

**PENERAPAN METODE CLASSIFICATION BASED ON FUZZY
ASSOCIATION RULE (CFAR) UNTUK EVALUASI KINERJA
KARYAWAN PT. SIERAD PRODUCE, Tbk**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
Gelar Sarjana dalam bidang Ilmu Komputer



Disusun oleh :
Nike Kartika Sari
NIM. 0810960016

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

MALANG

2013

LEMBAR PERSETUJUAN
PENERAPAN METODE *CLASSIFICATION BASED ON FUZZY*
***ASSOCIATION RULE* (CFAR) UNTUK EVALUASI KINERJA**
KARYAWAN PT. SIERAD PRODUCE, Tbk

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
Gelar Sarjana dalam bidang Ilmu Komputer



Disusun oleh :
Nike Kartika Sari
NIM. 0810960016

Telah diperiksa dan disetujui oleh :

Pembimbing I,

Pembimbing II,

Dian Eka Ratnawati, S.Si., M.Kom.
NIP. 19730619 200212 2 001

Lailil Muflikhah, S.Kom., M.Sc.
NIP. 19741113 200501 2 001

**LEMBAR PENGESAHAN
PENERAPAN METODE *CLASSIFICATION BASED ON FUZZY
ASSOCIATION RULE* (CFAR) UNTUK EVALUASI KINERJA
KARYAWAN PT. SIERAD PRODUCE, Tbk**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
Gelar Sarjana dalam bidang Ilmu Komputer

Disusun oleh :

**NIKE KARTIKA SARI
NIM. 0810960016**

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 4 Januari 2013
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Penguji I,

Penguji II,

**Candra Dewi, S.Kom., M.Sc.
NIP. 19771114 200312 2 001**

**Suprpto, S.T., M.T.
NIP. 19710727 199603 1 001**

Penguji III,

**Budi Darma Setiawan, S.Kom., M.Cs.
NIK. 841015 06 1 1 0090**

**Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer**

**Drs. Marji, M.T.
NIP. 19670801 199203 1 001**

**PERNYATAAN
ORISINALITAS SKRIPSI**

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

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 4 Januari 2013

Mahasiswa,

Nike Kartika Sari

NIM. 0810960016

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode *Classification Based on Fuzzy Association Rule* (CFAR) untuk Evaluasi Kinerja Karyawan PT. Sierad Produce, Tbk”.

Skripsi ini merupakan salah satu syarat untuk memenuhi persyaratan akademis untuk menyelesaikan studi di program Sarjana Komputer Program Studi Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang. Tidak dapat dipungkiri bahwa tidak mungkin penulis dapat menyelesaikan skripsi ini tanpa bantuan dan dukungan dari banyak pihak. Untuk itu, dengan ketulusan dan kerendahan hati penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Dian Eka R, S.Si, M.Kom selaku pembimbing I. Terima kasih atas semua saran dan ilmu yang telah diberikan kepada penulis dalam penyusunan skripsi ini.
2. Lailil Muflikhah, S.Kom, M.Sc selaku pembimbing II. Terima kasih atas bimbingan yang telah diberikan.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Informatika / Ilmu Komputer Universitas Brawijaya.
5. Nanang Yudi Setiawan, ST, selaku dosen pembimbing akademik yang telah banyak memberikan perhatian dan bimbingannya.
6. Seluruh Dosen Bapak dan Ibu Dosen, khususnya Dosen Program Studi Informatika / Ilmu Komputer atas kesediaan membagi ilmunya kepada penulis.

7. Seluruh Civitas Akademi Informatika / Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika / Ilmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Kedua orang tua, ayah ibu dan kakak tersayang terima kasih atas kesabaran dan keikhlasan yang selalu mengiringi langkah hidupku, dukungan berupa materi, nasehat, motivasi, dan kepercayaan untukku menuntut ilmu disini.
9. Duaja Wira Putra tercinta terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.
10. Dian Cahyono, Tirana Noor Fatyanosa, Aries Kurniawan, I Putu Susila Handika, Karina Amalia Daeng Malino, Tebe, Resty, Nisa, Mutya dan sahabat-sahabat di Program Studi Informatika / Ilmu Komputer Universitas Brawijaya yang telah banyak memberikan dukungannya demi kelancaran pelaksanaan penyusunan skripsi ini.
11. Dan semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu terima kasih atas semua bantuan yang telah diberikan.

Penulis sadari bahwa dalam laporan ini kemungkinan masih ada kekurangan, oleh karena itu penulis sangat menghargai saran dan kritik yang sifatnya membangun dari pembaca. Semoga skripsi ini dapat bermanfaat.

Malang, 4 Januari 2013

Mahasiswa,

Nike Kartika Sari

NIM. 0810960016

ABSTRAK

Nike Kartika Sari. 2013. : Penerapan Metode *Classification based on Fuzzy Association Rule (CFAR)* untuk Evaluasi Kinerja Karyawan PT. Sierad Produce, Tbk.

Dosen Pembimbing : Dian Eka R, S.Si., M.Kom dan Lailil Muflikhah, S.Kom., M.Sc.

Evaluasi kinerja karyawan merupakan faktor penting untuk mendapatkan informasi dalam pengembangan Sumber Daya Manusia (SDM) suatu perusahaan. Evaluasi kinerja karyawan dilakukan berdasarkan kriteria-kriteria sesuai dengan kebijakan perusahaan. Setiap karyawan memiliki nilai yang berbeda untuk setiap kriterianya. Dengan mengetahui kemampuan masing-masing karyawan, perusahaan lebih mudah menentukan langkah-langkah berikutnya dalam usaha meningkatkan kinerja karyawan. Hal ini akan menjadi suatu permasalahan jika dikaitkan dengan keputusan dalam pengembangan SDM selanjutnya.

Penelitian ini mengimplementasikan *Classification based on Fuzzy Association Rule*(CFAR) untuk evaluasi kinerja karyawan. Metode CFAR mampu mengintegrasikan *classification* dan *fuzzy association rule*. Langkah-langkah metode CFAR adalah memetakan atribut-atribut kuantitatif ke dalam himpunan *fuzzy*, kemudian membangkitkan *rule* klasifikasi menggunakan algoritma apriori. *Rule* yang dihasilkan dilatih menggunakan data latih dan akan dimasukkan dalam *classifier* jika *rule* mengklasifikasi data latih dengan benar. *Rule* yang ada dalam *classifier* akan digunakan untuk menentukan kelas dari data uji. Hasil pengujian yang dilakukan menunjukkan tingkat akurasi tertinggi dengan prosentase 89%. Hasil ini diperoleh pada saat *minimum support* 15% dan saat *minimum confidence* 85% pada 300 data latih.

Kata kunci : evaluasi kinerja, *fuzzy*, *classification*, *fuzzy association rule*, algoritma *apriori*, CFAR.

ABSTRACT

Nike Kartika Sari. 2013. : *The Implementation Classification based on Fuzzy Association Rule (CFAR) Method for Employee Performance Evaluation (Case Study : PT. Sierad Produce, Tbk).*

Advisor : Dian Eka R, S.Si, M.Kom and Lailil Muflikhah, S.Kom, M.Sc.

Employee performance evaluation is a fundamental factor to get the information in term of Human Resources (HR) development in company. Employee performance evaluation is based on criteria in accordance with company policy. Each employee has a different value for each criteria. By knowing the capabilities of each employee, it will be easier for company to determine the next steps in effort to improve employee performance. It would be a problem if it is associated with subsequent decisions in human resource development.

This study implements Classification based on Fuzzy Association Rule (CFAR) for the evaluation of employee performance. CFAR method is capable to integrate fuzzy classification association rules. CFAR method is to map quantitative attributes into fuzzy set, and then generate classification rules using apriori algorithm. The results of generate classification rule are trained using trained data and will be put in classifier if the rule classify the data correctly. The rule in the classifier will be used to determine the class of the test data. The results of this research showed the highest level of accuracy with a percentage of 89%. This result was obtained at 15% minimum support and minimum confidence at 85% on 300 trained data.

Key words : Employee performance evaluation, fuzzy, classification, fuzzy association rule, apriori algorithm, CFAR.

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR SOURCECODE	xvi
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
 BAB II TINJAUAN MASALAH	 5
2.1 Manajemen Sumber Daya Manusia	5
2.2 Evaluasi Kinerja.....	5
2.2.1 Pelatihan (<i>training</i>).....	6
2.2.2 Data Evaluasi Kinerja Karyawan PT.Sierad Produce, Tbk.....	6
2.3 <i>Data Mining</i>	6
2.3.1 Pengertian <i>Data Mining</i>	6
2.3.2 Tugas <i>Data Mining</i>	7
2.3.3 Tahapan-tahapan <i>Data Mining</i>	8
2.3.4 Teknik <i>Data Mining</i>	9
2.4 Himpunan <i>Fuzzy</i>	10
2.4.1 Fungsi Keanggotaan	10
2.5 <i>Association Rule</i>	12
2.5.1 Definisi	12
2.5.2 <i>Quantitative Assosiation Rule</i>	14
2.5.3 Algoritma Apriori.....	15
2.5.4 <i>Fuzzy Association Rule (FAR)</i>	16
2.5.5 <i>Fuzzy Normalization</i>	16
2.5.6 Algoritma <i>Fuzzy Association Rule Mining</i>	18

2.5.7 Klasifikasi	19
2.5.8 <i>Associative Classification</i>	20
2.5.9 <i>Classification based on Fuzzy Association Rule</i>	21
2.6 Evaluasi.....	25
BAB III METODOLOGI DAN PERANCANGAN	26
3.1 Data yang Digunakan.....	27
3.2 Deskripsi Umum Sistem	27
3.3 Perancangan Sistem.....	27
3.3.1 Pembentukan Aturan	28
3.3.1.1 Proses Transformasi ke Data <i>Fuzzy</i>	29
3.3.1.2 Normalisasi	31
3.3.1.3 Proses <i>Mining FCAR</i>	32
3.3.1.4 Membangun <i>Classifier</i>	35
3.3.2 Proses Pengujian	36
3.3.2.1 Menghitung <i>Discriminant Function</i>	37
3.4 Perancangan Antarmuka	38
3.5 Perhitungan Manual	41
3.6 Perancangan Uji Coba.....	61
3.6.1 Pengujian jumlah <i>rule</i> yang dibangkitkan	61
3.6.2 Pengujian akurasi klasifikasi	62
BAB IV IMPLEMENTASI DAN PEMBAHASAN	63
4.1 Lingkungan Implementasi	63
4.1.1 Lingkungan Perangkat Keras	63
4.1.2 Lingkungan Perangkat Lunak.....	63
4.2 Implementasi Program	63
4.2.1 Implementasi Pembentukan <i>Rule</i>	73
4.2.1.1 Implementasi <i>Load Data</i>	73
4.2.1.2 Implementasi transformasi ke data <i>fuzzy</i>	74
4.2.1.3 Implementasi Perhitungan Normalisasi	75
4.2.1.4 Implementasi pembangkitan <i>Rule</i>	76
4.2.1.5 Implementasi <i>Classifier</i>	79
4.2.2 Implementasi Pengujian	81
4.2.2.1 Implementasi Proses Klasifikasi.....	81
4.3 Implementasi Antarmuka	82
4.3.1 Implementasi Antarmuka Pengujian	82
4.3.2 Implementasi Antarmuka Pembentukan Aturan	84

xi

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Tahapan Proses KDD	9
Gambar 2.2 Representasi Linear Naik	11
Gambar 2.3 Representasi Linear Turun	11
Gambar 2.4 Representasi Trapesium	12
Gambar 2.5 Langkah-langkah <i>Associative Classification</i>	21
Gambar 2.6 Algoritma FCAR	22
Gambar 2.7 Algoritma <i>Classifier</i>	24
Gambar 3.1 Langkah-langkah Penelitian	26
Gambar 3.2 <i>Flowchart</i> Proses Pembentukan Aturan.....	28
Gambar 3.3 <i>Flowchart</i> Proses Transformasi ke Data <i>Fuzzy</i>	29
Gambar 3.4 Himpunan <i>Fuzzy</i> Semua Atribut	30
Gambar 3.5 <i>Flowchart</i> Proses Normalisasi	31
Gambar 3.6 <i>Flowchart Mining FCAR</i>	32
Gambar 3.7 <i>Flowchart</i> Proses Membangkitkan <i>Frequent 1-itemset L1</i>	34
Gambar 3.8 <i>Flowchart</i> Proses Pembangkitan <i>Rule</i>	35
Gambar 3.9 <i>Flowchart</i> Membangun <i>Classifier</i>	36
Gambar 3.10 <i>Flowchart</i> Proses Pengujian	37
Gambar 3.11 <i>Flowchart</i> Menghitung <i>Discriminant Function</i>	38
Gambar 3.12 Antarmuka Pengujian Tunggal	39
Gambar 3.13 Antarmuka Pengujian Data Banyak.....	40
Gambar 3.14 Antarmuka Pembentukan <i>Rule</i>	40
Gambar 4.1 Antarmuka Proses Pengujian Data Tunggal	83
Gambar 4.2 Antarmuka Proses Pengujian Data Banyak.....	84
Gambar 4.3 Antarmuka <i>Panel Import Data</i> pada Proses Pembentukan Aturan	85
Gambar 4.4 Antarmuka <i>Panel View Data</i> pada Proses Pembentukan Aturan	86
Gambar 4.5 Antarmuka <i>Panel itemsets</i> pada Proses Pembentukan Aturan	87
Gambar 4.6 Antarmuka <i>Panel FCAR selected</i> Pada Proses Pembentukan Aturan	88
Gambar 4.7 Antarmuka <i>Panel Classifier</i> pada Proses Pembentukan Aturan	89
Gambar 4.8 Antarmuka <i>Panel Export Rule</i> pada Proses Pembentukan Aturan	90
Gambar 4.9 Grafik Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Jumlah <i>Rule</i> Dalam <i>Classifier</i> pada 100 Record	93
Gambar 4.10 Grafik Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Jumlah <i>Rule</i> Dalam <i>Classifier</i> pada 200 Record	95
Gambar 4.11 Grafik Pengaruh <i>Minimum support</i> dan <i>Minimum confidence</i> Terhadap Jumlah <i>Rule</i> Dalam <i>Classifier</i> pada Semua Data	97
Gambar 4.12 Grafik Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Akurasi (100 data latih)	99
Gambar 4.13 Grafik Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Akurasi (200 data latih)	100

Gambar 4.14 Grafik Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Akurasi (300 data latih)	102
Gambar 4.15 Grafik Pengaruh Akurasi Terhadap Jumlah <i>Rule</i> yang Terbentuk dalam <i>Classifier</i>	105



DAFTAR TABEL

	Halaman
Tabel 2.1 Contoh pemetaan atribut kuantitatif ke biner.....	14
Tabel 2.2 Tanpa <i>Fuzzy Normalization</i>	17
Tabel 2.3 Setelah Normalisasi	17
Tabel 3.1 Sampel Data Kinerja Karyawan PT.Sierad Produce.....	42
Tabel 3.2 Transformasi ke Data <i>Fuzzy</i> Atribut Kooperatif.....	42
Tabel 3.3 Transformasi ke Data <i>Fuzzy</i> Atribut Kehadiran dan Ketepatan Waktu	42
Tabel 3.4 Transformasi ke Data <i>Fuzzy</i> Atribut Inisiatif.....	43
Tabel 3.5 Transformasi ke Data <i>Fuzzy</i> Atribut Keandalan.....	43
Tabel 3.6 Transformasi ke Data <i>Fuzzy</i> Atribut Sikap.....	43
Tabel 3.7 Transformasi ke Data <i>Fuzzy</i> Atribut <i>Judgement</i>	44
Tabel 3.8 Transformasi ke Data <i>Fuzzy</i> Atribut Komunikasi	44
Tabel 3.9 Transformasi ke Data <i>Fuzzy</i> Atribut Hubungan antar Manusia.....	44
Tabel 3.10 Transformasi ke Data <i>Fuzzy</i> Atribut Keahlian Profesional	45
Tabel 3.11 Normalisasi dan perhitungan kardinalitas skalar atribut Kooperatif.	46
Tabel 3.12 Normalisasi dan perhitungan kardinalitas skalar atribut Kehadiran dan Ketepatan Waktu.....	47
Tabel 3.13 Normalisasi dan perhitungan kardinalitas skalar atribut Inisiatif.	47
Tabel 3.14 Normalisasi dan perhitungan kardinalitas skalar atribut Keandalan.....	47
Tabel 3.15 Normalisasi dan perhitungan kardinalitas skalar atribut Sikap	48
Tabel 3.16 Normalisasi dan perhitungan kardinalitas skalar atribut <i>Judgement</i>	48
Tabel 3.17 Normalisasi dan perhitungan kardinalitas skalar atribut Komunikasi.	48
Tabel 3.18 Normalisasi dan perhitungan kardinalitas skalar atribut Hubungan antar Manusia.....	49
Tabel 3.19 Normalisasi dan perhitungan kardinalitas skalar atribut Keahlian Profesional.	49
Tabel 3.20 <i>Frequent 1-itemset</i> (L1)	50
Tabel 3.21 Perhitungan <i>support</i> FCAR_CAND ₂	50
Tabel 3.22 Kandidat <i>2-itemset</i> dan Nilai <i>Support</i> nya.....	51
Tabel 3.23 <i>Frequent 2-itemset</i> (L2).....	52
Tabel 3.24 Kandidat <i>3-itemset</i> dan Nilai <i>Support</i> -nya.....	52
Tabel 3.25 <i>Frequent 3-itemset</i> (L3).....	53
Tabel 3.26 Kandidat <i>4-itemset</i> dan Nilai <i>Support</i> -nya.....	53
Tabel 3.27 <i>Frequent 4-itemset</i> (L4).....	54
Tabel 3.28 Kandidat <i>5-itemset</i> dan Nilai <i>Support</i> -nya.....	54
Tabel 3.29 FCAR dan <i>Confidence</i> Semua <i>k-itemset</i>	55
Tabel 3.30 FCAR yang Memenuhi <i>Minimum Confidence</i>	56
Tabel 3.31 Perhitungan <i>DF</i> Data ke-1	58
Tabel 3.32 Hasil Pengurutan <i>DF</i> Data ke-1.....	58
Tabel 3.33 Hasil Pelatihan <i>Rule</i>	59

Tabel 3.34 Hasil <i>Classifier</i>	59
Tabel 3.35 Tabel Uji Pengaruh <i>Minimum support</i> dan <i>Minimum confidence</i> Terhadap Jumlah <i>Rule</i> yang Dihasilkan	62
Tabel 3.36 Tabel uji Pengaruh <i>Minimum Support</i> dan <i>Minimum Confidence</i> Terhadap Akurasi.....	62
Tabel 4.1 Kelas-kelas yang Dibangun	64
Tabel 4.2 Kelas <code>ExcelImport.java</code>	65
Tabel 4.3 Kelas <code>ItemViewer.java</code>	66
Tabel 4.4 Kelas <code>Table.java</code>	66
Tabel 4.5 Kelas <code>Chooser.java</code>	66
Tabel 4.6 Kelas <code>Filter.java</code>	67
Tabel 4.7 Kelas <code>XMLProcessor.java</code>	67
Tabel 4.8 Kelas <code>CFAR.java</code>	68
Tabel 4.9 Kelas <code>MainFrame.java</code>	68
Tabel 4.10 Kelas <code>Tester.java</code>	69
Tabel 4.11 Kelas <code>Norm.java</code>	70
Tabel 4.12 Kelas <code>Term.java</code>	70
Tabel 4.13 Kelas <code>Attribute.java</code>	70
Tabel 4.14 Kelas <code>HashMap2D.java</code>	71
Tabel 4.15 Kelas <code>Itemset.java</code>	71
Tabel 4.16 Kelas <code>KeySet.java</code>	72
Tabel 4.17 Kelas <code>KeySetList.java</code>	72
Tabel 4.18 Kelas <code>ValueComparator.java</code>	73
Tabel 4.19 Rata-rata Jumlah <i>Rule</i> yang Terbentuk (100 record)	91
Tabel 4.20 Rata-rata Jumlah <i>Rule</i> Dalam <i>Classifier</i> (100 record)	92
Tabel 4.21 Rata-rata Jumlah <i>Rule</i> yang Terbentuk (200 record)	94
Tabel 4.22 Rata-rata Jumlah <i>Rule</i> Dalam <i>Classifier</i> (200 record)	95
Tabel 4.23 Pengujian Jumlah <i>Rule</i> yang Terbentuk (300 record).....	96
Tabel 4.24 Jumlah <i>Rule</i> Dalam <i>Classifier</i> (300 record).....	97
Tabel 4.25 Rata-rata Pengujian Akurasi (100 data latih).....	98
Tabel 4.26 Rata-rata Pengujian Akurasi (200 data latih).....	100
Tabel 4.27 Hasil Pengujian Akurasi (300 data latih)	101

DAFTAR SOURCE CODE

	Halaman
Source Code 4.1 Load Data	73
Source Code 4.2 Transformasi ke Data Fuzzy	74
Source Code 4.3 Perhitungan Normalisasi	75
Source Code 4.4 Pembentukan Kandidat <i>Itemset</i>	76
Source Code 4.5 Pembangkitan <i>Frequent 1-itemsets</i>	77
Source Code 4.6 Pembangkitan <i>Frequent k-itemsets</i>	78
Source Code 4.7 Membuat Rule	79
Source Code 4.8 Perhitungan DF	79
Source Code 4.9 Proses <i>Classifier</i>	81
Source Code 4.10 Proses Klasifikasi	82



BAB I

PENDAHULUAN

1.1 Latar Belakang

Evaluasi kinerja karyawan pada suatu perusahaan dilakukan dengan penilaian kinerja masing-masing karyawan. Penilaian kinerja karyawan merupakan faktor penting untuk mendapatkan informasi dalam pengembangan Sumber Daya Manusia (SDM) perusahaan tersebut. Berhasilnya suatu perusahaan sangat dipengaruhi oleh SDM yang berkualitas. Dengan mengetahui kemampuan masing-masing karyawan, perusahaan lebih mudah menentukan langkah-langkah berikutnya dalam usaha meningkatkan kinerja karyawan. Dalam penilaian kinerja karyawan PT. Sierad Produce,Tbk terdapat beberapa kriteria yaitu kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antar manusia, dan keahlian profesional. Kriteria-kriteria tersebut cukup sulit apabila dinyatakan secara tegas. Seorang karyawan yang bekerja dengan disiplin tentunya akan mendapatkan penilaian yang lebih tinggi dari karyawan yang tidak disiplin. Namun, bisa pula karyawan yang mempunyai keahlian komputer akan mempunyai nilai lebih besar bila dibanding dengan karyawan yang tidak punya keahlian komputer. Nilai dari perbandingan hal-hal diatas bersifat tidak tegas atau kabur (*fuzzy*). Data penilaian kinerja karyawan terdiri dari atribut-atribut dengan nilai kuantitatif. Banyaknya variabel dan penentuan parameter merupakan permasalahan utama yang memungkinkan terjadinya kesalahan dalam evaluasi kinerja karyawan suatu perusahaan. Salah satu metode yang dapat digunakan untuk mengklasifikasi evaluasi kinerja karyawan adalah dengan *associative classification*.

Associative classification merupakan salah satu teknik data mining yang mengintegrasikan *classification* dan *association rule*. Tugas penting *association rule* dalam *association classification* adalah menemukan sejumlah aturan yang mengacu pada sebuah atribut (kelas) dan memilih aturan terbaik untuk membentuk suatu sistem klasifikasi. Dalam membentuk sistem klasifikasi, hasil penelitian [LU-07] mengusulkan *classification based on association* (CBA) yang

diketahui bahwa integrasi klasifikasi dengan *association rule* menghasilkan sejumlah aturan kelas dan dapat melakukan klasifikasi yang lebih akurat daripada metode klasifikasi seperti C.45 dan *classification association rules* (CAR). Selain itu, banyak aturan yang ditemukan oleh metode *associative classification* tidak dapat ditemukan pada metode CAR.

Metode *associativeclassification* seperti CBA mempartisi atribut kuantitatif ke dalam beberapa interval dan memetakannya menjadi nilai biner agar sesuai dengan bentuk *association rule*. Partisi tersebut mengakibatkan masalah batas tegas karena nilai-nilai tertentu yang berada di dekat batas interval mungkin dibuang atau diabaikan. Diantara teknik partisi untuk *association rule*, himpunan *fuzzy* dinilai sesuai untuk mengatasi masalah batas tegas. Menurut [KUO-98], himpunan *fuzzy* memberikan transisi yang “halus” antara anggota dan non-anggota dalam himpunan. Dalam permasalahan yang terdiri dari atribut-atribut dengan nilai kuantitatif, [CHE-08] mengusulkan metode *classification based on fuzzy association rule* (CFAR) karena lebih efektif dan memiliki akurasi lebih baik dibandingkan dengan metode CBA. Himpunan *fuzzy* dalam CFAR digunakan sebagai partisi atribut kuantitatif. Langkah-langkah metode CFAR adalah memetakan atribut-atribut kuantitatif ke dalam himpunan *fuzzy* kemudian membangkitkan *rule* untuk membentuk sistem klasifikasi dengan akurasi yang baik. Dari penelitian [CHE-08] diketahui bahwa akurasi metode CFAR untuk *dataset* penyakit jantung sekitar 85,55% dan *dataset wine* sekitar 93,33% lebih baik dibandingkan dengan CBA dengan akurasi 83,33% untuk *dataset* penyakit jantung dan 86,57% untuk *dataset wine*. Selain itu, [LU-04] juga menggunakan metode *fuzzy association rule* untuk klasifikasi dan menunjukkan metode tersebut memiliki akurasi lebih baik dibandingkan dengan CBA. Pada skripsi ini mencoba mengimplementasikan metode CFAR untuk evaluasi kinerja karyawan.

Untuk meningkatkan keakuratan dalam evaluasi kinerja karyawan diterapkan metode CFAR. Maka dalam skripsi ini diambil judul **”Penerapan Metode Classification Based on Fuzzy Association Rule (CFAR) untuk Evaluasi Kinerja Karyawan PT. Sierad Produce, Tbk”**.

1.2 Rumusan Masalah

Rumusan masalah dalam penulisan skripsi ini adalah:

1. Bagaimana menerapkan *Classification based on Fuzzy Association Rule Mining* (CFAR) dalam evaluasi kinerja karyawan.
2. Bagaimana pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule* yang terbentuk dan akurasi metode CFAR dalam evaluasi kinerja karyawan.

1.3 Batasan Masalah

Batasan masalah dalam penulisan skripsi ini adalah:

1. Data yang dijadikan pelatihan dan pengujian pada skripsi ini adalah data penilaian kinerja karyawan berdasarkan *behavior* karyawan PT. Sierad Produce, Tbk semester 1 tahun 2011.
2. Kategori yang dihasilkan dari proses klasifikasi adalah evaluasi kinerja karyawan yang di *training* atau tidak.
3. Setiap atribut kecuali atribut kelas pada data memiliki 5 variabel linguistik.

1.4 Tujuan

Tujuan yang ingin dicapai dari skripsi ini adalah:

1. Menerapkan *Classification based on Fuzzy Association Rule Mining* (CFAR) dalam evaluasi kinerja karyawan.
2. Menghitung pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule* yang terbentuk dan akurasi metode CFAR dalam penilaian evaluasi kinerja karyawan.

1.5 Manfaat

Manfaat yang diperoleh dari penulisan skripsi yaitu hasil penerapan *Classification based on Fuzzy Association Rule* diharapkan dapat membantu pihak perusahaan sebagai pertimbangan dalam menilai kinerja karyawan dan menentukan langkah lanjutan yang harus dilakukan.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan skripsi.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori – teori yang berhubungan dengan *Classification based on Fuzzy Association Rule*.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

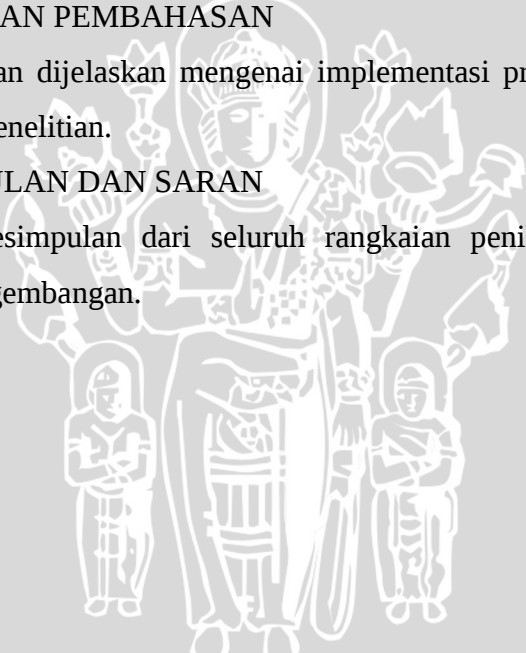
Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam membangun sistem klasifikasi.

4. BAB IV HASIL DAN PEMBAHASAN

Dalam bab ini akan dijelaskan mengenai implementasi program, pengujian dan analisa hasil penelitian.

5. BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangan.



BAB II

TINJAUAN PUSTAKA

2.1 Manajemen Sumber Daya Manusia

Sumber Daya Manusia (SDM) dalam sebuah perusahaan sangat berpengaruh terhadap berhasil tidaknya perusahaan tersebut. SDM dalam dunia bisnis adalah karyawan atau orang yang bekerja dalam suatu organisasi. SDM pada suatu perusahaan perlu dikelola secara profesional agar terwujud keseimbangan antara kebutuhan pegawai dengan tuntutan dan kemampuan organisasi perusahaan. Untuk meningkatkan efektivitas sumber daya manusia diperlukan Manajemen Sumber Daya Manusia (MSDM). Fungsi MSDM dalam suatu perusahaan adalah sebagai bahan-bahan perencanaan, penyusunan karyawan, pengembangan karyawan, pengelola karir, evaluasi kinerja, kompensasi karyawan dan hubungan ketenagakejaan yang baik sehingga dapat berfungsi secara produktif untuk tercapainya tujuan perusahaan [MAN-05].

2.2 Evaluasi Kinerja

Evaluasi kinerja merupakan proses penilaian kepada seseorang atau unit-unit kerja dalam satu perusahaan terhadap pelaksanaan tugas yg telah ditetapkan terlebih dahulu. Evaluasi kinerja dapat digunakan sebagai bahan pertimbangan dalam memberikan imbalan atau penghargaan kepada karyawan. Tujuan utama evaluasi kinerja ini adalah mengetahui tingkat pencapaian tujuan perusahaan, jika terjadi penyimpangan supaya segera diperbaiki sehingga tujuan tercapai.

Evaluasi kinerja dalam lingkup manajemen merupakan alat pengukur kontribusi seorang pekerja atau karyawan pada suatu organisasi. Bagi seorang pemimpin, melakukan evaluasi kinerja karyawan merupakan suatu tugas yang sulit dan tidak disukai. Tidak banyak orang yang suka memberikan evaluasi, karena evaluasi seakan-akan dianggap mencari kesalahan orang lain [SWA-11].

2.2.1Pelatihan (*training*)

Pelatihan adalah proses pendidikan jangka pendek yang bertujuan untuk memberikan pengetahuan baru dengan tujuan meningkatkan kinerja karyawan pada suatu perusahaan [MAN-05]. Fungsi utama pelatihan adalah proses merubah peserta yang tidak terlatih menjadi seorang yang terlatih. Pelatihan terhadap karyawan merupakan salah satu bentuk usaha suatu perusahaan dalam meningkatkan kinerja karyawan.

2.2.2Data Evaluasi Kinerja Karyawan PT. Sierad Produce, Tbk

DataEvaluasi Kinerja Karyawan PT. Sierad Produce, Tbk merupakan data hasil penilaian kinerja karyawan yang dilakukan oleh departemen *Human Resources Development* (HRD) PT Sierad Produce,Tbk. Data ini terdiri dari 9 atribut, yaitu kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antar manusia, dan keahlian profesional.

Atribut kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antar manusia, dan keahlian professional memiliki *linguistic term* yang sama dan dibagi menjadi 5 kelompok yaitu sangat kurang (nilai kurang dari 30), kurang (nilai antara 30 sampai dengan 50), cukup (nilai antara 50 sampai dengan 70), baik (nilai antara 70 sampai dengan 90), dan sangat baik (nilai lebih dari 90). Batasan-batasan nilai tersebut didapat dari HRD PT. Sierad Produce, Tbk.

2.3 Data Mining

2.3.1 Pengertian Data Mining

Data *mining* adalah serangkaian proses penemuan informasi baru dengan mencari pola atau aturan dari kumpulan data yang berukuran besar. Suatu pola yang menarik adalah pola yang tidak diketahui sebelumnya dan berguna. Informasi dari data *mining* dapat digunakan sebagai perbaikan pengambilan keputusan di masa mendatang berdasarkan pola yang ada [SAN-07].

Data *mining* merupakan teknologi yang menggabungkan metoda analisis tradisional dengan algoritma yang canggih untuk memproses data dengan *volume* besar [KUS-07]. Penemuan pengetahuan yang tidak terduga dan mengolah data

dalam jumlah besar dapat dilakukan dengan data *mining*. Menurut [HAN-01], data *mining* adalah solusi yang dapat menemukan informasi yang tersembunyi berupa pola dan aturan dari sekumpulan data yang besar sehingga mudah dipahami berguna bagi pemilik data [HAN-01].

2.3.2 Tugas Data Mining

Fungsionalitas dari data *mining* dan pola pencarian yang dapat ditemukan dalam data *mining* sebagai berikut [HAN-01] :

1. Deskripsi Kelas

Data dapat diasosiasikan dengan kelas, digunakan untuk menggambarkan kelas secara individual dan konsep secara tepat.

2. Analisis Asosiasi

Analisis Asosiasi merupakan penemuan *Association Rule* yang menunjukkan nilai kondisi dari atribut yang terjadi secara bersama-sama dan terus-menerus dalam membentuk sekumpulan data.

3. Klasifikasi dan Prediksi

Klasifikasi adalah proses untuk menemukan sekumpulan model yang menggambarkan dan membedakan kelas data yang bertujuan agar model tersebut dapat digunakan untuk memprediksi kelas suatu objek yang belum diketahui kelasnya. Klasifikasi dapat digunakan untuk memprediksi data nominal maupun data diskrit.

4. Analisis Cluster

Analisis *cluster* berbeda dengan klasifikasi, *clustering* menganalisis objek data dengan kelas yang belum diketahui, sedangkan klasifikasi digunakan untuk menganalisis objek data dari kelas yang telah diketahui.

5. Analisis Outlier

Outlier dapat dideteksi menggunakan tes yang bersifat statistik yang mengambil sebuah distribusi dan probabilitas model untuk data atau menggunakan ukuran jarak, dimana objek yang jaraknya jauh dari *cluster* yang lain dianggap sebagai *outlier*.

6. Analisis *Evolution*

Data analisis *evolution* ini mencari model atau objek yang memiliki kebiasaan berubah setiap waktu (*time-series*). Analisis *Evolution* dapat berupa *characterization, descrimition, association, classification, dan clustering*.

Fungsionalitas data *mining* digunakan untuk menentukan jenis pola yang dapat ditemukan dalam tugas data mining. Pada umumnya terdapat dua kategori dalam tugas data *mining*, yaitu deskriptif dan prediktif. Deskriptif *mining* mengkarakterisasi sifat-sifat umum dari data dalam *database*. Prediktif *mining* mengambil kesimpulan pada data yang ada, dalam rangka untuk membuat prediksi [HAN-01].

2.3.3 Tahapan-tahapan Data Mining

Data mining dan *Knowledge Discovery in Database* (KDD) merupakan istilah yang memiliki konsep berbeda, akan tetapi saling berkaitan karena datmin adalah bagian dalam proses *knowledge discovery in database* (KDD). Proses *knowledge discovery in database* (KDD) secara umum adalah sebagai berikut [KUS-07]:

1. Data Selection

Data *selection* adalah tahap seleksi data yang akan digunakan dalam proses data mining dari sejumlah besar data operasional. Hasil dari seleksi data disimpan dalam suatu berkas terpisah dari *database* operasional.

2. *Prepocessing* atau *Cleaning*

Pada tahap ini, dilakukan pembuangan duplikasi data, pemeriksaan data yang tidak konsisten, dan memperbaiki kesalahan pada data, serta memperkaya data yang sudah ada dengan data atau informasi eksternal.

3. Transformation

Tahap *transformation* sangat bergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. Data Mining

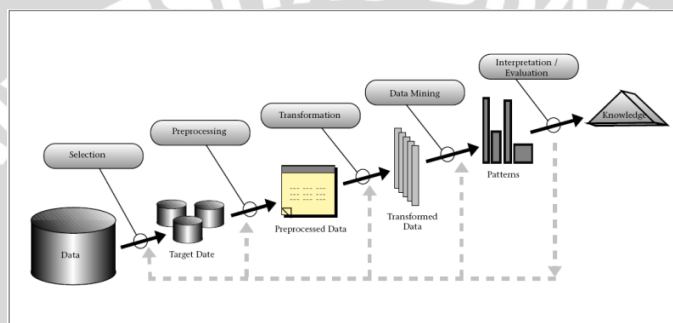
Data *mining* merupakan tahap pencarian pola atau informasi dari data yang terpilih dengan menggunakan metode atau teknik tertentu. Ketepatan metode

atau teknik yang dipilih bergantung pada tujuan dari proses *knowledge discovery in database* (KDD) secara keseluruhan.

5. *Interpretation* atau *Evaluation*

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

Keseluruhan dari proses *Knowledge Discovery in Database* (KDD) diilustrasikan pada gambar 2.1.



