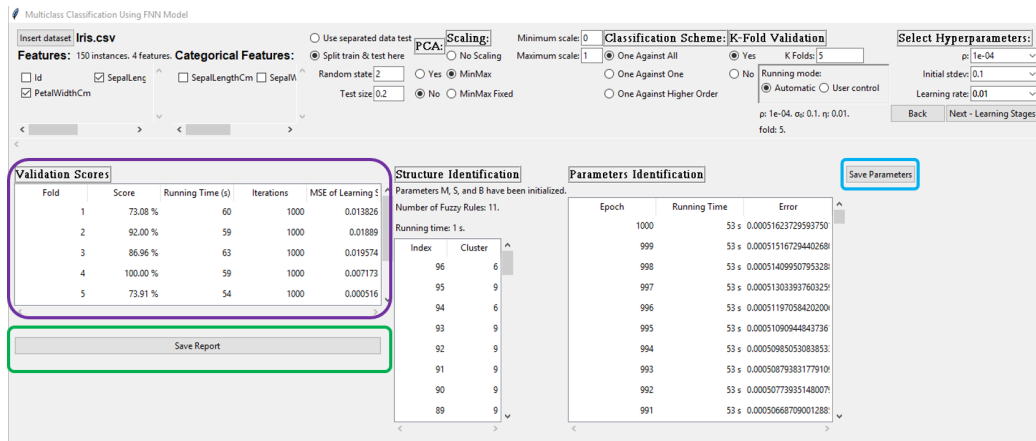
(b) Skema OAO

(c) Skema OAH

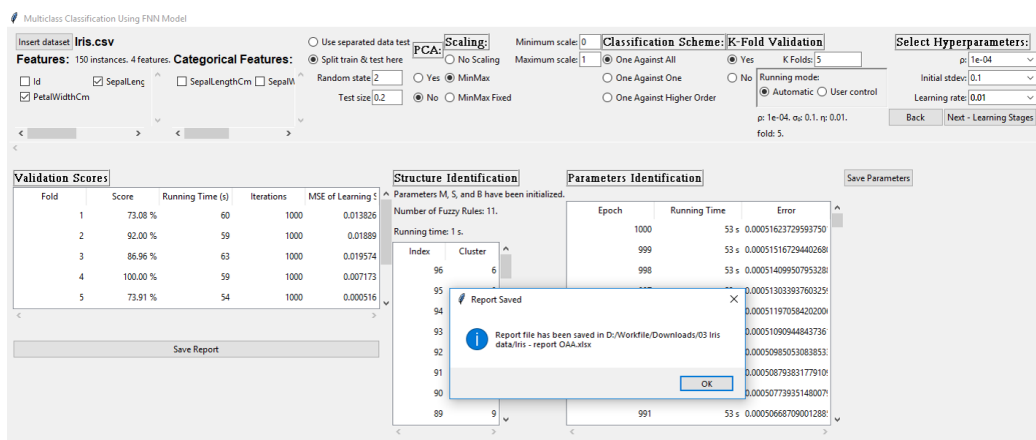
Gambar IV.4: Proses pembangunan dan pengujian model jaringan saraf fuzzy

Proses pembangunan model ditampilkan seperti pada Gambar IV.4. Pada gambar tersebut ditampilkan contoh proses pembangunan model JSF dengan skema yang berbeda. Setiap skema klasifikasi multikelas memiliki tampilan yang berbeda-beda

karena banyaknya jaringan yang terbentuk juga berbeda. Dalam kotak berwarna ungu pada Gambar IV.4a, skema OAA hanya melakukan satu kali fase identifikasi struktur dan identifikasi parameter. Ini dikarenakan skema OAA hanya membentuk satu jaringan dari model JSF. Sebagaimana telah dijelaskan pada Subbab III.3 bahwa skema OAO dan OAHO menghasilkan model JSF dengan jaringan berganda. Akibatnya, skema OAO dan OAHO (Gambar IV.4b dan Gambar IV.4c) melakukan lebih dari satu kali identifikasi struktur dan identifikasi parameter.



(a)



(b)

Gambar IV.5: Proses penyimpanan hasil uji

Jika satu kali validasi silang telah diproses, maka akan ditampilkan skor akurasi dari data validasi seperti yang tertera di dalam kotak ungu pada Gambar IV.5a. Setelah semua validasi silang selesai, maka pengguna dapat menekan tombol 'Save Report' (yang ada di dalam kotak berwarna hijau) untuk menyimpan skor akurasi dari setiap data validasi berdasarkan hiperparameter tertentu ke dalam *Ms. Excel*. Gambar IV.5b menunjukkan bahwa proses penyimpanan file yang diinginkan



berjalan dengan lancar. Selain itu, juga diperlihatkan lokasi dan nama file yang telah disimpan. Tombol ‘**Save Parameters**’ di dalam lingkaran biru pada Gambar IV.5a berfungsi untuk menyimpan nilai-nilai dari parameter yang diperoleh, yaitu matriks  $M$ ,  $S$ , dan kumpulan matriks  $B_i$  ke dalam *Ms. Excel*.

Proses validasi silang tersebut dilakukan berulang-ulang menggunakan nilai hiperparameter yang berbeda. Setelah model JSF memiliki skor akurasi yang optimal, maka hiperparameter yang terlibat dalam skema pembangunan model JSF tersebut digunakan untuk mendapatkan model JSF berdasarkan data latih secara keseluruhan, bukan dari beberapa sub data latih. Setelah itu, model yang diperoleh akan diuji terhadap data uji untuk mendapatkan skor akurasi model pada data uji.

## **IV.2 Deskripsi dan Analisis Data**

Data yang digunakan di antaranya adalah data koordinat I, data koordinat II, data tanaman iris, dan data evaluasi mobil. Dua data pertama dibangkitkan menggunakan bahasa pemrograman python. Dua data terakhir diambil dari situs web *UCI Machine Learning Repository*. Tiga data pertama merupakan data yang seimbang. Hal ini dikarenakan setiap tiga data tersebut memiliki kardinalitas yang sama untuk setiap label kelasnya. Selain itu, nilai dari fitur-fitur pada tiga data pertama berupa bilangan real. Nilai dari fitur-fitur pada data terakhir berupa kategori. Data-data tersebut belum dipisahkan menjadi data latih dan data uji. Oleh karena itu, untuk setiap data akan dipisahkan menjadi data latih dan data uji pada tahap pra pemrosesan data.

Sebelum setiap data digunakan untuk membangun dan menguji model dan skema JSF, akan dilakukan analisis terhadap masing-masing data terlebih dahulu. Pada analisis data ini, dari masing-masing data akan dicari rata-rata, simpangan baku, median, nilai minimal, dan nilai maksimal untuk setiap fitur yang berupa bilangan real. Untuk fitur yang berupa kategori, akan dicari kardinalitas setiap kategorinya.

### **IV.2.1 Data Koordinat I**

Data koordinat I terdiri dari 1.000 observasi. Setiap observasi mempunyai tiga fitur yang masing-masing menjelaskan koordinat observasi tersebut pada sumbu  $x$ ,  $y$ , dan  $z$ . Nilai-nilai untuk setiap fitur pada setiap observasi hanya berupa bilangan bulat. Label pada data koordinat I berupa nomor oktan. Banyaknya nomor oktan

yang berbeda ada 8. Dengan demikian, data ini terdiri dari 8 label kelas.

Data ini terdiri dari tiga fitur. Akibatnya, setiap observasi yang ada di dalam data ini merupakan rangkap tiga terurut dari bilangan bulat. Pada data koordinat I, rangkap tiga terurut ini dibangkitkan dengan cara menentukan permutasi tiga angka yang diambil dari himpunan  $U_1 = \{-5 \leq u \leq -1 \text{ atau } 1 \leq u \leq 5 : u \in \mathbb{Z}\}$ . Banyaknya permutasi yang mungkin adalah 1.000. Dengan demikian, semua permutasi termuat di dalam data koordinat I. Selanjutnya, setiap observasi diberi label berupa nomor oktan berdasarkan tanda pada masing-masing fiturnya. Karena semua permutasi termuat di dalam data, maka kardinalitas untuk masing-masing label kelas pada data ini adalah sama, yaitu 125. Data koordinat I dilampirkan pada Lampiran A. Berikut ini hanya ditampilkan statistika deskriptifnya saja.

Tabel IV.1: Statistika deskriptif untuk setiap fitur dari data koordinat I

Statistika deskriptif	Fitur		
	$x$	$y$	$z$
<b>Median</b>	0	0	0
<b>Rata-rata</b>	0	0	0
<b>Simpangan baku</b>	3,318	3,318	3,318
<b>Minimum</b>	-5	-5	-5
<b>Maksimum</b>	5	5	5

#### IV.2.2 Data Koordinat II

Berikut ini adalah statistika deskriptif untuk masing-masing fitur dari data koordinat II.