Gambar 2.1 Tahapan Proses *Knowledge Discovery in Database*

Sumber : [HAN-01]

2.3.4 Teknik *Data Mining*

Teknik-teknik dalam data *mining* yang paling populer antara lain [PRA-03]:

1. *Association Rule Mining*

Association Rule Mining adalah teknik data *mining* untuk menemukan aturan asosiasi antara suatu kombinasi *item*.

2. *Classification*

Classification merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui sebelumnya. Sebagai contoh, klasifikasi dalam penelitian adalah mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

3. *Clustering*

Proses pengelompokan data tanpa berdasarkan kelas data tertentu. *Clustering* dapat digunakan untuk memberikan label pada kelas data yang belum diketahui.

2.4 Himpunan Fuzzy

Himpunan tegas (*crisp*) A didefinisikan oleh *item-item* yang ada pada himpunan itu. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu *item* x dalam suatu himpunan $A(\mu_A[x])$, memiliki dua kemungkinan, yaitu [KUS-10]:

1. Satu (1) yang berarti bahwa suatu *item* menjadi anggota dalam suatu himpunan.
2. Nol (0) yang berarti bahwa suatu *item* tidak menjadi anggota dalam suatu himpunan.

Nilai keanggotaan *fuzzy* menunjukkan bahwa suatu *item* dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga terdapat nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran suatu *item* tidak hanya benar atau salah. Masih ada nilai-nilai yang terletak antara benar dan salah. Himpunan *fuzzy* memiliki 2 atribut, yaitu [KUS-10] :

1. *Linguistic*, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: muda, parobaya, tua.
2. *Numerics*, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable seperti: 40, 25, 50, dan sebagainya.

2.4.1 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melakukan pendekatan fungsi.

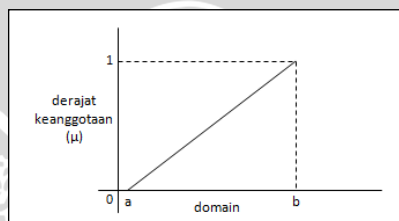
Terdapat beberapa fungsi yang dapat digunakan untuk mendapatkan nilai keanggotaan. Seperti representasi linear, representasi kurva segitiga, representasi kurva trapesium, dan representasi kurva bahu [KUS-10].

Berikut penjelasan tentang masing-masing representasi [KUS-10]:

a. Representasi Linier

Pada representasi linier, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Terdapat 2 keadaan himpunan *fuzzy* yang linier:

1. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Representasi linear naik dapat dilihat pada gambar 2.2.

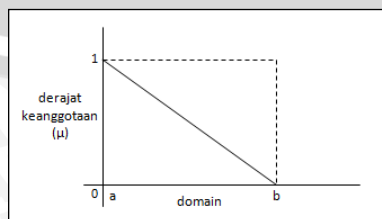


Gambar 2.2 Representasi Linear Naik

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

2. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Representasi linear turun dapat dilihat pada gambar 2.3



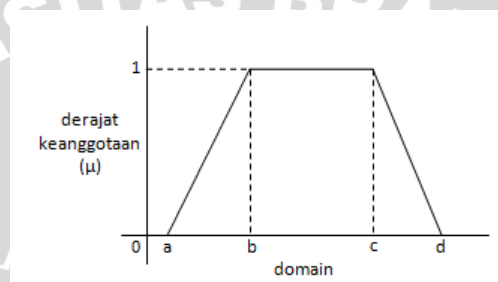
Gambar 2.3 Representasi Linear Turun

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 1; & x \leq a \\ \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.2)$$

b. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja terdapat beberapa titik yang memiliki nilai keanggotaan 1. Representasi kurva trapesium dapat dilihat pada gambar 2.4.



Gambar 2.4 Representasi Kurva Trapesium

Pada kurva trapesium terdapat 4 titik, dimana titik a dan d merupakan nilai domain terendah dengan derajat keanggotaan 1 ($\mu=0$), sedangkan titik b dan c merupakan nilai domain tertinggi dengan derajat keanggotaan 1 ($\mu=1$). Dimana titik a bergerak menuju titik b, yang mana merupakan representasi linier naik. Titik c bergerak menuju titik d, yang mana merupakan representasi linier turun. Untuk mencari derajat keanggotaan dari titik a hingga titik d dapat dirumuskan menggunakan fungsi keanggotaan yang dinyatakan pada aturan 2.3

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c \leq x \leq d \end{cases} \quad (2.3)$$

2.5 Association Rule

2.5.1 Definisi

Association Rule merupakan bentuk dari data mining yang melakukan proses menggali sesuatu yang sangat berharga dari jumlah data yang besar,

dimana sesuatu yang sangat berharga adalah *rule* yang nilai *confidence*-nya lebih besar dari nilai yang telah ditetapkan sebagai batas *minimum*. *Rule* tersebut menggambarkan isi dari basis data yang belum diketahui dan mungkin tidak dapat diungkapkan secara eksplisit [KAN-03].

Dalam data *mining*, tugas *association rule* adalah mencari pola atau aturan yang tersembunyi untuk mengukur hubungan antara dua atribut atau lebih [SHO-09]. *Association Rule* digunakan untuk menemukan hubungan diantara data atau bagaimana suatu kelompok data mempengaruhi suatu keberadaan data yang lain. Metode ini dapat digunakan untuk mengenali pola-pola tertentu dalam basis data yang besar. Dalam *association rule*, suatu kelompok item dinamakan *itemset*.

Association rule meliputi 2 tahap, yaitu [HAN-01]:

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Membangkitkan *association rule* dari *frequent itemset* yang telah dibuat sebelumnya.

Dalam menentukan suatu *association rule*, terdapat suatu *interesting measure* (ukuran kepercayaan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu [BUD-05] :

1. *Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset*.
2. *Confidence* adalah suatu ukuran yang menunjukkan hubungan antara dua atau lebih *item* secara *conditional*.

Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rule*, yaitu untuk membandingkan dengan batasan (*threshold*) yang ditentukan oleh pengguna. Batasan tersebut umumnya terdiri dari *minimum support* dan *confidence* ditunjukkan oleh persamaan 2.4 dan 2.5.

$$\text{Support} = P(A \cup B)$$

$$\text{Confidence} = P(A|B)$$

dimana,

$P(A \cup B)$: probabilitas bahwa sebuah *record* mengandung A dan B

$P(A|B)$: probabilitas kondisional yang menunjukkan probabilitas bahwa sebuah *record* jika mengandung A maka mengandung B juga

$$Support(A) = \frac{\text{jumlah transaksi mengandung A}}{\text{total transaksi}} \quad (2.4)$$

$$Confidence = P(A|B) = \frac{\text{jumlah transaksi mengandung A dan B}}{\text{jumlah transaksi mengandung A}} \quad (2.5)$$

2.5.2 Quantitative Association Rule

Dalam menemukan *association rule*, *quantitative association rule* melakukan partisi domain atribut kuantitatif dengan menggabungkan partisi yang berdekatan dan kemudian mengubah masalah menjadi biner satu [GYE-01]. Dalam proses partisi domain atribut kuantitatif akan dihasilkan interval-interval, dimana interval tersebut dipetakan menjadi 1 dan 0. Nilai kuantitatif yang masuk kedalam interval akan dipetakan menjadi 1 dan sebaliknya.

Tabel 2.1 Contoh Memetaan Atribut Kuantitatif ke Biner

RI D	Age (11..20)	Age (21..30)	Age (31..40)	Income (1000..1800)	Income (2000..2800)	Status Unmarried	Status Married
1	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0
3	0	0	0	1	1	1	0
4	1	0	1	0	0	0	1
5	0	1	1	0	0	0	1
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
19	1	0	0	1	0	1	0
20	0	0	1	0	1	0	1

Dalam menyelesaikan atribut kuantitatif, *quantitative association rule* menimbulkan masalah *sharp boundary* yaitu masalah batas tajam antar interval. Dalam proses *mining*, *quantitative association rule* mengabaikan elemen-elemen di dekat batasan (*boundary*). Sebagai contoh, metode interval mungkin mengklasifikasikan seseorang muda jika usia kurang dari 35 dan tua jika umur lebih besar dari 35. Ini jelas tidak sesuai dengan persepsi manusia tentang muda dan tua. Selain itu, tidak mudah untuk membedakan derajat keanggotaan untuk metode interval. Misalnya, dengan menggunakan metode interval, usia 40 dan usia 80 dari keduanya akan diklasifikasikan menjadi tua. Namun, secara intuitif dapat dikatakan bahwa usia 80 jauh lebih tua dari usia 40 [GYE-01].

Untuk mengatasi masalah *sharp boundary* tersebut, digunakan konsep himpunan *fuzzy*. Dengan menggunakan pemodelan himpunan *fuzzy*, data kuantitatif dapat ditangani tanpa batas partisi tajam disebabkan oleh metode interval karena memberikan transisi yang halus antara keanggotaan dan non-keanggotaan dari suatu himpunan sehingga sedikit elemen-elemen batas yang dikecualikan. Selain itu, penggunaan konsep himpunan *fuzzy* pada *association rule* lebih dimengerti karena istilah linguistik yang berhubungan dengan himpunan *fuzzy* [GYE-01].

2.5.3 Algoritma Apriori

Pendekatan yang umum digunakan untuk menemukan aturan asosiasi adalah algoritma *Apriori* [AGR-94]. Algoritma *apriori* adalah suatu algoritma yang mencari *frequent itemset*. *Itemset* adalah sekumpulan *item*. *Frequent itemset* adalah suatu *itemset* yang memenuhi *min_support*. Suatu *itemset* yang memuat k *item* disebut k -*itemset*. Algoritma ini menggunakan pengetahuan awal yaitu *itemset* mana saja yang paling sering muncul dengan melakukan pendekatan iteratif dimana k -*itemset* digunakan untuk mengeksplorasi $(k+1)$ *itemset*. Pertama-tama cari *frequent 1-itemset* (L_1), gunakan L_1 untuk mencari *frequent 2-itemset* (L_2) dan seterusnya sampai tidak ada lagi *frequent k-itemset* ditemukan. Secara umum algoritma *apriori* terdiri 2 langkah utama, yaitu [HAN-03]:

1. *Join* (Penggabungan)

Untuk menemukan L_k , sebuah himpunan kandidat k -*itemset* dibuat dengan menggabungkan (*join*) L_{k-1} dengan L_{k-1} . Himpunan kandidat dinotasikan dengan C_k . Misal l_1 dan l_2 menjadi *itemset* L_{k-1} . Notasi $l_i[j]$ merujuk item ke- j di l_i (misal, $l_1[k-2]$ menunjuk pada ke-2 terakhir dari l_1).

2. *Prune* (Pemangkasan)

C_k adalah super set dari L_k , anggotanya boleh atau tidak boleh *frequent*, tapi semua *frequent k-itemset* dimasukkan kedalam C_k . *Scan* basis data menunjukkan jumlah masing-masing kandidat di C_k menghasilkan determinasi L_k (misalnya semua kandidat berjumlah tidak kurang dari jumlah *minimum support* adalah *frequent*, maka itu menjadi L_k).

2.5.4 Fuzzy Association Rule (FAR)

Metode untuk menemukan *association rule* dengan menggunakan konsep himpunan *fuzzy* disebut *Fuzzy Association Rule* yang mampu menangani batas tajam antara interval-interval atribut kuantitatif. Dalam *dataset*, setiap *record* dinotasikan dengan $T = \{t_1, t_2, \dots, t_n\}$ dan untuk setiap atribut yang muncul di T dinotasikan dengan $I = \{i_1, i_2, \dots, i_m\}$. I terdiri dari semua *item* yang mungkin pada *dataset*, kombinasi berbeda dari *item* tersebut disebut *itemsets*. Untuk merepresentasikan himpunan *fuzzy* yang berasosiasi dengan i_j kita dapat menggunakan $F_{i_j} = \{f_{ij}^1, f_{ij}^2, \dots, f_{ij}^k\}$ dimana f_{ij}^k merepresentasikan *fuzzy set* ke- k pada F_{i_j} . *Fuzzy association rule* merupakan bentuk :

“IF X is A then Y is B ”,

dimana $X = \{x_1, x_2, \dots, x_p\}$ dan $Y = \{y_1, y_2, \dots, y_q\}$ adalah subset dari atribut *itemset* I , $A = \{f_{x_1}, f_{x_2}, \dots, f_{x_p}\}$ dan $B = \{f_{y_1}, f_{y_2}, \dots, f_{y_q}\}$ mengandung *fuzzy set* yang terkait dengan atribut yang sesuai pada X dan Y . “ X is A ” merupakan *antecedent* dan “ Y is B ” adalah konsekuensi [KUO-98].

Support dan *confidence* juga digunakan untuk mengukur kemenarikan *fuzzy association rule*. *Support* dari FAR didefinisikan seperti pada persamaan 2.6 [LU-03]:

$$Sup(X \Rightarrow Y) = \frac{\sum_{i=1}^n \prod_{j=1}^{p+q} t_i[i_j]}{n} \quad (2.6)$$

dimana $t_i[i_j]$ adalah nilai derajat keanggotaan dari *fuzzy* atribut i_j pada *record* ke- i , n adalah jumlah *record* pada *dataset*. Sedangkan *confidence* didefinisikan seperti pada persamaan 2.7 :

$$Conf(X \Rightarrow Y) = \frac{Sup(X \Rightarrow Y)}{Sup(X)} \quad (2.7)$$

2.5.5 Fuzzy Normalization

Misalnya sebuah transaksi t , nilai atribut ke- j untuk $t[i_j]$, dapat dipetakan menjadi derajat keanggotaan q_k dan kita dapat menggunakan $t[i_j]$ untuk merepresentasikan derajat keanggotaan $t[i_j]$ pada atribut kategori atau kuantitatif

(fuzzy) pada *record* ke- i pada i_j . Jika i_j adalah atribut kategori maka $t_i[i_j] \in \{0,1\}$.

Jika i_j adalah atribut kuantitatif (fuzzy) maka $0 \leq t_i[i_j] \leq 1$ [CHE-08].

Jika i_j adalah atribut kategori maka :

$$\sum_{j=1}^{q_k} t_i[i_j] = 1 \quad (2.8)$$

Jika i_j adalah atribut kuantitatif, persamaan 2.8 tidak selalu sesuai karena pada transaksi yang terjadi sekali, mungkin total nilai derajat keanggotaan lebih dari 1 sehingga harus dilakukan normalisasi. Normalisasi dapat dilakukan dengan cara seperti pada persamaan 2.9 [CHE-08] :

$$t'_i[i_j] = \begin{cases} t_i[i_j] / \sum_{j=1}^{q_k} t_i[i_j] & \text{jika } i_j \text{ adalah fuzzy} \\ t_i[i_j] & \text{jika } i_j \text{ kategori} \end{cases} \quad (2.9)$$

Sebagai contoh, misalnya $I = \{\text{umur, bald}\}$ yang mana umur adalah atribut kuantitatif dengan tiga *fuzzy set* {umur_muda, umur_paruhsbaya, umur_tua} dan *bald* adalah atribut kategori dengan domain {0,1}. Diberikan sebuah transaksi $t = \{40, 1\}$ maka t' dengan nilai baru dapat dilihat pada tabel 2.2.

Tabel 2.2 Tanpa Fuzzy Normalization

ID	muda	paruhbaya	tua	bald
ID1	0.7	0.6	0.1	1

Tanpa normalisasi (tabel 2.2), untuk *itemset* {bald, muda}, $\text{support}(\{\text{bald, muda}\}) = 0.7$, $\text{support}(\{\text{bald, paruhbaya}\}) = 0.6$ dan $\text{support}(\{\text{bald, tua}\}) = 0.1$. Artinya, transaksi $t = \{40, 1\}$ pada (bald, umur) sesuai dengan 3 *itemset* dalam t' dengan jumlah total *support* sebesar $0.7 + 0.6 + 0.1 = 1.4$. Maka perlu dilakukan normalisasi agar jumlah totalnya menjadi 1 yang hasilnya dapat dilihat pada tabel 2.3.

Tabel 2.3 Setelah Normalisasi

ID	muda	paruhbaya	tua	bald
ID1	0.5	0.429	0.071	1

2.5.6 Algoritma Fuzzy Association Rule Mining

Algoritma *fuzzy association rule mining* merupakan pengembangan dari algoritma *apriori* oleh [AGR-94]. Algoritma ini pertama mengubah atau mentransformasi nilai kuantitatif menjadi nilai derajat keanggotaan *fuzzy* berdasarkan ketentuan linguistik yang didefinisikan pada domain atribut masing-masing. Atribut-atribut yang telah ditransformasi tersebut disebut sebagai atribut *fuzzy*. Notasi yang digunakan dalam algoritma ini yaitu :

1. $T = \{T_i\} (i = 1, \dots, n)$ adalah set *record* dimana T_i menyatakan *record* ke- i ;
2. x_{ij} adalah nilai atribut I_j pada *record* ke- i ;
3. $\min_sup$ adalah nilai *minimumsupport* yang telah ditentukan;
4. $\min_conf$ adalah nilai *minimumconfidence* yang telah ditentukan;
5. L_k adalah *frequentk-itemset* dengan nilai *fuzzy support* melebihi $\min_sup$;
6. C_k adalah kandidat *itemset* (variabel linguistik)

Sedangkan algoritma *fuzzy association rule mining* adalah sebagai berikut [JIN-04] :

1. Nilai *crisp* dipetakan menjadi nilai keanggotaan *fuzzy* untuk *record* T_i pada linguistik ke- k untuk atribut I_j : $\mu_{a_j^k}^l(x_{ij}) (k = 1, \dots, l)$ dimana l merupakan jumlah linguistik pada atribut I_j .
2. Linguistik ke- k untuk setiap atribut I_j dihitung kardinalitas skalarnya. Kardinalitas skalar himpunan *fuzzy* A merupakan penjumlahan dari nilai keanggotaan dari semua elemen dalam A .

$$Card(a_j^k) = \sum_{i=1}^n \mu_{a_j^k}^l(x_{ij}) ; \text{dimana } (j = 1, \dots, m; k = 1, \dots, l) \quad (2.10)$$

3. Linguistik dari atribut I_j yang kardinalitas skalarnya melebihi nilai *minimumsupport* ($\min_sup$) yang telah ditentukan dimasukkan dalam *frequent 1-itemset* L_1 .
4. Atur $k=1$, dimana k merupakan jumlah linguistik pada *frequent k-itemset* L_k .
5. $A_i = \{A_i(1), \dots, A_i(k)\}$ ditunjukkan sebagai himpunan linguistik ke- i pada L_k dengan $A_i(j)$ sebagai linguistik ke- j pada A_i . Membangkitkan himpunan kandidat C_k dengan menggabungkan linguistik ke- i A_i dan linguistik ke- j A_j pada L_{k-1} .

Dengan memeriksa nama dari linguistik, linguistik pada C_k tidak boleh memiliki atribut yang sama.

6. Jika ada subset $k-1$ dari C_k yang tidak termasuk dalam *frequent* $(k-1)$ -*itemset* L_{k-1} , maka terapkan *downward closure property* untuk memangkas kandidat *itemset* C_k . *Downward closure property* menyatakan bahwa jika sebuah *itemset* adalah *frequent* (yang *support*-nya diatas *minimumsupport*) maka semua subset dari *itemset* juga harus *frequent*.
7. Hitung *support* untuk setiap himpunan kandidat C_k menggunakan persamaan 2.6. Jika *support* lebih besar atau sama dengan *minimumsupport* yang telah ditentukan (*min_sup*), maka masukkan C_k di *frequent k-itemset* L_k .
8. Jika L_k kosong, maka lakukan langkah selanjutnya, jika tidak atur $k = k+1$ dan ulangi langkah 5, 6 dan 7.
9. Bangkitkan *fuzzy association rule* untuk semua *frequent k-itemset* L_k dengan cara berikut :

Untuk semua linguistik A_i pada L_k , $A_i = \{A_i(1), \dots, A_i(k)\}$ ($A_i(k)$ adalah linguistik ke- k pada A_i), bentuk semua kemungkinan *fuzzy association rule*.

Contoh :

$$\begin{aligned} & "X_1 \text{ is } A_1(1) \text{ AND } \dots \text{ AND } X_{r-1} \text{ is } A_1(r-1) \text{ AND } X_{r+1} \text{ is } A_1(r+1) \\ & \text{AND } \dots \text{ AND } X_k \text{ is } \tilde{A}_1(k) \Rightarrow X_r \text{ is } \tilde{A}_1(r)" \quad (r = 1, \dots, k) \end{aligned} \quad (2.11)$$

Hitung nilai *confidence* untuk setiap aturan menggunakan persamaan 2.7.

Output adalah aturan yang *confidence*-nya lebih besar atau sama dengan *minimum confidence* yang telah ditentukan (*min_conf*).

2.5.7 Klasifikasi

Klasifikasi merupakan suatu teknik yang dapat memberikan klasifikasi pada data baru dengan manipulasi data yang telah diklasifikasikan dan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari *record* yang terklasifikasi untuk menentukan kelas data baru [KUS-07].

Proses klasifikasi biasanya dibagi menjadi dua fase, yaitu [PRA-03]:

1. *Learning*
2. Pada fase ini sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan.
3. *Test*

Pada fase ini model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi, model ini dapat dipakai untuk prediksi kelas data yang belum diketahui.

Terdapat beberapa metode klasifikasi. Diantaranya *decision tree*, *Bayesian*, *neural network*, *k-nearest neighbor*, *associative classification* dan lain-lain. Pada skripsi ini digunakan metode klasifikasi yaitu *associative classification*.

2.5.8 Associative Classification

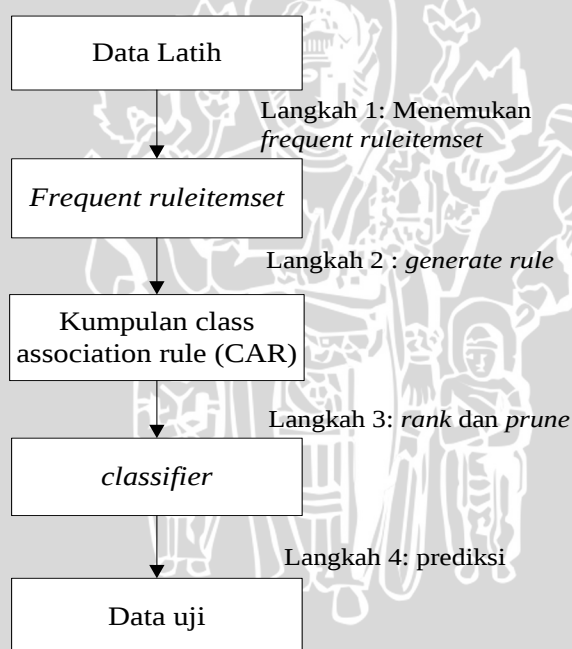
Associative classification (AC) merupakan kasus khusus dalam pencarian *association rule* yang mana bagian *right-hand-side* atau konsekuen mengacu pada sebuah atribut kelas. Algoritma AC menghasilkan aturan *if-then* sederhana yang mudah dipahami dan diinterpretasikan oleh *end-user* [THA-07].

Langkah utama dalam membangun *classifier* dengan menggunakan AC, sebagai berikut [THA-07] :

1. Menemukan semua *frequent rule items*.
2. Membangkitkan semua CAR yang memiliki *confidence* di atas *minconf threshold* dari *frequent rule items* yang diekstrak pada langkah 1.
3. Pemilihan salah satu subset dari CAR untuk membentuk *classifier* dari yang dihasilkan pada langkah 2.
4. Mengukur kualitas dari *classifier* yang diperoleh pada objek data uji.

Terdapat beberapa metode penerapan *associative classification* diantaranya adalah CBA, *Classification based on Multiple Association Rule* (CMAR) dan *Classification based on Predictive Association Rule* (CPAR). Jika data yang digunakan memiliki nilai kuantitatif, semua metode tersebut melakukan partisi atribut menjadi beberapa interval dan memetakan nilai kuantitatif dalam atribut menjadi nilai biner agar sesuai dengan bentuk umum *association rule*. Seperti yang telah dijelaskan dalam subbab 2.3.2 partisi tersebut

menimbulkan masalah *sharp boundary* atau batas tajam yang menyebabkan hilangnya informasi terutama pada batas-batas partisi yang tajam. [LU-03] mengintegrasikan himpunan *fuzzy* dalam *associative classification* untuk mengatasi masalah batas tegas tersebut melalui metode *Classification System of Fuzzy Association Rule* (CFAR). Dari penelitian yang dilakukannya pada *dataset wine* dan *diabetes*, diketahui bahwa CFAR memiliki akurasi 97% dan 77,2% lebih tinggi daripada CBA dan C4.5 yang akurasinya 91% dan 92% untuk *dataset wine* dan 74% pada *dataset diabetes*. Penelitian lain yang dilakukan oleh [JIN-04] serta [CHE-08] pada beberapa *dataset* lain menunjukkan bahwa menggunakan *fuzzy association rule* untuk membangun sistem klasifikasi memiliki akurasi yang baik dengan rata-rata akurasi 85,48%. Berikut langkah-langkah *associative classification* pada gambar 2.5.



Gambar 2.5 Langkah-langkah Associative Classification

Sumber :[THA-07]

2.5.9 Classification based on Fuzzy Association Rule

Dalam sistem klasifikasi, *Fuzzy association rule* (FAR) merupakan bentuk aturan:

$$\text{"IF } X \text{ is } A \Rightarrow \text{Class } C_i \text{"},$$

dimana X is A adalah *antecedent* aturan dan *Class* C_i adalah konsekuensi aturan yang menunjukkan label kelas $C_i (C_i \in C, i = 1, \dots, q)$. Bentuk aturan tersebut merupakan subset dari FAR yang disebut FCAR (*fuzzy class association rule*).

Classification based on Fuzzy Association Rule (CFAR) merupakan metode yang mengintegrasikan masalah klasifikasi dengan teknik *fuzzy association rule mining*. Secara umum, CFAR terdiri dari dua tahap yaitu menemukan atau membangkitkan FCAR dan memilih atau memangkas FCAR untuk membangun sistem klasifikasi (*classifier*).

Algoritma yang digunakan untuk menemukan FCAR atau disebut algoritma FCAR adalah algoritma *fuzzy association rule* pada subbab 2.3.6 yang disesuaikan untuk masalah klasifikasi. Algoritma FCAR dapat dilihat pada gambar 2.6.

```

1   $L_1 = \text{find\_frequent1-item\_linguistic\_terms}(T);$ 
2  for ( $k=2; L_{k-1} \neq \emptyset; k++$  do begin
3     $FCAR\_CAND_k = FCAR\_CAND\_GEN(L_{k-1}, \text{min\_sup});$ 
4    for setiap kandidat  $fcar\_cand \in FCAR\_CAND_k$  do
      begin
5      if  $C_1 \subset fcar\_cand$  then
6        for setiap record  $T_i \in T$  do begin
7           $\text{compability}(i) = f_i(fcar\_cand, T_i);$ 
8           $fcar\_cand.FS = \frac{\sum_{i=1}^n \text{compability}(i)}{n};$ 
9        end
10       endif
11     end
12    $L_k = \{fcar\_cand \in FCAR\_CAND_k | fcar\_cand.count \geq \text{min\_sup}\}$ 
13 end
14  $L = \bigcup_k L_k;$ 
     $FCAR = \text{genFCAR}(L);$ 

```

Gambar 2.6Algoritma FCAR

Sumber :[JIN-04]

Input atau masukan yang diperlukan dalam algoritma ini ialah sejumlah *record* data $T = \{T_i\}$, ($i = 1, \dots, n$), sejumlah atribut dan kelas $\{I_j\}$ himpunan linguistik $= \{a_1, \dots, a_n\}$ yang berasosiasi dengan atribut I_j , *minimum support* dan *minimum confidence* yang telah ditentukan. Sedangkan *output* atau keluaran dari algoritma FCAR ialah sejumlah *Fuzzy Class Association Rule* (FCAR).

Algoritma FCAR pertama kali mengubah nilai kuantitatif menjadi nilai keanggotaan *fuzzy* yang telah didefinisikan untuk setiap atribut. Kemudian menghitung kardinalitas skalar untuk setiap linguistik menggunakan persamaan 2.10 dan membentuk *frequent 1-itemset* dengan memilih *itemset* atau linguistik yang memenuhi *minimum support*. Algoritma selanjutnya menggunakan *frequent (k-1)-itemset* untuk membentuk kandidat *k-itemset* $FCAR_CAND_k$ dengan menggunakan fungsi $FCAR_CAND_GEN$ (baris 2 pada gambar 2.6). Fungsi ini juga memastikan bahwa tidak ada linguistik pada kandidat yang berasosiasi pada atribut yang sama. Tidak seperti *fuzzy association rule mining*, algoritma FCAR hanya mengekstrak kandidat *itemset* yang mengandung atribut kelas (baris 5 pada gambar 2.6), karena hanya *itemset* tersebut yang dapat membangkitkan FCAR yang mengandung atribut kelas sebagai konsekuensi aturan. Selanjutnya dilakukan perhitungan *support* menggunakan persamaan 2.6 untuk setiap kandidat linguistik pada semua *record* (baris 7 dan 8 pada gambar 2.6). Semua kandidat yang *support*nya memenuhi *minimum support* dimasukkan dalam himpunan *frequent k-itemset* (baris 12 pada gambar 2.6). Terakhir, menggabungkan semua *frequent itemset* L dan membangkitkan FCAR menggunakan fungsi $genFCAR$ (baris 15 pada gambar 2.6) [JIN-04].

Aturan (FCAR) yang dibangkitkan dilatih menggunakan data latih berdasarkan pada ukuran kecocokan (disebut DF) antara aturan dan data latih untuk membangun *classifier*. Misalnya terdapat aturan $r: F \Rightarrow C$ dan sebuah *record* d , *confidence* untuk mengklasifikasikan d dengan aturan r adalah sebagai berikut :

$$DF = (d) * Conf(r) \quad (2.12)$$

dimana $Conf(r)$ adalah nilai *confidence* aturan r dan $\mu_F(d)$ derajat keanggotaan atribut pada *record* d yang bersesuaian dengan aturan r . Misalnya R adalah kumpulan aturan atau FCAR dan D adalah data latih maka proses membangun *classifier* dapat dilihat pada gambar 2.7 [CHE-08].

```

1.  for each  $d'' \in D''$  do
2.      for each  $r \in R$  do
3.   $r.DF = \mu_{d''} Dconf\ r$ 
4.       $r.use=0$ 
5.       $r.RightN=0$ 
6.       $r.WrongN=0$ 
7.  end for
8.  sort  $R$  by  $r.DF$  desc
9.   $flag=0$ 
10. while( $flag=0$ ) do
11.      $r$ =fisrt rule of  $R$ 
12.     if  $r.use=1$  then
13.         Break
14.      $r.use=1$ 
15.     if clas( $r, d''$ ) then
16.          $r.RightN++$ 
17.          $flag=1$ 
18.         break
19.     else
20.          $r.WrongN++$ 
21.         move  $r$  to the
           bottom of  $R$ 
22. end for

```

Gambar 2.7 Algoritma Classifier

Sumber :[CHE-08]

Proses pelatihan *rule* terdiri dari tiga tahap yaitu [CHE-08] :

1. Menghitung DF setiap aturan dan mengurutkannya dari terbesar hingga terkecil (baris 2-8),
2. Memilih aturan secara berurutan hingga *record* terklasifikasi dengan benar (baris 9-22), untuk setiap aturan $RightN$ dan $WrongN$ digunakan untuk menyimpan jumlah data yang diklasifikasikan benar atau salah,
3. Mengulangi langkah 1 dan 2 untuk setiap *record*. Aturan terburuk yaitu aturan yang nilai $RightN$ -nya sama dengan 0 dan satu aturan dengan nilai $(RightN/(RightN+WrongN))$ terkecil akan dihapus dari R .

Dalam mengklasifikasikan data baru, dilakukan perhitungan *discriminant function* untuk setiap setiap kelas. Misalnya terdapat k FCAR sebagai berikut :

$$R_k: IF\ X(1,k)\ is\ B(1,k) AND \dots AND\ X(l_k,k)\ is\ B(l_k,k)\ THEN\ i\ is\ C_k$$

dimana $k = 1, 2, \dots, M$, $X(1, k), \dots, X(l_k, k) \in$ atribut $\{i_1, i_2, \dots, i_m\}$, $B(l_k, k)$ adalah fuzzy set untuk atribut $X(l_k, k)$ dan $C_k \in \{C_1, \dots, C_q\}$ merupakan kelas. Untuk mengklasifikasikan sampel baru (y) maka digunakan persamaan sebagai berikut :

$$g_h(y) = \frac{\sum_{1 \leq k \leq M, i=c_k} \prod_{j=1}^{l_k} B(i,k)[X(j,k)(y)]}{\sum_{k=1}^M \prod_{j=1}^{l_k} B(i,k)[X(j,k)(y)]} \quad (2.13)$$

dimana $X(j,k)(y)$ adalah nilai atribut $X(j,k)$ pada sampel y , $B(i,k)[X(j,k)(y)]$ adalah derajat keanggotaan $X(j,k)(y)$ pada fuzzy set $B(j,k)$. Setelah *discriminant function* untuk setiap kelas dihitung, maka diambil label kelas dengan nilai *discriminant function* yang terbesar untuk mengklasifikasi sampel y [LU-03].

2.6 Evaluasi

Evaluasi dilakukan untuk mengetahui tingkat akurasi dari hasil klasifikasi, dengan cara menghitung jumlah *record* uji yang kelasnya diprediksi secara tepat (akurat). Akurasi dapat diperoleh dari persentase kebenaran, yaitu perbandingan antara jumlah diagnosa yang tepat dengan jumlah data keseluruhan NUG-09]. Akurasi dinyatakan dalam persamaan (2.14).

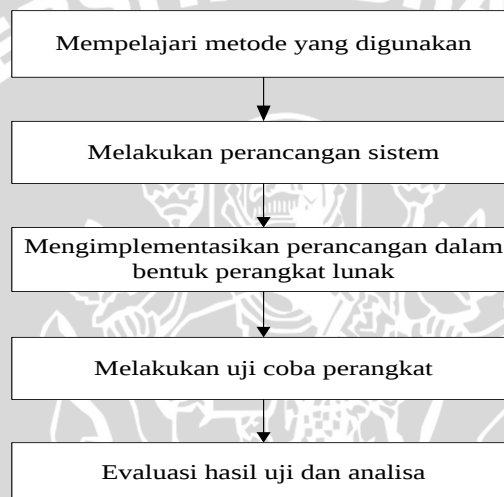
$$Akurasi = \frac{\text{jumlah prediksi benar}}{\text{jumlah total prediksi}} \times 100\% \quad (2.14)$$

Jumlah prediksi benar adalah jumlah *record* data uji yang diprediksi kelasnya menggunakan metode klasifikasi dan hasilnya sama dengan kelas sebenarnya. Sedangkan jumlah total prediksi adalah jumlah keseluruhan *record* yang diprediksi kelasnya (seluruh data uji).

BAB III

METODOLOGI DAN PERANCANGAN

Pada bab ini berisi penjelasan mengenai metode dan langkah-langkah perancangan yang dilakukan dalam penelitian tentang *fuzzy association rule* dengan metode *Classification Based on Fuzzy Association Rule* (CFAR) untuk Evaluasi Kinerja Karyawan PT. Sierad Produce, Tbk. Langkah-langkah yang dijalankan dalam penelitian ini dijelaskan pada gambar 3.1, sebagai berikut :



Gambar 3.1Langkah-langkah Penelitian

Berdasarkan gambar 3.1, langkah langkah yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Melakukan studi literatur yang berhubungan dengan evaluasi kinerja karyawan dan metode yang digunakan yaitu *classification based on association rule* (CFAR) serta literatur lain yang berkaitan seperti yang telah dijelaskan pada bab 2.
2. Melakukan perancangan sistem.
3. Mengimplementasikan hasil analisis dan perancangan yang telah dilakukan dalam bentuk perangkat lunak.
4. Melakukan uji coba terhadap perangkat lunak.
5. Mengevaluasi *output* yang dihasilkan oleh sistem.

3.1 Data yang Digunakan

Data yang digunakan dalam penelitian ini adalah data evaluasi kinerja karyawan pada PT. Sierad Produce, Tbk yang merupakan data hasil penilaian kinerja karyawan semester I tahun 2011 yang dilakukan oleh departemen *Human Resources Development* (HRD) sebanyak 300 data. Pada dataset evaluasi kinerja karyawan, atribut yang digunakan ialah kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antara manusia, dan keahlian profesional. Sedangkan kelas *output* yaitu: yaitu:

- 1 = Mengikuti *training*/pelatihan.
- 2 = Tidak mengikuti *training*/pelatihan.

3.2 Deskripsi Umum Sistem

Secara umum sistem yang dibangun adalah perangkat lunak yang mengimplementasikan *classification based on fuzzy association rule* (CFAR) untuk klasifikasi kinerja karyawan. *Fuzzy association rule* digunakan untuk menemukan sejumlah aturan yang mana bagian konsekuennya berupa atribut kelas. Kemudian aturan-aturan yang terbentuk akan dipilih untuk membentuk *classifier*. Aturan-aturan yang ada dalam *classifier* nantinya akan digunakan untuk mengidentifikasi data uji.

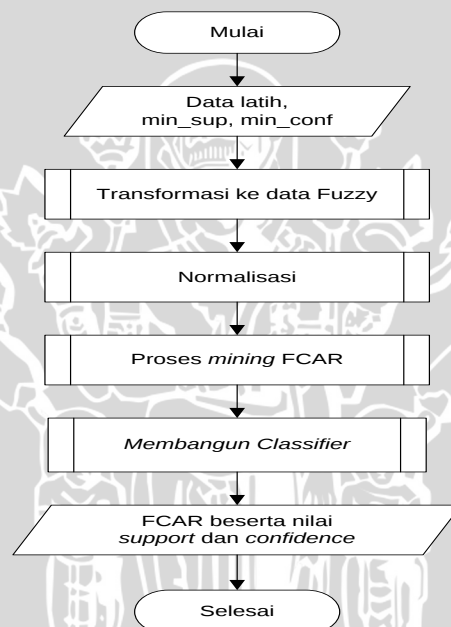
Perangkat lunak yang dibangun menggunakan data evaluasi kinerja karyawan pada PT. Sierad Produce, Tbk. Perangkat lunak ini akan menguji keakuratan hasil klasifikasi data evaluasi kinerja karyawan terhadap data sebenarnya. Parameter yang digunakan dalam pengujian yaitu *minimum support* dan *minimum confidence*. Sistem ini bertujuan untuk menentukan apakah seorang karyawan mengikuti *training*/pelatihan atau tidak mengikuti *training*/pelatihan berdasarkan beberapa atribut yang digunakan.

3.3 Perancangan Sistem

Pada perancangan sistem klasifikasi data kinerja karyawan, diperlukan dua proses utamayaitu proses pembentukan aturan dan proses pengujian. Perancangan alur dari proses pembentukan aturan dan pengujian dapat dilihat pada gambar 3.2 dan 3.10.

3.3.1 Pembentukan Aturan

Proses pembentukan aturan (*rules*) merupakan proses untuk mendapatkan sejumlah aturan *fuzzy* menggunakan algoritma FCAR. Proses ini terdiri dari 4 subproses yaitu transformasi ke data *fuzzy*, normalisasi, *mining* FCAR dan proses membangun *classifier*. Masukan yang diperlukan pada proses ini yaitu data kinerja karyawan pada PT. Sierad Produce,Tbk, *minimum support* (*min_sup*) dan *minimum confidence* (*min_conf*) yang diberikan oleh pengguna sedangkan keluaran proses berupa aturan *fuzzy* yang disebut *fuzzy class association rule* (FCAR). Alur proses pembentukan aturan (*rules*) pada gambar 3.2 adalah sebagai berikut :



Gambar 3.2 Flowchart Proses Pembentukan Aturan

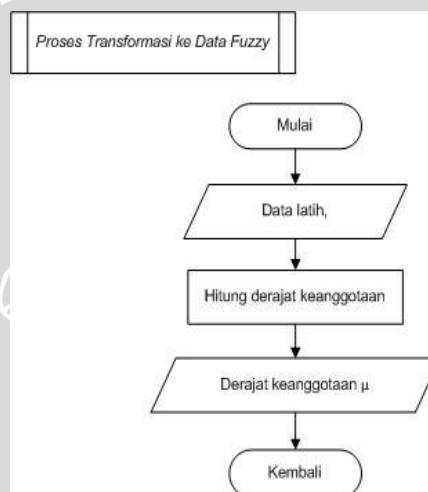
Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Sistem mendapatkan *input* berupa data latih, *minimum support* (*min_sup*) dan *minimum confidence* (*min_conf*). Data latih terdiri dari 10 atribut yaitu kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antara manusia, keahlian profesional dan kelas.
2. Mengubah data latih menjadi linguistik dan derajat keanggotaan *fuzzy*.
3. Melakukan normalisasi data *fuzzy*.

4. Melakukan *mining* aturan (*rule*) menggunakan algoritma FCAR.
5. Membangun *classifier* menggunakan aturan yang dihasilkan pada langkah 3.
6. Keluarannya berupa aturan klasifikasi *fuzzy* yang disebut *fuzzy class association rule* (FCAR) beserta nilai *support* dan *confidence*.

3.3.1.1 Proses Transformasi ke Data Fuzzy

Proses ini mengubah nilai kuantitatif pada setiap atribut menjadi derajat keanggotaan *fuzzy* beserta linguistiknya berdasarkan fungsi keanggotaan masing-masing atribut. Alur proses transformasi data ke data *fuzzy* dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart Proses Tranformasi ke Data Fuzzy

Terdapat 9 atribut yang akan diubah ke dalam atribut *fuzzy* yaitu kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antara manusia, dan keahlian professional. Masing-masing dari 9 atribut tersebut dibagi menjadi lima parameter atau variabel linguistik yaitu sangat kurang, kurang, cukup, baik, dan sangat baik. Fungsi keanggotaan semua atribut dapat dilihat pada gambar 3.4. Dan untuk menghitung fungsi keanggotaan tiap variabel dihitung seperti pada persamaan 3.1 parameter sangat kurang, persamaan 3.2 parameter kurang, persamaan 3.3 parameter cukup, persamaan 3.4 parameter baik, dan persamaan 3.5 parameter sangat baik.

$$\mu_{\text{sangat kurang}}(x) = \begin{cases} 1 & ; \quad x < 20 \\ \frac{35-x}{(35-20)} & ; \quad 20 \leq x \leq 35 \\ 0 & ; \quad x > 35 \end{cases} \quad (3.1)$$

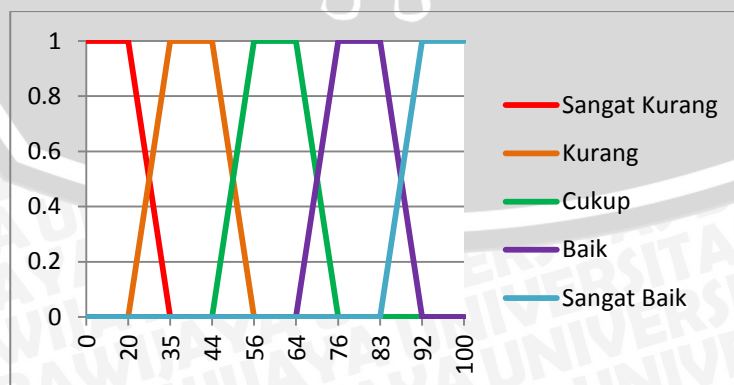
$$\mu_{\text{kurang}}(x) = \begin{cases} 0 & ; \quad x < 20 \vee x > 56 \\ \frac{x-20}{(35-20)} & ; \quad 20 \leq x \leq 35 \\ 1 & ; \quad 35 < x < 44 \\ \frac{56-x}{(56-44)} & ; \quad 44 \leq x \leq 56 \end{cases} \quad (3.2)$$

$$\mu_{\text{cukup}}(x) = \begin{cases} 0 & ; \quad x < 44 \vee x > 76 \\ \frac{x-44}{(56-44)} & ; \quad 44 \leq x \leq 56 \\ 1 & ; \quad 56 < x < 64 \\ \frac{76-x}{(76-64)} & ; \quad 64 \leq x \leq 76 \end{cases} \quad (3.3)$$

$$\mu_{\text{baik}}(x) = \begin{cases} 0 & ; \quad x < 64 \vee x > 92 \\ \frac{x-64}{(76-64)} & ; \quad 64 \leq x \leq 76 \\ 1 & ; \quad 76 < x < 83 \\ \frac{92-x}{(92-83)} & ; \quad 83 \leq x \leq 92 \end{cases} \quad (3.4)$$

$$\mu_{\text{sangat baik}}(x) = \begin{cases} 0 & ; \quad x < 83 \\ \frac{x-83}{(92-83)} & ; \quad 83 \leq x \leq 92 \\ 1 & ; \quad x > 92 \end{cases} \quad (3.5)$$

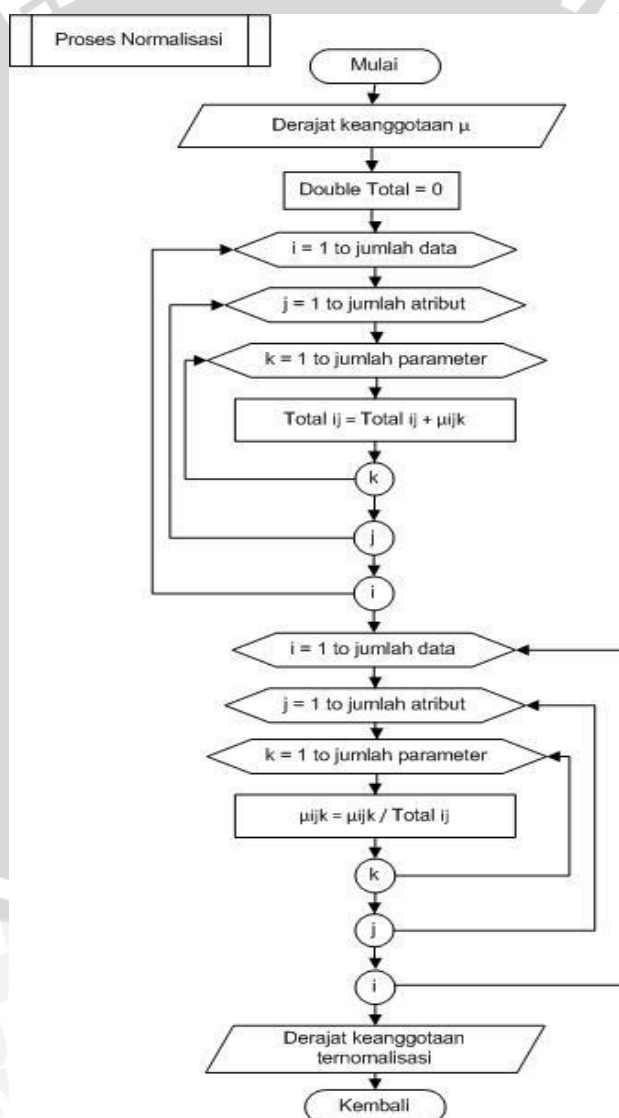
Himpunan fuzzy untuk setiap *linguistic term* semua atribut menggunakan kurva berbentuk trapesium seperti pada gambar 3.4



Gambar 3.4 Himpunan Fuzzy Semua Atribut

3.3.1.2 Normalisasi

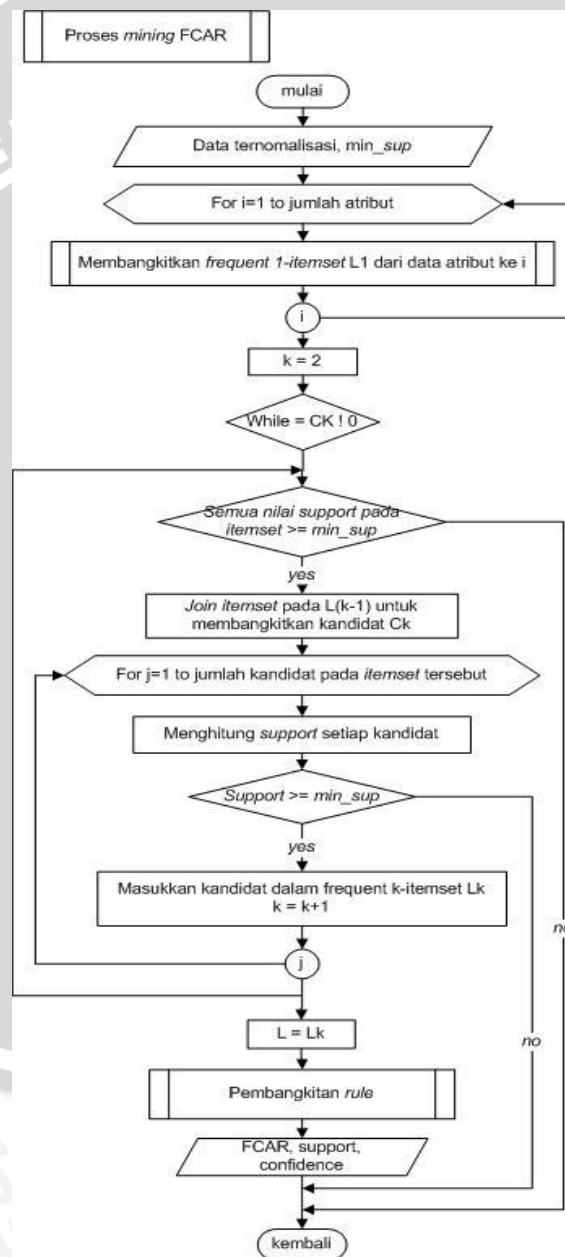
Normalisasi dilakukan untuk setiap linguistik pada setiap atribut agar nilai derajat keanggotaan atribut tersebut berjumlah 1. Perhitungan normalisasi dilakukan dengan membagi derajat keanggotaan untuk linguistik ke- k pada atribut ke- j pada baris (data) ke- i dengan jumlah total derajat keanggotaan atribut ke- j pada data ke- i seperti pada persamaan 2.9. Dalam sistem ini, setiap atribut memiliki 5 parameter atau variabel linguistik sehingga k dibatasi sampai 5. Jumlah atribut dalam sistem yang akan dinormalisasi adalah 9 atribut untuk setiap data. Alur proses normalisasi dapat dilihat pada gambar 3.5.



Gambar 3.5 Flowchart Proses Normalisasi

3.3.1.3 Proses *Mining* FCAR

Proses ini merupakan proses penerapan algoritma FCAR untuk membentuk aturan dari data yang telah dinormalisasi. Masukan yang diperlukan pada proses *mining* FCAR ialah data yang ternormalisasi, *minimum support* (*min_sup*) dan *minimum confidence* (*min_conf*). Proses ini memiliki 2 subproses yaitu membangkitkan *frequent 1-itemset* L1 dan pembangkitan *rule*. Alur proses *mining* FCAR dapat dilihat pada gambar 3.6.

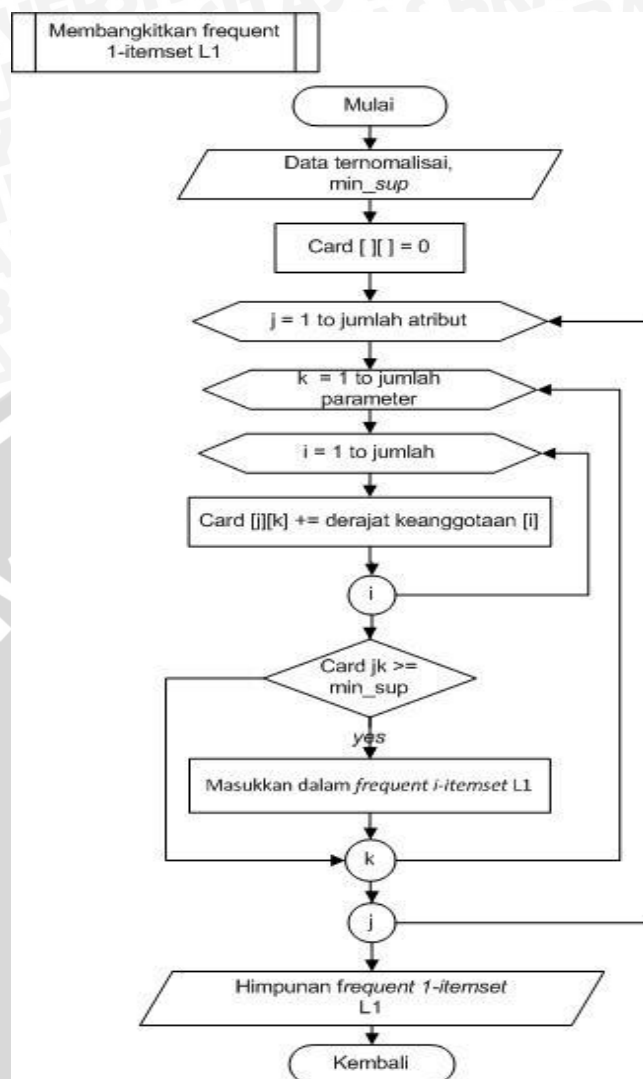


Gambar 3.6 Flowchart Proses Mining FCAR

Langkah awal yang dilakukan pada proses ini yaitu subproses membangkitkan *frequent 1-itemset* L1. Subproses ini merupakan proses untuk mendapatkan *item* yang memenuhi nilai *minimum support* yang telah ditentukan pengguna dengan cara menghitung kardinalitas skalar setiap linguistik untuk setiap atribut menggunakan persamaan 2.10. Perhitungan kardinalitas skalar dilakukandengan menjumlahkan derajat keanggotaan pada setiap *record* untuk setiap linguistik pada atribut. Kardinalitas skalar yang lebih atau sama dengan *minimumsupport* (*min_sup*) akan dimasukkan dalam *frequent 1-itemset* L1. Alur proses ini dapat dilihat pada gambar 3.7. Keluaran yang dihasilkan dari proses ini ialah himpunan linguistik atribut yang memenuhi nilai *min_sup* (*frequent 1-itemset* L1).

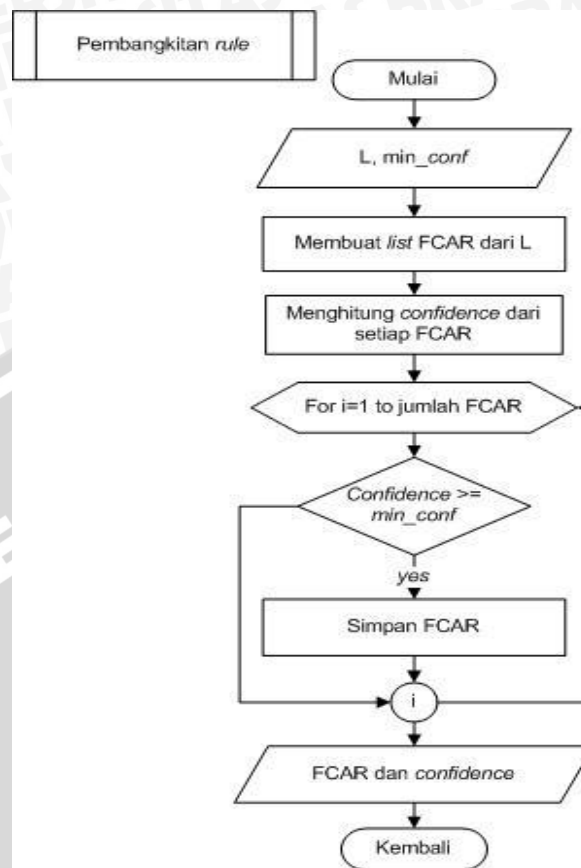
Langkah selanjutnya ialah membentuk kandidat Ck dari *frequent 1-itemset* L1 dan menghitung *support* setiap kandidat. *Support* yang lebih atau sama dengan *minimum support* selanjutnya akan dimasukkan dalam L2. Proses ini berulang hingga tidak ada lagi himpunan *itemset* dalam Lk. Penjelasan dari gambar 3.5 adalah sebagai berikut :

1. Inisialisasi $k=2$.
2. Pengecekan semua nilai *support* pada *itemset*, jika nilai *support* lebih besar dari *minimumsupport* maka dilanjutkan ke langkah 3.
3. Membentuk kandidat *itemset* Ck dari *frequent 1-itemset*L1 dengan cara menggabungkan setiap linguistik pada Lk-1
4. Menghitung *support* setiap kandidat yang ada pada Ck menggunakan persamaan 2.6.
5. Membandingkan nilai *support* tersebut dengan *minimum support* (*min_sup*) yang telah ditentukan. Jika nilainya lebih atau sama dengan *min_sup* maka akan di masukkan dalam *frequent k-itemset* Lk $k= k+1$.
6. Ulangi langkah 2 sampai 5 hingga tidak ada lagi *itemset* dalam Lk.
7. Setelah iterasi berhenti, semua *frequent k-itemset* Lk digabung dalam satu himpunan besar L. Himpunan *itemset* L akan digunakan untuk membentuk aturan (FCAR).



Gambar 3.7 Flowchart Proses Membangkitkan Frequent 1-Itemset L1

Subproses selanjutnya pada proses *mining* FCAR adalah pembangkitan *rule*. Subproses ini merupakan proses untuk membuat aturan dari himpunan *frequent itemset* yang dihasilkan tahapan sebelumnya. Aturan yang dibuat ialah aturan klasifikasi *fuzzy* yang mana bagian konsekuennya merupakan atribut kelas. Setiap aturan yang dibentuk akan dihitung nilai *confidence*-nya menggunakan persamaan 2.7. Aturan dengan *confidence* yang lebih besar atau sama dengan *minimum confidence* (*min_conf*) yang telah ditentukan akan disimpan untuk digunakan pada proses selanjutnya yaitu membangun *classifier*. Alur proses pembangkitan *rule* dapat dilihat pada gambar 3.8.

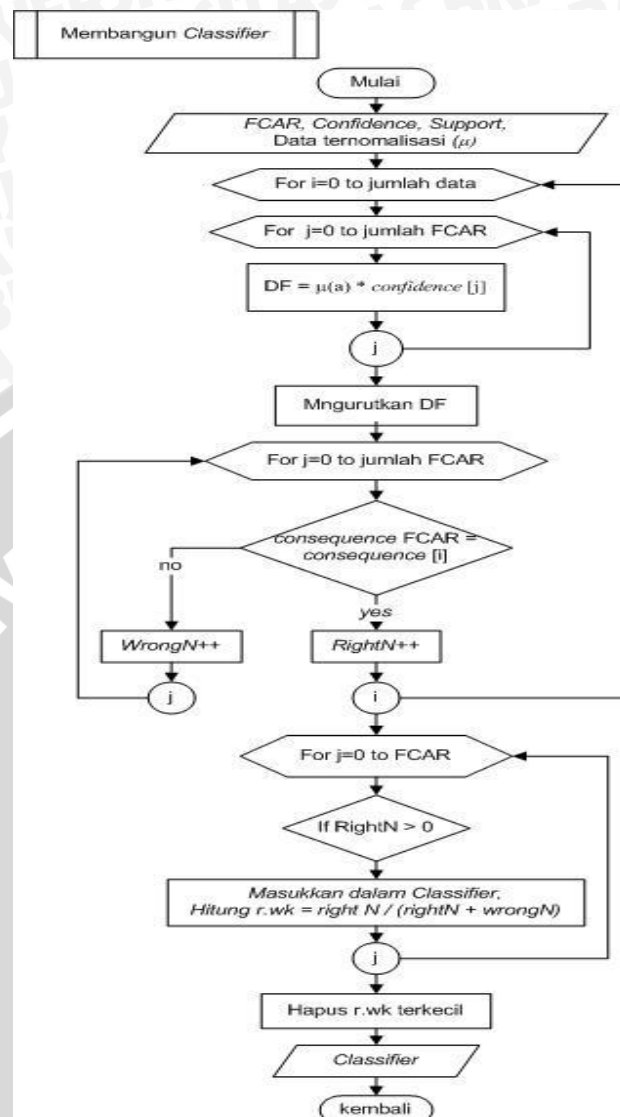


Gambar 3.8 Flowchart Proses Pembangkitan Rule

3.3.1.4 Membangun Classifier

Proses membangun *classifier* bertujuan untuk melatih aturan (FCAR) yang diperoleh dari proses pembentukan aturan (*rules*). Pelatihan dilakukan seperti pada prosedur pelatihan *rule* pada subbab 2.5.9 dengan menghitung nilai *DF* setiap *rule*. *DF* merupakan perkalian derajat keanggotaan *items* pada data ke-*n* yang bersesuaian dengan *rule* ke-*r* seperti pada rumus 2.12. Nilai *DF* kemudian diurutkan dari nilai terbesar ke terkecil. Jika *rule* mengklasifikasikan data latih dengan benar maka nilai *rightN rule* tersebut akan bertambah.

Sebaliknya, jika tidak maka nilai *wrongN* yang bertambah. Pada skirpsi ini, *rule* dengan nilai *rightN* lebih dari 0 akan dimasukkan dalam *classifier*. Kemudian *rule* dengan nilai *wk* terkecil akan dihapus dari *classifier* selama *rule* yang ada dalam *classifier* lebih dari satu. Alur proses membangun *classifier* ditampilkan pada gambar 3.9.

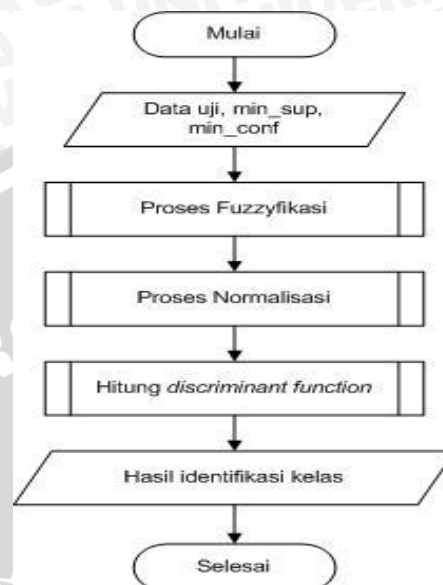


Gambar 3.9 Flowchart Proses Membangun classifier

3.3.2 Proses Pengujian

Proses pengujian merupakan proses untuk mengetahui tingkat keakuratan aturan yang terbentuk. Proses ini terdiri dari 3 subproses yaitu transformasi ke data *fuzzy*, normalisasi dan perhitungan *discriminant function*. Dua subproses pada proses pengujian ini (transformasi ke data *fuzzy* dan normalisasi) sama dengan subproses yang ada pada proses pembentukan aturan (*rules*). Masukan yang diperlukan pada proses pengujian berupa data uji kinerja karyawan PT Sierad Produce, Tbk dan aturan *fuzzy* yang terbentuk dari proses sebelumnya. Sedangkan keluarannya berupa hasil prediksi atau identifikasi kelas yaitu

menentukan apakah seorang karyawan mengikuti *training*/pelatihan atau tidak mengikuti *training*/pelatihan. Alur proses pengujian pada gambar 3.10 adalah sebagai berikut :



Gambar 3.10 Flowchart Proses Pengujian

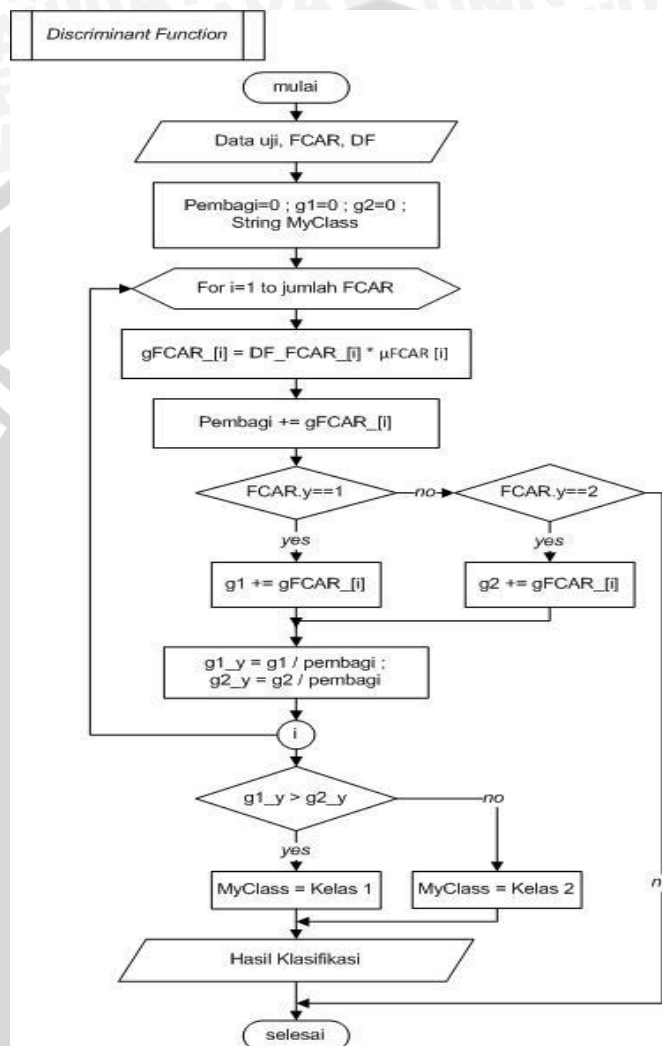
Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Sistem mendapatkan *input* berupa data uji yang terdiri atribut kooperatif, kehadiran dan ketepatan waktu, keandalan, inisiatif, sikap, *judgement*, komunikasi, hubungan antara manusia, dan keahlian profesional serta FCAR, *confidence* dan *support* yang didapatkan pada proses pembentukan aturan.
2. Mengubah data latih menjadi linguistik dan derajat keanggotaan *fuzzy*.
3. Melakukan normalisasi data *fuzzy*.
4. Menghitung *discriminant function* untuk semua FCAR yang dihasilkan dan memilih nilai *discriminant function* yang paling besar untuk mengklasifikasi pada data uji.
5. Keluarannya berupa hasil identifikasi seorang karyawan mengikuti *training*/pelatihan atau tidak mengikuti *training*/pelatihan.

3.3.2.1 Menghitung *Discriminant Function*

Untuk mengidentifikasi atau mengklasifikasikan data uji, dilakukan perhitungan *discriminant function* setiap kelas untuk setiap FCAR sesuai dengan

persamaan 2.13. Kemudian membandingkan *discriminant function* setiap kelas dan kelas yang nilai *discriminant function*nya tertinggi akan digunakan untuk mengklasifikasi data uji. Alur perhitungan *discriminant function* dapat dilihat pada gambar 3.11.



Gambar 3.11 Flowchart Menghitung Discriminant Function

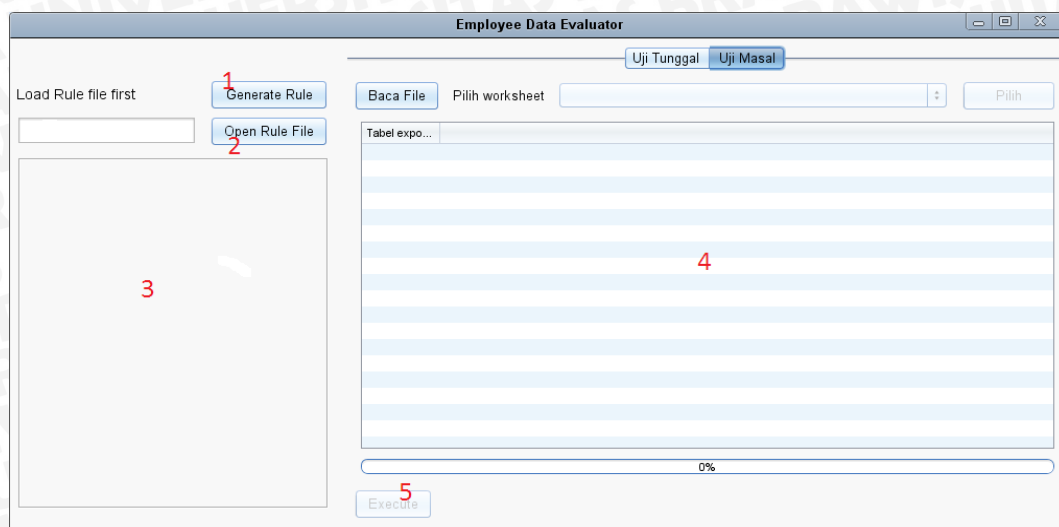
3.4 Perancangan Antarmuka

Antarmuka (*interface*) untuk sistem ini terdiri dari dua bagian utama, yaitu bagian untuk pembentukan aturan yang bertujuan untuk membangkitkan aturan fuzzy (FCAR) dan bagian untuk pengujian yang bertujuan untuk melakukan pengujian terhadap aturan yang terbentuk. Antarmuka dari sistem ditunjukkan pada gambar 3.12 dan 3.13. untuk proses pengujian, gambar 3.14 untuk proses pembentukan aturan.

Gambar 3.12 Antarmuka Pengujian Tunggal

Pada gambar 3.12, antarmuka pengujian tunggal terdiri dari :

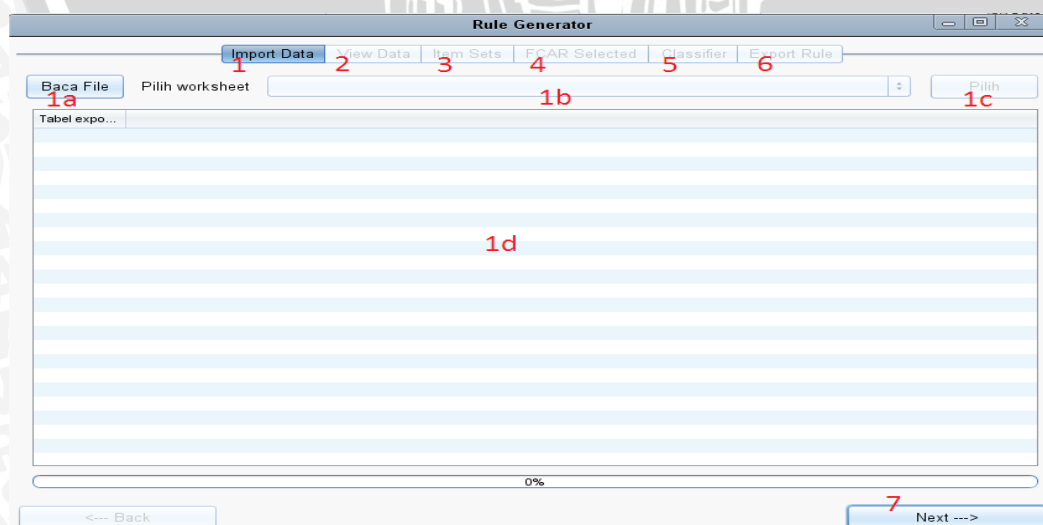
1. Tombol *Generate Rule* untuk menampilkan *form* proses pembentukan aturan.
2. Tombol *Open Rule File* untuk mengambil data (*rule*) yang telah disimpan setelah proses pembentukan aturan.
3. *Text field* untuk menampilkan lokasi penyimpanan data (*rule*) yang telah terbentuk.
4. *Text area* untuk menampilkan data (*rule*) yang telah terbentuk.
5. Untuk memasukkan nilai-nilai setiap atribut yang akan dilakukan pengujian (*uji tunggal*).
6. Tombol *Excute* digunakan untuk memulai proses pengujian dan hasilnya akan ditampilkan pada *text field* nomor 8.
7. *Text Area* untuk menampilkan proses pengujian.
8. *Text field* untuk menampilkan hasil pengujian.



Gambar 3.13 Antarmuka Pengujian Data Banyak

Pada gambar 3.13, antarmuka pengujian data banyak terdiri dari :

1. Tombol *Generate Rule* untuk menampilkan *form* proses pembentukan aturan.
2. Tombol *Open Rule File* untuk mengambil data (*rule*) yang telah disimpan setelah proses pembentukan aturan.
3. *Text area* untuk menampilkan data (*rule*) yang telah terbentuk.
4. *Text Area* untuk menampilkan data yang akan dilakukan pengujian dan data yang telah dilakukan pengujian.
5. Tombol *Excute* digunakan untuk memulai proses pengujian dan hasilnya akan ditampilkan pada *text area* nomor 4.



Gambar 3.14 Antarmuka Pembentukan Rule

Pada gambar 3.14, antarmuka pengujian terdiri dari :

1. *Panel Import Data* menampilkan data yang akan dilakukan pembentukan aturan dan data yang *valid* dapat melakukan proses selanjutnya.
 - 1a. Tombol *Baca File* untuk mengambil data yang akan dilakukan proses pembentukan aturan.
 - 1b. *Combo box* menampilkan pilihan *worksheet* data yang akan dilakukan proses pembentukan aturan.
 - 1c. Tombol pilih untuk memulai proses pembacaan data.
 - 1d. *Text area* untuk menampilkan data yang akan dilakukan proses pembentukan aturan.
2. *Panel View Data* menampilkan proses data setelah dilakukan transformasi ke data *fuzzy* dan normalisasi, serta terdapat perintah untuk memasukan minimal *support*.
3. *Panel Item Sets* menampilkan *itemset* yang terbentuk dan terdapat perintah untuk memasukkan minimal *confidence*.
4. *Panel FCAR selected* menampilkan aturan yang terbentuk sementara.
5. *Panel Classifier* menampilkan hasil pembentukan aturan yang telah diseleksi (*classifier*).
6. *Panel Export Rule* untuk menyimpan *rule* yang telah terbentuk.
7. *Tombol Next* untuk melakukan proses ke *panel* selanjutnya.

3.5 Perhitungan Manual

Perhitungan manual ini menggunakan sampel data kinerja karyawan pada PT Sierad Produce, Tbk sebanyak 10 *record* yang terdiri dari 9 atribut. Sampel data ini dapat dilihat pada tabel 3.1 sebagai data latih. Prosedur perhitungan manualnya adalah sebagai berikut :

1. Menentukan nilai *minimum support* dan *minimum confidence*.
Inisialisasi awal :
 - a. Jumlah data (n) = 10
 - b. *Minimum support* (min_sup) = 0,25
 - c. *Minimum confidence* (min_conf) = 65%

Tabel 3.1 Sampel Data Kinerja Karyawan pada PT. Sierad Produce, tbk

No	Kooperatif	Kehadiran dan Ketepatan Waktu	Inisiatif	Keandalan	Sikap	Judgement	Komunikasi	Hubungan antar manusia	Keahlian Profesional	Kelas
1	83.4	98.1	83.9	92.7	81	75.9	84.8	82.9	89.6	Tidak
2	73.6	66.8	91.5	89.3	73.1	76.3	93.4	84.6	91.4	Tidak
3	71	95.5	85.5	73.7	83.5	70.6	58.3	83.1	72.5	Training
4	64	80.2	62.5	73.8	86.5	69.3	71.6	69.2	73.2	Training
5	69.8	89.6	65.1	80.2	75.2	66.8	76.2	87.5	80.9	Training
6	61.9	71.7	65.9	52.8	82.9	57.5	54.6	75	64.5	Training
7	66.4	91.6	72.1	89.9	88.5	69	71.3	70.2	79.6	Tidak
8	68.6	66.9	50	75.6	74.1	73.9	69.3	63.8	70	Training
9	64	77.4	67.3	85.3	79.1	68.6	66.8	65.2	87.6	Training
10	82.1	75.6	74.8	81.6	79.2	71	85.2	84.6	81.1	Tidak

2. Transformasi ke data *fuzzy*, mengubah nilai kuantitatif pada setiap atribut untuk setiap *record* menjadi nilai derajat keanggotaan *fuzzy* menggunakan fungsi keanggotaan seperti pada subbab 3.3.1. Hasil proses transformasi ke data *fuzzy* masing-masing atribut dapat dilihat pada tabel 3.2-3.10.