Tabel IV.2: Statistika deskriptif untuk setiap fitur dari data koordinat II

Statistika deskriptif	Fitur		
	$x$	$y$	$z$
<b>Median</b>	0	0	0
<b>Rata-rata</b>	-0,091	0,255	0,022
<b>Simpangan baku</b>	56,61	58,32	59,04
<b>Minimum</b>	-100	-100	-100
<b>Maksimum</b>	100	100	100

Data koordinat II mirip dengan data koordinat I. Perbedaannya terletak pada bilangan bulat rangkap tiga terurut yang dibangkitkan. Pada data koordinat II,

rangkap tiga terurut adalah sebanyak 1.000 permutasi tiga angka dari himpunan  $U_2 = \{-100 \leq u \leq -1 \text{ atau } 1 \leq u \leq 100 : u \in \mathbb{Z}\}$ . Banyaknya permutasi adalah  $200^3 = 8 \times 10^6$ . Dengan demikian, hanya 0,0125% dari semua kemungkinan permutasi yang termuat di dalam data koordinat II. Kardinalitas untuk masing-masing label kelas pada data ini dibuat sama, yaitu 125. Data ini dilampirkan pada Lampiran B.

### IV.2.3 Data Tanaman Iris

Data ini memuat 3 kelas yang masing-masing terdiri dari 50 observasi. Setiap kelas mengacu pada jenis tanaman iris, yaitu: iris *setosa*, iris *versicolor*, dan iris *virginica*. Setiap observasi memiliki 4 fitur, yaitu: panjang sepal (cm), lebar sepal (cm), panjang kelopak (cm), dan lebar kelopak (cm). Data tanaman iris merupakan data yang paling dikenal dan paling sering ditemukan dalam literatur pengenalan pola yang termasuk ke dalam masalah klasifikasi multikelas (Dua dan Graff, 2019). Data ini dilampirkan pada Lampiran C. Berikut ini hanya ditampilkan statistika deskriptifnya saja.

Tabel IV.3: Statistika deskriptif untuk setiap fitur dari data tanaman iris

Statistika deskriptif	Fitur			
	Panjang sepal	Lebar sepal	Panjang kelopak	Lebar kelopak
<b>Median</b>	5,8	3	4,35	1,3
<b>Rata-rata</b>	5,843	3,054	3,759	1,199
<b>Simpangan baku</b>	0,8281	0,4336	1,764	0,7632
<b>Minimum</b>	4,3	2	1	0,1
<b>Maksimum</b>	7,9	4,4	6,9	2,5

### IV.2.4 Data Evaluasi Mobil

Data evaluasi mobil menceritakan spesifikasi dan klasifikasi dari 1.728 mobil. Setiap mobil pada data ini diklasifikasikan ke dalam 4 label kelas berdasarkan kualitasnya. Empat label kelas ini adalah kualitas mobil yang *unaccepted*, *accepted*, *good*, dan *very good*. Setiap label kelas memiliki banyaknya observasi yang berbeda-beda. Banyaknya mobil dengan kualitas yang *unaccepted*, *accepted*, *good*, dan *very good* berturut-turut adalah 1210, 384, 69, dan 65. Setiap mobil yang ada di dalam data ini dideskripsikan oleh 6 kategori spesifikasi, yaitu:

- *Buying*, yaitu kategori harga beli mobil yang meliputi *very high*, *high*, *medium*, dan *low*,
- *Maintenance*, yaitu biaya pemeliharaan mobil yang kategorinya sama dengan kategori pada spesifikasi *buying*,
- *Doors*, yaitu kategori banyaknya pintu mobil, terdiri dari 4 kategori yang berbeda: dua pintu, tiga pintu, empat pintu, dan kategori lima pintu atau lebih,
- *Persons*, yaitu kategori banyaknya maksimal orang yang berada di dalam mobil, meliputi: kategori dua orang, kategori empat orang, dan kategori lebih dari empat orang,
- *Lug boot*, yaitu kategori ukuran bagasi mobil yang meliputi ukuran kecil, sedang, dan besar,
- *Safety*, yaitu tingkat keamanan bagi pengendara dan penumpang, meliputi tingkat keamanan rendah, sedang, dan tinggi.

Tabel IV.4: Kardinalitas setiap kategori pada setiap fitur dari data evaluasi mobil

<i>Buying</i>		<i>Maintenance</i>		<i>Doors</i>	
Kategori	Kardinalitas	Kategori	Kardinalitas	Kategori	Kardinalitas
<i>med</i>	432	<i>med</i>	432	<i>two</i>	432
<i>high</i>	432	<i>high</i>	432	<i>four</i>	432
<i>low</i>	432	<i>low</i>	432	<i>three</i>	432
<i>vhigh</i>	432	<i>vhigh</i>	432	<i>5more</i>	432

<i>Persons</i>		<i>Lug boot</i>		<i>Safety</i>	
Kategori	Kardinalitas	Kategori	Kardinalitas	Kategori	Kardinalitas
<i>two</i>	576	<i>big</i>	576	<i>med</i>	576
<i>four</i>	576	<i>med</i>	576	<i>high</i>	576
<i>more</i>	576	<i>small</i>	576	<i>low</i>	576

Berdasarkan Tabel IV.4, data evaluasi mobil memiliki kardinalitas yang seimbang antara setiap kategori pada masing-masing fiturnya. Karena semua fitur pada data ini berupa kategori, maka harus dilakukan pengkodean terlebih dahulu untuk setiap fiturnya. Pada tugas akhir ini, hanya akan dilakukan pengkodean dengan metode *one hot encoding* yang telah dijelaskan pada Subbab III.2. Setelah dilakukan *one hot encoding*, banyaknya fitur pada data evaluasi mobil menjadi 21 dengan setiap entrinya berupa bilangan biner 0 atau 1. Data lengkap untuk data evaluasi mobil ini dilampirkan pada Lampiran D.

### IV.3 Hasil Uji

Pada tugas akhir ini, setiap data akan digunakan untuk membangun model JSF dengan skema tertentu dari klasifikasi multikelas. Tiga skema klasifikasi multikelas yang telah dijelaskan pada Subbab III.3 akan digunakan untuk masing-masing data. Dengan demikian, terdapat 12 model JSF karena setiap data menghasilkan 3 model JSF dengan skema klasifikasi multikelas yang berbeda-beda.

Untuk setiap data dan skema klasifikasi, dilakukan validasi silang *K-fold* untuk mendapatkan hiperparameter terbaik. Pada tugas akhir ini, hanya akan digunakan  $K = 5$ , sehingga data latih dipartisi menjadi 5 sub data latih. Selanjutnya, dari 5 sub data latih ini akan diperoleh 5 *learning set* dan 5 data validasi yang saling bersesuaian. Metode pembuatan 5 *learning set* dan 5 data validasi ini telah dijelaskan pada Subbab III.4.

Pada validasi silang *5-fold*, setiap *learning set* akan digunakan untuk membangun model JSF dengan skema klasifikasi tertentu. Setiap model JSF yang diperoleh ini akan diuji pada data validasi. Hasil uji ini akan menghasilkan skor akurasi dari prediksi label kelas pada setiap observasi di dalam data validasi. Dengan demikian, akan ada 5 model JSF dan 5 skor akurasi yang diperoleh. Untuk meringkas performa dari 5 model JSF ini, digunakan nilai rata-rata dan simpangan baku dari 5 skor akurasi, sebagaimana telah dijelaskan pada Subbab III.4.

Model JSF yang paling optimal akan dihasilkan ketika menggunakan hiperparameter terbaik untuk membangun model JSF tersebut. Karena terdapat lebih dari satu kombinasi hiperparameter yang mungkin, maka validasi silang *5-fold* harus dilakukan secara berulang dengan kombinasi hiperparameter yang berbeda-beda. Setelah melakukan validasi silang *5-fold* secara berulang, maka dapat dipilih hiperparameter yang dapat membangun 5 model JSF dengan rata-rata akurasi terbesar, simpangan baku akurasi yang cukup kecil, dan *running time* yang lebih singkat. Meskipun validasi silang *5-fold* dilakukan berulang, hasil partisi data latih tidak akan berbeda karena partisinya hanya dilakukan di awal.

#### IV.3.1 Data Koordinat I

Berdasarkan Tabel IV.1, nilai minimal dan maksimal dari fitur pada data koordinat I masih dapat diterima oleh model JSF. Selain itu, setiap fiturnya memiliki satuan yang sama. Oleh karena itu, tidak akan ada proses normalisasi terhadap fitur-

fitur pada data ini. Pra pemrosesan data yang dilakukan terhadap data ini hanya pemisahan data latih dan data uji. Setelah dilakukan pemisahan data latih dan data uji yang dilanjutkan dengan partisi data latih ke dalam 5 bagian, diperoleh banyaknya observasi yang dikelompokkan berdasarkan label kelasnya (Tabel IV.5).

Tabel IV.5: Kardinalitas label kelas pada data koordinat I

	Oktan I	Oktan II	Oktan III	Oktan IV
<b>Data asli</b>	125	125	125	125
<b>Data latih</b>	98	89	101	102
<b>Sub data latih 1</b>	20	18	21	21
<b>Sub data latih 2</b>	20	18	20	21
<b>Sub data latih 3</b>	20	18	20	20
<b>Sub data latih 4</b>	19	18	20	20
<b>Sub data latih 5</b>	19	17	20	20
<b>Data uji</b>	27	36	24	23

	Oktan V	Oktan VI	Oktan VII	Oktan VIII
<b>Data asli</b>	125	125	125	125
<b>Data latih</b>	106	102	101	101
<b>Sub data latih 1</b>	22	21	21	21
<b>Sub data latih 2</b>	21	21	20	20
<b>Sub data latih 3</b>	21	20	20	20
<b>Sub data latih 4</b>	21	20	20	20
<b>Sub data latih 5</b>	21	20	20	20
<b>Data uji</b>	19	23	24	24

### Skema OAA

Tabel IV.6 adalah daftar ringkasan dari performa semua model JSF dengan skema OAA berdasarkan 5 *learning set* pada data koordinat I. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda. Rata-rata dan simpangan baku akurasi dihitung menggunakan (3.6) dan (3.7) berturut-turut.