Tabel 3.2 Transformasi ke Data Fuzzy Atribut Kooperatif

No	Kooperatif	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	83.4	0	0	0	0.781818	0.036364	0	1
2	73.6	0	0	0.2	0.8	0	0	1
3	71	0	0	0.416667	0.583333	0	1	0
4	64	0	0	1	0	0	1	0
5	69.8	0	0	0.516667	0.483333	0	1	0
6	61.9	0	0	1	0	0	1	0
7	66.4	0	0	0.8	0.2	0	0	1
8	68.6	0	0	0.616667	0.383333	0	1	0
9	64	0	0	1	0	0	1	0
10	82.1	0	0	0	1	0	0	1

Tabel 3.3 Transformasi ke Data Fuzzy Atribut Kehadiran dan Ketepatan Waktu

No	KdKW	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	98.1	0	0	0	0	1	0	1
2	66.8	0	0	0.766667	0.233333	0	0	1
3	95.5	0	0	0	0	1	1	0
4	80.2	0	0	0	1	0	1	0
5	89.6	0	0	0	0.218182	0.66	1	0
6	71.7	0	0	0.358333	0.641667	0	1	0
7	91.6	0	0	0	0.036364	0.86	0	1
8	66.9	0	0	0.758333	0.241667	0	1	0
9	77.4	0	0	0	1	0	1	0
10	75.6	0	0	0.033333	0.966667	0	0	1

Tabel 3.4 Transformasi ke Data Fuzzy Atribut Inisiatif

No	Inisiatif	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	83.9	0	0	0	0.736364	0.081818	0	1
2	91.5	0	0	0	0.045455	0.772727	0	1
3	85.5	0	0	0	0.590909	0.227273	1	0
4	62.5	0	0	1	0	0	1	0
5	65.1	0	0	0.908333	0.091667	0	1	0
6	65.9	0	0	0.841667	0.158333	0	1	0
7	72.1	0	0	0.325	0.675	0	0	1
8	50	0	0.5	0.5	0	0	1	0
9	67.3	0	0	0.725	0.275	0	1	0
10	74.8	0	0	0.1	0.9	0	0	1

Tabel 3.5 Transformasi ke Data Fuzzy Atribut Keandalan

No	Keandalan	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	92.7	0	0	0	0	1	0	1
2	89.3	0	0	0	0.245455	0.572727	0	1
3	73.7	0	0	0.191667	0.808333	0	1	0
4	73.8	0	0	0.183333	0.816667	0	1	0
5	80.2	0	0	0	1	0	1	0
6	52.8	0	0.266667	0.733333	0	0	1	0
7	89.9	0	0	0	0.190909	0.627273	0	1
8	75.6	0	0	0.033333	0.966667	0	1	0
9	85.3	0	0	0	0.609091	0.209091	1	0
10	81.6	0	0	0	1	0	0	1

Tabel 3.6 Transformasi ke Data Fuzzy Atribut Sikap

No	Sikap	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	81	0	0	0	1	0	0	1
2	73.1	0	0	0.241667	0.758333	0	0	1
3	83.5	0	0	0	0.772727	0.045455	1	0
4	86.5	0	0	0	0.5	0.318182	1	0
5	75.2	0	0	0.066667	0.933333	0	1	0
6	82.9	0	0	0	1	0	1	0
7	88.5	0	0	0	0.318182	0.5	0	1
8	74.1	0	0	0.158333	0.841667	0	1	0
9	79.1	0	0	0	1	0	1	0
10	79.2	0	0	0	1	0	0	1

Tabel 3.7 Transformasi ke Data Fuzzy Atribut Judgement

No	Judgement	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	75.9	0	0	0.008333	0.991667	0	0	1
2	76.3	0	0	0	1	0	0	1
3	70.6	0	0	0.45	0.55	0	1	0
4	69.3	0	0	0.558333	0.441667	0	1	0
5	66.8	0	0	0.766667	0.233333	0	1	0
6	57.5	0	0	1	0	0	1	0
7	69	0	0	0.583333	0.416667	0	0	1
8	73.9	0	0	0.175	0.825	0	1	0
9	68.6	0	0	0.616667	0.383333	0	1	0
10	71	0	0	0.416667	0.583333	0	0	1

Tabel 3.8 Transformasi ke Data Fuzzy Atribut Komunikasi

No	Komunikasi	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	84.8	0	0	0	0.654545	0.163636	0	1
2	93.4	0	0	0	0	1	0	1
3	58.3	0	0	1	0	0	1	0
4	71.6	0	0	0.366667	0.633333	0	1	0
5	76.2	0	0	0	1	0	1	0
6	54.6	0	0.116667	0.883333	0	0	1	0
7	71.3	0	0	0.391667	0.608333	0	0	1
8	69.3	0	0	0.558333	0.441667	0	1	0
9	66.8	0	0	0.766667	0.233333	0	1	0
10	85.2	0	0	0	0.618182	0.2	0	1

Tabel 3.9 Transformasi ke Data Fuzzy Atribut Hubungan antar Manusia

No	HaM	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	82.9	0	0	0	1	0	0	1
2	84.6	0	0	0	0.672727	0.145455	0	1
3	83.1	0	0	0	0.809091	0.009091	1	0
4	69.2	0	0	0.566667	0.433333	0	1	0
5	87.5	0	0	0	0.409091	0.409091	1	0
6	75	0	0	0.083333	0.916667	0	1	0
7	70.2	0	0	0.483333	0.516667	0	0	1
8	63.8	0	0	1	0	0	1	0
9	65.2	0	0	0.9	0.1	0	1	0
10	84.6	0	0	0	0.672727	0.145455	0	1

Tabel 3.10 Transformasi ke Data Fuzzy Atribut Keahlian Profesional

No	KP	Derajat Keanggotaan					Kelas	
		SK	K	C	B	SB	C1	C2
1	89.6	0	0	0	0.218182	0.6	0	1
2	91.4	0	0	0	0.054545	0.763636	0	1
3	72.5	0	0	0.291667	0.708333	0	1	0
4	73.2	0	0	0.233333	0.766667	0	1	0
5	80.9	0	0	0	1	0	1	0
6	64.5	0	0	0.958333	0.041667	0	1	0
7	79.6	0	0	0	1	0	0	1
8	70	0	0	0.5	0.5	0	1	0
9	87.6	0	0	0	0.4	0.418182	1	0
10	81.1	0	0	0	1	0	0	1

3. Melakukan normalisasi untuk setiap linguistik agar nilai fungsi keanggotaan untuk setiap atribut berjumlah 1. Perhitungan normalisasi pada atribut kooperatif pada *record* 1 sebagai berikut :

$$(1, SK) = \frac{0}{0+0+0+0,7818+0,0364} = 0$$

$$(1, K) = \frac{0}{0+0+0+0,7818+0,0364} = 0$$

$$(1, C) = \frac{0}{0+0+0+0,7818+0,0364} = 0$$

$$(1, B) = \frac{0,7818}{0+0+0+0,7818+0,0364} = 0.9556$$

$$(1, SB) = \frac{0,0364}{0+0+0+0,7818+0,0364} = 0.0444$$

Keterangan:

- 1 : kooperatif
- 2 : kehadiran dan ketepatan waktu
- 3 : inisiatif
- 4 : keandalan
- 5 : sikap
- 6 : *judgement*
- 7 : komunikasi
- 8 : hubungan antar manusia
- 9 : keahlian profesional

SK : sangat kurang

K : kurang

C : cukup

B : baik

SB : sangat baik

Hasil normalisasi masing-masing atribut dapat dilihat pada tabel 3.11-3.19.

4. Menghitung kardinalitas skalar setiap atribut menggunakan persamaan 2.10.

Contoh perhitungan kardinalitas skalar pada atribut kooperatif :

$$\text{Card}(1, \text{SB}) = 0.0444 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0.0444$$

Hasil perhitungan kardinalitas skalar masing-masing atribut dapat dilihat pada tabel 3.11-3.19.

Tabel 3.11 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Kooperatif

No	kooperatif					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0.955556	0.044444	0	1
2	0	0	0.2	0.8	0	0	1
3	0	0	0.416667	0.583333	0	1	0
4	0	0	1	0	0	1	0
5	0	0	0.516667	0.483333	0	1	0
6	0	0	1	0	0	1	0
7	0	0	0.8	0.2	0	0	1
8	0	0	0.616667	0.383333	0	1	0
9	0	0	1	0	0	1	0
10	0	0	0	1	0	0	1
card =	0	0	5.55	4.405556	0.044444	6	4

Tabel 3.12 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Kehadiran dan Ketepatan Waktu

No	Kehadiran dan Ketepatan waktu					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0	1	0	1
2	0	0	0.766667	0.233333	0	0	1
3	0	0	0	0	1	1	0
4	0	0	0	1	0	1	0
5	0	0	0	0.248447	0.751553	1	0
6	0	0	0.358333	0.641667	0	1	0
7	0	0	0	0.040568	0.959432	0	1
8	0	0	0.758333	0.241667	0	1	0
9	0	0	0	1	0	1	0
10	0	0	0.033333	0.966667	0	0	1
card =	0	0	1.916667	4.372348	3.710985	6	4

Tabel 3.13 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Inisiatif

No	Inisiatif					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0.9	0.1	0	1
2	0	0	0	0.055556	0.944444	0	1
3	0	0	0	0.722222	0.277778	1	0
4	0	0	1	0	0	1	0
5	0	0	0.908333	0.091667	0	1	0
6	0	0	0.841667	0.158333	0	1	0
7	0	0	0.325	0.675	0	0	1
8	0	0.5	0.5	0	0	1	0
9	0	0	0.725	0.275	0	1	0
10	0	0	0.1	0.9	0	0	1
card =	0	0.5	4.4	3.777778	1.322222	6	4

Tabel 3.14 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Keandalan

No	Keandalan					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0	1	0	1
2	0	0	0	0.3	0.7	0	1
3	0	0	0.191667	0.808333	0	1	0
4	0	0	0.183333	0.816667	0	1	0
5	0	0	0	1	0	1	0
6	0	0.266667	0.733333	0	0	1	0
7	0	0	0	0.233333	0.766667	0	1
8	0	0	0.033333	0.966667	0	1	0
9	0	0	0	0.744444	0.255556	1	0
10	0	0	0	1	0	0	1
card =	0	0.266667	1.141667	5.869444	2.722222	6	4

Tabel 3.15 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Sikap

No	Sikap					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	1	0	0	1
2	0	0	0.241667	0.758333	0	0	1
3	0	0	0	0.944444	0.055556	1	0
4	0	0	0	0.611111	0.388889	1	0
5	0	0	0.066667	0.933333	0	1	0
6	0	0	0	1	0	1	0
7	0	0	0	0.388889	0.611111	0	1
8	0	0	0.158333	0.841667	0	1	0
9	0	0	0	1	0	1	0
10	0	0	0	1	0	0	1
card =	0	0	0.466667	8.477778	1.055556	6	4

Tabel 3.16 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Judgement

No	Judgement					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0.008333	0.991667	0	0	1
2	0	0	0	1	0	0	1
3	0	0	0.45	0.55	0	1	0
4	0	0	0.558333	0.441667	0	1	0
5	0	0	0.766667	0.233333	0	1	0
6	0	0	1	0	0	1	0
7	0	0	0.583333	0.416667	0	0	1
8	0	0	0.175	0.825	0	1	0
9	0	0	0.616667	0.383333	0	1	0
10	0	0	0.416667	0.583333	0	0	1
card =	0	0	4.575	5.425	0	6	4

Tabel 3.17 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Komunikasi

No	Komunikasi					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0.8	0.2	0	1
2	0	0	0	0	1	0	1
3	0	0	1	0	0	1	0
4	0	0	0.366667	0.633333	0	1	0
5	0	0	0	1	0	1	0
6	0	0.116667	0.883333	0	0	1	0
7	0	0	0.391667	0.608333	0	0	1
8	0	0	0.558333	0.441667	0	1	0
9	0	0	0.766667	0.233333	0	1	0
10	0	0	0	0.755556	0.244444	0	1
card =	0	0.116667	3.966667	4.472222	1.444444	6	4

Tabel 3.18 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Hubungan Antar Manusia

No	Hubungan antar Manusia					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	1	0	0	1
2	0	0	0	0.822222	0.177778	0	1
3	0	0	0	0.988889	0.011111	1	0
4	0	0	0.566667	0.433333	0	1	0
5	0	0	0	0.5	0.5	1	0
6	0	0	0.083333	0.916667	0	1	0
7	0	0	0.483333	0.516667	0	0	1
8	0	0	1	0	0	1	0
9	0	0	0.9	0.1	0	1	0
10	0	0	0	0.822222	0.177778	0	1
card =	0	0	3.033333	6.1	0.866667	6	4

Tabel 3.19 Normalisasi dan Perhitungan Kardinalitas Skalar Atribut Keahlian Profesional

No	Keahlian Profesional					Kelas	
	SK	K	C	B	SB	C1	C2
1	0	0	0	0.266667	0.733333	0	1
2	0	0	0	0.066667	0.933333	0	1
3	0	0	0.291667	0.708333	0	1	0
4	0	0	0.233333	0.766667	0	1	0
5	0	0	0	1	0	1	0
6	0	0	0.958333	0.041667	0	1	0
7	0	0	0	1	0	0	1
8	0	0	0.5	0.5	0	1	0
9	0	0	0	0.488889	0.511111	1	0
10	0	0	0	1	0	0	1
card =	0	0	1.983333	5.838889	2.177778	6	4

Nilai kardinalitas skalar yang lebih atau sama dengan *minimum support* (*min_sup*) dimasukkan dalam *frequent 1-itemset* L1 (Tabel 3.20).

No	Itemset	Card
1	(1, C)	5.55
2	(1, B)	4.4056
3	(1, SB)	0.0444
4	(2, C)	1.9167
5	(2, B)	4.3723
6	(2, SB)	3.711
7	(3, K)	0.5
8	(3, C)	4.4
9	(3, B)	3.7778
10	(3, SB)	1.3222
11	(4, K)	0.2667
12	(4, C)	1.1417
13	(4, B)	5.8694
14	(4, SB)	2.7222
15	(5, C)	0.4667
16	(5, B)	8.4778

No	Itemset	Card
17	(5, SB)	1.0556
18	(6, C)	4.575
19	(6, B)	5.425
20	(7, K)	0.1167
21	(7, C)	3.9667
22	(7, B)	4.4722
23	(7, SB)	1.4444
24	(8, C)	3.0333
25	(8, B)	6.1
26	(8, SB)	0.8667
27	(9, C)	1.9833
28	(9, B)	5.8389
29	(9, SB)	2.1778
30	(Kelas, 1)	6
31	(Kelas, 2)	4

- Tabel 3.21 Perhitungan *Support* FCAR_CAND₂**

Record	(1, K)	C1	Compatibility
1	0	1	0
2	0	1	0
3	0	1	0
4	0	0	0
5	0	1	0
6	0	1	0
7	0	0	0
8	0	1	0
9	0	1	0
10	0	0	0
11	0,675	1	0,675
23	0	1	0
24	0	0	0
25	0	0	0
26	1	1	1
27	0	1	0
28	0	0	0

Record	(1, K)	C1	Compability
12	0,037	1	0,037
13	0	0	0
14	0	0	0
15	0,05	1	0,05
16	0	0	0
17	0	1	0
18	0	1	0
19	0	0	0
20	0	0	0
21	0	1	0
22	0	0	0
29	0,082	0	0
30	0,097	1	0.097
Jumlah			1,861
FS ($X \rightarrow Y$)			0,062

Perhitungan *support* salah satu kandidat menggunakan persamaan 2.6 adalah sebagai berikut :

$$\begin{aligned}
 & \text{Support } \{(1, K), (\text{kelas}, C1)\} \\
 &= ((0.7333 \times 0) + (0.9333 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) \\
 &\quad + (0 \times 0) + (0 \times 1) + (0.5111 \times 1) + (0 \times 0)) \\
 &= \frac{0.5111}{10} \\
 &= 0,0511
 \end{aligned}$$

Tabel 3.22 Kandidat 2-itemset dan Nilai Support-nya

No	Itemset	Support	No	Itemset	Support
1	(1, C), (Kelas, C1)	0.45	30	(5, C), (Kelas, C2)	0.0242
2	(1, C), (Kelas, C2)	0.1	31	(5, B), (Kelas, C1)	0.5331
3	(1, B), (Kelas, C1)	0.145	32	(5, B), (Kelas, C2)	0.3147
4	(1, B), (Kelas, C2)	0.2956	33	(5, SB), (Kelas, C1)	0.0445
5	(1, SB), (Kelas, C1)	0	34	(5, SB), (Kelas, C2)	0.0611
6	(1, SB), (Kelas, C2)	0.0444	35	(6, C), (Kelas, C1)	0.3567
7	(2, C), (Kelas, C1)	0.1117	36	(6, C), (Kelas, C2)	0.1008
8	(2, C), (Kelas, C2)	0.08	37	(6, B), (Kelas, C1)	0.2433
9	(2, B), (Kelas, C1)	0.3132	38	(6, B), (Kelas, C2)	0.2992
10	(2, B), (Kelas, C2)	0.1241	39	(7, K), (Kelas, C1)	0.0117
11	(2, SB), (Kelas, C1)	0.1752	40	(7, K), (Kelas, C2)	0
12	(2, SB), (Kelas, C2)	0.1959	41	(7, C), (Kelas, C1)	0.3575
13	(3, K), (Kelas, C1)	0.05	42	(7, C), (Kelas, C2)	0.0392
14	(3, K), (Kelas, C2)	0	43	(7, B), (Kelas, C1)	0.2308
15	(3, C), (Kelas, C1)	0.3975	44	(7, B), (Kelas, C2)	0.2164
16	(3, C), (Kelas, C2)	0.0425	45	(7, SB), (Kelas, C1)	0
17	(3, B), (Kelas, C1)	0.1247	46	(7, SB), (Kelas, C2)	0.1444
18	(3, B), (Kelas, C2)	0.2531	47	(8, C), (Kelas, C1)	0.255
19	(3, SB), (Kelas, C1)	0.0278	48	(8, C), (Kelas, C2)	0.0483
20	(3, SB), (Kelas, C2)	0.1044	49	(8, B), (Kelas, C1)	0.2939
21	(4, K), (Kelas, C1)	0.0267	50	(8, B), (Kelas, C2)	0.3161
22	(4, K), (Kelas, C2)	0	51	(8, SB), (Kelas, C1)	0.0511
23	(4, C), (Kelas, C1)	0.1142	52	(8, SB), (Kelas, C2)	0.0356
24	(4, C), (Kelas, C2)	0	53	(9, C), (Kelas, C1)	0.1983
25	(4, B), (Kelas, C1)	0.4336	54	(9, C), (Kelas, C2)	0
26	(4, B), (Kelas, C2)	0.1533	55	(9, B), (Kelas, C1)	0.3506
27	(4, SB), (Kelas, C1)	0.0255	56	(9, B), (Kelas, C2)	0.2333
28	(4, SB), (Kelas, C2)	0.2467	57	(9, SB), (Kelas, C1)	0.0511
29	(5, C), (Kelas, C1)	0.0225	58	(9, SB), (Kelas, C2)	0.1667

Support yang lebih atau sama dengan *minimum support* dimasukkan dalam *frequent2-itemset* L2 (Tabel 3.23).

Tabel 3.23 Frequent2-itemset (L2)

No	Itemset	Support
1	(1, C), (Kelas, C1)	0.45
2	(1, B), (Kelas, C2)	0.2956
3	(2, B), (Kelas, C1)	0.3132
4	(3, C), (Kelas, C1)	0.3975
5	(3, B), (Kelas, C2)	0.2531
6	(4, B), (Kelas, C1)	0.4336
7	(5, B), (Kelas, C1)	0.5331
8	(5, B), (Kelas, C2)	0.3147

No	Itemset	Support
9	(6, C), (Kelas, C1)	0.3567
10	(6, B), (Kelas, C2)	0.2992
11	(7, C), (Kelas, C1)	0.3575
12	(8, C), (Kelas, C1)	0.255
13	(8, B), (Kelas, C1)	0.2939
14	(8, B), (Kelas, C2)	0.3161
15	(9, B), (Kelas, C1)	0.3506

6. Membentuk kembali kandidat 3-itemset FCAR_CAND₃ dari L2 dan menghitung support untuk setiap kandidat. Himpunan kandidat 3-itemset FCAR_CAND3 dapat dilihat pada tabel 3.24.

Tabel 3.24 Kandidat 3-itemset dan Nilai Support-nya

No	Itemset	Support
1	(1, C), (2, B), (Kelas, C1)	0.2919
2	(1, C), (3, C), (Kelas, C1)	0.3344
3	(1, C), (4, B), (Kelas, C1)	0.3011
4	(1, C), (5, B), (Kelas, C1)	0.4006
5	(1, C), (6, C), (Kelas, C1)	0.2867
6	(1, C), (7, C), (Kelas, C1)	0.2778
7	(1, C), (8, C), (Kelas, C1)	0.2167
8	(1, C), (8, B), (Kelas, C1)	0.212
9	(1, C), (9, B), (Kelas, C1)	0.2417
10	(1, B), (3, B), (Kelas, C2)	0.1939
11	(1, B), (5, B), (Kelas, C2)	0.264
12	(1, B), (6, B), (Kelas, C2)	0.2414
13	(1, B), (8, B), (Kelas, C2)	0.2539
14	(2, B), (3, C), (Kelas, C1)	0.2612
15	(2, B), (4, B), (Kelas, C1)	0.2042
16	(2, B), (5, B), (Kelas, C1)	0.2688
17	(2, B), (6, C), (Kelas, C1)	0.2049
18	(2, B), (7, C), (Kelas, C1)	0.1835
19	(2, B), (8, C), (Kelas, C1)	0.1762
20	(2, B), (8, B), (Kelas, C1)	0.1246
21	(2, B), (9, B), (Kelas, C1)	0.1652
22	(3, C), (4, B), (Kelas, C1)	0.2748
23	(3, C), (5, B), (Kelas, C1)	0.3446
24	(3, C), (6, C), (Kelas, C1)	0.2631
25	(3, C), (7, C), (Kelas, C1)	0.1945
26	(3, C), (8, C), (Kelas, C1)	0.1789
27	(3, C), (8, B), (Kelas, C1)	0.1732
28	(3, C), (9, B), (Kelas, C1)	0.2315

No	Itemset	Support
29	(3, B), (5, B), (Kelas, C2)	0.2105
30	(3, B), (6, B), (Kelas, C2)	0.1754
31	(3, B), (8, B), (Kelas, C2)	0.2034
32	(4, B), (5, B), (Kelas, C1)	0.3754
33	(4, B), (6, C), (Kelas, C1)	0.2215
34	(4, B), (7, C), (Kelas, C1)	0.2218
35	(4, B), (8, C), (Kelas, C1)	0.2099
36	(4, B), (8, B), (Kelas, C1)	0.1728
37	(4, B), (9, B), (Kelas, C1)	0.3046
38	(5, B), (6, C), (Kelas, C1)	0.3246
39	(5, B), (7, C), (Kelas, C1)	0.3288
40	(5, B), (8, C), (Kelas, C1)	0.2171
41	(5, B), (8, B), (Kelas, C1)	0.2682
42	(5, B), (9, B), (Kelas, C1)	0.3022
43	(5, B), (6, B), (Kelas, C2)	0.2495
44	(5, B), (8, B), (Kelas, C2)	0.2647
45	(6, C), (7, C), (Kelas, C1)	0.2109
46	(6, C), (8, C), (Kelas, C1)	0.1129
47	(6, C), (8, B), (Kelas, C1)	0.2049
48	(6, C), (9, B), (Kelas, C1)	0.1944
49	(6, B), (8, B), (Kelas, C2)	0.2509
50	(7, C), (8, C), (Kelas, C1)	0.1529
51	(7, C), (8, B), (Kelas, C1)	0.2034
52	(7, C), (9, B), (Kelas, C1)	0.168
53	(8, C), (8, B), (Kelas, C1)	0.0412
54	(8, C), (9, B), (Kelas, C1)	0.1378
55	(8, B), (9, B), (Kelas, C1)	0.1619

Support yang lebih atau sama dengan *minimum support* dimasukkan dalam *frequent3-itemset* L3 (Tabel 3.25).

Tabel 3.25 Frequent3-itemset (L3)

No	Itemset	Support	No	Itemset	Support
1	(1, C), (2, B), (Kelas, C1)	0.2919	12	(3, C), (5, B), (Kelas, C1)	0.3446
2	(1, C), (3, C), (Kelas, C1)	0.3344	13	(3, C), (6, C), (Kelas, C1)	0.2631
3	(1, C), (4, B), (Kelas, C1)	0.3011	14	(4, B), (5, B), (Kelas, C1)	0.3754
4	(1, C), (5, B), (Kelas, C1)	0.4006	15	(4, B), (9, B), (Kelas, C1)	0.3046
5	(1, C), (6, C), (Kelas, C1)	0.2867	16	(5, B), (6, C), (Kelas, C1)	0.3246
6	(1, C), (7, C), (Kelas, C1)	0.2778	17	(5, B), (7, C), (Kelas, C1)	0.3288
7	(1, B), (5, B), (Kelas, C2)	0.264	18	(5, B), (8, B), (Kelas, C1)	0.2682
8	(1, B), (8, B), (Kelas, C2)	0.2539	19	(5, B), (9, B), (Kelas, C1)	0.3022
9	(2, B), (3, C), (Kelas, C1)	0.2612	20	(5, B), (8, B), (Kelas, C2)	0.2647
10	(2, B), (5, B), (Kelas, C1)	0.2688	21	(6, B), (8, B), (Kelas, C2)	0.2509
11	(3, C), (4, B), (Kelas, C1)	0.2748			

7. Membentuk kembali kandidat 4-itemset FCAR_CAND₃ dari L3 dan menghitung *support* untuk setiap kandidat. Himpunan kandidat 4-itemset FCAR_CAND4 dapat dilihat pada tabel 3.26.

Tabel 3.26 Kandidat 4-itemset dan Nilai Support-nya

No	Itemset	Support	No	Itemset	Support
1	(1, C), (2, B), (3, C), (Kelas, C1)	0.2456	23	(2, B), (3, C), (6, C), (Kelas, C1)	0.174
2	(1, C), (2, B), (4, B), (Kelas, C1)	0.1834	24	(2, B), (5, B), (6, C), (Kelas, C1)	0.1813
3	(1, C), (2, B), (5, B), (Kelas, C1)	0.2498	25	(2, B), (5, B), (7, C), (Kelas, C1)	0.1671
4	(1, C), (2, B), (6, C), (Kelas, C1)	0.1941	26	(2, B), (5, B), (8, B), (Kelas, C1)	0.1069
5	(1, C), (2, B), (7, C), (Kelas, C1)	0.1783	27	(2, B), (5, B), (9, B), (Kelas, C1)	0.1318
6	(1, C), (3, C), (4, B), (Kelas, C1)	0.2124	28	(3, C), (4, B), (5, B), (Kelas, C1)	0.2293
7	(1, C), (3, C), (5, B), (Kelas, C1)	0.2875	29	(3, C), (4, B), (6, C), (Kelas, C1)	0.157
8	(1, C), (3, C), (6, C), (Kelas, C1)	0.2261	30	(3, C), (4, B), (9, B), (Kelas, C1)	0.204
9	(1, C), (3, C), (7, C), (Kelas, C1)	0.1838	31	(3, C), (5, B), (6, C), (Kelas, C1)	0.2354
10	(1, C), (4, B), (5, B), (Kelas, C1)	0.2546	32	(3, C), (5, B), (7, C), (Kelas, C1)	0.1758
11	(1, C), (4, B), (6, C), (Kelas, C1)	0.1567	33	(3, C), (5, B), (8, B), (Kelas, C1)	0.1533
12	(1, C), (4, B), (7, C), (Kelas, C1)	0.154	34	(3, C), (5, B), (9, B), (Kelas, C1)	0.1916
13	(1, C), (4, B), (9, B), (Kelas, C1)	0.2043	35	(4, B), (5, B), (9, B), (Kelas, C1)	0.2627
14	(1, C), (5, B), (6, C), (Kelas, C1)	0.2595	36	(4, B), (5, B), (6, C), (Kelas, C1)	0.1939
15	(1, C), (5, B), (7, C), (Kelas, C1)	0.2557	37	(4, B), (5, B), (7, C), (Kelas, C1)	0.1971
16	(1, C), (5, B), (8, B), (Kelas, C1)	0.1912	38	(4, B), (5, B), (8, B), (Kelas, C1)	0.1512
17	(1, C), (5, B), (9, B), (Kelas, C1)	0.202	39	(5, B), (6, C), (7, C), (Kelas, C1)	0.1988
18	(1, C), (6, C), (7, C), (Kelas, C1)	0.1809	40	(5, B), (6, C), (8, B), (Kelas, C1)	0.1904
19	(1, B), (5, B), (8, B), (Kelas, C2)	0.2317	41	(5, B), (6, C), (9, B), (Kelas, C1)	0.1695
20	(1, B), (6, B), (8, B), (Kelas, C2)	0.2128	42	(5, B), (7, C), (8, B), (Kelas, C1)	0.1917
21	(2, B), (3, C), (5, B), (Kelas, C1)	0.2189	43	(5, B), (7, C), (9, B), (Kelas, C1)	0.1487
22	(2, B), (3, C), (4, B), (Kelas, C1)	0.1699	44	(5, B), (8, B), (9, B), (Kelas, C1)	0.1418

Support yang lebih atau sama dengan *minimum support* dimasukkan dalam *frequent4-itemset* L4 (Tabel 3.27).

Tabel 3.27*Frequent4-itemset* (L4)

No	Itemset	Support
1	(1, C), (3, C), (5, B), (Kelas, C1)	0.2875
2	(1, C), (4, B), (5, B), (Kelas, C1)	0.2546
3	(1, C), (5, B), (6, C), (Kelas, C1)	0.2595
4	(1, C), (5, B), (7, C), (Kelas, C1)	0.2557
5	(4, B), (5, B), (9, B), (Kelas, C1)	0.2627

8. Membentuk kandidat *5-itemset* FCAR_CAND5 dari L4 dan menghitung *support* untuk setiap kandidat. *Support* yang lebih atau sama dengan *minimum support* dimasukkan dalam *frequent5-itemset* L5 (Tabel 3.27).

Tabel 3.28Kandidat *5-itemset* dan Nilai *Support*-nya

No	Itemset	Support
1	(1, C), (3, C), (4, B), (5, B), (Kelas, C1)	0.1728
2	(1, C), (3, C), (5, B), (6, C), (Kelas, C1)	0.2011
3	(1, C), (3, C), (5, B), (7, C), (Kelas, C1)	0.1668
4	(1, C), (4, B), (5, B), (6, C), (Kelas, C1)	0.1338
5	(1, C), (4, B), (5, B), (7, C), (Kelas, C1)	0.1352
6	(1, C), (4, B), (5, B), (9, B), (Kelas, C1)	0.1705
7	(1, C), (5, B), (6, C), (7, C), (Kelas, C1)	0.171

9. Berhenti, karena *support*-nya tidak memenuhi nilai *minimum support*.
10. Membangkitkan aturan (*rule*) dari semua *frequent k-itemset* L_k mulai dari *2-itemset* (L2) dari tabel 2.23 sampai *5-itemset* dari tabel 3.27 dan menghitung *confidence*-nya. Aturan yang dibangkitkan adalah *fuzzy class association rule* (FCAR) yang mana bagian konsekuennya adalah sebuah kelas. Hasil dari pembangkitan aturan dapat dilihat pada tabel 3.29.

Confidence dihitung menggunakan persamaan 2.7. Aturan yang nilai *confidencenya* lebih atau sama dengan *minimum confidence* dapat dilihat pada tabel 3.30. Contoh perhitungan *confidence* untuk $\{(1, C)\} \rightarrow (\text{kelas}, C1)$:

$$\text{Sup } \{(1, C)\} \Rightarrow (\text{kelas}, C1) = \frac{4.5}{10} = 0,45$$

$$\text{Sup} \{(1, C)\} = \frac{5.55}{10} = 0,555$$

$$\text{Conf}\{(1, C)\} \Rightarrow (\text{kelas}, C1) = \frac{0,45}{0,555} \times 100\% = 81,81\%$$

Keterangan:

- 1 : kooperatif
- 2 : kehadiran dan ketepatan waktu
- 3 : inisiatif
- 4 : keandalan
- 5 : sikap
- 6 : *judgement*
- 7 : komunikasi
- 8 : hubungan antar manusia
- 9 : keahlian professional

SK : sangat kurang

K : kurang

C : cukup

B : baik

SB : sangat baik

Tabel 3.29 FCAR dan Confidence Semua k-itemset

No.	Rule	Sup (X → Y)	Sup (X)	Confidence
1	(1, C), (Kelas, C1)	0.45	0.555	0.8108108
2	(1, B), (Kelas, C2)	0.2956	0.4406	0.6709033
3	(2, B), (Kelas, C1)	0.3132	0.4272	0.7331461
4	(3, C), (Kelas, C1)	0.3975	0.44	0.9034091
5	(3, B), (Kelas, C2)	0.2531	0.3778	0.6699312
6	(4, B), (Kelas, C1)	0.4336	0.5869	0.7387971
7	(5, B), (Kelas, C1)	0.5331	0.8478	0.628804
8	(5, B), (Kelas, C2)	0.3147	0.8478	0.371196
9	(6, C), (Kelas, C1)	0.3567	0.4575	0.7796721
10	(6, B), (Kelas, C2)	0.2992	0.5425	0.5515207
11	(7, C), (Kelas, C1)	0.3575	0.3967	0.9011848
12	(8, C), (Kelas, C1)	0.255	0.3033	0.8407517
13	(8, B), (Kelas, C1)	0.2939	0.61	0.4818033
14	(8, B), (Kelas, C2)	0.3161	0.61	0.5181967
15	(9, B), (Kelas, C1)	0.3506	0.5839	0.6004453
16	(1, C), (2, B), (Kelas, C1)	0.2919	0.2998	0.9736491
17	(1, C), (3, C), (Kelas, C1)	0.3344	0.3604	0.9278579
18	(1, C), (4, B), (Kelas, C1)	0.3011	0.3257	0.9244704

19	(1, C), (5, B), (Kelas, C1)	0.4006	0.4469	0.8963974
20	(1, C), (6, C), (Kelas, C1)	0.2867	0.3333	0.860186
21	(1, C), (7, C), (Kelas, C1)	0.2778	0.3091	0.8987383
22	(1, B), (5, B), (Kelas, C2)	0.264	0.3965	0.665826
23	(1, B), (8, B), (Kelas, C2)	0.2539	0.3357	0.7563301
24	(2, B), (3, C), (Kelas, C1)	0.2612	0.2721	0.9599412
25	(2, B), (5, B), (Kelas, C1)	0.2688	0.3847	0.6987263
26	(3, C), (4, B), (Kelas, C1)	0.2748	0.2924	0.9398085
27	(3, C), (5, B), (Kelas, C1)	0.3446	0.3673	0.9381977
28	(3, C), (6, C), (Kelas, C1)	0.2631	0.2862	0.9192872
29	(4, B), (5, B), (Kelas, C1)	0.3754	0.5072	0.740142
30	(4, B), (9, B), (Kelas, C1)	0.3046	0.4299	0.7085369
31	(5, B), (6, C), (Kelas, C1)	0.3246	0.3898	0.8327347
32	(5, B), (7, C), (Kelas, C1)	0.3288	0.3441	0.9555362
33	(5, B), (8, B), (Kelas, C1)	0.2682	0.5329	0.5032839
34	(5, B), (9, B), (Kelas, C1)	0.3022	0.4728	0.6391709
35	(5, B), (8, B), (Kelas, C2)	0.2647	0.5329	0.4967161
36	(6, B), (8, B), (Kelas, C2)	0.2509	0.3399	0.7381583
37	(1, C), (3, C), (5, B), (Kelas, C1)	0.2875	0.2976	0.9660618
38	(1, C), (4, B), (5, B), (Kelas, C1)	0.2546	0.2664	0.9557057
39	(1, C), (5, B), (6, C), (Kelas, C1)	0.2595	0.2777	0.9344616
40	(1, C), (5, B), (7, C), (Kelas, C1)	0.2557	0.2679	0.9544606
41	(4, B), (5, B), (9, B), (Kelas, C1)	0.2627	0.3733	0.7037235

Tabel 3.30 FCAR yang Memenuhi *Minimum Confidence*

No	Rule	Sup (X → Y)	Confidence
1	JIKA kooperatif cukup MAKA mengikuti <i>training</i>	0.45	0.810811
2	JIKA inisiatif cukup MAKA mengikuti <i>training</i>	0.3975	0.903409
3	JIKA <i>judgement</i> cukup MAKA mengikuti <i>training</i>	0.3567	0.779672
4	JIKA komunikasi cukup MAKA mengikuti <i>training</i>	0.3575	0.901185
5	JIKA hubungan antar manusia cukup MAKA mengikuti <i>training</i>	0.255	0.840752
6	JIKA kooperatif cukup DAN kehadiran & ketepatan waktu baik MAKA mengikuti <i>training</i>	0.2919	0.973649
7	JIKA kooperatif cukup DAN inisiatif cukup MAKA mengikuti <i>training</i>	0.3344	0.927858
8	JIKA kooperatif cukup DAN keandalan baik MAKA mengikuti <i>training</i>	0.3011	0.92447
9	JIKA kooperatif cukup DAN sikap baik MAKA mengikuti <i>training</i>	0.4006	0.896397
10	JIKA kooperatif cukup DAN <i>judgement</i> cukup MAKA mengikuti <i>training</i>	0.2867	0.860186
11	JIKA kooperatif cukup DAN komunikasi cukup MAKA mengikuti <i>training</i>	0.2778	0.898738
12	JIKA kooperatif baik DAN hubungan antar manusia baik MAKA tidak mengikuti <i>training</i>	0.2539	0.75633
13	JIKA kehadiran & ketepatan waktu baik DAN inisiatif cukup MAKA mengikuti <i>training</i>	0.2612	0.959941
14	JIKA inisiatif cukup DAN keandalan baik MAKA mengikuti <i>training</i>	0.2748	0.939808
15	JIKA inisiatif cukup DAN sikap baik MAKA mengikuti <i>training</i>	0.3446	0.938198
16	JIKA inisiatif cukup DAN <i>judgement</i> cukup MAKA	0.2631	0.919287

	mengikuti <i>training</i>		
17	JIKA sikap baik DAN <i>judgement</i> cukup MAKA mengikuti <i>training</i>	0.3246	0.832735
18	JIKA sikap baik DAN komunikasi cukup MAKA mengikuti <i>training</i>	0.3288	0.955536
19	JIKA kooperatif cukup DAN inisiatif cukup DAN sikap baik MAKA mengikuti <i>training</i>	0.2875	0.966062
20	JIKA kooperatif cukup DAN keandalan baik DAN sikap baik MAKA mengikuti <i>training</i>	0.2546	0.955706
21	JIKA kooperatif cukup DAN sikap baik DAN <i>judgement</i> cukup MAKA mengikuti <i>training</i>	0.2595	0.934462
22	JIKA kooperatif cukup DAN sikap baik DAN komunikasi cukup MAKA mengikuti <i>training</i>	0.2557	0.954461

11. 22 *rule* yang dibangkitkan akan dilatih menggunakan data latih dengan cara menghitung nilai *DF* setiap *rule* untuk setiap data latih. Nilai *DF rule* kemudian diurutkan dari nilai terbesar ke nilai terkecil. Jika *rule* pertama mengklasifikasikan data latih dengan benar maka nilai *rightN* nya bertambah. Jika tidak maka *rule* diletakkan di urutan terakhir dan nilai *wrongN*-nya bertambah. Prosesnya adalah sebagai berikut :

Contoh perhitungan *DF* data ke-1 (kelas 2:

Rule1 :

Confidence = 0.81

Nilai kooperatif cukup pada data ke-1 = 0

$DF = 0 * 0.81 = 0$

Rule2 :

Confidence = 0.90

Nilai inisiatif cukup pada data ke-1 = 0

$DF = 0 * 0.90 = 0$

Rule3 :

Confidence = 0.78

Nilai *judgement* cukup pada data ke-1 = 0.0083

$DF = 0.0083 * 0.78 = 0.0065$

Hasil perhitungan untuk data ke-1 dapat dilihat pada tabel 3.31

Tabel 3.31 Perhitungan *DF* Data ke-1

Rule	DF
1	0
2	0
3	0.0065
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0

Rule	DF
12	0.7227
13	0
14	0
15	0
16	0
17	0.0069
18	0
19	0
20	0
21	0
22	0

Selanjutnya mengurutkan nilai *DF* secara menurun yang dapat dilihat pada tabel 3.32

Tabel 3.32 Hasil Pengurutan *DF* Data ke-1

Rule	DF
12	0.7227
17	0.0069
3	0.0065
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0

Rule	DF
1	0
13	0
14	0
15	0
16	0
2	0
18	0
19	0
20	0
21	0
22	0

Cek apakah kelas *rule* ke-12 sama dengan kelas data ke-1. Karena kelas *rule* ke-12 adalah 2, sedangkan kelas data ke-1 adalah 2 maka *rule* ke-12 sehingga dan nilai *R12.rightN* menjadi 1. Proses berpindah ke data ke-2

Jika semua *rule* telah dilatih pada semua data latih maka dilakukan pengecekan pada setiap *rule*. Jika nilai *rightN rule* = 0 maka *rule* akan di buang. Jika lebih dari 0 maka akan dimasukkan dalam *classifier* dan menghitung nilai *r.wk*. *r.wk* adalah hasil bagi *rightN* dengan *rightN+wrongN*. Selama jumlah *rule* dalam *classifier* lebih dari 1 maka *rule* dengan nilai *r.wk* terkecil akan di buang. Tabel 3.33 menunjukkan jumlah *rightN* dan *wrongN* hasil pelatihan *rule*. Dan hasil *classifier* dapat dilihat pada tabel 3.34.

Tabel 3.33 Hasil Pelatihan *Rule*

Rule	RightN	WrongN	r.wk
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	1	1	0.5
6	2	0	1
7	0	1	0
8	0	1	0
9	0	1	0
10	0	1	0
11	0	1	0
12	3	0	1
13	0	0	0
14	1	0	1
15	0	1	0
16	0	1	0
17	0	1	0
18	2	1	0.67
19	0	1	0
20	0	0	0
21	1	1	0.5
22	0	1	0

Tabel 3.34 Hasil *Classifier*

No	Rule	Keterangan	Sup ($X \rightarrow Y$)	Confidence
1	6	JIKA kooperatif cukup DAN kehadiran & ketepatan waktu baik MAKA mengikuti training	0.2919	0.973649
2	12	JIKA kooperatif baik DAN hubungan antar manusia baik MAKA tidak mengikuti training	0.2539	0.75633
3	14	JIKA inisiatif cukup DAN keandalan baik MAKA mengikuti training	0.2748	0.939808
4	18	JIKA sikap baik DAN komunikasi cukup MAKA mengikuti training	0.3288	0.955536
5	21	JIKA kooperatif cukup DAN sikap baik DAN judgement cukup MAKA mengikuti training	0.2595	0.934462

12. Untuk mengklasifikasikan data baru dilakukan perhitungan *discriminant function* menggunakan persamaan 2.13 untuk setiap kelas. Kelas yang nilai *discriminant function*nya paling tinggi digunakan untuk mengklasifikasikan data. Misalnya diberikan 2 data uji yang diambil secara acak dari data evaluasi kinerja karyawan sebagai berikut :

1	2	3	4	5	6	7	8	9	kelas	Klasifikasi kelas
64	77.4	67.3	85.3	79.1	68.6	66.8	65.2	87.6	1	?
82.1	75.6	74.8	81.6	79.2	71	85.2	84.6	81.1	2	?

Setelah ditransformasi menjadi data *fuzzy* dan normalisasi :

1 = KOOPERATIF					2 = KEHADIRAN					3 = INISIATIF				
SK	K	C	B	SB	SK	K	C	B	SB	SK	K	C	B	SB
0	0	1	0	0	0	0	0	1	0	0	0	0.725	0.275	0
0	0	0	1	0	0	0	0.0333	0.9667	0	0	0	0.1	0.9	0

4= KEANDALAN					5 = SIKAP					6 = JUDGEMENT				
SK	K	C	B	SB	SK	K	C	B	SB	SK	K	C	B	SB
0	0	0	0.744	0.2556	0	0	0	1	0	0	0	0.6167	0.3833	0
0	0	0	1	0	0	0	0	1	0	0	0	0.4167	0.5833	0

7 = KOMUNIKASI					8 = HUB.ANTAR MANUSIA					9 = KEAHLIANPROFESIONAL				
SK	K	C	B	SB	SK	K	C	B	SB	SK	K	C	B	SB
0	0	0.766	0.2333	0	0	0	0.9	0.1	0	0	0	0	0.489	0.5111
0	0	0	0.7556	0.244	0	0	0	0.822	0.1778	0	0	0	1	0

1. Menghitung *DF* untuk setiap FCAR :

$$\text{FCAR 1} = 0.9736 * 1 * 1 = 0.9736$$

$$\text{FCAR 2} = 0.7563 * 0 * 0.1 = 0$$

$$\text{FCAR 3} = 0.9398 * 0.725 * 0.744 = 0.5069$$

$$\text{FCAR 4} = 0.9555 * 1 * 0.766 = 0.7319$$

$$\text{FCAR 5} = 0.9345 * 1 * 1 * 0.6167 = 0.5763$$

Discriminant function pada kelas 1 :

$$g_1(y) = \frac{\text{FCAR 1} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}}{\text{FCAR 1} + \text{FCAR 2} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}}$$

$$g_1(y) = \frac{0.9736 + 0.5069 + 0.7319 + 0.5763}{0.9736 + 0 + 0.5069 + 0.7319 + 0.5763}$$

$$= 1$$

Discriminant function pada kelas 2:

$$g_2(y) = \frac{\text{FCAR 2}}{\text{FCAR 1} + \text{FCAR 2} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}}$$

$$g_2(y) = \frac{0}{0.9736 + 0 + 0.5069 + 0.7319 + 0.5763} = 0$$

Karena *discriminant function* untuk kelas 1 lebih besar dari pada kelas 2 maka data uji diklasifikasikan dalam kelas 1 yaitu mengikuti *training*.

2. Menghitung *DF* untuk setiap FCAR :

$$\begin{aligned}
 \text{FCAR 1} &= 0.9736 * 0 * 0.9667 &= 0 \\
 \text{FCAR 2} &= \mathbf{0.7563 * 1 * 0.822} &= \mathbf{0.6217} \\
 \text{FCAR 3} &= 0.9398 * 0.1 * 1 &= 0.0939 \\
 \text{FCAR 4} &= 0.9555 * 1 * 0 &= 0 \\
 \text{FCAR 5} &= 0.9345 * 0 * 1 * 0.4167 &= 0
 \end{aligned}$$

Discriminant function pada kelas 1 :

$$\begin{aligned}
 g_1(y) &= \frac{\text{FCAR 1} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}}{\text{FCAR 1} + \text{FCAR 2} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}} \\
 g_1(y) &= \frac{0 + 0.0939 + 0 + 0}{0 + 0.6217 + 0.0939 + 0 + 0} \\
 &= 0.1312
 \end{aligned}$$

Discriminant function pada kelas 2:

$$\begin{aligned}
 g_2(y) &= \frac{\text{FCAR 2}}{\text{FCAR 1} + \text{FCAR 2} + \text{FCAR 3} + \text{FCAR 4} + \text{FCAR 5}} \\
 g_2(y) &= \frac{0.6217}{0 + 0.6217 + 0.0939 + 0 + 0} \\
 &= 0.8688
 \end{aligned}$$

Karena *discriminant function* untuk kelas 2 lebih besar dari pada kelas 1 maka data uji diklasifikasikan dalam kelas 2 yaitu tidak mengikuti *training*

3.6 Perancangan Uji Coba

Pengujian dilakukan untuk mengetahui tingkat akurasi dari hasil klasifikasi evaluasi kinerja karyawan PT. Sierad Produce, Tbk menggunakan metode *Classification Based on Fuzzy Association Rule* (CFAR) serta mengetahui pengaruh nilai *minimum support* dan *minimum confidence* terhadap jumlah aturan (*rule*) yang dihasilkan.

3.6.1 Pengujian jumlah *rule* yang dibangkitkan

Pengujian ini untuk mengetahui jumlah *rule* yang dibangkitkan dari pasangan nilai *minimum support* dan *minimum confidence* yang berbeda pada proses pembangkitan *rule*. Nilai *minimum support* yang digunakan adalah 15%, 20%, 25%, 30%, 35%, 40%, 45% dan 50%. Sedangkan *minimum confidence* yang

digunakan 50%, 55%, 60%, 65%, 70%, 80%, 85%, 90%, 95% dan 100%. Perubahan nilai *minimum support* dan *minimum confidence* dilakukan pada 3 data latih dengan jumlah yang berbeda yaitu 100 *record*, 200 *record*, dan 300 *record* (semua data). Pada dua data latih pertama, data dipilih secara acak sebanyak 10 kali perulangan. Kemudian rata-rata jumlah *rule* yang terbentuk dalam setiap proses akan dicatat dalam tabel 3.35. Begitu pula untuk *rule* yang terbentuk menggunakan semua data, hasilnya akan dicatat dalam tabel 3.35.

Tabel 3.35 Tabel Uji Pengaruh *Minimum support* dan *Minimum confidence* Terhadap Jumlah *Rule* yang Dihasilkan

MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50								
55								
60								
...								
90								
95								
100								

3.6.2. Pengujian akurasi klasifikasi

Pengujian ini menggunakan *rule* yang dibangkitkan untuk setiap perubahan nilai *minimum support* dan *minimum confidence* yang dilakukan pada 3 data latih dan telah melewati proses membangun *classifier* pada pengujian sebelumnya. *Rule* dalam *classifier* diuji menggunakan data uji masing-masing atau data itu sendiri untuk 3 data latih. Hasil perbandingan tersebut kemudian dihitung dengan rumus 2.19. Hasil perhitungan akurasi kemudian disimpan dalam tabel 3.36.

Tabel 3.36 Tabel Uji Pengaruh *Minimum Support* dan *Minimum Confidence* Terhadap Akurasi

MinConf (%)	Akurasi							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50								
55								
60								
...								
90								
95								
100								

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Implementasi merupakan representasi rancangan berupa aplikasi klasifikasi penilaian kinerja karyawan yang menerapkan metode *Classification based on Fuzzy Association Rule* (CFAR). Adapun yang akan dijelaskan dalam subbab ini meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan dan pengujian aplikasi klasifikasi penilaian kinerja karyawan ini adalah sebagai berikut :

1. Prosesor Intel(R) Core(TM) 2 Duo CPU T6600 @ 2.20 GHz.
2. Memori 2 Gb
3. Harddisk 320 Gb
4. Monitor 12"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan program enkripsi citra digital dan uji coba adalah:

1. Sistem operasi *Windows 7* 64-bit sebagai tempat aplikasi dijalankan.
2. *NetBeans IDE 7.0* sebagai *programming software development* dalam pembuatan aplikasi klasifikasi penilaian kinerja karyawan.
3. Bahasa pemrograman *Java*

4.2 Implementasi Program

Berdasarkan perancangan perangkat lunak yang telah diuraikan pada bab 3, maka akan dibahas mengenai implementasi program sesuai dengan rancangan.

Sistem diimplementasikan pada kelas utama yang menampung seluruh proses. Selanjutnya tiap-tiap proses dilakukan oleh kelas lain sesuai dengan fungsi dan pembagiannya. Kelas-kelas yang dibangun akan dijelaskan dalam tabel 4.1

Tabel 4.1 Kelas-kelas yang Dibangun

No.	Nama class	Keterangan
1.	ExcelImport.java	Kelas ini digunakan untuk <i>import</i> data dari <i>excel</i> (.xls)
2.	ItemViewer.java	Kelas ini digunakan untuk menampilkan bentuk tabel dari Obyek <i>Attribute</i>
3.	Table.java	Kelas merupakan bentuk struktur tabel yang menurun sifat dari <i>DefaultTableModel</i> yang mana merupakan komponen penyusun <i>grid</i> dalam <i>JTable</i>
4.	Chooser.java	<i>Subclass</i> dari <i>JFileChooser</i> yang digunakan untuk memilih dan menyimpan <i>file</i>
5.	Filter.java	Kelas ini dibutuhkan <i>Chooser</i> untuk menyaring <i>file</i> yang dibutuhkan
6.	XMLProcessor.java	Kelas ini digunakan untuk membaca dan menulis xml
7.	CFAR.java	Kelas ini digunakan untuk menjalankan seluruh proses
8.	Mainframe.java	Kelas ini digunakan untuk menjalankan proses <i>generate rule</i>
9.	Tester.java	Kelas ini digunkkan untuk mengklasifikasikan data berdasarkan <i>rule</i>
10.	Norm.java	Kelas ini digunakan untuk mendapatkan data ternormalisasi
11.	Term.java	Kelas ini digunakan untuk menghitung nilai derajat keanggotaan tiap parameter
12.	Atrribute.java	Kelas ini digunakan untuk menyimpan data

		atribut, indeksnya berupa nomor dan <i>fuzzy set</i> var linguistik, dan nilainya berupa data ternormalisasi
13.	HashMap2D.java	Kelas ini merupakan bentuk <i>map</i> dengan dua indeks
14.	ItemSet.java	Kelas <i>ItemSet</i> merupakan representasi dari <i>itemset</i> strukturnya berupa <i>map</i> yang mana <i>keysetlist</i> sebagai index dan kardinalitas sebagai nilainya
15.	Keyset.java	Kelas <i>Keyset</i> merupakan representasi dari kunci penyusun <i>itemset</i>
16.	KeySetList	Kelas <i>KeySetList</i> merupakan kumpulan dari <i>keyset</i> dan merupakan kunci pada <i>itemset</i>
17.	ValueComparator	<i>Comparator</i> merupakan pembanding nilai pada <i>TreeMap</i> sehingga kelas ini dapat digunakan untuk mengurutkan nilai-nilai dalam map

Untuk penjelasan *method* yang terdapat pada masing-masing kelas akan diuraikan dalam tabel 4.2-4.18.

Tabel 4.2 Kelas ExcelImport.java

No	Method	Keterangan
1.	ExceiImport()	Menampilkan komponen untuk tampilan GUI
2.	pilihActionPerfor med(ActionEvent evt)	Membaca <i>sheet</i> yang dipilih
3.	read(Sheet sheet)	Membaca <i>sheet</i> yang terpilih dari <i>workbook</i>
4.	read()	Membaca <i>file excel</i> kemudian mendaftar <i>sheet</i> apa saja yang ada di dalamnya
5.	readActionPerfor med(ActionEvent evt)	Membuka data (<i>file chooser</i>), mendapatkan dan membaca <i>file</i>

Tabel 4.3 Kelas ItemViewer.java

No	Method	Keterangan
1.	ItemViewer(Atrribute att, byte I)	Konstruktor dengan argumen <i>attribute</i> dan tipe isi <i>attribute</i> karena obyek <i>Attribute</i> dapat berisi data atribut dan target maka disediakan dua macam I
2.	ItemViewer(ItemSet set)	Menampilkan hasil gabungan <i>key1</i> dan <i>key2</i> sejumlah <i>KeySet</i> yang ada dalam <i>itemset</i>
3.	setTitle(String con)	Menampilkan judul pada label

Tabel 4.4 Kelas Table.java

No	Method	Keterangan
1.	Table(Object[][] data, String[] title)	Disusun dari judul tabel dan data tabel
2.	Editable() : Boolean[]	Mengedit isi kolom
3.	getColumnClass(int column) : Class	Mengembalikan objek kelas yang mewakili kelas <i>runtime</i> dari objek
4.	isCellEditable(int rowIndex, int columnIndex) : Boolean	Mengedit data pada baris dan kolom tertentu

Tabel 4.5 Kelas Chooser.java

No	Method	Keterangan
1.	Chooser(byte filter)	Disusun berdasarkan jenis <i>filter</i> yang digunakan
2.	getFile() : File	Mendapatkan <i>file</i> hasil membuka
3.	getStatus() : int	Mendapatkan status pengambilan <i>file</i> (<i>return</i> diterima atau gagal)
4.	open()	Membuka dialog untuk membuka <i>file</i>
5.	save()	Membuka dialog untuk menyimpan <i>file</i> dan

	menambahkan ekstensi pada <i>file</i> jika tidak terdapat ekstensi pada namanya
--	---

Tabel 4.6 Kelas `Filter.java`

No	Method	Keterangan
1.	<code>Filter(byte con)</code>	Hanya terdapat dua jenis <i>file</i> yang dapat disaring, yakni <i>file</i> .xls dan <i>file</i> .rule
2.	<code>accept(File f) : boolean</code>	Menetapkan <i>file</i> yang dapat ditampilkan pada <i>chooser</i> , kemudian <i>file</i> yang ditampilkan hanyalah <i>file</i> dengan ekstensi yang ditentukan
3.	<code>getDescription() : String</code>	Menampilkan deskripsi ekstensi <i>file</i> yang disaring / diperbolehkan ditampilkan

Tabel 4.7 Kelas `XMLProcessor.java`

No	Method	Keterangan
1.	<code>XMLReader(String filename) : Itemset</code>	Membaca <i>file XML</i> kemudian dibentuk kembali berupa <i>itemset</i>
2.	<code>Convert(KeySetList k) : String</code>	Mendapatkan bentuk <i>String</i> dari <i>KeySetList</i>
3.	<code>getTagValue(String sTag, Element eElement) : String</code>	Membaca <i>fileXML</i> kemudian dibentuk kembali berupa <i>itemset</i>
4.	<code>Parse(String tagValue) : KeySetList</code>	Mendapatkan <i>KeySetList</i> dari <i>String</i>
5.	<code>Write(ItemSet set, String namaFile)</code>	Menuliskan data dalam <i>itemset</i> ke dalam <i>XML</i>

Tabel 4.8 Kelas CFAR.java

No	Method	Keterangan
1.	Main(String[] args)	Menjalankan program (mengubah tampilan antarmuka dengan fitur <i>look and feel java</i> dan memanggil kelas <i>tester</i>)

Tabel 4.9 Kelas MainFrame.java

No	Method	Keterangan
1.	MainFrame	Jika <i>main</i> nilainya <i>null</i> maka kelas ini diawali dengan inisialisasi <i>main</i>
2.	DF(KeySetList key, int index) : double	Menghitung DF
3.	backButtonPerformed(ActionEvent evt)	Kembali ke <i>tab</i> sebelumnya
4.	compatibility(ArrayList<KeySet> keySets) : double	Menghitung nilai kompatibilitas suatu <i>keysetlist</i>
5.	count_support(ItemSet set) : ItemSet	Proses perhitungan <i>support</i> untuk <i>itemset</i> selanjutnya
6.	create_fcar()	Proses menciptakan FCAR berdasarkan <i>itemset</i> yang telah terseleksi
7.	enableTab(int idx)	Sistem tampilan berupa tabulasi <i>enableTab</i> merupakan pengontrol <i>tab</i> mana yang bisa dibuka
8.	exportActionPerformed(ActionEvent evt)	Mengekspor <i>file</i>
9.	First_count_support(ItemSet set) : Itemset	proses perhitungan <i>support</i> untuk <i>itemset-1</i> , dipisahkan karena <i>itemset-1</i> belum memiliki <i>keyset</i> target
10.	getInstance() : MainFrame	Menampilkan <i>mainframe</i> pada kelas <i>tester</i>

11.	mainTabStateChanged (ChangedEvent evt)	Proses setiap <i>tab</i>
12.	nextButtonActionPerformed (ActionEvent evt)	Menuju <i>tab</i> selanjutnya
13.	normalize_data()	Proses normalisasi data
14.	print (ItemSet set)	Mencetak <i>itemset</i>
15.	select_rule() : boolean	menyeleksi <i>itemset</i> yang akan dijadikan <i>rule</i> melalui proses <i>classifier</i>
16.	Create_candidate	Membentuk kandidat untuk setiap <i>itemset</i>

Tabel 4.10 Kelas `Tester.java`

No	Method	Keterangan
1.	Tester()	Menampilkan komponen untuk tampilan GUI
2.	DF (KeySetList key, int index) : double	Menghitung nilai DF
3.	exeButtonActionPerformed (ActionEvent evt)	Mengklasifikasikan tiap data pada uji banyak data
4.	genButtonActionPerformed (ActionEvent evt)	Menampilkan <i>mainframe</i>
5.	initComponents()	Menampilkan komponen untuk tampilan GUI
6.	openButtonActionPerformed (ActionEvent evt)	Membuka (memilih) <i>file</i>
7.	process_single (int index) : boolean	Mengklasifikasikan data tunggal

Tabel 4.11 Kelas Norm.java

No	Method	Keterangan
1.	<code>getNormalizedData(String[] imported_expert, Object[] [] imported_data) : HashMap<String, Attribute></code>	Menghitung nilai normalisasi, kemudian disimpan pada <i>HashMap</i>

Tabel 4.12 Kelas Term.java

No	Method	Keterangan
1.	<code>ups_B(double x) : double</code>	Menghitung nilai derajat keanggotaan untuk parameter Baik
2.	<code>ups_C(double x) : double</code>	Menghitung nilai derajat keanggotaan untuk parameter Cukup
3.	<code>ups_K(double x) : double</code>	Menghitung nilai derajat keanggotaan untuk parameter Kurang
4.	<code>ups_SB(double x) : double</code>	Menghitung nilai derajat keanggotaan untuk parameter Sangat Baik
5.	<code>ups_SK(double x) : double</code>	Menghitung nilai derajat keanggotaan untuk parameter Sangat Kurang

Tabel 4.13 Kelas Attribute.java

No	Method	Keterangan
1.	<code>Attribute(String expert)</code>	Mengambil atribut (setiap atribut memiliki nama)
2.	<code>get_card_value(String key) : double</code>	mendapatkan nilai kardinalitas dari suatu atribut berdasarkan variabel <i>linguistik</i>
3.	<code>get_expert_name() : String</code>	Mengambil nama atribut

Tabel 4.14 Kelas `HashMap2D.java`

No	Method	Keterangan
1.	<code>HashMap2D()</code>	Memanggil nilai dan menyimpan pada <i>HashMap</i>
2.	<code>clear()</code>	Mengosongkan isi <i>map</i>
3.	<code>column_keys(Row_Key key) : Set<Column_Key></code>	Mendapatkan daftar indeks kedua dari suatu indeks pertama
4.	<code>column_length(Row_Key key) : int</code>	Mendapatkan ukuran kolom <i>map</i>
5.	<code>containsKeys(Row_Key key) : Boolean</code>	Mengecek apakah <i>map</i> memiliki kunci yang dicari
6.	<code>containsKeys(Row_Key key1, Column_Key key2) : boolean</code>	Pengecekan <i>key</i> dalam baris dan kolom
7.	<code>get(Row_Key key1, Column_Key key2) : Value</code>	Mendapatkan nilai atau <i>value</i> dari <i>map</i> dengan kedua indeks sebagai argumen
8.	<code>put(Row_Key key1, Column_Key key2, Value value) : Value</code>	Memasukkan suatu nilai ke dalam <i>map</i>
9.	<code>row length() : int</code>	Mendapatkan ukuran baris <i>map</i>
10.	<code>toString() : String</code>	Mendapatkan bentuk <i>string</i> dari seluruh elemen <i>map</i>

Tabel 4.15 Kelas `Itemset.java`

No	Method	Keterangan
1.	<code>Itemset(Itemset itemset)</code>	Memanggil konstraktor pada <i>superclass</i> dengan satu parameter
2.	<code>Itemset()</code>	Memanggil konstraktor pada <i>superclass</i>
3.	<code>isUncontinueable() : Boolean</code>	Mengecek apakah <i>itemset</i> dapat diteruskan ke <i>itemset</i> selanjutnya

4.	<code>put(KetSet key, Double val)</code>	Memasukkan <i>keysetlist</i> ke dalam <i>itemset</i> , <i>keysetlist</i> -nya dibentuk dari <i>keyset</i> tunggal
5.	<code>select(double value) : ItemSet</code>	Mendapatkan <i>itemset</i> baru melalui seleksi nilai kardinalitas masukkan berupa nilai <i>support</i> atau <i>confidence</i>

Tabel 4.16 Kelas `keySet.java`

No	Method	Keterangan
1.	<code>KeySet(String key1, String key2)</code>	Mengambil nilai parameter yang dimasukkan di method ini dan disimpan pada <i>variablekey1</i> dan <i>key2</i>
2.	<code>KeySet(Object key1, String key2)</code>	Mengambil nilai parameter yang dimasukkan di method ini dan disimpan pada <i>variablekey1</i> dan <i>key2</i>
3.	<code>equals(Object obj) : boolean</code>	Dua <i>keyset</i> dinyatakan sama jika <i>key1</i> dan <i>key2</i> -nya sama
4.	<code>hashCode() : int</code>	Mendapatkan kode <i>hash</i> untuk mencocokkan <i>object</i>
5.	<code>toString() : String</code>	Mendapatkan bentuk <i>string</i> dari <i>keyset</i>

Tabel 4.17 Kelas `keySetList.java`

No	Method	Keterangan
1.	<code>KetSetList(KeySetList key)</code>	Membentuk <i>keysetlist</i> melalui <i>keysetlist</i> lain
2.	<code>KeySetList()</code>	Memanggil konstruktor pada <i>superclass</i>
3.	<code>combine(KeySetList A, KeySetList B) : KeySetList</code>	Membentuk <i>keysetlist</i> baru melalui dua <i>keysetlist</i> * dengan syarat tidak ada yang dimasukkan ke dalam <i>keysetlist</i> baru adalah <i>keyset</i> yang unik (tidak ganda)
4.	<code>getNonTargetKeySets() :</code>	Mendapatkan <i>keyset-keyset</i> non target (yang berhubungan dengan atribut)

	KeySetList	
5.	getTarget() : String	Mendapatkan <i>String</i> dari <i>keyset</i> target (C1 / C2)

Tabel 4.18 Kelas `ValueComparator.java`

No	Method	Keterangan
1.	<code>ValueComparator(</code> <code>Itemset base)</code>	Mendapatkan nilai-nilai dalam <i>map</i>
2.	<code>Compare(Object</code> <code>a, Object b) :</code> <code>int</code>	Mengurutkan nilai-nilai dalam <i>map</i>

4.2.1 Implementasi Pembentukan *Rule*

4.2.1.1 Implementasi *Load Data*

Tahap awal yang dilakukan dalam proses pelatihan data evaluasi kinerja karyawan ini adalah mengambil data yang dijadikan data latih dengan membaca *file excel*. Setelah itu data dalam *file* tersebut disimpan dalam suatu kelas atribut yang berisi variabel-variabel untuk penyimpanan setiap atribut dalam evaluasi kinerja karyawan. Implementasi baca *file* data latih ditunjukkan pada *source code* 4.1.

```
Private void readActionPerformed(java.awt.event.ActionEvent
evt) {
    Chooser open = new Chooser(Filter.EXCEL);
    //membuka file chooser
    open.open();
    if(open.getStatus() == Chooser.APPROVE_OPTION) {
        inputWorkbook = open.getFile();
    }
    //mendapatkan file
    read();
    //membaca file
}
}
```

Source Code 4.1 Load Data

4.2.1.2 Implementasi transformasi ke data *fuzzy*

Tahap selanjutnya adalah transformasi ke data *fuzzy* untuk mengubah nilai *crisp* menjadi nilai *fuzzy* dari setiap atribut sehingga memiliki derajat keanggotaan di setiap atribut *value*. Setiap atribut memiliki 5 atribut *value* yang sama, yaitu sangat kurang, kurang, cukup, baik, dan sangat baik. Implementasi transformasi ke data *fuzzy* ditunjukkan pada *source code* 4.2.

```
public class Term {
    public static double ups_SK(double x) {
        if (x < 20) {
            return 1;
        } else if (x >= 20 && x <= 35) {
            return (35 - x) / 15.0;
        } else {
            return 0;
        }
    }

    public static double ups_K(double x) {
        if (x > 35 && x < 44) {
            return 1;
        } else if (x >= 20 && x <= 35) {
            return (x - 20) / 15.0;
        } else if (x >= 44 && x <= 56) {
            return (56 - x) / 12.0;
        } else {
            return 0;
        }
    }

    public static double ups_C(double x) {
        if (x > 56 && x < 64) {
            return 1;
        } else if (x >= 44 && x <= 56) {
            return (x - 44) / 12.0;
        } else if (x >= 64 && x <= 76) {
            return (76 - x) / 12.0;
        } else {
            return 0;
        }
    }

    public static double ups_B(double x) {
        if (x > 76 && x < 83) {
            return 1;
        } else if (x >= 64 && x <= 76) {
            return (x - 64) / 12.0;
        } else if (x >= 83 && x <= 92) {
            return (92 - x) / 9.0;
        } else {
            return 0;
        }
    }

    public static double ups_SB(double x) {
        if (x > 92) {
            return 1;
        } else if (x >= 83 && x <= 92) {
            return (x - 83) / 9.0;
        } else {
            return 0;
        }
    }
}
```

Source Code 4.2 Transformasi ke Data Fuzzy

4.2.1.3 Implementasi Perhitungan Normalisasi

Proses normalisasi dilakukan untuk semua atribut kecuali atribut kelas. Tahap yang dilakukan adalah menjumlahkan derajat keanggotaan setiap Term.ups menggunakan fungsi denm. Kemudian, derajat keanggotaan setiap Term.ups dibagi dengan hasil penjumlahan tersebut. Proses ini dapat dilihat pada *source code* 4.3.

```
//kelas norm digunakan untuk mendapatkan data ternormalisasi
public class Norm {
    public static HashMap<String, Attribute>
    getNormalizedData(String[] imported_expert, Object[][]
    imported_data){
        HashMap<String, Attribute> normalized_data = new
        HashMap<String, Attribute>();
        //untuk setiap attribute
        for(int i = 0; i < imported_expert.length - 1; i++){
            String key = imported_expert[i+1];
            Attribute value = new Attribute(key);
            for(int j = 0; j < imported_data.length; j++){
                if(imported_data[j][0].toString().equals(""))
                continue;
                double fuzzy =
                Double.parseDouble(imported_data[j][i+1].toString());
                //proses perhitungan derajat keanggotaan
                double sk = Term.ups_SK(fuzzy),
                k = Term.ups_K(fuzzy),
                c = Term.ups_C(fuzzy),
                b = Term.ups_B(fuzzy),
                sb = Term.ups_SB(fuzzy),
                denm = sk + k + c + b + sb;
                //nilai derajat keanggotaan ternormalisasi disimpan dalam
                Attribute
                value.put(j, Attribute.SK, sk / denm);
                value.put(j, Attribute.K, k / denm);
                value.put(j, Attribute.C, c / denm);
                value.put(j, Attribute.B, b / denm);
                value.put(j, Attribute.SB, sb / denm);
            }
            //dan attribute disimpan ke dalam map
            normalized_data.put(key, value);
        }
        return normalized_data;
    }
}
```

Source Code 4.3 Perhitungan Normalisasi

4.2.1.4 Implementasi Pembangkitan Rule

Proses pembangkitan *rule* diawali dengan membangkitkan frequent 1-*itemsets* yang merupakan proses untuk mendapatkan atribut atau *item* yang memenuhi batasan *minimum support*. Frequent 1-*itemsets* yang telah didapat dari proses sebelumnya digunakan untuk membentuk kombinasi kandidat 2-*itemsets* menggunakan fungsi `create_candidate()` untuk menemukan frequent 2-*itemsets*. Setiap kombinasi kandidat *itemsets* yang dibentuk menggunakan fungsi `create_candidate()` akan diperiksa apakah berasal dari atribut yang sama, memiliki atribut kelas *support* kandidat lebih atau sama dengan *minimum support*. Jika kandidat memenuhi syarat tersebut maka akan dimasukkan dalam new `ItemSet()`. Proses pembangkitan ditunjukkan oleh fungsi `create_candidate()` pada *source code 4.4*.

```
//membentuk kandidat untuk setiap itemset
private void create_candidate(){
    double support = Double.parseDouble(supField.getText());
    all_key_set = new ItemSet();
    itemPanel.removeAll();
//itemset-1
    ItemSet newSet = card_data_set.select(support);
    if(!newSet.isUncontinuable()){
        int i = 1;
        ItemViewer view = new ItemViewer(newSet);
        view.setTitle("Itemset-"+(i++));
        itemPanel.add(view);
//itemset-2 setelah itemset-1 melalui perhitungan support
//dan seleksi support
        newSet =
            first_count_support(newSet).select(support);
        while(!newSet.isUncontinuable()){
//memasukkan itemset baru ke dalam daftar itemset (all_key_set)
            all_key_set.putAll(newSet);
            view = new ItemViewer(newSet);
            view.setTitle("Itemset-"+(i++));
            itemPanel.add(view);
//itemset selanjutnya diperoleh melalui perhitungan support
//dan seleksi support
            ItemSet newSet2 =
                count_support(newSet).select(support);
            newSet = new ItemSet(newSet2);
        }
    }
}
```

Source Code 4.4 Pembentukan Kandidat Itemset

Frequent 1-itemsets didapat dengan cara menghitung kardinalitas skalar setiap *items*. Perhitungan kardinalitas skalar dilakukan untuk setiap *KeySet* dengan menjumlahkan derajat keanggotaan setiap *record* data. Nilai hasil perhitungan kardinalitas skalar setiap *KeySet* kemudian dibandingkan dengan *minimum support* yang diberikan. Jika nilai kardinalitas skalar lebih besar atau sama dengan *minimum support* maka *items* akan disimpan dalam *new ItemSet()*. Proses ini dapat dilihat pada *source code* 4.5.

```
//mendapatkan itemset baru melalui seleksi nilai kardinalitas
//inputan berupa nilai support atau confidence
public ItemSet select(double value){
    ItemSet newSet = new ItemSet();
    for(KeySetList i : keySet()){
        double val = get(i);
        if(val >= value)
            newSet.put(i, val);
    }
    return newSet;
}

//proses perhitungan support untuk itemset-1 dipisahkan karena
//itemset-1 belum memiliki keyset target
private ItemSet first_count_support(ItemSet set){
    ItemSet newSet = new ItemSet();
    for(KeySetList i : set.keySet()){
        KeySetList list1 = new KeySetList(i),
        list2 = new KeySetList(i);
//ditambahkan keyset target
        list1.add(KeySetList.keySet1);
        list2.add(KeySetList.keySet2);
        newSet.put(list1, compatibility(list1));
        newSet.put(list2, compatibility(list2));
    }
    return newSet;
}
```

Source Code 4.5 Pembangkitan *Frequent 1-itemsets*

Proses membentuk *rule* diawali dengan mencari *frequent k-itemsets*. Proses ini diimplementasikan dalam kelas membentuk *rule.Frequent 1-itemsets* yang telah didapat dari proses sebelumnya digunakan untuk membentuk kombinasi kandidat *2-itemsets* menggunakan fungsi *create_candidate()* untuk menemukan *frequent 2-itemsets*. Setelah mendapatkan *frequent 2-itemsets*, membentuk kandidat *k-itemsets* dan seterusnya hingga tidak ada lagi kandidat dalam *frequent k-itemsets* yang dihasilkan. Proses pencarian *frequent k-itemsets* dapat dilihat pada *source code* 4.6.

```

private ItemSet count_support(ItemSet set){
    ItemSet newSet = new ItemSet();
    int i = 0;
    for(KeySetList a : set.keySet()){
        int j = 0;
        for(KeySetList b : set.keySet()){
            //untuk membentuk itemset baru diperlukan pengecekan apakah
            //kelas target sudah sama atau tidak, jika sama maka dapat
            //dihitung supportnya
            if(j > i && ((a.contains(KeySetList.keySet1) &&
                b.contains(KeySetList.keySet1)) ||
                (a.contains(KeySetList.keySet2) &&
                b.contains(KeySetList.keySet2)))){
                KeySetList _new = KeySetList.combine(a, b);
                newSet.put(_new, compatibility(_new));
            }
            j++;
        }
        i++;
    }
    return newSet;
}

```

Source Code 4.6 Pembangkitan Frequentk-itemsets

Setelah tidak ditemukan lagi kandidat yang *frequent*, maka proses selanjutnya adalah membuat *rule* dari *frequent k-itemsets* mulai dari 2-itemsets. Setiap *rule* yang dibentuk kemudian dihitung nilai *confidencenya*. Nilai *confidence* yang lebih atau sama dengan *minimum support* akan disimpan dalam *conf_data_set*. Proses membangkitkan *rule* dapat dilihat pada *source code 4.7*.

```

private void create_fcar(){
    double confidence =
    Double.parseDouble(conField.getText());
    conf_data_set = new ItemSet();
    //menyeleksi seluruh isi all_key_set berdasarkan nilai
    //confidence, jika lolos maka dimasukkan ke dalam conf_data_set
    for(KeySetList i : all_key_set.keySet()){
        KeySetList j = i.getNonTargetKeySets();
        double conf = all_key_set.get(i) / compatibility(j);
        if(conf >= confidence)
            conf_data_set.put(i, conf);
    }
    //membentuk tabel untuk menampilkan isi conf_data_set dan hasil
    //perhitungan confidence
    int n = 0;
    Object[][] card_data = new
    Object[conf_data_set.size()][4];
    tgt = new HashSet<KeySet>();
    for(KeySetList i : conf_data_set.keySet()){
        tgt.addAll(i);
        card_data[n][0] = n + 1;
        card_data[n][1] = i.toString();
    }
}

```

```

        card_data[n][2] = all_key_set.get(i);
        card_data[n][3] = conf_data_set.get(i);
        n++;
    }
    Table tab = new Table(card_data, conf_title);
    fcarTable.setRowHeight(30);
    fcarTable.setModel(tab);
    fcarTable.revalidate();
}

```

Source Code 4.7 Membuat Rule

4.2.1.5 Implementasi Classifier

Proses membangun *classifier*, langkah pertama yang dilakukan ialah menghitung nilai DF untuk setiap *rule* dan hasilnya disimpan dalam dfs. Perhitungan nilai DF ditunjukkan pada *source code 4.8*

```

private double DF(KeySetList key, int index){
    double sum = 0;
    KeySetList _selected = key.getNonTargetKeySets();
    for(KeySet i : _selected){
        Attribute att = normalized_data.get(i.key1);
        double val = att.get(index, i.key2);
        sum += val * conf_data_set.get(key);
    }
    return sum / (double) _selected.size();
}

```

Source Code 4.8 Perhitungan DF

Kemudian nilai DF diurutkan dari terbesar ke terkecil menggunakan *method* `srt.putAll(dfs).Rule` pertama dalam urutan digunakan untuk mengklasifikasi data uji. Jika kelas *rule* pertama sama dengan kelas pada data latih maka nilai *rRightrule* bertambah. Jika tidak maka *rule* tersebut diletakkan pada akhir urutan dan *rWrongrule* bertambah. Proses ini terdapat *method* `select_rule()` dan dapat dilihat pada *sourcecode 4.9*.