Karena telah diperoleh rata-rata akurasi yang besarnya 100%, maka perulangan berhenti. Dengan demikian, hiperparameter  $\theta = (\rho, s_0, \eta) = (10^{-8}, 1, 0,01)$  adalah hiperparameter terbaik untuk data koordinat I yang dimodelkan menggunakan skema OAA. Selanjutnya, hiperparameter ini digunakan untuk membangun model JSF dengan skema OAA berdasarkan data latih dari data koordinat I. Kemudian, parameter yang telah diperoleh berdasarkan skema pembangunan model

Tabel IV.6: Hasil validasi silang 5-fold data latih pada data koordinat I dengan skema OAA

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-4}$	1	0,01	90,05%	5,15%	4657
$10^{-4}$	1	0,05	90,05%	5,15%	4669
$10^{-8}$	1	0,01	100,00%	0,00%	2178

JSF, digunakan untuk memprediksi label kelas pada data uji. Setelah dilakukan prediksi pada data uji, maka diperoleh akurasi dari prediksi tersebut. Akurasi yang diperoleh adalah 100% juga dan durasi waktu yang dibutuhkan untuk membangun model JSF adalah selama 499 detik.

#### Skema OAO

Performa semua model JSF dari data latih pada data koordinat I dengan skema OAO diringkas pada Tabel IV.7. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Tabel IV.7: Hasil validasi silang 5-fold data latih pada data koordinat I dengan skema OAO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-4}$	1	0,01	91,92%	4,59%	2215
$10^{-8}$	1	0,01	100,00%	0,00%	1373

Berdasarkan Tabel IV.7, diperoleh hiperparameter  $\theta = (\rho, s_0, \eta) = (10^{-8}, 1, 0,01)$  sebagai hiperparameter terbaik untuk data koordinat I yang dimodelkan dengan skema OAO. Hiperparameter ini menyebabkan rata-rata akurasi yang dicapai oleh model sebesar 100%. Meskipun memiliki hiperparameter terbaik yang sama dengan skema OAA, skema OAO memiliki durasi waktu yang lebih singkat untuk membangun model JSF. Hal ini dikarenakan setiap jaringan dari 28 jaringan yang dibangun memiliki *learning set* yang sedikit. Selain itu, iterasi yang diper-

lukan pada fase identifikasi parameter untuk setiap jaringannya banyak yang tidak mencapai 1.000 kali iterasi.

Setelah hiperparameter  $\theta$  digunakan untuk membangun model JSF pada data latih dengan skema OAO, diperoleh akurasi dari data uji sebesar 100%. Waktu yang diperlukan untuk membangun model JSF dengan skema OAO adalah selama 331 detik.

### Skema OAHO

Performa semua model JSF dengan skema OAHO berdasarkan 5 *learning set* pada data koordinat I diringkas pada Tabel IV.8. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Tabel IV.8: Hasil validasi silang 5-fold data latih pada data koordinat I dengan skema OAHO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-8}$	1	0,01	95,09%	4,98%	2016
$10^{-12}$	1	0,01	95,09%	4,98%	2170
$10^{-12}$	3	0,01	90,12%	12,05%	1937
$10^{-4}$	1	0,01	89,78%	6,00%	3812
$10^{-4}$	3	0,01	91,88%	9,86%	2299
$10^{-25}$	3	0,01	89,23%	13,82%	1690

Tabel IV.8 memberikan informasi bahwa jika nilai  $\rho$  semakin menjauh dari  $10^{-8}$  atau  $10^{-12}$ , maka dapat mengakibatkan rata-rata akurasi mengecil. Oleh karena itu, perulangan dari validasi silang dihentikan setelah  $\rho = 10^{-25}$ . Dengan demikian, hiperparameter  $\theta = (\rho, s_0, \eta) = (10^{-8}, 1, 0,01)$  adalah hiperparameter terbaik untuk model JSF dari data latih pada data koordinat I dengan skema OAHO dan akurasinya sebesar  $95,09\% \pm 4,98\%$ . Meskipun akurasi model JSF dengan hiperparameter  $\theta$  sama dengan akurasi dari model JSF dengan hiperparameter  $(10^{-12}, 1, 0,01)$ , tetapi hiperparameter  $\theta$  memiliki durasi waktu yang lebih singkat.

Selanjutnya, akan dibangun model JSF dari data latih pada data koordinat I dengan skema klasifikasi OAHO dan menggunakan hiperparameter  $\theta = (10^{-8}, 1, 0,01)$ .



Hasil dari model ini berupa parameter-parameter yang terlibat pada  $p - 1$  jaringan saraf fuzzy, yaitu matriks  $M_{k,k^+}$ ,  $S_{k,k^+}$ , dan  $B_{k,k^+}$  untuk  $k = 1, 2, \dots, p$ . Kemudian nilai dari parameter-parameter ini diuji terhadap data uji dan diperoleh akurasi pada data uji sebesar 98%. Durasi waktu yang diperlukan untuk membangun model JSF dari data latih tersebut adalah selama 490 detik.

### IV.3.2 Data Koordinat II

Tabel IV.9 menampilkan kardinalitas masing-masing label kelas pada data koordinat II dan pada bagian-bagiannya.

Tabel IV.9: Kardinalitas label kelas pada data koordinat II

	Oktan I	Oktan II	Oktan III	Oktan IV
<b>Data asli</b>	125	125	125	125
<b>Data latih</b>	99	100	104	102
<b>Sub data latih 1</b>	20	20	21	21
<b>Sub data latih 2</b>	20	20	21	21
<b>Sub data latih 3</b>	20	20	21	20
<b>Sub data latih 4</b>	20	20	21	20
<b>Sub data latih 5</b>	19	20	20	20
<b>Data uji</b>	26	25	21	23

	Oktan V	Oktan VI	Oktan VII	Oktan VIII
<b>Data asli</b>	125	125	125	125
<b>Data latih</b>	95	103	99	98
<b>Sub data latih 1</b>	19	21	20	20
<b>Sub data latih 2</b>	19	21	20	20
<b>Sub data latih 3</b>	19	21	20	20
<b>Sub data latih 4</b>	19	20	20	19
<b>Sub data latih 5</b>	19	20	19	19
<b>Data uji</b>	30	22	26	27

Berdasarkan Tabel IV.2, nilai minimal dan maksimal dari fitur pada data koordinat II sangat jauh dari 0. Karena simpangan baku setiap fiturnya bernilai cukup besar dan tidak seragam, maka dapat dikatakan bahwa nilai-nilai pada setiap fitur tersebar cukup luas dan penyebaran nilai antar fiturnya sangat beragam. Akibatnya, perlu dilakukan normalisasi terhadap fitur-fitur pada data ini. Karena dimensinya hanya tiga, maka tidak perlu dilakukan PCA terhadap fitur-fitur pada data ini. Dengan demikian, normalisasi yang akan dilakukan adalah normalisasi minimal-maksimal.

Supaya tidak kehilangan makna yang berhubungan dengan nomor oktan, maka akan digunakan skala minimal  $-1$  dan skala maksimal  $1$  pada proses normalisasi.

Sebelum dilakukan normalisasi, dilakukan terlebih dahulu pemisahan data menjadi data latih dan data uji. Hal ini dikarenakan proses normalisasi menggunakan nilai minimal dan maksimal dari setiap fitur pada data latih. Setelah normalisasi, dilakukan partisi data latih ke dalam 5 bagian. Banyaknya observasi pada data koordinat II dan pada bagian-bagiannya yang dikelompokkan berdasarkan label kelas didaftarkan pada Tabel IV.9.

### Skema OAA

Performa semua model JSF dengan skema OAA berdasarkan 5 *learning set* pada data koordinat II diringkas pada Tabel IV.10. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Tabel IV.10: Hasil validasi silang 5-fold data latih pada data koordinat II dengan skema OAA

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-12}$	0,5	0,01	86,59%	14,29%	2896
$10^{-20}$	0,5	0,01	87,76%	11,44%	2143
$10^{-20}$	0,2	0,01	87,50%	11,92%	2132
$10^{-20}$	1,2	0,01	88,77%	9,81%	2129
$10^{-25}$	1,2	0,01	89,04%	8,38%	2136
$10^{-50}$	1,2	0,01	88,78%	8,83%	1859
$10^{-125}$	1,2	0,01	89,28%	8,98%	1680
$10^{-200}$	1,2	0,01	89,28%	8,98%	1674
$10^{-200}$	0,6	0,01	89,15%	9,22%	1673
$10^{-275}$	1,2	0,01	89,28%	8,98%	1684
$10^{-350}$	1,2	0,01	89,40%	8,96%	1525
$10^{-350}$	1	0,01	89,15%	9,13%	1518

Berdasarkan Tabel IV.10, jika nilai  $s_0$  dijadikan tetap, maka rata-rata akurasi model akan membesar seiring dengan mengecilnya nilai  $\rho$ . Jika nilai  $\rho$  dijadikan tetap, maka rata-rata akan membesar seiring dengan membesarnya nilai  $s_0$ . Dengan demikian, rata-rata akurasi yang optimal diperoleh ketika nilai  $\rho$  dan  $s_0$  berturut-

berturut adalah nilai terkecil dan terbesar dari himpunan nilai-nilai  $\rho$  dan  $s_0$  yang tersedia. Jadi, berdasarkan Tabel IV.10 dan penjelasan di atas,  $\theta = (10^{-350}, 1, 2, 0, 01)$  adalah hiperparameter paling optimal untuk model JSF dari kombinasi empat sub data latih pada data koordinat II dengan skema OAA. Akurasi yang diperoleh sebesar  $89,4\% \pm 8,96$ .