Setelah semua *rule* dilatih, *rule* yang nilai *rRight* lebih dari 0 akan dimasukkan dalam *classifier* dan dihitung nilai *wk*. Selama *rule* dalam *classifier* lebih dari 1, *rule* dengan nilai *wk* terkecil akan dihapus dari *classifier*. Proses *Classifier* ditunjukkan pada *source code 4.9*.

```

//menyeleksi itemset yang akan dijadikan rule melalui proses
classifier
    private boolean select_rule(){
        HashMap2D<KeySetList, Boolean, Integer>   rwk = new
HashMap2D<KeySetList, Boolean, Integer>();
        Attribute cluster = normalized_data.get("Kelas");
        //rwk menyimpan hasil classifier untuk setiap nRight dan nWrong
        for(KeySetList i : conf_data_set.keySet()){
            rwk.put(i, true, 0);
            rwk.put(i, false, 0);
        }
        //proses perhitungan classifier
        for(int i = 0; i < imported_data.length; i++){
            if(imported_data[i][0].toString().equals(""))
continue;
            ItemSet dfs = new ItemSet();
            for(KeySetList j : conf_data_set.keySet())
                dfs.put(j, DF(j, i));
            TreeMap<KeySetList, Double>   srt = new
TreeMap<KeySetList, Double>(new ValueComparator(dfs));
            srt.putAll(dfs);
            for(KeySetList j : srt.keySet()){
                if((cluster.get(i, Attribute.target1) == 1 &&
j.contains(KeySetList.keySet1)) ||
                (cluster.get(i, Attribute.target2) == 1 &&
j.contains(KeySetList.keySet2))){
                    int n = rwk.get(j, true) + 1;
                    rwk.put(j, true, n);
                    break;
                }
                else{
                    int n = rwk.get(j, false) + 1;
                    rwk.put(j, false, n);
                }
            }
        }
        //keysetlist dengan rWk terkecil akan dibuang
        KeySetList removed = null;
        double min = Double.MAX_VALUE;
        int n = 0, idx = 0;
        selected_data_set = new ItemSet(conf_data_set);
        Object[][] card_data = new Object[conf_data_set.size()][];
        for(KeySetList i : conf_data_set.keySet()){
            boolean sel = false;
            int rRight = rwk.get(i, true), rWrong = rwk.get(i,
false);

            double wk = Double.NaN;
            if(rRight == 0) selected_data_set.remove(i);
            else {
                wk = rRight / (double)(rRight + rWrong);
                if(wk < min){
                    min = wk;
                    removed = i;
                    idx = n;
                }
                sel = true;
            }
        }
    }

```

```

        card_data[n] = new Object[]{n+1, i.toString(), rRight,
rWrong, wk, sel};
        n++;
    }
    if(removed != null && !selected_data_set.isEmpty())
        selected_data_set.remove(removed);
    card_data[idx][5] = false;

    Table table = new Table(card_data, selected_title);
    ItemViewer view = new ItemViewer(selected_data_set);
    car1.setViewportView(new JTable(table));
    car2.setViewportView(view);

    return      tgt.contains(KeySetList.keySet1)      &&
tgt.contains(KeySetList.keySet2);
    }

    private void print(ItemSet set){
        System.out.println("-----Itemset-----");
        for(KeySetList i : set.keySet())
            System.out.println(i.toString()+" "+set.get(i));
    }

```

Source Code 4.9 Proses Classifier

4.2.2 Implementasi Pengujian

4.2.2.1 Implementasi Proses Klasifikasi

Klasifikasi terhadap data baru/uji bisa dilakukan selama ada *rule* yang dibangkitkan. Tahap awal yang dilakukan ialah memilih dan melakukan pembacaan data uji. Kemudian data uji di transformasi ke data *fuzzy* dan dinormalisasi seperti pada proses pelatihan. Setelah itu, dilakukan perhitungan *discriminant function* untuk setiap *rule* pada setiap *record* data. Hasil perhitungan *discriminant function* pada *record* ke-n kemudian dijumlahkan sesuai dengan kelas masing-masing. Jumlah *discriminant function* masing-masing kelas ini kemudian dibagi dengan jumlah seluruh *discriminant function* untuk menentukan hasil klasifikasi data uji. Jika jumlah *discriminant function* pada kelas C1 lebih besar daripada C2 maka data uji diklasifikasikan ke kelas C1. Sebaliknya, jika jumlah *discriminant function* pada kelas C2 lebih besar daripada C1 maka data uji diklasifikasikan ke kelas C2. Jika *discriminant function* kelas C1 sama dengan kelas C2 maka dilihat jumlah *rule* kelas mana yang paling banyak dibangkitkan. Setelah seluruh data uji diklasifikasi maka dihitung akurasi. Proses klasifikasi dapat dilihat pada *source code* 4.10.

```

private boolean process_single(int index){
    double target1 = 0, target2 = 0;
    for(KeySetList i : itemset.keySet()){
        double df = DF(i, index);
        if(itemset.get(i) == 1)
            target1 += df;
        else if(itemset.get(i) == 2)
            target2 += df;
    }
    double denom = target1 + target2;
    return target1/denom > target2/denom;
}

private double DF(KeySetList key, int index){
    double sum = 1;
    for(KeySet i : key){
        try{
            Attribute att = normalized_data.get(i.key1);
            sum *= att.get(index, i.key2);
        }
        catch (NullPointerException ex){}
    }
    return sum;
}

```

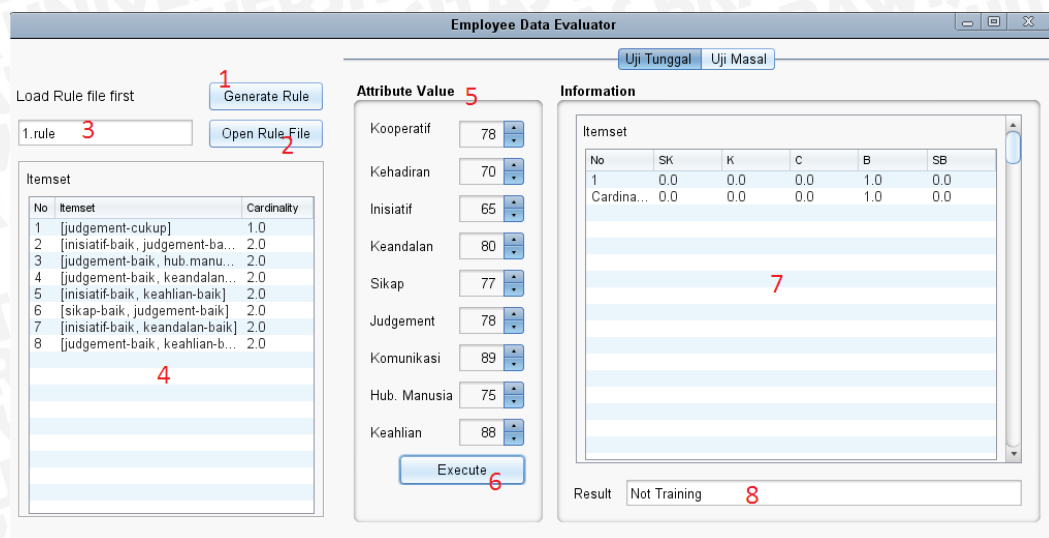
Source Code 4.10Proses Klasifikasi

4.3 Implementasi Antarmuka

Antarmuka (*interface*) untuk sistem ini terdiri dari dua bagian utama, yaitu bagian untuk pembentukan aturan yang bertujuan untuk membangkitkan aturan *fuzzy* (FCAR) dan bagian untuk pengujian yang bertujuan untuk melakukan pengujian terhadap aturan yang terbentuk.

4.3.1 Implementasi Antarmuka Pengujian

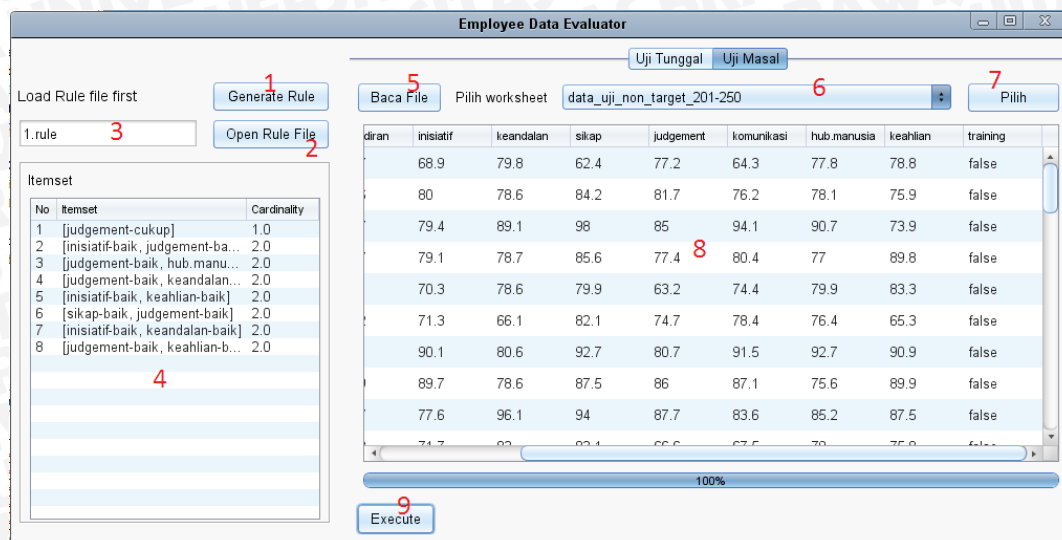
Dalam antarmuka pengujian akan dilakukan pengujian terhadap aturan yang terbentuk. Langkah awal dalam proses pengujian adalah mengambil *rule*(data) yang telah terbentuk, kemudian memasukkan data yang akan dilakukan pengujian. Hasil dari pengujian adalah target *training* atau tidak *training*. Data yang akan dilakukan pengujian dapat berbentuk data tunggal yaitu dengan memasukkan nilai atribut (satu data) yang ditunjukkan pada gambar 4.1. Untuk pengujian data massal, data yang akan dilakukan pengujian lebih dari dua dan bertipe .xls yang ditunjukkan pada gambar 4.2.



Gambar 4.1Antarmuka Proses Pengujian Data Tunggal

Pada gambar 4.1, antarmuka proses pengujian data tunggal :

1. Tombol *Generate Rule* untuk proses pembentukan aturan.
2. Tombol *Open Rule File* untuk mengambil *rule* yang telah disimpan(aturan yang terbentuk).
3. *Text field* untuk menampilkan lokasi data yang ditampilkan pada text area nomor 4.
4. *Panel* untuk menampilkan *rule* yang akan digunakan untuk pengujian data baru.
5. Untuk mengisi nilai untuk setiap atribut (data tunggal).
6. Tombol *excute* untuk melakukan proses pengujian.
7. *Panel* untuk menampilkan hasil proses pengujian data.
8. *Text field* menampilkan hasil pengujian.



Gambar 4.2Antarmuka Proses Pengujian Data Banyak

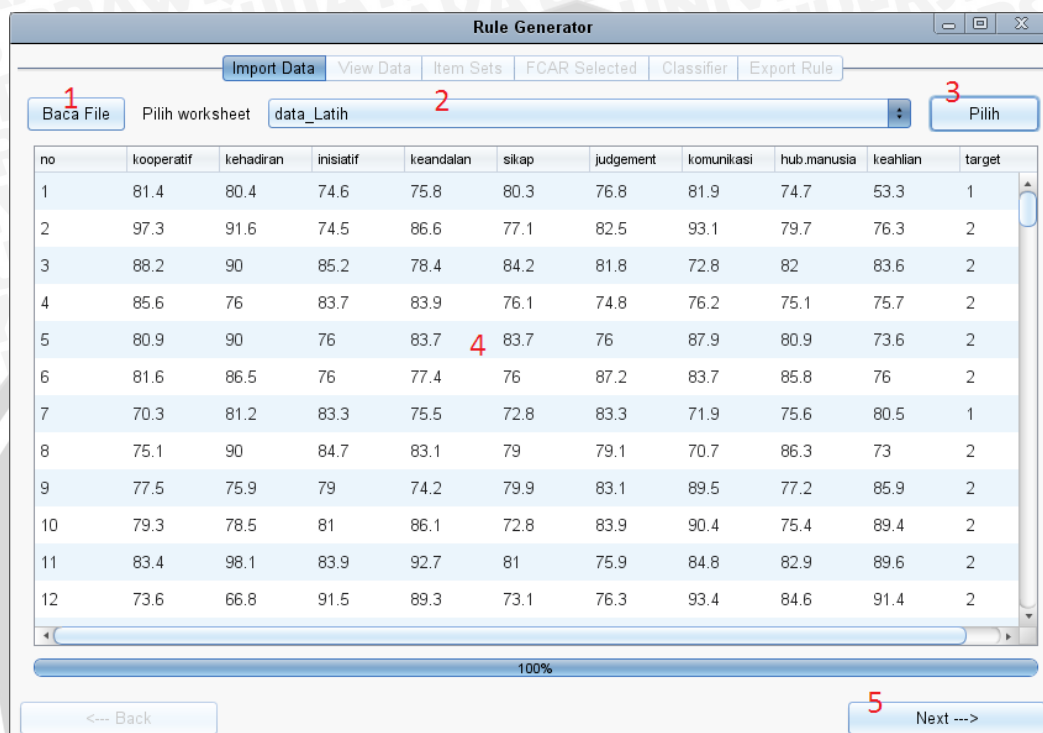
Pada gambar 4.2, antarmuka proses pengujian data banyak :

1. Tombol *Generate Rule* untuk proses pembentukan aturan.
2. Tombol *Open Rule File* untuk mengambil *rule* yang telah disimpan(aturan yang terbentuk).
3. *Text field* untuk menampilkan lokasi data yang ditampilkan pada *text area* nomor 4.
4. *Panel* untuk menampilkan *rule* yang akan digunakan untuk pengujian data baru.
5. Tombol *Baca File* untuk mengambil data yang akan dilakukan pengujian.
6. *Combo box* untuk menampilkan pilihan *worksheet* data yang akan diuji.
7. Tombol pilih untuk mengambil data yang akan dilakukan pengujian.
8. *Scrollpane* untuk menampilkan data yang akan dilakukan pengujian.
9. Tombol *Excute* untuk proses menampilkan hasil pengujian dan akan ditampilkan pada nomor 8.

4.3.2 Implementasi Antarmuka Pembentukan Aturan

Dalam antarmuka pembentukan aturan akan dilakukan pengujian terhadap aturan yang terbentuk. Langkah awal pada proses pembentukan aturan adalah mentransformasi data ke data *fuzzy*, proses normalisasi, pembentukan *itemsets*,

kemudian terbentuk aturan yang selanjutnya dilakukan proses *classifier*. Rulehasil *classifier* dapat disimpan untuk pengujian data baru. Antarmuka proses pembentukan aturan ditunjukkan pada gambar 4.3-4.8.



Gambar 4.3Antarmuka *Panel Import Data* pada Proses Pembentukan Aturan

Pada gambar 4.3, antarmuka *panelimport* data pada proses pembentukan aturan :

1. Tombol *Baca File* untuk mengambil data yang akan dilakukan pembentukan aturan.
2. *Combo box* menampilkan pilihan *worksheet* data yang akan ditampilkan.
3. Tombol *Pilih* untuk memulai menampilkan data.
4. *Scrollpane* menampilkan data yang akan dilakukan pembentukan aturan.
5. Tombol *Next* untuk proses selanjutnya yaitu *view data*.

Rule Generator

Import Data View Data Item Sets FCAR Selected Classifier Export Rule

Atribut: Kelas

No	Target 1	Target 2
1	1.0	0.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0
5	0.0	1.0
6	0.0	1.0
7	1.0	0.0
8	0.0	1.0
9	0.0	1.0
10	0.0	1.0
11	0.0	1.0
12	0.0	1.0
13	1.0	0.0
14	1.0	0.0
15	1.0	0.0
16	1.0	0.0
17	0.0	1.0

Atribut: kooperatif

No	SK	K	C	B	SB
1	0.0	0.0	0.0	1.0	0.0
2	0.0	0.0	0.0	0.0	1.0

Minimum Support : 0.25

<--- Back Next --->

Gambar 4.4Antarmuka *Panel View Data* pada Proses Pembentukan Aturan

Pada gambar 4.4, antarmuka *panel view* data pada proses pembentukan aturan :

1. *Scrollpane* untuk menampilkan data setelah dilakukan proses transformasi ke datafuzzy dan normalisasi.
2. *Text field* untuk mengisi nilai *minimumsupport*.
3. Tombol *back* untuk kembali pada proses sebelumnya.
4. Tombol *next* untuk proses selanjutnya yaitu *Itemsets*.

Rule Generator

Import Data View Data **Item Sets** FCAR Selected Classifier Export Rule

Itemset-1

No	Itemset	Cardinality
1	[judgement-cukup]	95.37641440057376
2	[hub.manusia-kurang]	0.4894181636168794
3	[sikap-kurang]	5.610378585570397
4	[kooperatif-baik]	192.8228358978472
5	[hub.manusia-baik]	190.52964426451587
6	[judgement-kurang]	7.861960150243385
7	[kooperatif-cukup]	41.76099621778068
8	[judgement-sangat baik]	23.43249975185048
9	[komunikasi-sangat baik]	55.23768043296271
10	[inisiatif-sangat baik]	30.2495345/6050277
11	[judgement-baik]	173.20379932657522
12	[komunikasi cukup]	78.19026226952719
13	[inisiatif-cukup]	79.20083010856894
14	[inisiatif-baik]	185.80199647161956
15	[keandalan-sangat baik]	46.7627669639915
16	[keandalan-cukup]	77.85258218687638
17	[kooperatif-kurang]	0.9417626907505452

Itemset-2

No	Itemset	Cardinality
1	[komunikasi-baik, Kelas-C2]	0.3251778013310356
2	[sikap-baik, Kelas-C1]	0.2854500115173471

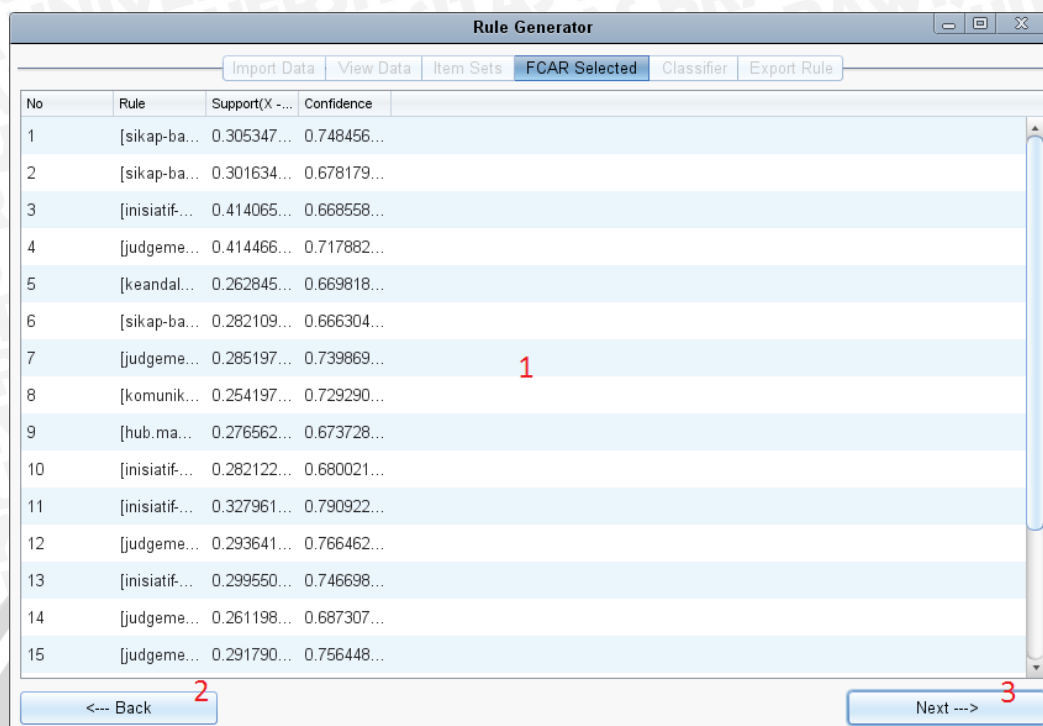
Minimum Confidence : 0.65

<--- Back Next --->

Gambar 4.5Antarmuka *Panel itemsets* pada Proses Pembentukan Aturan

Pada gambar 4.5, antarmuka *panel Item Sets* pada proses pembentukan aturan :

1. *Scrollpane* untuk menampilkan proses pembentukan *itemsets*.
2. *Text field* untuk mengisi nilai *minimumconfidence*.
3. Tombol *back* untuk kembali pada proses sebelumnya.
4. Tombol *next* untuk proses selanjutnya yaitu *FCAR selected*.



No	Rule	Support(X ...)	Confidence
1	[sikap-ba...	0.305347...	0.748456...
2	[sikap-ba...	0.301634...	0.678179...
3	[inisiatif...	0.414065...	0.668558...
4	[judgeme...	0.414466...	0.717882...
5	[keandal...	0.262845...	0.669818...
6	[sikap-ba...	0.282109...	0.666304...
7	[judgeme...	0.285197...	0.739869...
8	[komunik...	0.254197...	0.729290...
9	[hub.ma...	0.276562...	0.673728...
10	[inisiatif...	0.282122...	0.680021...
11	[inisiatif...	0.327961...	0.790922...
12	[judgeme...	0.293641...	0.766462...
13	[inisiatif...	0.299550...	0.746698...
14	[judgeme...	0.261198...	0.687307...
15	[judgeme...	0.291790...	0.756448...

Gambar 4.6Antarmuka *Panel FCAR selected* Pada Proses Pembentukan Aturan

Pada gambar 4.6, antarmuka panel *FCAR selected* pada proses pembentukan aturan :

1. *Panel* untuk menampilkan hasil pembentukan aturan (sementara).
2. Tombol *back* untuk kembali pada proses sebelumnya.
3. Tombol *next* untuk proses selanjutnya yaitu *Classifier*.

Rule Generator

Import Data View Data Item Sets FCAR Selected **Classifier** Export Rule

No	Rule	nRight	nWrong	rWk	Selected
1	[sikap-baik, Kelas-...	14	51	0.2153846153846154	true
2	[sikap-baik, Kelas-...	1	52	0.01886792452830...	false
3	[inisiatif-baik, Kela...	3	40	0.06976744186046...	true
4	[judgement-baik, K...	3	47	0.06	true
5	[keandalan-baik, K...	2	47	0.04081632653061...	true
6	[sikap-baik, Kelas-...	4	50	0.07407407407407...	true
7	[judgement-baik, K...	7	47	0.12962962962962...	true
8	[komunikasi-baik, ...	2	47	0.04081632653061...	true
9	[hub.manusia-baik, ...	3	48	0.05882352941176...	true
10	[inisiatif-baik, Kela...	1	49	0.02	true
11	[inisiatif-baik, Kela...	47	51	0.47959183673469...	true
12	[judgement-baik, K...	23	49	0.3194444444444444	true

Itemset

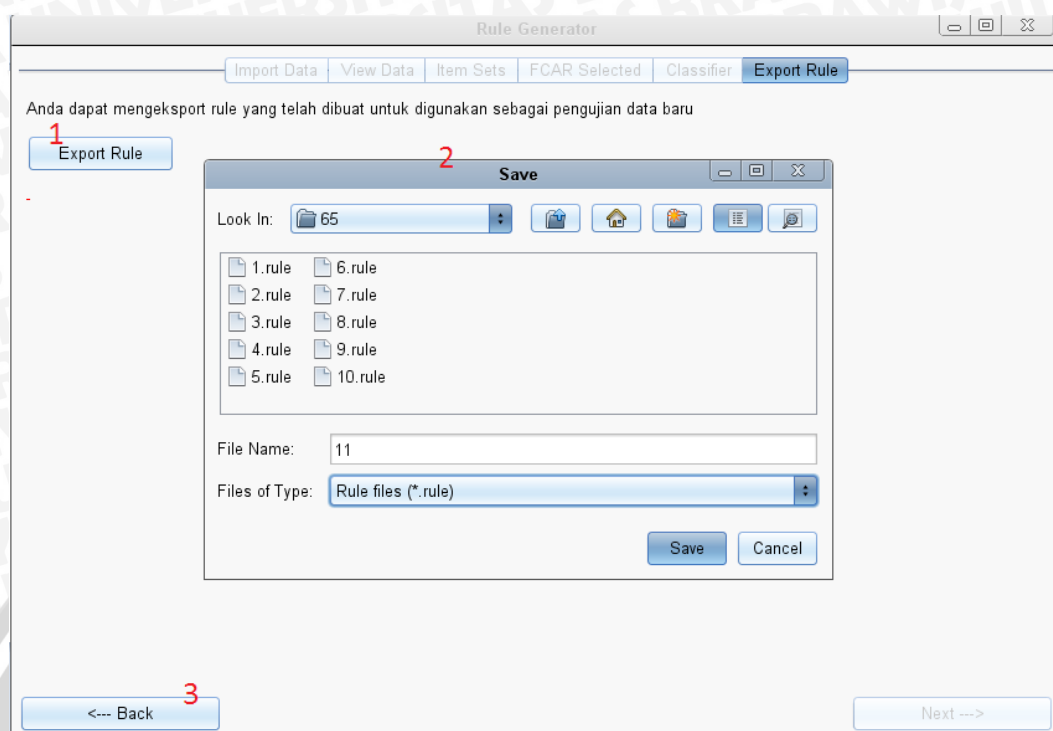
No	Itemset	Cardinality
1	[sikap-baik, Kelas-C2, judgement-baik]	0.7484560417659437
2	[inisiatif-baik, Kelas-C2]	0.6685589437978454
3	[keandalan-baik, Kelas-C2, hub.manusia...	0.6698180195858163
4	[judgement-baik, Kelas-C2]	0.7178820631372937
5	[sikap-baik, Kelas-C2, keahlian-baik]	0.6663042267102938
6	[judgement-baik, Kelas-C2, hub.manusia...	0.7398698410321817
7	[hub.manusia-baik, Kelas-C2, keahlian-b...	0.6737282027955381
8	[komunikasi-baik, Kelas-C2, judgement-...	0.729290441475454
9	[inisiatif-baik, Kelas-C2, hub.manusia-baik]	0.6800210502987525
10	[inisiatif-baik, Kelas-C2, judgement-baik]	0.7909224791147321
11	[inisiatif-baik, Kelas-C2, keandalan-baik]	0.7466986318250091

<--- Back Next >-->

Gambar 4.7Antarmuka *Panel Classifier* pada Proses Pembentukan Aturan

Pada gambar 4.7, antarmuka *panel classifier* pada proses pembentukan aturan :

1. *Scrollpane* untuk menampilkan aturan yang telah terbentuk (sebelum proses klasifikasi).
2. *Scrollpane* untuk menampilkan hasil pembentukan aturan setelah dilakukan proses *classifier* (klasifikasi).
3. Tombol *back* untuk kembali pada proses sebelumnya.
4. Tombol *next* untuk proses selanjutnya yaitu *Export rule*.



Gambar 4.8 Antarmuka *Panel Export Rule* pada Proses Pembentukan Aturan

Pada gambar 4.8, antarmuka *panel Export Rule* pada proses pembentukan aturan :

1. Tombol *Export Rule* untuk proses penyimpanan *rule* yang terklasifikasi.
2. Untuk menentukan lokasi penyimpanan *rule*.
3. Tombol *back* untuk kembali pada proses sebelumnya.

4.4 Implementasi pengujian data

Berdasarkan rancangan pengujian pada bab III, terdapat 2 pengujian yang dilakukan yaitu pengujian untuk mengetahui pengaruh perubahan nilai *minimum support* dan *minimum confidence* terhadap jumlah *rule* yang terbentuk melalui proses membangkitkan *rule* dan pengujian untuk mengetahui pengaruh *minimum support* dan *minimum confidence* terhadap akurasi klasifikasi.

Data latih yang digunakan dibagi menjadi tiga, yaitu 100 *record* data, 200 *record* data, dan semua data (300 *record*). Pada dua data latih pertama, data diambil secara acak sebanyak 10 kali perulangan. Data uji yang digunakan untuk 100, 200 dan 300 *record* data latih adalah data latih itu sendiri atau masing-

masing. Nilai *minimum support* yang digunakan adalah 15%, 20%, 25%, 30%, 35%, 40%, 45% dan 50%. Sedangkan *minimum confidence* yang digunakan 50%, 55%, 60%, 65%, 70%, 80%, 85%, 90%, 95% dan 100%.

4.4.1 Pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule*

Pengujian jumlah *rule* yang terbentuk dilakukan pada 3 macam data latih dengan jumlah data berbeda. Pengujian untuk masing-masing data latih dilakukan sebanyak 88 kali dengan mengubah nilai *minimum support* dan *minimum confidence* untuk mendapatkan jumlah *rule* yang terbentuk.

4.4.1.1 Data latih 1: 100 *record* data

Data latih sebanyak 100 *record* dipilih secara acak dari 300 *record* dan diulang sebanyak 10 kali sehingga terdapat 10 data latih dengan jumlah 100 *record*. Masing-masing data latih tersebut diuji dengan mengubah nilai *minimum support* dan *minimum confidence* yang telah ditentukan. *Rule* yang terbentuk untuk masing-masing dijumlahkan dan diambil nilai rata-ratanya. Rata-rata jumlah jumlah *rule* yang terbentuk dari 100 *record* data dapat dilihat pada tabel 4.19.

Tabel 4.19 Rata-rata Jumlah Rule yang Terbentuk (100 *record*)

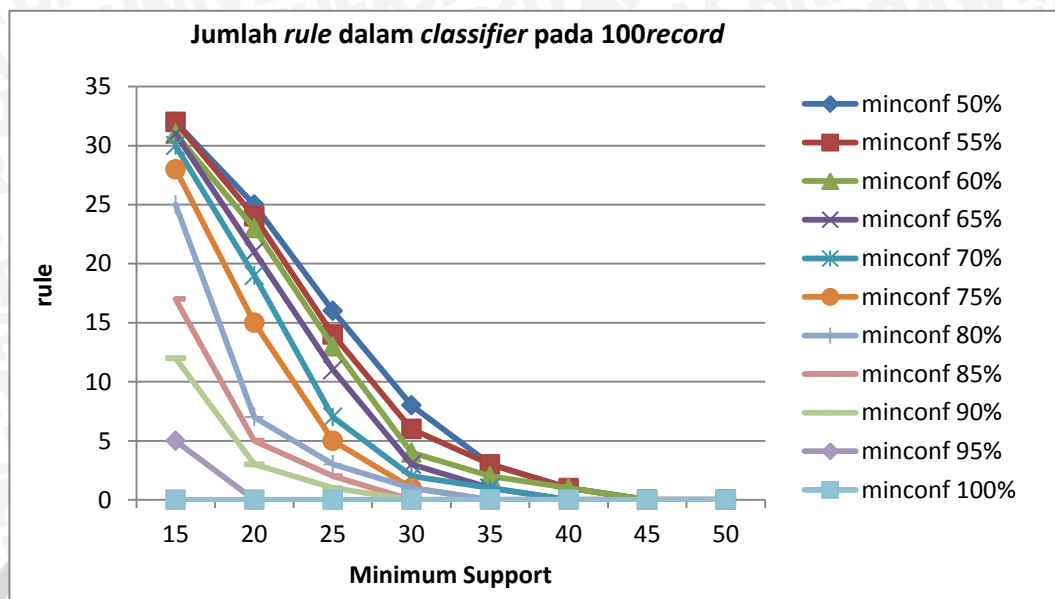
MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	537	95	28	12	4	1	0	0
55	515	86	28	9	4	1	0	0
60	485	77	23	7	2	1	0	0
65	348	64	18	5	1	1	0	0
70	371	50	12	3	1	0	0	0
75	248	32	9	2	1	0	0	0
80	116	18	5	1	0	0	0	0
85	23	7	3	1	0	0	0	0
90	14	3	2	0	0	0	0	0
95	7	1	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Berdasarkan tabel 4.19, ditunjukkan bahwa rata-rata jumlah *rule* terbanyak yang dibentuk adalah 537 *rule* saat *minimum support* 15% dan *minimum confidence* 50%. Sedangkan jumlah *rule* terkecil adalah 0 atau tidak menghasilkan *rule* sama sekali. Semakin tinggi *minimum support* maupun *minimum confidence* yang digunakan semakin sedikit *rule* yang dibentuk. Perubahan nilai *minimum support* dan *minimum confidence* berpengaruh terhadap jumlah *rule* yang terbentuk. Nilai *minimum support* yang merupakan batasan kemunculan suatu *items* bersamaan yang semakin tinggi akan menghasilkan *frequent itemsets* yang semakin sedikit sehingga mengakibatkan semakin kecil kemungkinan kandidat *rule* yang akan dibentuk. Ditambah lagi jika nilai *minimum confidence* yang digunakan semakin tinggi, kemungkinan kandidat *rule* yang memenuhi nilai tersebut juga akan semakin sedikit. Begitu pula pada jumlah *rule* yang masuk dalam *classifier*.