Perhatikan bahwa durasi waktu untuk membangun model JSF cenderung semakin singkat seiring dengan mengecilnya nilai  $\rho$ . Hal ini dikarenakan untuk fitur-fitur dengan variansi yang cukup besar, nilai  $\rho$  yang lebih besar akan mengakibatkan lebih banyak implikasi pada aturan fuzzy yang terbentuk dari fase identifikasi struktur. Akibatnya, kompleksitas waktu pada fase identifikasi parameter semakin besar.

Selanjutnya, hiperparameter  $\theta = (10^{-350}, 1, 2, 0, 01)$  digunakan untuk membangun model JSF dari data latih pada data koordinat II dengan skema klasifikasi OAA. Hasil dari model ini berupa parameter-parameter yang terlibat pada JSF tunggal, yaitu matriks  $\mathbf{M}$  dan  $\mathbf{S}$ , serta kumpulan matriks  $\mathbf{B}_i$  untuk  $i = 1, 2, \dots, r$  dengan  $r$  menyatakan banyaknya implikasi pada aturan fuzzy. Setelah nilai dari parameter-parameter ini digunakan untuk menentukan label kelas dari setiap observasi pada data uji, diperoleh akurasi sebesar 96%. Durasi waktu yang diperlukan adalah selama 378 detik.

### Skema OAO

Performa semua model JSF dengan skema OAO berdasarkan 5 *learning set* pada data koordinat II diringkas pada Tabel IV.11. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Berdasarkan penjelasan pada Subbab III.3, skema OAO menandingkan setiap label kelas dengan label kelas yang lain satu per satu. Akibatnya, setiap jaringan dari model JSF berganda yang dibangun dengan skema ini lebih mudah mengenali label kelas dari observasi-observasi pada *learning set* berdasarkan nilai dari setiap fiturnya, meskipun nilai  $\rho$  tidak terlalu kecil. Jika nilai  $\rho$  terlalu besar, maka setiap jaringan dari model JSF berganda yang dibangun dengan skema OAO ini akan sangat membedakan observasi-observasi yang label kelasnya sama pada *learning set*. Hal ini disebabkan oleh nilai variansi yang sangat besar untuk setiap fitur dari observasi-observasi dengan label kelas yang sama. Oleh karena itu, nilai  $\rho$  yang

Tabel IV.11: Hasil validasi silang 5-fold data latih pada data koordinat II dengan skema OAO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-8}$	0,2	0,01	80,75%	14,11%	2958
$10^{-12}$	0,2	0,01	83,54%	11,26%	2658
$10^{-20}$	0,2	0,01	83,04%	12,89%	1977
$10^{-20}$	1,2	0,01	85,33%	10,66%	2018
$10^{-25}$	1,2	0,01	85,60%	9,96%	2065
$10^{-50}$	1,2	0,01	84,69%	11,20%	1873
$10^{-50}$	0,6	0,01	84,70%	10,73%	1885
$10^{-25}$	0,6	0,01	83,55%	11,90%	2033
$10^{-75}$	1,2	0,01	84,81%	12,08%	1807

mungkin menyebabkan model JSF mendekati optimal adalah nilai  $\rho$  yang tidak terlalu besar dan tidak terlalu kecil, sesuai dengan performa model pada Tabel IV.11.

Tabel IV.11 menunjukkan bahwa nilai hiperparameter terbaik adalah  $\theta = (10^{-25}, 1,2, 0,01)$ . Hiperparameter  $\theta$  ini mengakibatkan akurasi seluruh model JSF yang dibangun dari 5 *learning set* dengan skema OAO adalah 85,6%.

Selanjutnya, hiperparameter  $\theta$  tersebut digunakan untuk membangun model JSF dari data latih dengan skema OAO. Skema pembangunan model ini menghasilkan parameter-parameter yang berupa kumpulan matriks  $M_{h,k}$ ,  $S_{h,k}$ , dan  $B_{h,k}$  untuk setiap  $h = 1, 2, \dots, p - 1$  dan  $k = h + 1, h + 2, \dots, p$  dengan  $p$  menyatakan banyaknya label kelas yang berbeda pada data koordinat II, yaitu  $p = 8$ . Proses pembangunan model JSF dengan skema OAO pada data ini memerlukan waktu selama 559 detik.

Kemudian, model JSF dengan skema OAO melakukan prediksi terhadap data uji menggunakan parameter-parameter yang telah disebutkan di atas. Setelah membandingkan hasil prediksi dengan label kelas yang sebenarnya dari setiap observasi pada data uji, diperoleh akurasi sebesar 96%.

## Skema OAHO

Performa semua model JSF dengan skema OAHO berdasarkan 5 *learning set* pada data koordinat II diringkaskan pada Tabel IV.12. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Tabel IV.12: Hasil validasi silang 5-fold data latih pada data koordinat II dengan skema OAHO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-25}$	0,2	0,1	77,31%	15,74%	1229
$10^{-8}$	0,2	0,1	76,82%	14,72%	2273
$10^{-8}$	0,6	0,1	79,81%	16,66%	1837
$10^{-12}$	0,6	0,1	75,41%	16,05%	1695
$10^{-12}$	0,5	0,1	73,17%	15,00%	1697
$10^{-8}$	0,6	0,05	77,93%	18,04%	1860
$10^{-50}$	0,2	0,01	78,11%	17,83%	1195
$10^{-50}$	1,2	0,01	77,99%	17,07%	1174
$10^{-50}$	0,6	0,01	78,23%	17,50%	1172
$10^{-8}$	1,2	0,01	78,01%	14,84%	1843
$10^{-75}$	1,2	0,01	79,10%	16,98%	1119
$10^{-125}$	1,2	0,01	79,10%	16,98%	1120
$10^{-200}$	1,2	0,01	79,10%	16,98%	1122
$10^{-75}$	0,6	0,01	79,35%	16,85%	1122
$10^{-125}$	0,6	0,01	79,35%	16,85%	1128

Berdasarkan Tabel IV.12, diperoleh hiperparameter terbaik, yaitu  $\theta = (10^{-8}, 0,6, 0,1)$ . Hiperparameter  $\theta$  dapat membangun 5 model JSF dengan rata-rata akurasi 79,81% dan simpangan baku akurasi 16,66%. Artinya, model JSF dari data latih pada data koordinat II dengan skema OAHO memiliki akurasi sebesar  $79,81\% \pm 16,66\%$ .

Selanjutnya, hiperparameter ini digunakan untuk membangun model JSF dari data latih dengan skema OAHO. Kemudian, model JSF ini diuji terhadap data uji untuk memprediksi label kelas dari setiap observasinya dan diperoleh akurasi sebesar 84,5% dengan durasi waktu pembangunan model selama 472 detik.

### IV.3.3 Data Tanaman Iris

Data tanaman iris memiliki nilai-nilai fitur dengan satuan yang sama. Tetapi, Tabel IV.3 menunjukkan bahwa nilai minimum dari masing-masing fitur berbeda, begitu juga dengan nilai maksimumnya. Akibatnya, jangkauan nilai setiap fiturnya berbeda-beda juga. Untuk menyamakan jangkauan tersebut, dilakukan normalisasi minimal-maksimal pada tahap pra pemrosesan data. Karena semua nilai dari fitur-fiturnya berupa bilangan real positif, maka normalisasinya menggunakan 0 sebagai skala minimal dan 1 sebagai skala maksimal. Dengan kata lain, normalisasi minimal-maksimal ini mentransformasi nilai-nilai dari setiap fiturnya menjadi bilangan real pada selang  $[0, 1]$ .

Sama seperti yang dilakukan pada data koordinat II dalam memulai tahap pra pemrosesan, data tanaman iris juga memulainya dengan memisahkan observasi-observasinya menjadi dua kelompok data, yaitu data latih dan data uji. Setelah dilakukan pra pemrosesan data, data latih akan dipartisi ke dalam 5 bagian untuk keperluan skema pemilihan model. Sebagaimana telah dijelaskan di awal subbab ini, 5 partisi data latih ini akan menghasilkan 5 *learning set* dan 5 data validasi. Banyaknya observasi yang dikelompokkan berdasarkan label kelas pada data tanaman iris dan pada bagian-bagiannya telah didaftarkan pada Tabel IV.13.

Tabel IV.13: Kardinalitas label kelas pada data tanaman iris

	<b>Iris- setosa</b>	<b>Iris- virginica</b>	<b>Iris- versicolor</b>
<b>Data asli</b>	50	50	50
<b>Data latih</b>	36	42	42
<b>Sub data latih 1</b>	8	9	9
<b>Sub data latih 2</b>	7	9	9
<b>Sub data latih 3</b>	7	8	8
<b>Sub data latih 4</b>	7	8	8
<b>Sub data latih 5</b>	7	8	8
<b>Data uji</b>	14	8	8

### Skema OAA

Performa semua model JSF dengan skema OAA berdasarkan 5 *learning set* pada data tanaman iris diringkaskan pada Tabel IV.14. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang *5-fold* dengan hiperparameter yang berbeda-beda.

Tabel IV.14: Hasil validasi silang 5-fold data latih pada data tanaman iris dengan skema OAA

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-12}$	0,1	0,01	100,00%	0,00%	31

Melalui percobaan hiperparameter yang pertama, yaitu  $\theta = (10^{-12}, 0,1, 0,01)$ , langsung diperoleh performa model JSF yang sangat memuaskan. Oleh karena itu, proses perulangan untuk proses validasi silang cukup dilakukan satu kali. Dengan demikian, dapat dikatakan bahwa hiperparameter  $\theta$  ini adalah hiperparameter terbaik.

Selanjutnya, akan dibangun model JSF dari data latih pada data tanaman iris dengan menggunakan skema OAA. Hiperparameter  $\theta$  digunakan sebagai *input* untuk membangun model JSF ini. Model JSF yang diperoleh akan diuji pada data uji dari data tanaman iris ini. Proses pengujian ini menghasilkan nilai akurasi model JSF tersebut sebesar 100%. Durasi waktu yang dibutuhkan untuk membangun model JSF dari data latih adalah selama 31 detik.

### Skema OAO

Performa semua model JSF dengan skema OAO berdasarkan 5 *learning set* pada data tanaman iris diringkas pada Tabel IV.15. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Berdasarkan Tabel IV.15, sebagian besar hiperparameternya mengakibatkan model JSF memiliki rata-rata akurasi yang sama, yaitu 99,13%. Karena nilai rata-rata akurasi tersebut merupakan nilai rata-rata akurasi paling besar, maka hiperparameter terbaik adalah hiperparameter yang mempunyai *running time* paling singkat pada proses pembangunan model JSF. Dengan demikian, diperoleh hiperparameter terbaik  $\theta = (10^{-12}, 0,3, 0,01)$ .

Selanjutnya, hiperparameter  $\theta$  digunakan untuk membangun model JSF dari data latih dengan skema klasifikasi OAO dan berdurasi selama 51 detik. Kemudian, model JSF ini diuji terhadap data uji untuk memprediksi label kelas dari setiap

Tabel IV.15: Hasil validasi silang 5-fold data latih pada data tanaman iris dengan skema OAO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-4}$	0,1	0,01	96,05%	5,96%	31
$10^{-4}$	0,2	0,01	97,76%	3,01%	168
$10^{-4}$	0,25	0,01	97,59%	3,13%	236
$10^{-4}$	0,3	0,01	99,13%	1,74%	230
$10^{-8}$	0,1	0,01	98,36%	2,01%	13
$10^{-8}$	0,2	0,01	98,36%	2,01%	13
$10^{-8}$	0,25	0,01	98,36%	2,01%	14
$10^{-8}$	0,3	0,01	98,36%	2,01%	15
$10^{-12}$	0,1	0,01	99,13%	1,74%	54
$10^{-12}$	0,2	0,01	99,13%	1,74%	55
$10^{-12}$	0,25	0,01	99,13%	1,74%	54
$10^{-12}$	0,3	0,01	99,13%	1,74%	51
$10^{-20}$	0,1	0,01	99,13%	1,74%	54
$10^{-20}$	0,2	0,01	99,13%	1,74%	54
$10^{-20}$	0,25	0,01	99,13%	1,74%	55
$10^{-25}$	0,1	0,01	99,13%	1,74%	54
$10^{-25}$	0,2	0,01	99,13%	1,74%	54
$10^{-4}$	0,3	0,05	99,13%	1,74%	203
$10^{-8}$	0,3	0,05	98,36%	2,01%	7
$10^{-12}$	0,3	0,05	99,13%	1,74%	52

observasinya. Label kelas hasil prediksi ini dibandingkan dengan label kelas yang sebenarnya pada data uji. Perbandingan ini menghasilkan skor akurasi, yaitu sebesar 100%.

### Skema OAHO

Performa semua model JSF dengan skema OAHO berdasarkan 5 *learning set* pada data tanaman iris diringkas pada Tabel IV.16. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Berdasarkan Tabel IV.16, sebagian besar hiperparameternya mengakibatkan model JSF memiliki rata-rata akurasi yang sama, yaitu 99,13%. Karena nilai rata-rata akurasi tersebut merupakan nilai rata-rata akurasi paling besar, maka hiperpa-



Tabel IV.16: Hasil validasi silang 5-fold data latih pada data tanaman iris dengan skema OAHO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-4}$	0,1	0,01	96,05%	5,96%	29
$10^{-4}$	0,2	0,01	96,76%	3,01%	75
$10^{-4}$	0,25	0,01	95,25%	5,64%	121
$10^{-4}$	0,3	0,01	97,59%	3,13%	116
$10^{-8}$	0,1	0,01	98,36%	2,01%	14
$10^{-8}$	0,2	0,01	98,36%	2,01%	14
$10^{-8}$	0,25	0,01	98,36%	2,01%	15
$10^{-8}$	0,3	0,01	98,36%	2,01%	15
$10^{-12}$	0,1	0,01	99,13%	1,74%	52
$10^{-12}$	0,2	0,01	99,13%	1,74%	55
$10^{-12}$	0,25	0,01	99,13%	1,74%	55
$10^{-20}$	0,1	0,01	99,13%	1,74%	54
$10^{-25}$	0,1	0,01	99,13%	1,74%	53
$10^{-4}$	0,1	0,1	96,82%	4,51%	7
$10^{-8}$	0,1	0,1	98,36%	2,01%	8
$10^{-12}$	0,1	0,1	99,13%	1,74%	55

parameter terbaik adalah hiperparameter yang mempunyai *running time* paling singkat pada proses pembangunan model JSF. Dengan demikian, diperoleh hiperparameter terbaik  $\theta = (10^{-12}, 0,1, 0,01)$ .

Selanjutnya, hiperparameter  $\theta$  digunakan untuk membangun model JSF dari data latih dengan skema klasifikasi OAO dan berdurasi selama 52 detik. Kemudian, model JSF ini diuji terhadap data uji untuk memprediksi label kelas dari setiap observasinya. Label kelas hasil prediksi ini dibandingkan dengan label kelas yang sebenarnya pada data uji. Perbandingan ini menghasilkan skor akurasi, yaitu sebesar 100%.

#### IV.3.4 Data Evaluasi Mobil

Data evaluasi mobil merupakan data yang semua fiturnya berupa kategori. Oleh karena itu, proses awal pada tahap pra pemrosesan data adalah pengkodean setiap fiturnya menggunakan metode *one hot encoding*. Berdasarkan penjelasan pada

Subbab IV.2.4, banyaknya fitur pada data evaluasi mobil setelah proses pengkodean adalah 21. Meskipun data evaluasi mobil yang akan digunakan ini memiliki dimensi yang besar, tetap tidak akan ada proses ekstraksi fitur menggunakan PCA. Hal ini bertujuan supaya makna dari 21 fitur ini tetap merepresentasikan data evaluasi mobil yang asli. Karena setiap fitur hanya memiliki nilai antara 0 atau 1 dan kardinalitasnya seimbang, maka variansi dari setiap fiturnya cukup kecil.

Dimensi yang besar pada data evaluasi mobil ini mengakibatkan perkalian fungsi gauss yang didefinisikan pada (2.39) akan memiliki nilai yang cukup kecil. Oleh karena itu, nilai-nilai  $\rho$  yang dipilih juga cukup kecil.

Setelah dilakukan pengkodean pada setiap fitur dari data evaluasi mobil, maka dilakukan pemisahan observasi-observasinya menjadi data latih dan data uji. Untuk kebutuhan skema pemilihan model, dilakukan juga partisi terhadap data latih menjadi 5 bagian. Kemudian, 5 bagian ini menghasilkan 5 *learning set* dan 5 data validasi yang saling bersesuaian. Banyaknya observasi untuk setiap label kelas pada data evaluasi mobil dan bagian-bagiannya didaftarkan pada Tabel IV.17.

Tabel IV.17: Kardinalitas label kelas pada data evaluasi mobil

	<i>unaccpted</i>	<i>accepted</i>	<i>good</i>	<i>very good</i>
<b>Data asli</b>	1210	384	69	65
<b>Data latih</b>	962	312	51	57
<b>Sub data latih 1</b>	193	63	11	12
<b>Sub data latih 2</b>	193	63	10	12
<b>Sub data latih 3</b>	192	62	10	11
<b>Sub data latih 4</b>	192	62	10	11
<b>Sub data latih 5</b>	192	62	10	11
<b>Data uji</b>	248	72	18	8

### Skema OAA

Performa semua model JSF dengan skema OAA berdasarkan 5 *learning set* pada data evaluasi mobil diringkaskan pada Tabel IV.18. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang *5-fold* dengan hiperparameter yang berbeda-beda.