Rule yang dibangkitkan pada semua data latih akan mengalami proses membangun *classifier*. Proses ini bertujuan untuk menyaring atau menyeleksi *rule* yang akan digunakan untuk mengklasifikasi data uji sehingga dapat mengurangi waktu komputasi. *Rule* yang ada dalam *classifier* merupakan *rule* yang paling sering digunakan oleh data latih. Rata-rata jumlah *rule* yang masuk dalam *classifier* dapat dilihat pada tabel 4.20 dan gambar 4.9.

Tabel 4.20 Rata-rata Jumlah *Rule* Dalam *Classifier* (100 record)

MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	32	25	16	8	3	1	0	0
55	32	24	14	6	3	1	0	0
60	31	23	13	4	2	1	0	0
65	31	21	11	3	1	0	0	0
70	30	19	7	2	1	0	0	0
75	28	15	5	1	0	0	0	0
80	25	7	3	1	0	0	0	0
85	17	5	2	0	0	0	0	0
90	12	3	1	0	0	0	0	0
95	5	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0



Gambar 4.9 Grafik Pengaruh Minimum Support dan Minimum Confidence Terhadap Jumlah Rule Dalam Classifier pada 100 Record

Pada pengujian ini diketahui jumlah *rule* yang masuk dalam *classifier*, lebih sedikit daripada *rule* awal yang dibangkitkan. Jumlah *rule* dalam *classifier* cenderung menurun seiring dengan kenaikan *minimum support* pada setiap *minimum confidence* yang digunakan. Berdasarkan tabel 4.20, rata-rata jumlah *rule* terbanyak dalam *classifier* adalah 32 *rule* saat *minimum support* 15% dan *minimum confidence* 50%. Sedangkan jumlah *rule* terkecil adalah 0 *rule*. Hal ini terjadi karena tidak ada *rule* yang terbentuk pada tahap awal sehingga tidak ada *rule* yang bisa dimasukkan dalam *classifier*. Kurva yang dihasilkan pada semua *minimum confidence* terus menurun hingga ke titik nol untuk setiap perubahan *minimum support*.

4.4.1.2 Data latih 2 : 200 record data

Data latih sebanyak 200 *record* dipilih secara acak dari 300 *record* dan diulang sebanyak 10 kali sehingga terdapat 10 data latih dengan jumlah 100 *record*. Masing-masing data latih tersebut diuji dengan mengubah nilai *minimum support* dan *minimum confidence* yang telah ditentukan. *Rule* yang terbentuk untuk masing-masing dijumlahkan dan diambil nilai rata-ratanya. Rata-rata

jumlah *rule* yang terbentuk dari 200 *record* data dapat dilihat pada tabel 4.21.

Tabel 4.21 Rata-rata Jumlah *Rule* yang Terbentuk (200 *record*)

MinConf (%)	Jumlah <i>Rule</i>							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	616	89	30	11	6	1	0	0
55	591	83	26	8	4	1	0	0
60	562	72	21	6	3	1	0	0
65	507	58	14	4	1	1	0	0
70	396	39	7	1	0	0	0	0
75	247	19	3	1	0	0	0	0
80	66	8	2	0	0	0	0	0
85	8	4	2	0	0	0	0	0
90	2	1	1	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

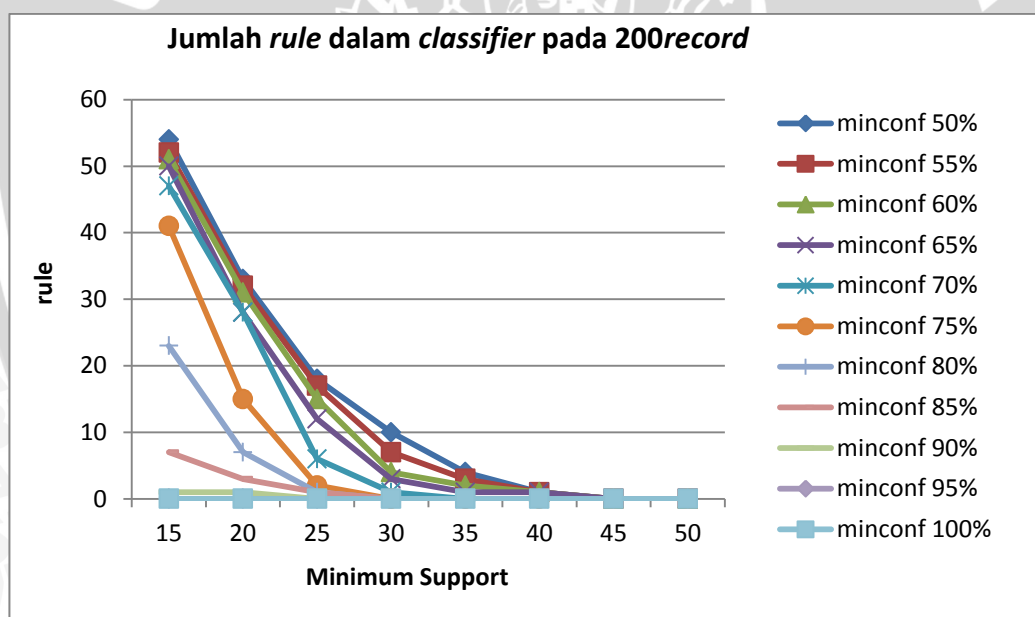
Berdasarkan tabel 4.21, diketahui bahwa hasil uji coba menggunakan 200 *record* data, jumlah rata-rata *rule* terbanyak diperoleh saat *minimum support* 15% dan *minimum confidence* 50% yaitu 616 *rule*. Sedangkan, *rule* yang paling sedikit ialah tidak menghasilkan *rule* sama sekali. Sama seperti pada pengujian dengan 100 *record* data, jumlah *rule* yang terbentuk semakin menurun seiring dengan peningkatan nilai *support* dan nilai *confidence* yang digunakan. Perubahan nilai *minimum support* dan *minimum confidence* berpengaruh terhadap jumlah *rule* yang terbentuk. Nilai *minimum support* yang merupakan batasan kemunculan suatu *items* bersamaan yang semakin tinggi akan menghasilkan *frequent itemsets* yang semakin sedikit sehingga mengakibatkan semakin kecil kemungkinan kandidat *rule* yang akan dibentuk. Ditambah lagi jika nilai *minimum confidence* yang digunakan semakin tinggi, kemungkinan kandidat *rule* yang memenuhi nilai tersebut juga akan semakin sedikit. Begitu pula pada jumlah *rule* yang masuk dalam *classifier*.

Rule yang dibangkitkan pada semua data latih akan mengalami proses membangun *classifier*. Proses ini bertujuan untuk menyaring atau menyeleksi *rule* yang akan digunakan untuk mengklasifikasi data uji sehingga dapat mengurangi waktu komputasi. *Rule* yang ada dalam *classifier* merupakan *rule* yang paling

sering digunakan oleh data latih. Rata-rata jumlah *rule* yang masuk dalam *classifier* dapat dilihat pada tabel 4.22 dan gambar 4.10.

Tabel 4.22 Rata-rata Jumlah Rule Dalam Classifier (200 record)

MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	54	33	18	10	4	1	0	0
55	52	32	17	7	3	1	0	0
60	51	31	15	4	2	1	0	0
65	50	28	12	3	1	1	0	0
70	47	28	6	1	0	0	0	0
75	41	15	2	0	0	0	0	0
80	23	7	1	0	0	0	0	0
85	7	3	1	0	0	0	0	0
90	1	1	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0



Gambar 4.10 Grafik Pengaruh Minimum Support dan Minimum Confidence Terhadap Jumlah Rule Dalam Classifier pada 200 Record

Berdasarkan tabel 4.22, rata-rata jumlah *rule* terbanyak dalam *classifier* adalah 54 *rule* saat *minimum support* 15% dan *minimum confidence* 50%. Sedangkan jumlah *rule* terkecil adalah 0 *rule*. Hal ini terjadi karena tidak ada *rule* yang terbentuk pada tahap awal sehingga tidak ada *rule* yang bisa dimasukkan

dalam *classifier*. Rata-rata jumlah *rule* yang masuk dalam *classifier* dalam bentuk grafik dapat dilihat pada gambar 4.10. Kurva yang dihasilkan pada semua *minimum confidence* terus menurun hingga ke titik nol untuk setiap perubahan *minimum support*.

4.4.1.3 Data latih 3 : 300 record data

Hasil pengujian pada semua *record* data dapat dilihat pada tabel 4.23. Diketahui bahwa *rule* terbanyak yang dibangkitkan sama seperti pada 2 jumlah *record* data sebelumnya yaitu saat *minimum support* 15% dengan *minimum confidence* 50%. Sedangkan *rule* terkecil yang dibangkitkan adalah 0 atau tidak menghasilkan *rule*. Jumlah *rule* yang terbentuk terus menurun seiring dengan semakin tingginya nilai *minimum support* maupun *minimum confidence* yang digunakan.

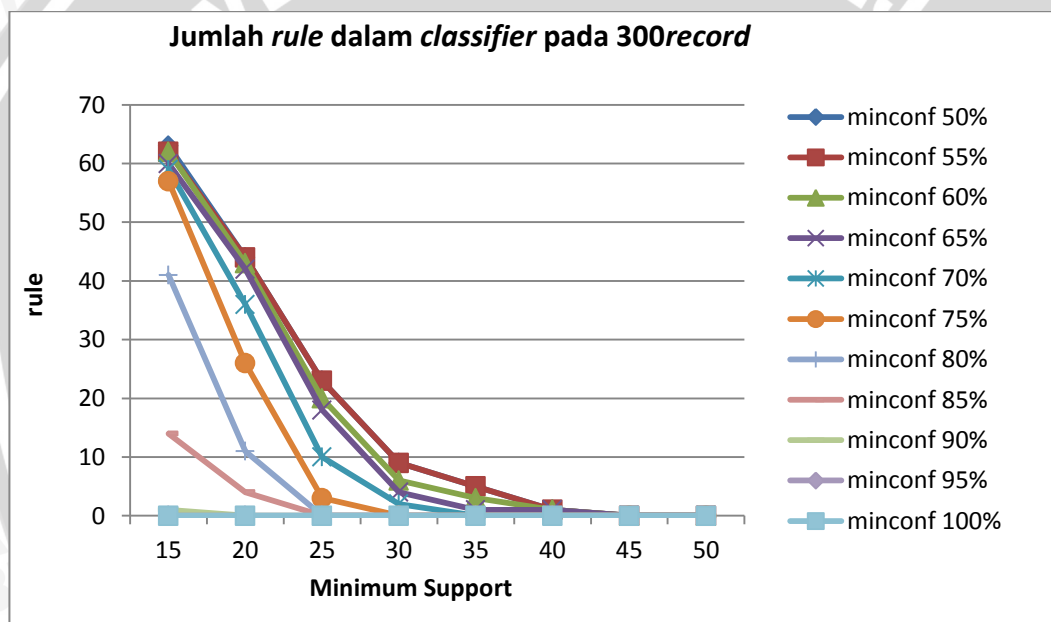
Tabel 4.23 Pengujian Jumlah Rule yang Terbentuk (300 record)

MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	427	94	29	11	6	2	0	0
55	425	93	28	10	6	2	0	0
60	408	84	24	7	4	2	0	0
65	384	76	20	5	2	2	0	0
70	339	54	11	3	1	1	0	0
75	234	36	4	1	0	0	0	0
80	130	13	1	0	0	0	0	0
85	17	5	1	0	0	0	0	0
90	2	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Perubahan nilai *minimum support* dan *minimum confidence* juga berpengaruh terhadap jumlah *rule* yang masuk dalam *classifier*. Sama seperti data latih sebelumnya, *rule* yang masuk dalam *classifier* lebih sedikit daripada *rule* awal yang dibangkitkan. Jumlah *rule* dalam *classifier* cenderung menurun seiring dengan kenaikan *minimum support* pada setiap *minimum confidence* yang digunakan. Rata-rata jumlah *rule* yang masuk dalam *classifier* dapat dilihat pada tabel 4.24 dan gambar 4.11.

Tabel 4.24Jumlah Rule Dalam Classifier (300 record)

MinConf (%)	Jumlah Rule							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	63	44	23	9	5	1	0	0
55	62	44	23	9	5	1	0	0
60	62	43	20	6	3	1	0	0
65	60	42	18	4	1	1	0	0
70	59	36	10	2	0	0	0	0
75	57	26	3	0	0	0	0	0
80	41	11	0	0	0	0	0	0
85	14	4	0	0	0	0	0	0
90	1	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0



Gambar 4.11 Grafik Pengaruh *Minimum support* dan *Minimum confidence* Terhadap Jumlah Rule Dalam Classifier pada Semua Data

Berdasarkan tabel 4.24, jumlah *rule* terbanyak dalam *classifier* adalah 63 *rule* saat *minimum support* 15% dan *minimum confidence* 50%. Sedangkan jumlah *rule* terkecil adalah 0 *rule*. Hal ini terjadi karena tidak ada *rule* yang terbentuk pada tahap awal sehingga tidak ada *rule* yang bisa dimasukkan dalam *classifier*. Rata-rata jumlah *rule* yang masuk dalam *classifier* dalam bentuk grafik dapat dilihat pada gambar 4.11. Kurva yang dihasilkan pada semua *minimum confidence* terus menurun hingga ke titik nol untuk setiap perubahan *minimum support*.

4.4.2 Pengujian pengaruh *minimum support* dan *minimum confidence* terhadap akurasi

Pengujian pengaruh *minimum support* dan *minimum confidence* terhadap akurasi dilakukan pada data uji 100 *record*, 200 *record* dan 300 *record* data menggunakan *rule* yang dibangkitkan oleh 3 data latih pada pengujian pertama yang telah melewati proses membangun *classifier*. Hasil klasifikasi sistem kemudian dibandingkan dengan kelas aktual pada data uji.

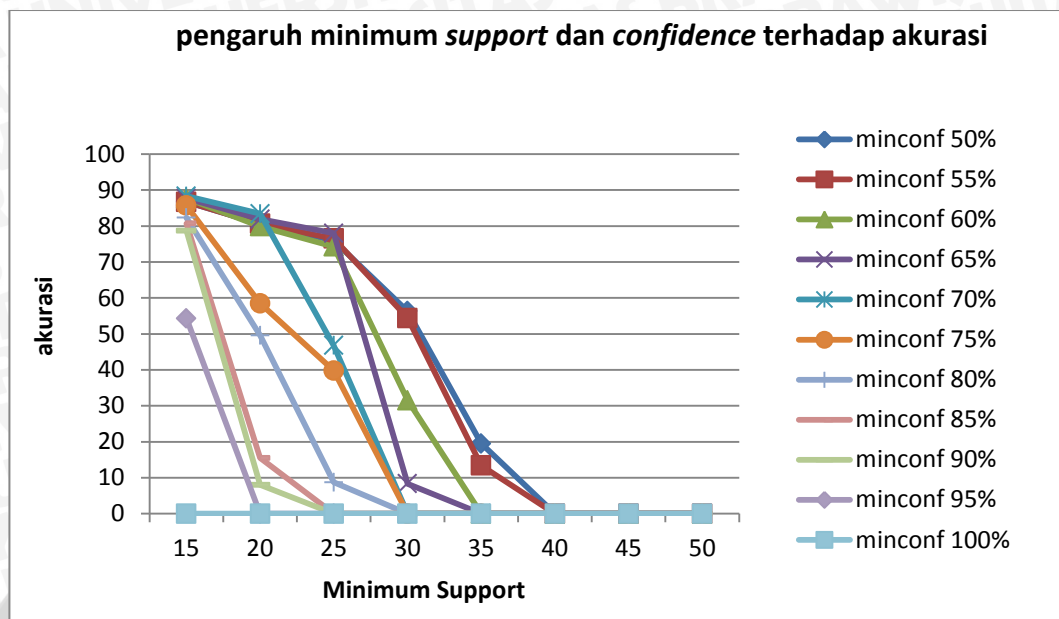
4.4.2.1 Data latih 1 : 100 *record* data

Pengujian akurasi menggunakan *rule* dari 100 *record* data dilakukan pada 100 *record* data uji dari data itu sendiri. Rata-rata hasil pengujian akurasi menggunakan *rule* dalam *classifier* dari 100 data latih dapat dilihat pada tabel 4.25.

Tabel 4.25 Rata-rata Pengujian Akurasi (100 data latih)

MinConf (%)	Akurasi							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	86,7	80,6	76	56,3	19,5	0	0	0
55	86,7	80,6	76,5	54,4	13,4	0	0	0
60	87,7	79,9	74,3	31,5	0	0	0	0
65	87,7	81,9	78	8,3	0	0	0	0
70	88,3	83,4	46,8	0	0	0	0	0
75	85,7	58,5	39,8	0	0	0	0	0
80	82,3	49,6	8,7	0	0	0	0	0
85	80,7	15,5	0	0	0	0	0	0
90	78,7	8	0	0	0	0	0	0
95	54,3	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Berdasarkan tabel 4.25, ditunjukkan bahwa akurasi tertinggi yang diperoleh adalah 88,3% yaitu saat *minimum support* 15% dan *minimum confidence* 70%. Sedangkan akurasi terendah adalah 0%. Akurasi sistem menjadi 0% atau tidak ada akurasi karena pada proses pembangkitan *rule* tidak ada *rule* yang terbentuk yang memenuhi batasan nilai *minimum confidence* sehingga tidak dapat dilakukan proses klasifikasi pada data uji. Hasil pengujian akurasi sistem dapat pula dilihat pada gambar 4.12.



Gambar 4.12 Grafik Pengaruh *Minimum Support* dan *Minimum Confidence* Terhadap Akurasi (100 data latih)

Berdasarkan gambar 4.12, diketahui bahwa rata-rata akurasi pada semua *minimum confidence* cenderung menurun pada setiap kenaikan *minimum support*. Penurunan akurasi ke titik nol menunjukkan bahwa tidak ada *rule* yang dapat digunakan untuk mengklasifikasi data uji sehingga tidak dapat dilakukan perhitungan akurasi. Jika *rule* yang dibangkitkan sedikit akibat nilai *minimum support* dan *confidence* yang tinggi maka kemungkinan *rule* yang dibangkitkan dengan bagian konsekuen hanya pada salah satu kelas menjadi besar sehingga peluang data terklasifikasi ke kelas.

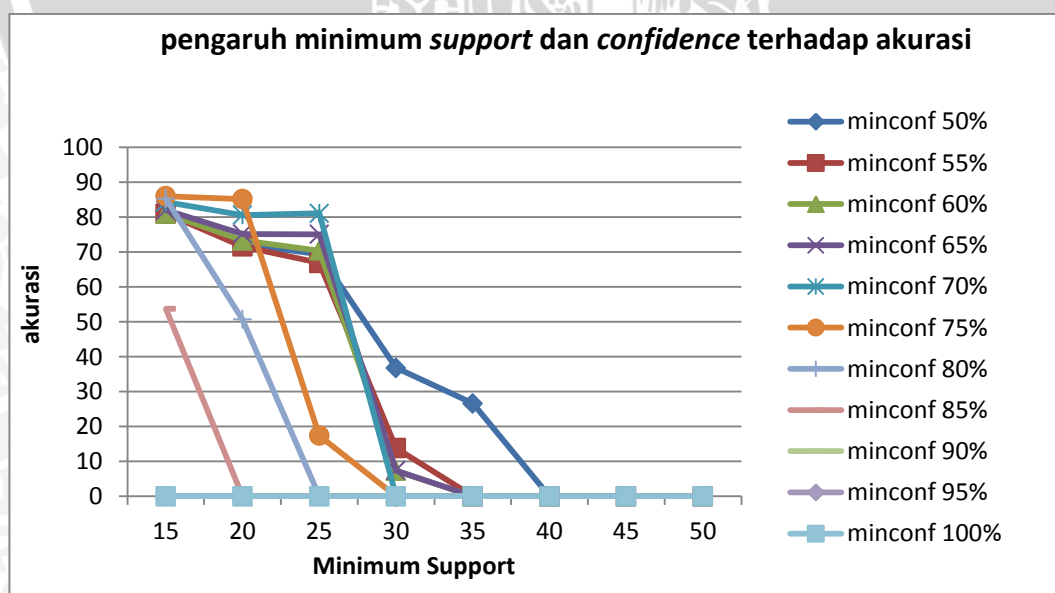
4.4.2.2 Data latih 2 : 200 record data

Pengujian akurasi menggunakan *rule* dari 200 *record* data juga dilakukan pada 200 *record* data uji dari data itu sendiri. Hasil pengujian ini dapat dilihat pada tabel 4.26.

Tabel 4.26 Rata-rata Pengujian Akurasi (200 data latih)

MinConf (%)	Akurasi							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	81,6	72,2	69,4	36,8	32,3	0	0	0
55	81	71,6	66,9	13,8	0	0	0	0
60	81	73,3	70,4	7,3	0	0	0	0
65	82	75,2	75,1	7,4	0	0	0	0
70	84,3	80,6	81,1	0	0	0	0	0
75	86	85,2	17,4	0	0	0	0	0
80	85,3	50,7	0	0	0	0	0	0
85	53,7	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Berdasarkan tabel 4.26 dapat dilihat bahwa rata-rata akurasi tertinggi yang diperoleh sebesar 86% yaitu saat *minimum support* 15% dan *minimum confidence* 75%. Akurasi terendah yang diperoleh sama seperti pada 100 data latih yaitu sebesar 0% atau dapat dikatakan tidak terjadi proses klasifikasi. Hasil pengujian akurasi sistem dapat pula dilihat pada gambar 4.13.



Gambar 4.13 Grafik Pengaruh Minimum Support dan Minimum Confidence Terhadap Akurasi (200 data latih)

Dari gambar 4.13 diketahui bahwa rata-rata akurasi pada semua *minimum confidence* cenderung menurun stabil pada setiap kenaikan *minimum support*. Penurunan akurasi ke titik nol menunjukkan bahwa tidak ada *rule* yang dapat digunakan untuk mengklasifikasi data uji sehingga tidak dapat dilakukan perhitungan akurasi. Jika *rule* yang dibangkitkan sedikit akibat nilai *minimum support* dan *minimum confidence* yang tinggi maka kemungkinan *rule* yang dibangkitkan dengan bagian konsekuen hanya pada salah satu kelas menjadi besar sehingga peluang data terklasifikasi ke kelas.

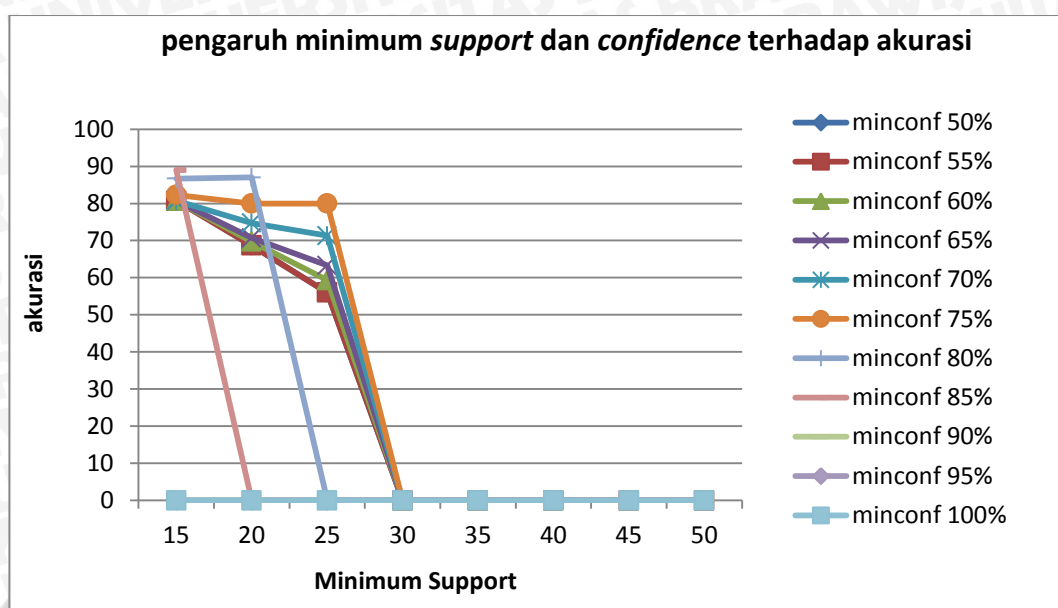
4.4.2.3 Data latih 3 : 300 record data

Pengujian menggunakan *rule* yang dibangkitkan dari 300 data dilakukan pada 300 *record* data uji dari data itu sendiri. Hasil dari pengujian ini dapat dilihat pada tabel 4.27

Tabel 4.27 Hasil Pengujian Akurasi (300 data latih)

MinConf (%)	Akurasi							
	MinSup (%)							
	15	20	25	30	35	40	45	50
50	81	68,7	56	0	0	0	0	0
55	80,7	68,7	56	0	0	0	0	0
60	80,7	69,7	59,3	0	0	0	0	0
65	80,7	70,7	63,3	0	0	0	0	0
70	80,7	74,7	71,3	0	0	0	0	0
75	82,3	80	80	0	0	0	0	0
80	86,7	87	0	0	0	0	0	0
85	89	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Berdasarkan tabel 4.27, akurasi tertinggi yang diperoleh sebesar 89% yaitu saat *minimum support* 15% dengan *minimum confidence* 85%. Sedangkan akurasi terendah adalah 0% atau tidak terjadi proses klasifikasi karena tidak ada *rule* yang dibangkitkan. Hasil pengujian akurasi sistem dapat pula dilihat pada gambar 4.14.



Gambar 4.14 Grafik Pengaruh *Minimum Support* dan *Minimum Confidence* Terhadap Akurasi (300 data)

Berdasarkan gambar 4.14, diketahui bahwa rata-rata akurasi pada semua *minimum confidence* cenderung menurun stabil pada setiap kenaikan *minimum support*. Penurunan akurasi ke titik nol menunjukkan bahwa tidak ada *rule* yang dapat digunakan untuk mengklasifikasi data uji sehingga tidak dapat dilakukan perhitungan akurasi. Jika *rule* yang dibangkitkan sedikit akibat nilai *minimum support* dan *minimum confidence* yang tinggi maka kemungkinan *rule* yang dibangkitkan dengan bagian konsekuen hanya pada salah satu kelas menjadi besar sehingga peluang data terklasifikasi ke kelas.

4.5 Analisis Hasil

4.5.1 Pengujian pengaruh *minimum support* dan *minimum confidence* terhadap jumlah *rule*

Pada proses pengujian menggunakan 3 macam data latih dengan jumlah data yang berbeda, diketahui bahwa jumlah *rule* terbanyak yang dibangkitkan adalah saat *minimum support* 1% dengan *minimum confidence* 50%. Jumlah *rule* yang dibangkitkan menurun seiring dengan kenaikan *minimum support* maupun *minimum confidence*. Nilai *minimum support* yang merupakan batasan kemunculan suatu *items* bersamaan yang semakin tinggi akan menghasilkan

frequent itemsets yang semakin sedikit sehingga mengakibatkan semakin kecil kemungkinan kandidat *rule* yang akan dibentuk. Ditambah lagi jika nilai *minimum confidence* yang digunakan semakin tinggi, kemungkinan kandidat *rule* yang memenuhi nilai tersebut juga akan semakin sedikit.

Rule yang dibangkitkan pada semua data latih akan mengalami proses membangun *classifier*. Proses ini bertujuan untuk menyaring atau menyeleksi *rule* yang akan digunakan untuk mengklasifikasi data uji sehingga dapat mengurangi waktu komputasi. *Rule* yang ada dalam *classifier* merupakan *rule* yang paling sering digunakan oleh data latih. Berdasarkan hasil uji coba diketahui bahwa jumlah *rule* yang masuk dalam *classifier* lebih sedikit dari pada jumlah *rule* awal yang dibangkitkan. Jumlah *rule* dalam *classifier* pada semua *minimum confidence* yang digunakan cenderung menurun seiring dengan kenaikan nilai *minimum support*.

Pada *rule* yang dihasilkan terdapat atribut yang lebih banyak kemunculannya, ini dikarenakan atribut ini melebihi *minimum support* dan *minimum confidence* yang diinputkan. Sehingga mengakibatkan kemungkinan kemunculan atribut dan kombinasi dengan atribut lain menjadi lebih besar.

4.5.2 Pengujian pengaruh *minimum support* dan *minimum confidence* terhadap akurasi

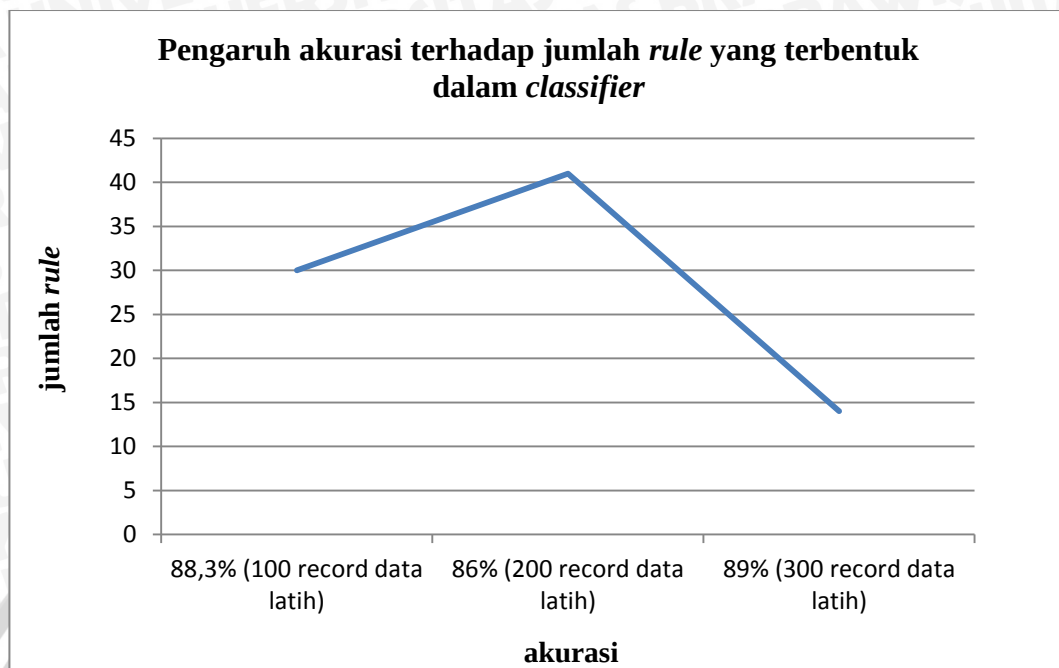
Berdasarkan hasil pengujian akurasi pada data uji menggunakan *rule* yang dibangkitkan dari 3 macam data latih, diketahui bahwa rata-rata akurasi tertinggi adalah 88,3% saat *minimum support* 15% dengan *minimum confidence* 70% pada data latih 1, 86% saat *minimum support* 15% dengan *minimum confidence* 75% pada data latih 2 dan 89% saat *minimum support* 15% dengan *minimum confidence* 85% pada data latih 3. Dari ketiga data latih yang memiliki jumlah *record* berbeda diketahui bahwa akurasi tertinggi diperoleh saat *minimum support* yang digunakan relatif kecil.

Berdasarkan hasil uji coba pada setiap perulangan jumlah *record* data latih, akurasi yang diperoleh tidak berpengaruh banyak terhadap beberapa pasangan nilai *minimum support* dan *minimum confidence* karena akurasi yang diperoleh cenderung pada nilai sama. Akan tetapi, rata-rata akurasi dari hasil

perulangan tersebut menunjukkan bahwa rata-rata akurasi pada semua *minimum confidence* cenderung menurun pada setiap kenaikan *minimum support*. Hal ini dikarenakan pada uji coba untuk setiap perulangan jumlah *record* data, pada pasangan nilai *minimum support* maupun *minimum confidence* yang tinggi kemungkinan tidak ada *rule* yang terbentuk makin besar sehingga kemungkinan tidak terjadi proses klasifikasi juga makin besar.

Jika *rule* yang dibangkitkan sedikit akibat nilai *minimum support* dan *confidence* yang tinggi maka kemungkinan *rule* yang dibangkitkan dengan bagian konsekuensi hanya pada salah satu kelas menjadi besar sehingga peluang data terklasifikasi ke kelas tersebut semakin besar. Jika *minimum support* maupun *minimum confidence* yang digunakan relatif kecil maka *rule* yang dibangkitkan lebih banyak dan variatif baik yang bagian konsekuennya kelas 1 maupun kelas 2 sehingga peluang data uji terklasifikasi ke kelas 1 maupun kelas 2 besar. Akurasi terendah yang diperoleh sistem adalah 0% atau dapat dikatakan tidak terjadi proses klasifikasi. Hal ini dikarenakan pada proses pembangkitan *rule* tidak ada *rule* yang terbentuk yang memenuhi *minimum confidence* sehingga tidak ada pula *rule* dalam *classifier* yang dapat digunakan untuk mengklasifikasi data uji.

Jumlah *rule* yang dihasilkan tidak mempengaruhi akurasi yang dihasilkan. Dapat dilihat pada gambar 4.15, diketahui pada data latih 100 *record* akurasi tertinggi yaitu 88,3% dengan jumlah *rule* yang terbentuk dalam *classifier* sebanyak 30 *rule*, pada data latih 200 *record* akurasi tertinggi yaitu 86% dengan jumlah *rule* yang terbentuk dalam *classifier* sebanyak 41 *rule*, dan pada data latih 300 *record* akurasi tertinggi yaitu 89% dengan jumlah *rule* yang terbentuk dalam *classifier* sebanyak 14 *rule*.



Gambar 4.15 Grafik Pengaruh akurasi terhadap jumlah *rule* yang terbentuk dalam *classifier*

Berdasarkan gambar 4.15, diketahui bahwa hasil pengujian jumlah *rule* dan pengujian akurasi, hasilnya tidak berkesinambungan. Hal ini dikarenakan padaperhitungan akurasi pengecekan dilakukan dengan penggabungan seluruh *rule* dan dilakukan pengecekan kebenaran terhadap data. Oleh karena itubenar atau salahnya *rule*, tergantung dari kombinasi atribut di dalam satu *rule* yang dihasilkan sehingga tidak dapat diprediksi dengan semakin banyak *rule* maka semakin baik pula akurasinya. Hasil *rule* yang terbentuk dengan akurasi tertinggi yaitu pada *minimum support* 15% dan *minimum confidence* 85% dapat dilihat pada lampiran 1.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

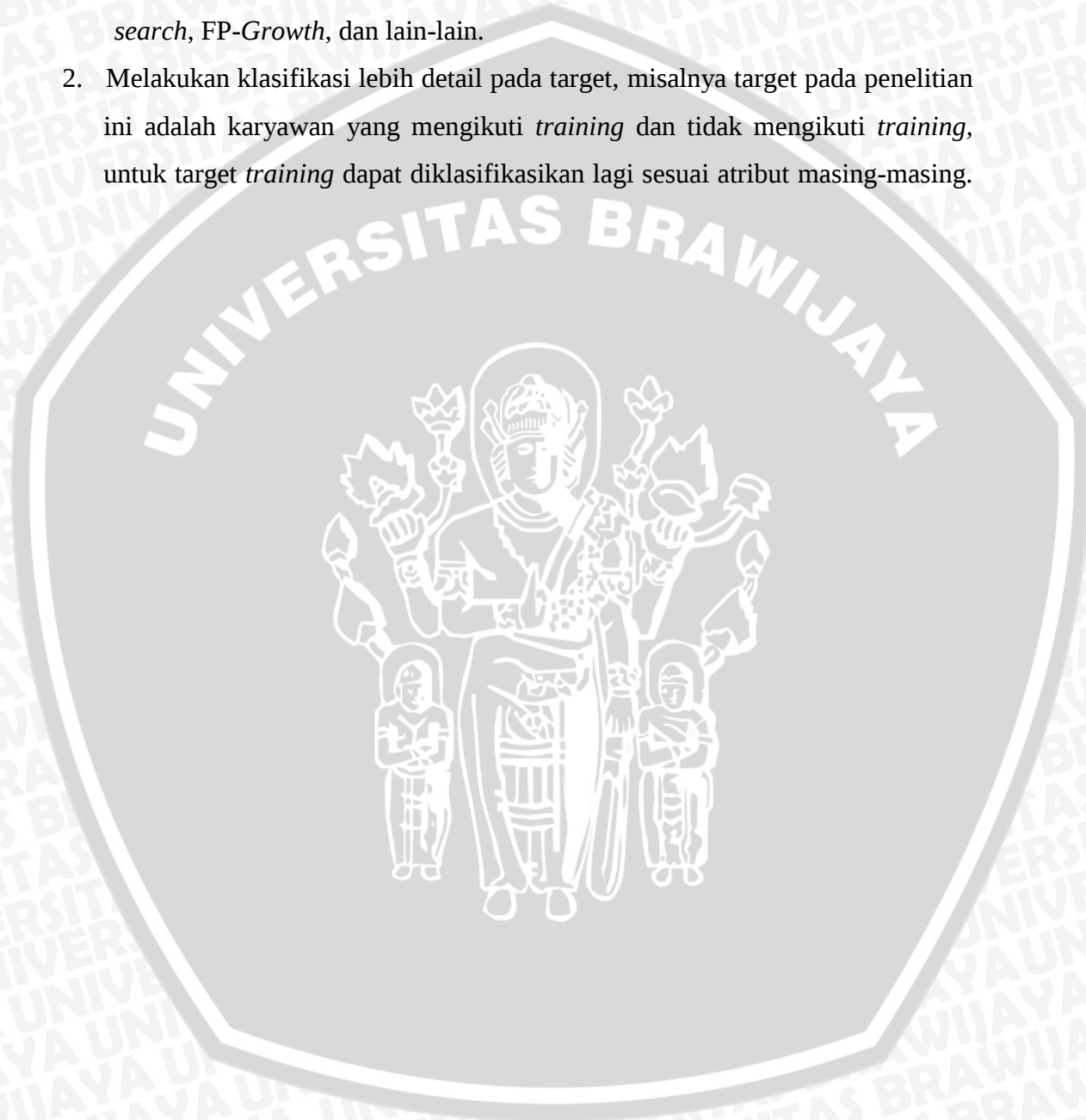
Berdasarkan hasil pengujian dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Metode *classification based on fuzzy association rule* (CFAR) dapat diimplementasikan untuk mengklasifikasi evaluasi kinerja karyawan. Proses awal yang dilakukan adalah dengan mentransformasi data latih menjadi data *fuzzy* menggunakan fungsi keanggotaan untuk masing-masing atribut. Kemudian melakukan pembangkitan *rule* yang mana bagian konsekuennya merupakan atribut kelas. *Rule* yang dibangkitkan dilatih menggunakan data latih dan *rule* yang mengklasifikasikan data latih dengan benar dimasukkan dalam *classifier*. *Rule* dalam *classifier* ini yang akan digunakan untuk mengidentifikasi atau menentukan kelas *output* data uji. Data uji yang digunakan berasal dari data latih.
2. Parameter *minimum support* dan *minimum confidence* berpengaruh pada proses pembangkitan *rule*. Semakin tinggi nilai *minimum support* dan *minimum confidence* yang digunakan semakin sedikit *rule* yang dihasilkan. Sedangkan terhadap akurasi, *minimum support* dan *minimum confidence* berpengaruh ketika nilai yang digunakan semakin tinggi karena pada pasangan *minimum support* dan *minimum confidence* yang tinggi kemungkinan *rule* yang dihasilkan hanya mewakili salah satu kelas besar dan kemungkinan tidak menghasilkan *rule* juga besar. Parameter *minimum support* dan *minimum confidence* berpengaruh pada proses pembangkitan *rule*. Semakin tinggi nilai *minimum support* dan *minimum confidence* yang digunakan semakin sedikit *rule* yang dihasilkan. Sedangkan terhadap akurasi, *minimum support* dan *minimum confidence* berpengaruh ketika nilai yang digunakan semakin tinggi karena pada pasangan *minimum support* dan *minimum confidence* yang tinggi kemungkinan *rule* yang dihasilkan hanya mewakili salah satu kelas besar dan kemungkinan tidak menghasilkan *rule* juga besar.