Tabel IV.18 menunjukkan bahwa hiperparameter terbaik untuk membangun model JSF dengan skema OAA dari data latih pada data evaluasi mobil adalah  $\theta = (10^{-200}, 0,3, 0,1)$ . Hiperparameter ini menyebabkan model JSF memiliki akurasi

Tabel IV.18: Hasil validasi silang 5-fold data latih pada data evaluasi mobil dengan skema OAA

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-200}$	0,1	0,01	87,05%	1,67%	9114
$10^{-200}$	0,6	0,01	91,02%	0,85%	2440
$10^{-275}$	0,1	0,01	89,65%	1,60%	3721
$10^{-275}$	0,5	0,01	90,95%	0,83%	2474
$10^{-350}$	0,6	0,01	91,02%	0,83%	2449
$10^{-200}$	0,3	0,1	93,56%	1,46%	2622

sebesar  $93,56\% \pm 1,46\%$ . Selanjutnya, hiperparameter ini digunakan untuk membangun model JSF dengan skema OAA dari data latih secara keseluruhan. Durasi waktu yang diperlukan untuk membangun model JSF tersebut adalah selama 624 detik. Model JSF tersebut diuji terhadap data uji dan didapatkan akurasi pada data uji sebesar 95,38%.

### Skema OAO

Performa semua model JSF dengan skema OAO berdasarkan 5 *learning set* pada data evaluasi mobil diringkaskan pada Tabel IV.19. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-fold dengan hiperparameter yang berbeda-beda.

Tabel IV.19: Hasil validasi silang 5-fold data latih pada data evaluasi mobil dengan skema OAO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-200}$	0,1	0,01	25,33%	6,35%	11372
$10^{-275}$	0,6	0,01	93,92%	1,10%	2639
$10^{-350}$	0,3	0,01	57,91%	7,93%	2732
$10^{-350}$	0,5	0,01	93,92%	1,10%	2716
$10^{-350}$	0,6	0,01	93,92%	1,10%	2694

Tabel IV.19 menunjukkan bahwa hiperparameter terbaik untuk membangun model

JSF dengan skema OAO dari data latih pada data evaluasi mobil adalah  $\theta = (10^{-275}, 0,6, 0,01)$ . Hiperparameter ini menyebabkan model JSF memiliki akurasi sebesar  $93,92\% \pm 1,1\%$ . Selanjutnya, hiperparameter ini digunakan untuk membangun model JSF dengan skema OAO dari data latih secara keseluruhan. Durasi waktu yang diperlukan untuk membangun model JSF tersebut adalah selama 559 detik. Model JSF tersebut diuji terhadap data uji dan didapatkan akurasi pada data uji sebesar 94%.

### Skema OAHO

Performa semua model JSF dengan skema OAHO berdasarkan 5 *learning set* pada data evaluasi mobil diringkas pada Tabel IV.20. Masing-masing ringkasan ini diperoleh dengan cara melakukan validasi silang 5-*fold* dengan hiperparameter yang berbeda-beda.

Tabel IV.20: Hasil validasi silang 5-*fold* data latih pada data evaluasi mobil dengan skema OAHO

Hiperparameter			Performa model JSF		
$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Total Running Time (s)
$10^{-125}$	0,3	0,01	90,66%	0,96%	3695
$10^{-200}$	0,1	0,01	88,64%	2,20%	11884
$10^{-200}$	0,3	0,01	90,66%	0,96%	3539
$10^{-200}$	0,6	0,01	90,81%	1,27%	3202
$10^{-275}$	0,1	0,01	89,80%	1,41%	6856
$10^{-275}$	0,3	0,01	90,66%	0,96%	3226
$10^{-275}$	0,6	0,01	90,81%	1,27%	3230
$10^{-350}$	0,1	0,01	90,66%	0,96%	3277
$10^{-350}$	0,2	0,01	90,66%	0,96%	3241
$10^{-350}$	0,6	0,01	90,81%	1,27%	3210

Tabel IV.20 menunjukkan bahwa hiperparameter terbaik untuk membangun model JSF dengan skema OAHO dari data latih pada data evaluasi mobil adalah  $\theta = (10^{-200}, 0,6, 0,01)$ . Hiperparameter ini menyebabkan model JSF memiliki akurasi sebesar  $90,66\% \pm 1,27\%$ . Selanjutnya, hiperparameter ini digunakan untuk membangun model JSF dengan skema OAHO dari data latih secara keseluruhan. Durasi waktu yang diperlukan untuk membangun model JSF tersebut adalah selama

845 detik. Model JSF tersebut diuji terhadap data uji dan didapatkan akurasi pada data uji sebesar 90,46%.

#### IV.3.5 Ringkasan Hasil Uji

Berdasarkan penjelasan pada empat bagian sebelumnya dari subbab ini, performa model JSF pada masing-masing data dengan skema tertentu telah diringkaskan pada Tabel IV.21.

Tabel IV.21: Ringkasan performa model jaringan saraf fuzzy

Data	Skema klasifikasi multikelas	Hiperparameter			Performa model JSF dalam proses validasi Silang			Performa model JSF pada data	
		$\rho$	$s_0$	$\eta$	Rata-rata akurasi	Simpangan baku akurasi	Running Time (s)	Akurasi model JSF pada data uji	Running Time (s)
Data koordinat I	OAA	$10^{-8}$	1	0.01	100.00%	0.00%	2178	100.00%	499
	OAO	$10^{-8}$	1	0.01	100.00%	0.00%	1373	100.00%	331
	OAHO	$10^{-8}$	1	0.01	95.09%	4.98%	2016	98.00%	490
Data koordinat II	OAA	$10^{-350}$	1.2	0.01	89.40%	8.96%	1525	96.00%	378
	OAO	$10^{-25}$	1.2	0.01	85.60%	9.96%	2065	94.00%	559
	OAHO	$10^{-8}$	0.6	0.1	79.81%	16.66%	1837	84.50%	472
Data iris	OAA	$10^{-12}$	0.1	0.01	100.00%	0.00%	31	100.00%	31
	OAO	$10^{-12}$	0.3	0.01	99.13%	1.74%	51	100.00%	21
	OAHO	$10^{-12}$	0.1	0.01	99.13%	1.74%	52	100.00%	20
Data evaluasi mobil	OAA	$10^{-200}$	0.3	0.1	93.56%	1.46%	2622	95.38%	624
	OAO	$10^{-275}$	0.6	0.01	93.92%	1.10%	2639	94.00%	559
	OAHO	$10^{-200}$	0.6	0.01	90.81%	1.27%	3202	90.46%	845

Akurasi yang diambil adalah akurasi model JSF pada data uji. Tabel IV.21 menunjukkan bahwa akurasi dari model JSF dengan skema klasifikasi OAA selalu bernilai paling tinggi. Sementara itu, akurasi dari model JSF dengan skema klasifikasi OAHO selalu bernilai paling rendah. Dengan demikian, diperoleh urutan skema klasifikasi multikelas pada model JSF dari yang paling kuat, yaitu: skema klasifikasi OAA, skema klasifikasi OAO, dan skema klasifikasi OAHO.

## **Bab V    Kesimpulan dan Saran**

### **V.1    Kesimpulan**

Pada tugas akhir ini, sebuah program yang dapat mengimplementasikan model jaringan saraf fuzzy (JSF) dengan tiga skema klasifikasi multikelas berhasil dirancang. Program ini dapat membangun model JSF dari empat data dengan tiga skema klasifikasi multikelas. Selain itu, melalui program ini, sebagian besar akurasi model JSF yang diperoleh dari masing-masing data cukup besar.

Setiap model JSF dari masing-masing data dengan skema klasifikasi multikelas tertentu telah diuji. Hasil uji ini menunjukkan bahwa model JSF dapat digunakan untuk menyelesaikan masalah klasifikasi multikelas dengan berbagai skema. Adapun urutan skema klasifikasi multikelas dari yang paling kuat yang dapat diterapkan untuk membangun model JSF adalah skema satu melawan semua, skema satu melawan satu, dan skema satu melawan orde yang lebih tinggi. Skema klasifikasi dikatakan lebih kuat jika memiliki nilai yang lebih besar pada akurasi dari model JSF yang dihasilkan.

### **V.2    Saran**

Program yang telah dirancang oleh penulis masih harus dikembangkan dan diperbaiki. Pengembangan dan perbaikan yang diperlukan di antaranya adalah pada kemampuan program untuk menampilkan data. Program ini belum bisa menampilkan data yang akan diolah, data latih, data uji, semua *learning set*, dan semua data validasi. Supaya pengguna dapat melakukan tahap pra pemrosesan data menjadi lebih mudah, sebaiknya disediakan tombol untuk melihat statistika deskriptif dari data yang akan diolah. Perbaikan program juga diperlukan pada kemampuan program untuk menampilkan hasil prediksi pada data uji dan data validasi. Program ini baru dapat menampilkan akurasi.

Model jaringan saraf fuzzy memiliki nilai neuron pada lapisan keluaran yang belum

tentu berada pada selang  $[0, 1]$ . Padahal untuk menyelesaikan masalah klasifikasi multikelas, diperlukan nilai neuron pada lapisan keluaran yang berada pada selang  $[0, 1]$ . Ini bertujuan supaya fungsi keputusan dari masing-masing skema klasifikasi multikelas dapat menginterpretasikan nilai-nilai neuron pada lapisan keluaran menjadi label kelas dengan lebih tepat. Untuk mengatasi hal ini, disarankan untuk mengganti fungsi dari setiap bagian konsekuensi pada aturan fuzzy. Fungsi yang disarankan untuk digunakan adalah fungsi logistik.

## DAFTAR PUSTAKA

- Alexandropoulos, S.-A. N., Kotsiantis, S. B. dan Vrahatis, M. N. (2019): Data preprocessing in predictive data mining, *The Knowledge Engineering Review*, **34**, 1–33. DOI: 10.1017/S026988891800036X.
- Alpaydn, E. (2010): *Introduction to Machine Learning*, 2 edn, The MIT Press, Cambridge, Massachusetts.
- Deng, Y., Ren, Z., Kong, Y., Bao, F. dan Dai, Q. (2017): A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification, *IEEE Transactions on Fuzzy Systems*, **25**(4), 1006–1012. DOI: 10.1109/TFUZZ.2016.2574915.
- Dua, D. dan Graff, C. (2019), ‘UCI Machine Learning Repository’.  
**URL:** <http://archive.ics.uci.edu/ml>
- Fullér, R. (2000): *Introduction to Neuro-Fuzzy Systems*, 1 edn, Physica-Verlag, Heidelberg. DOI: 10.1007/978-3-7908-1852-9.
- Ghongade, R. dan Ghatol, D. A. (2007): A multi-class heartbeat classifier employing hybrid fuzzy -neural network, *dalam 2007 International Conference on Intelligent and Advanced Systems*, 18–21. DOI: 10.1109/ICIAS.2007.4658340.
- Golub, G. H. dan Loan, C. F. V. (2013): *Matrix Computations*, 4 edn, Johns Hopkins University Press, Baltimore, Maryland.
- Graupe, D. (2013): *Principles of Artificial Neural Networks*, 3 edn, World Scientific Publishing Co. Pte. Ltd, Singapore. DOI: 10.1142/8868.
- Kotsiantis, S. B., Kanellopoulos, D. dan Pintelas, P. E. (2006): Data Preprocessing for Supervised Learning, *International Journal of Computer and Information Engineering*, **1**(12), 111–117. DOI: 10.1017/S026988891800036X.
- Kraipeerapun, P., Fung, C. C. dan Wong, K. W. (2006): Multiclass classification using neural networks and interval neutrosophic sets, *International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics - Proceedings*, **1**, 123–128.
- Kriesel, D. (2007): *A Brief Introduction to Neural Networks*.  
**URL:** <http://www.dkriesel.com>
- Lee, S.-J. dan Ouyang, C.-S. (2003): A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning, *IEEE Transactions on Fuzzy Systems*, **11**(3), 341 – 353. DOI: 10.1109/TFUZZ.2003.812693.
- Liu, H. dan Cocea, M. (2017): Semi-random partitioning of data into training and test sets in granular computing context, *Granular Computing*, **2**(4), 357–386. DOI: 10.1007/s41066-017-0049-2.
- Mendel, J. M. (2017): *Uncertain Rule-based Fuzzy Systems*, 2 edn, Springer, Cham. DOI: 10.1007/978-3-319-51370-6.



- Mohri, M., Rostamizadeh, A. dan Talwalkar, A. (2018): *Foundations of Machine Learning*, 2 edn, The MIT Press, Cambridge, Massachusetts.
- Nedjah, N. dan de Macedo Mourelle, L. (2005): Introducing You to Fuzziness, *dalam* Nedjah, N. dan de Macedo Mourelle, L., eds, *Fuzzy Systems Engineering*, 1 edn, Vol. 181 of *Studies in Fuzziness and Soft Computing*, Springer, Berlin, Heidelberg, 1–21. DOI: 10.1007/b102051.
- Ou, G. dan Murphey, Y. L. (2007): Multi-class pattern classification using neural networks, *Pattern Recognition*, **40**(1), 4–18. DOI: 10.1016/j.patcog.2006.04.041.
- Potdar, K., Pardawala, T. dan Pai, C. (2017): A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers, *International Journal of Computer Applications*, **175**, 7–9. DOI: 10.5120/ijca2017915495.
- Rangkuti, A. H., Ayuliana, A. dan Fahri, M. (2018): Improving Image Classification using Fuzzy Neural Network and Backtracking Algorithm, *Journal of Telecommunication, Electronic, and Computer Engineering*, **10**(2-3), 123–128.
- Rencher, A. C. (2002): *Methods of Multivariate Analysis*, 2 edn, John Wiley & Sons, New York. DOI: 10.1002/0471271357.
- Rogers, S. dan Girolami, M. (2017): *A First Course in Machine Learning*, 2 edn, CRC Press, Boca Raton.
- Schmidhuber, J. (2015): Deep learning in neural networks: An overview, *Neural Networks*, **61**, 85–117. DOI: 10.1016/j.neunet.2014.09.003.
- Sugeno, M. dan Kang, G. T. (1988): Structure identification of fuzzy model, *Fuzzy Sets and Systems*, **28**(1), 15 – 33. DOI: 10.1016/0165-0114(88)90113-3.
- Takagi, T. dan Sugeno, M. (1985): Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics*, **15**(1), 116 – 132. DOI: 10.1109/TSMC.1985.6313399.
- Wuest, T., Weimer, D., Irgens, C. dan Thoben, K.-D. (2016): Machine learning in manufacturing: advantages, challenges, and applications, *Production & Manufacturing Research*, **4**(1), 23–45.
- Yeh, C.-Y., Jeng, W.-H. dan Lee, S.-J. (2012): Data-Based System Modeling Using a Type-2 Fuzzy Neural Network With a Hybrid Learning Algorithm, *IEEE Transactions on Neural Networks*, **22**(12), 2296 – 2309. DOI: 10.1109/TNN.2011.2170095.
- Zadeh, L. A. (1965): Fuzzy sets, *Information and Control*, **8**(3), 338–353. DOI: 10.1016/S0019-9958(65)90241-X.
- Zadeh, L. A. (1973): Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Transactions on Systems, Man, and Cybernetics*, **3**(1), 28 – 44. DOI: 10.1109/TSMC.1973.5408575.

# LAMPIRAN

## Lampiran A Data Koordinat I

No	x	y	z	Oktan
1	-2	-3	1	Oktan III
2	3	3	4	Oktan I
3	2	4	5	Oktan I
4	-3	3	-3	Oktan VII
5	-1	4	-3	Oktan VII
6	-5	-5	4	Oktan III
7	1	-2	1	Oktan IV
8	5	5	-4	Oktan VIII
9	-4	1	3	Oktan II
10	5	3	-2	Oktan VIII
11	3	1	3	Oktan I
12	1	-1	4	Oktan IV
13	-3	5	-1	Oktan VII
14	1	2	-5	Oktan VIII
15	4	3	-2	Oktan VIII
16	1	3	3	Oktan I
17	1	3	-1	Oktan VIII
18	2	-3	1	Oktan IV
19	-2	1	3	Oktan II
20	-2	5	-2	Oktan VII
21	-5	-5	2	Oktan III
22	5	3	2	Oktan I
23	-2	-5	4	Oktan III
24	4	2	1	Oktan I
25	2	3	1	Oktan I
26	1	-3	1	Oktan IV
27	2	-5	1	Oktan IV
28	-4	2	4	Oktan II
29	5	-4	3	Oktan IV
30	-1	-2	3	Oktan III
31	-3	2	-2	Oktan VII
32	-3	4	-2	Oktan VII
33	-3	3	-4	Oktan VII
34	5	-3	-4	Oktan V
35	1	-4	5	Oktan IV
36	-2	-3	-4	Oktan VI
37	-4	-1	2	Oktan III
38	-1	-5	1	Oktan III
39	1	-5	-5	Oktan V
40	3	2	3	Oktan I

No	x	y	z	Oktan
41	-1	3	4	Oktan II
42	2	-1	1	Oktan IV
43	-3	-5	-2	Oktan VI
44	2	-3	-2	Oktan V
45	4	-1	-5	Oktan V
46	4	-1	5	Oktan IV
47	1	-5	-3	Oktan V
48	-3	-5	4	Oktan III
49	1	-2	-5	Oktan V
50	3	1	5	Oktan I
51	-1	-1	-1	Oktan VI
52	5	2	-3	Oktan VIII
53	-2	-3	4	Oktan III
54	2	-5	4	Oktan IV
55	4	4	-5	Oktan VIII
56	-1	5	2	Oktan II
57	3	-2	1	Oktan IV
58	3	-5	3	Oktan IV
59	-3	-5	1	Oktan III
60	-1	1	-5	Oktan VII
61	2	1	2	Oktan I
62	2	-5	-1	Oktan V
63	1	-1	-1	Oktan V
64	-1	5	1	Oktan II
65	4	-5	-2	Oktan V
66	5	-1	2	Oktan IV
67	4	5	3	Oktan I
68	-1	3	5	Oktan II
69	-3	2	-4	Oktan VII
70	-3	5	2	Oktan II
71	-2	3	4	Oktan II
72	3	-4	-3	Oktan V
73	-3	-1	-5	Oktan VI
74	-1	3	-3	Oktan VII
75	4	-1	-1	Oktan V
76	-4	3	-4	Oktan VII
77	-2	4	-2	Oktan VII
78	-5	4	4	Oktan II
79	-2	3	-3	Oktan VII
80	5	-2	4	Oktan IV

Data lengkap untuk data koordinat I dapat diakses di [http://bit.ly/data\\_koordinat\\_I](http://bit.ly/data_koordinat_I)