5.2 Saran

Dalam penelitian ini, terdapat beberapa hal yang dapat dijadikan bahan dalam penelitian lebih lanjut, antara lain adalah :

1. Menggunakan metode *association rule* selain *apriori* seperti ID3, *pincer search*, FP-Growth, dan lain-lain.
2. Melakukan klasifikasi lebih detail pada target, misalnya target pada penelitian ini adalah karyawan yang mengikuti *training* dan tidak mengikuti *training*, untuk target *training* dapat diklasifikasikan lagi sesuai atribut masing-masing.



DAFTAR PUSTAKA

- [AGR-94] Agrawal, R dan Srikant, R. 1994. *Fast Algorithms for Mining Association Rules*. Chile : Proceedings of the 20th International Conference on Very Large Data Bases.
- [BUD-05] Budhi, G.S., Lim, R., dan Prayitno, O. 2005. *Penggunaan Metode Fuzzy c-Covering untuk Analisa Market Basket pada Supermarket* (Online), (<http://puslit2.petra.ac.id/ejournal/index.php/inf/article/viewFile/16319/16311>, diakses tanggal 10 Maret 2012).
- [CHE-08] Chen, Zuoliang., dan Chen, Guoqing. 2008. *Building An Associative Classifier Based On Fuzzy Association Rules*. International Journal of Computational Intelligence Systems.
- [GYE-01] Gyenesei, Attila. 2001. *A Fuzzy Approach for Mining Quantitative Association Rule*. Finland : University of Turku.
- [HAN-01] Han, Jiawei and Khamber, Micheline. 2001. *Data mining : Concepts and Techniques*. San Francisco, USA : Morgan Kaufmann Publishers.
- [HAN-03] Han, Jiawei dan Kamber, Micheline. 2003. *Data Mining: Concepts and Techniques*. San Francisco, USA : Morgan Kaufmann Publishers.
- [JIN-04] Jin, Weiqing. 2004. *Fuzzy Classification Based on Fuzzy Association Rule Mining*. North Carolina State University.
- [KAN-03] Kantardzic, Mehmed. 2003. *Data Mining: Concepts, Models, Methods and Algorithm*. New York : John Wiley and Sons.
- [KUO-98] Kuok, C.M., Fu, A. dan Wong, M.H. 1998. *Mining Fuzzy Association Rule in Database* (Online), (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.684>, diakses tanggal 20 Februari 2012).
- [KUS-07] Kusnawi. 2007. *Pengantar Solusi Data Mining* (Online), (<http://p3m.amikom.ac.id/p3m/56%20->

%20PENGANTAR%20SOLUSI%20DATA%20MINING.pdf, diakses tanggal 20 Februari 2012).

- [KUS-10] Kusumadewi, S dan Purnomo, H. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan: Jilid 2*. Yogyakarta : Graha Ilmu.
- [KUS-09] Kusrini dan Luthfi, Emha Tufiq. 2009. *Algoritma Data Mining*. Yogyakarta : Penerbit ANDI.
- [LAR-05] Larose, Daniel. 2005. *Data Mining methods and Model*. New Jersey : Wiley-Interscience.
- [LU-03] Lu, Jianjiang., Xu, Baowen., dan Yang, Hongji. 2003. *A Classification Method of Fuzzy Association Rules*. Ukraine: IEEE International workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.
- [LUT-09] Luthfi, E.T dan Kusrini. 2009. *Algoritma Data Mining*. Yogyakarta : C.V Andi Offset.
- [MAN-05] Mangkunegara, Anwar P. 2005. *Manajemen sumber daya manusia perusahaan*. Bandung : PT Remaja Rosdakarya.
- [NUG-06] Nugraha, Dany, dkk. 2006. *Diagnosis Gangguan Sistem Urinari pada Anjing dan Kucing Menggunakan VFI 5*. Institut Pertanian Bogor.
- [PRA-03] Pramudiono, Iko. 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. Ilmu Komputer.Com.
- [SAN-07] Santosa, Budi. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta : Graha Ilmu.
- [SWA-11] Swasto, Bambang, Dr, ME. 2011. *Manajemen Sumber Daya Manusia*. Malang : UB Press.
- [REF-08] Refaeilzadeh, Payam, Tang, Lei dan liu, Huan. 2008. *Cross-Validation*. Arizona State University.
- [THA-07] Thabtah, Fadi. 2007. *A Review Of Associative Classification Mining*. United Kingdom : Cambridge University Press.

LAMPIRAN

Hasil pembangkitan *rule* terhadap 300 *record* data menggunakan *minSupport* 15% dan *minConfidence* 85%.

No	Rule	Target
1	[judgement-cukup]	1
2	[sikap-baik, judgement-baik, inisiatif-baik, keandalan-baik]	2
3	[kooperatif-sangat baik]	2
4	[keandalan-baik, inisiatif-baik, judgement-baik, sikap-baik]	2
5	[hub.manusia-cukup]	1
6	[sikap-baik, inisiatif-baik, judgement-baik, keandalan-baik]	2
7	[kooperatif-baik, judgement-cukup]	1
8	[komunikasi-sangat baik]	2
9	[keahlian-cukup]	1
10	[komunikasi-cukup]	1
11	[sikap-baik, judgement-baik, keandalan-baik, inisiatif-baik]	2
12	[keandalan-baik, inisiatif-baik, sikap-baik, judgement-baik]	2
13	[inisiatif-cukup]	1
14	[keandalan-cukup]	1

Data hasil pengujian pada 300 *record* menggunakan *minSupport* 15% dan *minConfidence* 85% dengan hasil akurasi 89%.

No.	Kooperatif	Kehadiran	Inisiatif	Keandalan	Sikap	Judgement	Komunikasi	Hub.manusia	Keahlian	Target	Hasil
1	81.4	80.4	74.6	75.8	80.3	76.8	81.9	74.7	53.3	1	2
2	97.3	91.6	74.5	86.6	77.1	82.5	93.1	79.7	76.3	2	2
3	88.2	90	85.2	78.4	84.2	81.8	72.8	82	83.6	2	2
4	85.6	76	83.7	83.9	76.1	74.8	76.2	75.1	75.7	2	2
5	80.9	90	76	83.7	83.7	76	87.9	80.9	73.6	2	2
6	81.6	86.5	76	77.4	76	87.2	83.7	85.8	76	2	2
7	70.3	81.2	83.3	75.5	72.8	83.3	71.9	75.6	80.5	1	2
8	75.1	90	84.7	83.1	79	79.1	70.7	86.3	73	2	2
9	77.5	75.9	79	74.2	79.9	83.1	89.5	77.2	85.9	2	2
10	79.3	78.5	81	86.1	72.8	83.9	90.4	75.4	89.4	2	2
11	83.4	98.1	83.9	92.7	81	75.9	84.8	82.9	89.6	2	2
12	73.6	66.8	91.5	89.3	73.1	76.3	93.4	84.6	91.4	2	2
13	71	95.5	85.5	73.7	83.5	70.6	58.3	83.1	72.5	1	1
14	64	80.2	62.5	73.8	86.5	69.3	71.6	69.2	73.2	1	1
15	69.8	89.6	65.1	80.2	75.2	66.8	76.2	87.5	80.9	1	1
16	61.9	71.7	65.9	52.8	82.9	57.5	54.6	75	64.5	1	1
17	66.4	91.6	72.1	89.9	88.5	69	71.3	70.2	79.6	2	1
18	68.6	66.9	50	75.6	74.1	73.9	69.3	63.8	70	1	1
19	64	77.4	67.3	85.3	79.1	68.6	66.8	65.2	87.6	1	1
20	82.1	75.6	74.8	81.6	79.2	71	85.2	84.6	81.1	2	2
21	77	76.6	67.6	87.8	86.8	69.1	75.6	71.6	81.2	1	1
22	80.4	87.5	80.9	72.3	82	81.7	83.2	88.7	84.8	2	2
23	83.8	93.1	65.2	80.1	76.7	61.9	82.6	89.6	61.8	1	1
24	86	96	88.9	78.7	80	75.9	76.4	95.8	75.2	2	2
25	83	92.8	69.7	74.1	74.8	82.5	85.8	80.9	81.1	2	2
26	82.2	95	67.9	81.8	67.9	75.9	61.1	73.9	86.6	1	1
27	82.4	89.9	81	68.2	76.8	74.2	73	77.2	59.5	1	1
28	74	89.9	73.6	76.4	56	59.1	70.9	93.4	86.8	1	1
29	77.9	81.8	75.7	72.4	85	64.7	75.5	76.6	83.2	1	1
30	75.3	81.6	66.9	56	82.1	50	78.2	81.4	66.7	1	1
31	84.6	73.3	87.8	83.1	78.9	85.9	73.6	78.6	74.3	2	2
32	81.4	79.6	79.6	74.2	60.3	72.6	51.2	70.9	58.9	1	1
33	83.6	72.2	62.8	71.7	75	58.1	65.2	84.2	79.5	1	1

34	74.1	80.3	73.3	85.8	82.5	81	80.2	83.5	83.2	2	2
35	82.1	96.3	74.3	77.4	81	73.2	83.6	67.9	77.3	2	2
36	82.3	83.5	71.1	53.7	68.4	73.3	83.1	61.7	59.7	1	1
37	88.3	81.2	73.2	78	77.5	79	86.4	71	72.2	2	2
38	85.1	84	74.1	80.8	88.5	77.9	93.8	88.4	84.7	2	2
39	92.1	78	84.8	90.7	80.8	78.7	93.7	86.8	73.9	2	2
40	93.3	73.7	70.1	71.7	80.1	72.3	96.7	74.6	73.5	2	2
41	84.6	70	82.9	79.7	88.4	86.3	82.9	71.3	80.1	2	2
42	81.5	84.3	86.9	75	88.5	79.4	94.4	78.6	84.3	2	2
43	93.4	84.1	83.4	77.3	88.5	72.5	94.7	84.3	77.1	2	2
44	86.6	82.7	88.7	79.5	77.1	77.5	93.6	77.7	87.8	2	2
45	76.8	85.8	80.4	82.6	71	80.4	83.2	87.2	74.8	2	2
46	86.2	93	76.1	73.1	78.3	84.8	76	84.1	83.6	2	2
47	89.4	84.1	74.1	78	80.1	73.7	74.2	80.4	91	2	2
48	82.7	78.8	81.4	79.2	92.2	74.8	85.9	76.1	86.3	2	2
49	76.5	72.7	84.7	56.9	88	74.3	75.4	57.3	69.2	1	1
50	88.1	81.8	83.3	90.1	82.5	86.6	84.3	97.6	91.6	2	2
51	76.7	80.7	75.3	79.5	76.3	86	78.3	82.8	78.8	2	2
52	89.2	73.7	75.9	90.3	77.3	86.3	97.5	77.8	87.2	2	2
53	86.8	83.2	81.2	78.9	92.8	75.4	85.9	81.3	82.6	2	2
54	77.2	85.7	83.7	86.2	78.9	83.1	85.9	78.7	74.8	2	2
55	76.2	72.4	78.7	78.7	82.2	86.1	94.1	86.8	72.4	2	2
56	90.8	85.7	71.7	83.1	85.3	79.4	88.3	86.3	77.6	2	2
57	100	83.5	95.2	90.6	79.5	88.5	90.7	74.2	79.7	2	2
58	93	80.3	81.4	81.9	73.1	70.3	91.7	72	73.5	2	2
59	97	83.4	78.7	87.1	84.1	81.8	92.3	87.7	84.6	2	2
60	91	74	96.1	89.9	80.5	82.6	88.4	82	80.1	2	2
61	76.7	77.6	74.7	79.7	80.9	72.3	97	83.5	86.5	2	2
62	90	72.4	81.3	84.7	87.2	74.3	78.5	95.1	92.3	2	2
63	96.4	75.7	77.7	86.4	83.8	77.1	84	95.5	92.4	2	2
64	76.5	83.5	82.9	86	91.5	91.8	86.7	90.3	74.9	2	2
65	91.4	84.4	80.4	77.3	86.1	90.5	93.7	78.8	82.7	2	2
66	94.2	72.1	90	87.4	83.2	93.4	99.4	78.8	83.8	2	2
67	74.8	78.5	67.7	77.8	78.7	74.2	83.2	71.6	73.6	1	2
68	86	84.4	86	75.6	89.7	75.9	82.9	87.7	88	2	2
69	83.6	82.6	79.6	87.2	89.6	85.4	86	94.1	88.3	2	2
70	60.5	79.4	81.3	64.3	75.9	76.6	74.7	72.3	59.3	1	1
71	82.3	84.3	72.1	74	90.8	71.6	97.5	93.8	83.6	2	2
72	93.7	75.6	78.4	77.5	83.1	83.8	79.5	96.6	79.6	2	2
73	89.2	82.2	86.9	86.2	96.2	74.9	91.9	80.5	75.4	2	2
74	94.5	84.3	90.8	93	88.6	92.6	82.5	81.5	77	2	2
75	75.6	81.7	88.4	79.4	84.3	75.7	82.6	84.3	80.7	2	2
76	92.2	87.7	80.5	83.2	95	86.5	88	88.5	83	2	2
77	85.7	81.5	87.1	76.6	85.6	82.5	84.9	97.7	83.9	2	2
78	87.9	86.7	87	87.3	87.6	84.9	89.1	87.9	86.4	2	2
79	87	87.3	82.4	74.9	76.2	75.1	75.2	76.7	77.9	2	2
80	93	82.6	84.9	91.5	87	87	91.5	92.7	84.9	2	2
81	82.7	77.3	72.9	89	94.4	76.3	80	94	73.3	2	1
82	87.9	86.7	71	73.5	78.5	74.3	71.2	81.7	86.8	2	2
83	81.1	83.3	78.7	81	77.1	76.3	71.5	86.7	86.3	2	2
84	83.4	80.2	84.6	86.1	79.5	85.5	84.3	84.6	86.4	2	2
85	80.4	82.9	76	84.7	76.3	77.6	85.3	73.5	77.2	2	2
86	93.5	76.4	78.3	79.9	79.2	78	99.4	79.3	98.8	2	2
87	85.4	82.8	76.9	75	98.3	74.2	88.8	97.3	72.1	2	2
88	76.4	83.3	54.3	76.7	81.3	81.6	61.2	69.7	74.3	1	1
89	74.1	87.7	54.1	76.7	78.2	84.1	64.2	63.7	86.2	1	1
90	92.5	84.2	91.8	77.9	82.9	88.5	74.1	76.5	77.2	2	2
91	72.1	85.7	67.9	73.3	75.4	89.4	61.1	88.6	81.4	1	1
92	73.4	79.4	72.4	87.3	82	73.3	64.3	72.1	85.7	1	1
93	91.7	74.5	75.4	80.9	80.4	83.5	93.5	97	88.9	2	2
94	98.4	85.4	77	76.6	80.8	75.6	62.1	97.7	94.9	2	2
95	92	83.9	81.8	86.5	80.7	76.3	81.2	89	77.6	2	2
96	89.1	77.5	87.5	90.4	80.6	85.5	88.7	74	74.3	2	2
97	92.4	95.2	97	91.2	89.8	80.1	90.2	90.3	90.9	2	2
98	84.2	75.4	79.4	84.7	82.6	55.8	60.8	88.7	77.2	1	1
99	80.7	75.2	71.8	82	81.8	67.4	70.5	92	77.1	2	1
100	74.7	92	78.7	83.2	85.7	83.2	60.9	77.5	89.3	2	2

101	98.5	97.2	84.3	85.5	79.4	83.4	89.1	90	94.5	2	2
102	87.3	77.3	80.3	76.4	77.8	86.7	59.2	80.8	84	2	2
103	99.1	90	84.4	92.4	76.9	78.8	97.4	76.4	94.9	2	2
104	72.4	92.1	74.9	93.3	79.9	73.4	77.2	91.8	97	2	1
105	90	78.8	81.3	80.4	66.9	85.1	75.7	78.2	85.5	2	2
106	97.9	76.6	80.6	90.9	97.7	87.4	96.6	90	93.2	2	2
107	90	81.4	73.1	87.8	95.9	88.3	82.8	91.4	83.9	2	2
108	81.2	92.2	79.9	76.8	79.5	83.5	84.7	74.5	64.4	2	2
109	72.9	68.6	86.3	87	74.7	72.8	80.6	71.8	80	1	2
110	71.5	88.8	58.7	62	60.5	62.8	59.6	81.9	64.1	1	1
111	82.9	90.6	74.1	76.8	78.1	55.5	76.8	77.7	81.3	1	1
112	72.6	94	79.8	70.9	81	67.5	66.4	78.5	76.7	1	1
113	73	70.7	86.2	87.8	76.2	79	72.9	72.7	85.1	2	2
114	76	79.5	79.1	76.6	77.7	67.8	86.4	78.5	66.4	1	2
115	68.2	97.9	62.9	65.6	89.6	76.2	75.9	78.4	67	1	1
116	82.8	90.2	81.5	69.1	81.2	77.6	81.9	75.3	67.2	2	2
117	78.8	76.3	73.9	84.3	81.2	73.7	80.4	70.6	75.9	1	2
118	74.9	53	66.9	70.9	61.8	60.5	85.2	58.6	62.2	1	1
119	83.6	90.5	76.5	81.5	76	80.4	80.5	78.7	82.3	2	2
120	73.8	76	74.9	67.2	69.5	66.8	61.2	67.5	72.4	1	1
121	75.2	91.2	55.1	66.3	67.7	61.1	62.9	64.7	62.5	1	1
122	65.3	93.3	61.4	60.8	75.9	58.7	63.5	89	60	1	1
123	79.5	93	60.2	83.7	65.1	82.3	70.3	84.3	69.1	1	1
124	80	94.7	79.2	76.4	79.3	87.4	77	70.6	66.9	2	2
125	83.4	94.4	83.6	78.4	95.8	83.9	78.5	91.7	87.3	2	2
126	81.6	63.9	57.2	80.3	76.8	67.3	82.7	78.2	87.1	1	1
127	68.7	80.4	76.4	91.6	83.6	57.7	76.2	81	60.7	1	1
128	82.8	85.3	66.5	53	77.6	56.7	68.2	71.3	53.4	1	1
129	82.2	83.9	82.7	85	74.7	84.9	79.7	86.6	83.8	2	2
130	81.3	80.1	60	65.1	74.8	57.5	83.2	77	80.1	1	1
131	71.6	98	88.7	58.9	80.6	85.7	78.9	79.1	73.5	2	1
132	62.3	80.3	67.1	62.1	77.6	63.1	50.9	82.6	59	1	1
133	83.3	93.5	77.4	73.9	61.6	65.3	71.9	74.1	69	1	1
134	82.5	92.5	76	85	69.6	60.4	85	84.3	58.2	1	1
135	74.5	78.1	78.7	76.9	85.6	66.2	67.9	77.6	83.1	1	1
136	78.6	91.1	74.2	61.4	84.3	63	73.1	81.6	81.5	1	1
137	76.8	83.6	59.6	61.7	79.4	64.8	66.8	87.8	72.3	1	1
138	81.3	59.5	83.3	70.6	45.2	75.4	76.6	57.1	76.2	1	1
139	68.5	77.3	59.9	57.9	68.7	59.7	75.8	55.7	78.1	1	1
140	74.2	67	78.3	81.2	53.8	80	74.2	73	73.3	1	1
141	83.2	90.3	74.8	74.3	81.8	73.7	78.2	73.8	76.6	2	2
142	70.7	91.7	79.6	76.2	77.7	83.8	81	82.7	78.6	2	2
143	81.5	90.3	73.3	90.2	84.3	76.8	75.5	77.6	84.1	2	2
144	80.7	93.5	74.2	84.3	80.4	74.5	70.4	87.2	77.3	2	2
145	81.5	98.8	85.5	91.6	81.6	86.2	82.8	75.9	94.4	2	2
146	76.8	93.6	79.2	81.4	79.9	82.2	73	83.8	84	2	2
147	91.8	92.7	81.5	85.3	98.4	86.2	95.5	79	77.2	2	2
148	82.6	93	81.4	78.2	77.7	80.5	77.5	83.2	73	2	2
149	77.3	95.1	66.2	73.9	75.4	61	75.4	63.5	72.2	1	1
150	84.5	95.5	81.2	85.7	77.5	76.7	74.1	82	90.7	2	2
151	72	82.9	63.2	70.4	56.9	69.6	56.9	65.3	63.2	1	1
152	72.2	96.9	78.2	83.9	90	73.6	78.8	95.3	88.7	2	2
153	87.3	83.7	84.7	71.3	78.7	75.7	72.3	81.1	78.8	2	2
154	76.4	76.4	81.7	76.9	80.8	76.5	79.7	72.1	73.3	1	2
155	60.4	72	68.5	64.6	64.1	58.1	61.1	74.4	63.2	1	1
156	80.3	95.3	74.3	79.3	79.3	78.4	78.9	72.7	86.8	2	2
157	78.2	88.9	94.5	77.9	83.3	75.6	81.4	83	74.4	2	1
158	80.4	94.4	76.8	86.1	82.6	77.1	82.1	87.3	77.9	2	2
159	83.3	73.9	78	63.3	79.3	82.4	82	73.1	84.2	2	1
160	79.7	72.1	79.8	81.9	78	73.5	74.5	67.2	75.7	1	2
161	80.7	87.2	76.2	64	74.3	62.8	80.3	71.6	85.9	1	1
162	83.8	81.6	79.6	92.1	79.9	77.7	85.4	81.7	76.6	2	2
163	58.3	64.2	64.5	62.4	70.5	51.3	69.6	59.7	56.8	1	1
164	79.1	96.8	64.7	71.7	79.3	64.3	70	71	76.8	1	1
165	80	73.7	64.2	61	74.7	60.1	57.9	60	65.5	1	1
166	83.8	97.4	79.3	72.7	91.9	86.1	89.5	81.8	82.7	2	2
167	84.8	93.8	84	78.7	77	84.5	84	79	88.4	2	2

168	82.1	90.1	80.4	87.5	84.3	76.5	91.7	91.1	77.3	2	2
169	86.7	85.7	79	72.7	85.5	82.1	80	83.9	77.6	2	2
170	81.4	96.2	76.8	83.5	78.8	65.6	88.3	81.3	79.1	2	1
171	64.3	90.4	61.8	58.1	76	68.5	70.2	63.3	62.8	1	1
172	84.1	94.8	63.4	73.7	68.4	64.9	69.6	75.7	89.9	1	1
173	83.3	95.2	78.2	64.6	71.5	64.3	67.4	75.7	65.7	1	1
174	76.9	94.2	61.5	70.5	67.4	54.8	74.1	75.7	76.4	1	1
175	74.8	77.3	81.3	67	80.6	74.9	80.5	83.1	76	1	2
176	67.4	96	62.1	80.4	53.1	49.1	63	66.7	78.7	1	1
177	77.6	95.9	82.4	74.5	74.3	54.3	62.2	85.3	74.6	1	1
178	79.1	95.5	75.4	88.8	77.7	78.9	72.1	85.8	74.5	2	2
179	82.7	94.2	77.6	85.9	88.1	82.2	77.3	81.8	82.8	2	2
180	75	76.8	84.4	61.8	72.4	79	83.3	82.9	84.7	2	1
181	81.9	91.2	63	71.8	80.2	78.7	77.4	78.3	71.7	1	1
182	92.3	89.5	84.5	75.8	78.6	81.9	78.7	83.4	81.8	2	2
183	79.7	95.5	76.2	61.7	83.8	61.3	71.4	82.3	58.3	1	1
184	82.6	66.8	83.3	81.3	86.9	78	72.3	65.7	80.6	2	2
185	87.5	80.8	72.2	81.3	77.5	83.2	77.4	91.1	68.1	2	2
186	80.1	97.9	87.3	80	87.6	83.2	87.6	75.8	81.9	2	2
187	87	96.3	84	75	80.9	78.5	89.5	83.4	75.2	2	2
188	78.7	99.4	72.7	89.9	77.5	60.6	53.6	80.2	69.2	1	1
189	79	98.6	74.5	77.1	79.9	52.3	86.8	78.2	82.4	2	1
190	74	75.9	76	77.4	78.6	70.7	80.1	74.6	74.5	1	2
191	82	77.8	62.5	76.5	80.3	58	85	82.5	76.3	1	1
192	65.4	70.3	81.3	85.9	65.1	80.8	56.9	66.9	78.9	1	1
193	87.5	83.9	96.7	97.6	81.2	87.2	91.6	98.6	94.8	2	2
194	80.5	70	70	82.6	70	72.1	74.5	80.8	74.2	1	1
195	78.9	83.3	67	84.5	78.4	87.8	77.2	71.6	79.8	2	1
196	79.3	82.2	67.8	74	80.7	59	80.8	75.5	64.7	1	1
197	83.9	82.9	81.7	81.6	77.2	76.1	78.4	85.3	77.2	2	2
198	74.5	95.8	66.4	79	83.9	64.7	79.6	81.8	73.7	2	1
199	91.8	92.1	83	83.7	86.4	83.7	91.5	92.7	78.8	2	2
200	96	85.7	60.9	76.4	81.8	60.1	85.7	83.3	74	2	1
201	71	82.7	68.9	79.8	62.4	77.2	64.3	77.8	78.8	1	1
202	85.2	70.6	80	78.6	84.2	81.7	76.2	78.1	75.9	2	2
203	98.3	80.7	79.4	89.1	98	85	94.1	90.7	73.9	2	2
204	97.6	86.7	79.1	78.7	85.6	77.4	80.4	77	89.8	2	2
205	91.7	90	70.3	78.6	79.9	63.2	74.4	79.9	83.3	2	1
206	89	68.2	71.3	66.1	82.1	74.7	78.4	76.4	65.3	1	1
207	90.9	85	90.1	80.6	92.7	80.7	91.5	92.7	90.9	2	2
208	90.4	90.9	89.7	78.6	87.5	86	87.1	75.6	89.9	2	2
209	83.8	96.7	77.6	96.1	94	87.7	83.6	85.2	87.5	2	2
210	81	89.3	71.7	83	83.1	66.6	67.5	79	75.8	1	1
211	82.2	63.3	60.3	82	64.3	85	82.9	75.5	76.7	1	1
212	72.7	90.2	78.2	92.5	85	81.8	67.4	80.7	79.4	2	1
213	74.6	90.3	80	62.9	83.2	80.5	77.3	58.3	62.4	1	1
214	76.1	90.4	77.7	89.9	84.2	86	89.4	85	91.3	2	2
215	84.4	93.1	69.9	75.2	83.1	77.3	57.4	71.2	79.7	1	2
216	73.5	89.7	61.7	79.7	67.4	73.1	76.5	72.7	67.9	1	1
217	75.6	81.2	62.3	74	63.8	58.2	70.9	60.4	78.1	1	1
218	80.7	81.9	66.8	82.6	88.3	78.6	66.6	76.8	78.5	2	1
219	58.4	81	57.7	64.6	82.7	58.7	82.2	80.4	61.9	1	1
220	84.4	85.7	70.1	67.8	65.2	81.8	81.1	83.6	62.9	1	1
221	76.2	85.3	83.1	85.2	73.3	81.4	78.3	78	78.2	2	2
222	60.4	80.9	72.3	73.7	80.4	58.7	76.3	77.9	58.4	1	1
223	74.1	90.5	85.1	86.9	76.5	66.7	75.2	87.5	80.3	2	1
224	78.6	75.8	72	59.1	70.1	78	63	59.8	59.2	1	1
225	74.8	98.8	82.9	60.8	54	80.3	56.4	54.8	79.7	1	1
226	81.5	90.9	67.8	81.7	79.1	65.4	63.4	84.7	75.9	1	1
227	92.8	82.5	93.5	90.7	94.3	81.5	81.4	77.7	88.9	2	2
228	64.7	68.8	79	56	81.9	65.6	76.7	81.1	67.8	1	1
229	83.6	76.2	72.1	79.5	71.2	85.2	83.2	89.8	81.5	2	2
230	76.7	79.6	74.2	83.7	77.9	82.4	89.2	78.9	81.9	2	2
231	88.5	73.6	87.5	84.7	81.2	74.4	58.4	89.1	83.4	2	2
232	82.3	87.2	63.4	71	57.5	78.1	79.3	74.5	78.2	1	1
233	76.9	84.5	82.3	65	87.2	82.2	75.3	74.5	74.5	2	1
234	81.3	81.5	61.5	63.6	81.9	76.3	83.4	75.5	60.5	1	1

235	82.8	83.4	79.2	80.3	91.6	80.1	92	81.8	85.1	2	2
236	72.3	73.7	63.8	84.9	84.7	75.4	83.6	77.5	61.5	1	1
237	84.7	75.2	81	88.7	86.7	71.2	79.1	80.1	84.3	2	2
238	76.8	64.4	78.4	75.4	76.7	86.2	80.8	84.6	82.1	2	2
239	75	87.3	78.5	71.2	77	78.2	69.9	81.4	74	1	2
240	90.9	92.6	77.6	81.1	72	84.5	90.1	81.9	87.3	2	2
241	70.9	78.8	65	68.3	45.1	63.5	72.4	68.5	82.5	1	1
242	80.4	72	85.9	84.6	70.4	48.9	60.7	54.5	36.3	1	1
243	75.9	92	82.9	86.6	83.5	44.4	57.7	58	41	1	1
244	91.8	61.3	49.9	54.6	55.5	56.2	84.1	72.4	67.9	1	1
245	50.6	58.9	38	60.4	44.7	51.8	88.8	71.5	69.4	1	1
246	72.6	84.1	74.6	85.8	88.1	58.7	59.6	88	59.6	1	1
247	85.5	87.9	84	66.8	89.6	79	84.4	81.2	83.6	2	2
248	74.1	98.5	73.6	82.3	84.5	82.5	82.9	74.2	74.7	2	2
249	59.3	64.5	68.9	63.5	80.8	69.2	71.3	63.4	58.2	1	1
250	88.3	90	84.1	76.2	65.7	81.9	86.1	75.9	88.3	2	2
251	78.2	96.8	78.9	84.4	88	61.7	71.8	83.4	90.9	2	1
252	72.9	85.7	84.8	77.6	85.4	81.3	80.4	83.3	85.2	2	2
253	80.2	92.8	78.2	86.2	82.6	88.1	81.7	72.2	78.8	2	2
254	90.9	62.1	73.1	66.2	73.7	70.7	90.5	76.4	66.9	1	1
255	77.5	53.3	82	65.3	74.3	68.2	53.8	77	71	1	1
256	89	70.2	58.2	68.7	52.7	54.5	63.6	53.4	85.7	1	1
257	71	68.5	66.5	61.1	78.7	64.9	65.3	74.8	73	1	1
258	63.6	66.3	67.3	62.9	72.3	67.6	75.5	61.1	74.4	1	1
259	74.5	82.7	67.9	61.5	81.9	60.3	70.8	77.3	64.3	1	1
260	79.4	95.6	77.4	83.9	70.7	50.9	81.9	80.5	77.4	2	1
261	85.4	92.8	83.1	82.6	79.9	59.3	67.9	75.4	73.6	2	1
262	85.7	92.7	54.8	58.6	55.4	58.9	58.1	74.3	58.4	1	1
263	79.1	76.3	83.5	67.5	76.1	60	66.4	87	76.4	1	1
264	61.5	89.2	56.7	73	62.8	59.7	63.9	61.2	61	1	1
265	76.7	70.6	77.2	85.4	79	57.5	78.3	78.2	60.8	1	1
266	85	58.1	90.4	58.2	79.2	73	71.1	72.4	69	1	1
267	82.1	56	60.4	60.5	81.4	60.2	72	87.7	66.2	1	1
268	77.5	78.6	63.2	58.5	45.2	59.1	52.7	66.7	66.6	1	1
269	49.4	57.6	35.9	51	57.1	54.5	50.2	72.6	46.1	1	1
270	80.1	63.1	53.4	58.9	50.6	49.2	62.6	58.6	53.7	1	1
271	64	65.6	68.2	57.8	74.4	68.4	67.6	84	61.8	1	1
272	64.1	75.3	67.4	57.2	78	60.7	79.1	79.1	70.7	1	1
273	85.5	64.6	60.1	65.4	81.1	66.6	59.8	78.9	55.1	1	1
274	66.7	70	65.9	63	89	68.8	61.9	63	71.3	1	1
275	76.3	66.5	54.4	68.9	63.6	63.5	54.2	64.8	70	1	1
276	65.8	69.4	57.4	71.5	69.5	64	65	63.5	68.6	1	1
277	78.3	78.5	70.8	61.1	59.8	63.6	61.6	86.9	75.2	1	1
278	79.4	98.2	85.5	87.1	79.8	80.6	72.3	83.9	78.5	2	2
279	85.3	77	78.6	81.2	80.5	61	68.4	84.1	80.5	1	1
280	57.8	97.4	80.4	84.9	66.9	73.9	67.3	76.6	82.8	1	1
281	76.6	82.1	86.9	73.3	87	62.4	74.7	78.1	65.7	1	1
282	64	46.2	61.7	57.5	63.3	83	64.5	69.5	67.8	1	1
283	79.3	96.3	85.2	60.9	73	72.6	66.9	68.4	80.4	1	1
284	66.7	38	55.3	64.5	56.9	39.4	55.7	55.1	37.6	1	1
285	82.7	97.9	82.8	63.5	69.9	68.5	73.9	60.2	87.2	1	1
286	78.9	85.8	39.1	53	50.1	33.4	65	58.3	43.7	1	1
287	65.3	84.5	71.4	57.6	57.1	44.9	61.1	59.8	32.2	1	1
288	73.3	77.6	60.7	62.3	85.4	64	78.9	84.2	60.6	1	1
289	75.5	73.4	69.8	71.1	67.4	58.8	65	65.3	75.1	1	1
290	62.2	58.9	69.8	60.3	59.6	54.3	71.5	65.1	70.4	1	1
291	65.3	82.8	68.2	63.5	79.7	60.1	60.1	67.1	64.9	1	1
292	70.3	99	62.4	67.2	69.9	78.1	62.4	56	76.3	1	1
293	95.2	94.4	94.4	95.8	93.2	91.8	86.8	93.9	93.8	2	2
294	58.4	63.5	66.9	55.6	52.8	57.8	55.8	66.7	59.8	1	1
295	93.1	73.3	92.5	79.2	71.9	73.4	83.8	73.5	86.3	2	2
296	74.4	91.6	77.5	75.2	81	82.1	84.6	83.8	84.7	2	2
297	87.3	80.4	73.1	68.1	89.9	83	85	78.4	82.9	2	2
298	82.1	96.2	82.9	80.6	75.4	76.1	79.9	87.8	79.1	2	2
299	78.9	99.3	88	87.2	83.4	74.2	78	79.3	72.4	2	2
300	65.3	82.8	68.2	63.5	79.7	60.1	60.1	67.1	64.9	1	1