## Lampiran B Data Koordinat II

No	x	y	z	Oktan	No	x	y	z	Oktan
1	51	-18	-24	Oktan V	41	-11	39	26	Oktan II
2	-5	19	13	Oktan II	42	31	-15	-2	Oktan V
3	-1	32	-87	Oktan VII	43	-47	-100	-23	Oktan VI
4	12	85	49	Oktan I	44	5	-92	-22	Oktan V
5	7	30	75	Oktan I	45	-34	32	-68	Oktan VII
6	-36	-21	92	Oktan III	46	35	-86	-39	Oktan V
7	4	70	37	Oktan I	47	-64	-6	-86	Oktan VI
8	82	28	-64	Oktan VIII	48	-20	-66	-48	Oktan VI
9	-69	36	68	Oktan II	49	-94	-48	-13	Oktan VI
10	9	-75	-78	Oktan V	50	96	22	-19	Oktan VIII
11	-65	-44	100	Oktan III	51	-36	-47	3	Oktan III
12	85	-24	96	Oktan IV	52	-56	-20	96	Oktan III
13	-10	95	37	Oktan II	53	82	76	-82	Oktan VIII
14	10	-41	-97	Oktan V	54	-15	-20	-78	Oktan VI
15	14	39	50	Oktan I	55	-74	-88	-16	Oktan VI
16	26	-15	72	Oktan IV	56	-6	65	-7	Oktan VII
17	61	-37	-84	Oktan V	57	77	58	19	Oktan I
18	-80	59	24	Oktan II	58	9	1	54	Oktan I
19	-19	-21	41	Oktan III	59	87	68	56	Oktan I
20	28	-56	100	Oktan IV	60	-23	74	43	Oktan II
21	21	-10	-80	Oktan V	61	-28	-84	-47	Oktan VI
22	-87	89	44	Oktan II	62	-15	3	-25	Oktan VII
23	-33	61	-60	Oktan VII	63	56	12	-53	Oktan VIII
24	-71	-4	-32	Oktan VI	64	23	70	76	Oktan I
25	64	85	-80	Oktan VIII	65	-38	-70	-96	Oktan VI
26	-88	-69	46	Oktan III	66	-37	-44	-27	Oktan VI
27	-84	94	96	Oktan II	67	80	15	75	Oktan I
28	-47	64	15	Oktan II	68	86	-7	89	Oktan IV
29	-25	32	-81	Oktan VII	69	-92	-39	-32	Oktan VI
30	67	26	16	Oktan I	70	-11	24	53	Oktan II
31	-14	-62	35	Oktan III	71	-77	-24	-29	Oktan VI
32	-32	22	23	Oktan II	72	48	-80	-38	Oktan V
33	-77	98	44	Oktan II	73	-35	-58	52	Oktan III
34	78	39	74	Oktan I	74	-16	-5	-46	Oktan VI
35	-29	26	98	Oktan II	75	20	-57	63	Oktan IV
36	-7	48	-64	Oktan VII	76	-67	-56	45	Oktan III
37	-75	75	81	Oktan II	77	27	-67	47	Oktan IV
38	-98	47	56	Oktan II	78	18	-42	43	Oktan IV
39	35	-60	96	Oktan IV	79	-50	-33	95	Oktan III
40	9	93	77	Oktan I	80	83	-37	-83	Oktan V

Data lengkap untuk data koordinat II dapat diakses di [http://bit.ly/data\\_koordinat\\_II](http://bit.ly/data_koordinat_II)

## Lampiran C Data Tanaman Iris

No	Panjang sepal (cm)	Lebar sepal (cm)	Panjang kelopak (cm)	Lebar kelopak (cm)	Spesies
1	5,1	3,5	1,4	0,2	Iris-setosa
2	4,9	3	1,4	0,2	Iris-setosa
3	4,7	3,2	1,3	0,2	Iris-setosa
4	4,6	3,1	1,5	0,2	Iris-setosa
5	5	3,6	1,4	0,2	Iris-setosa
6	5,4	3,9	1,7	0,4	Iris-setosa
7	4,6	3,4	1,4	0,3	Iris-setosa
8	5	3,4	1,5	0,2	Iris-setosa
9	4,4	2,9	1,4	0,2	Iris-setosa
10	4,9	3,1	1,5	0,1	Iris-setosa
11	5,4	3,7	1,5	0,2	Iris-setosa
12	4,8	3,4	1,6	0,2	Iris-setosa
51	7	3,2	4,7	1,4	Iris-versicolor
52	6,4	3,2	4,5	1,5	Iris-versicolor
53	6,9	3,1	4,9	1,5	Iris-versicolor
54	5,5	2,3	4	1,3	Iris-versicolor
55	6,5	2,8	4,6	1,5	Iris-versicolor
56	5,7	2,8	4,5	1,3	Iris-versicolor
57	6,3	3,3	4,7	1,6	Iris-versicolor
58	4,9	2,4	3,3	1	Iris-versicolor
59	6,6	2,9	4,6	1,3	Iris-versicolor
60	5,2	2,7	3,9	1,4	Iris-versicolor
61	5	2	3,5	1	Iris-versicolor
62	5,9	3	4,2	1,5	Iris-versicolor
101	6,3	3,3	6	2,5	Iris-virginica
102	5,8	2,7	5,1	1,9	Iris-virginica
103	7,1	3	5,9	2,1	Iris-virginica
104	6,3	2,9	5,6	1,8	Iris-virginica
105	6,5	3	5,8	2,2	Iris-virginica
106	7,6	3	6,6	2,1	Iris-virginica
107	4,9	2,5	4,5	1,7	Iris-virginica
108	7,3	2,9	6,3	1,8	Iris-virginica
109	6,7	2,5	5,8	1,8	Iris-virginica
110	7,2	3,6	6,1	2,5	Iris-virginica
111	6,5	3,2	5,1	2	Iris-virginica
112	6,4	2,7	5,3	1,9	Iris-virginica

Data lengkap untuk data tanaman iris dapat diakses di <https://archive.ics.uci.edu/ml/datasets/iris>

## Lampiran D Data Evaluasi Mobil

No	Buying	Maintenance	Doors	Persons	Lug boot	Safety	Class
1	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>small</i>	<i>low</i>	<i>unacc</i>
2	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>small</i>	<i>med</i>	<i>unacc</i>
3	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>small</i>	<i>high</i>	<i>unacc</i>
4	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
5	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>med</i>	<i>med</i>	<i>unacc</i>
6	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>med</i>	<i>high</i>	<i>unacc</i>
7	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
8	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>big</i>	<i>med</i>	<i>unacc</i>
9	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>two</i>	<i>big</i>	<i>high</i>	<i>unacc</i>
10	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>small</i>	<i>low</i>	<i>unacc</i>
11	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>small</i>	<i>med</i>	<i>unacc</i>
12	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>small</i>	<i>high</i>	<i>unacc</i>
13	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
14	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>med</i>	<i>med</i>	<i>unacc</i>
15	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>med</i>	<i>high</i>	<i>unacc</i>
16	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
17	<i>vhigh</i>	<i>vhigh</i>	<i>two</i>	<i>four</i>	<i>big</i>	<i>med</i>	<i>unacc</i>
255	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>small</i>	<i>high</i>	<i>acc</i>
256	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
257	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>med</i>	<i>med</i>	<i>unacc</i>
258	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>med</i>	<i>high</i>	<i>acc</i>
259	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
260	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>big</i>	<i>med</i>	<i>acc</i>
261	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>four</i>	<i>big</i>	<i>high</i>	<i>acc</i>
262	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>small</i>	<i>low</i>	<i>unacc</i>
263	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>small</i>	<i>med</i>	<i>unacc</i>
264	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>small</i>	<i>high</i>	<i>acc</i>
265	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
266	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>med</i>	<i>med</i>	<i>acc</i>
267	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>med</i>	<i>high</i>	<i>acc</i>
268	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
269	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>big</i>	<i>med</i>	<i>acc</i>
270	<i>vhigh</i>	<i>med</i>	<i>three</i>	<i>more</i>	<i>big</i>	<i>high</i>	<i>acc</i>
271	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>small</i>	<i>low</i>	<i>unacc</i>
272	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>small</i>	<i>med</i>	<i>unacc</i>
273	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>small</i>	<i>high</i>	<i>unacc</i>
274	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
275	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>med</i>	<i>med</i>	<i>unacc</i>
276	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>med</i>	<i>high</i>	<i>unacc</i>
277	<i>vhigh</i>	<i>med</i>	<i>four</i>	<i>two</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
1720	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>small</i>	<i>low</i>	<i>unacc</i>
1721	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>small</i>	<i>med</i>	<i>acc</i>
1722	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>small</i>	<i>high</i>	<i>good</i>
1723	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>med</i>	<i>low</i>	<i>unacc</i>
1724	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>med</i>	<i>med</i>	<i>good</i>
1725	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>med</i>	<i>high</i>	<i>vgood</i>
1726	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>big</i>	<i>low</i>	<i>unacc</i>
1727	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>big</i>	<i>med</i>	<i>good</i>
1728	<i>low</i>	<i>low</i>	<i>5more</i>	<i>more</i>	<i>big</i>	<i>high</i>	<i>vgood</i>

Data lengkap untuk data evaluasi mobil dapat diakses di <https://archive.ics.uci.edu/ml/datasets/car+evaluation>

.....

**Info cetak .....**

Revisi/cetak terakhir: 1 April 2020, pukul 07:09

Nomor halaman: i–xiii, 1–97      Total: 110 halaman

.....