

**IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING*
UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA
REKOMENDASI PENERIMA BEASISWA**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

UNIVERSITAS BRAWIJAYA



Disusun Oleh :
Agung Putra Widiyanto
0910960023

**PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013**

LEMBAR PERSETUJUAN

**IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING*
UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA
REKOMENDASI PENERIMA BEASISWA**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :
Agung Putra Widiyanto
0910960023

Skripsi ini telah disetujui oleh dosen pembimbing
pada tanggal 22 April 2013

Dosen Pembimbing I,

Dosen Pembimbing II,

Lailil Muflikhah, S.Kom., M.Sc.
NIP. 197411132005012001

Budi Darma S., S.Kom., M.Cs.
NIK. 84101506110090

LEMBAR PENGESAHAN

**IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING*
UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA
REKOMENDASI PENERIMA BEASISWA**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

Agung Putra Widiyanto
0910960023

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 3 Mei 2013

Dosen Penguji I,

Dosen Penguji II,

Drs. Marji, MT.
NIP. 19670801 199203 1 001

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

Dosen Penguji III,

Ahmad Afif Supianto, S.Si., M.Kom.
NIK. 820623 16 1 1 0425

Mengetahui,

Ketua Program Studi Ilmu Komputer,

Drs. Marji, MT.
NIP. 19670801 199203 1 001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Agung Putra Widiyanto
NIM : 0910960023
Program Studi : Ilmu Komputer
Jurusan : Ilmu Komputer
Fakultas : Program Teknologi Informasi dan Ilmu Komputer
Penulis skripsi berjudul : IMPLEMENTASI ALGORITMA *SUBTRACTIVE*

CLUSTERING UNTUK PEMBANGKITAN
ATURAN *FUZZY* PADA REKOMENDASI
PENERIMA BEASISWA

Dengan ini menyatakan bahwa:

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam skripsi ini.
 2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia menanggung segala resiko yang akan saya terima.
- Demikian pernyataan ini dibuat dengan segala kesadaran dan penuh tanggung jawab dan digunakan sebagaimana mestinya.

Malang, 22 April 2013

Yang menyatakan,

Agung Putra Widiyanto
NIM. 0910960023

IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING* UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA REKOMENDASI PENERIMA BEASISWA

ABSTRAK

Pemerintah melalui Dikti sesuai kewenangannya dapat memberikan Beasiswa dengan persyaratan yang ditetapkan. Parameter yang ditetapkan untuk pemohon beasiswa diantaranya adalah IPK, SKS, prestasi, dan keadaan ekonomi. Pada aturan yang diterapkan pada proses penetapan penerima beasiswa terdapat unsur subjektifitas terhadap parameter tersebut sehingga membuat rumit dalam pengambilan keputusan penerima beasiswa. *Fuzzy inference system* merupakan algoritma inferensi yang dapat memberikan sudut pandang yang cukup luas dalam pengambilan keputusan. Proses pembelajaran untuk membentuk aturan dapat dibangkitkan secara otomatis tanpa perlu bantuan seorang pakar, yaitu dengan algoritma *clustering*. Konsep dasar dari *subtractive clustering* adalah menentukan titik data dengan densitas tertinggi sebagai pusat *cluster*, sehingga algoritma ini akan membentuk jumlah *cluster* secara otomatis tanpa perlu diinisialisasi diawal.

Penelitian ini menggunakan data set beasiswa tahun 2012 yang berasal dari PPTI-UB. Data set terdiri dari 100 data latih dan 30 data uji. Proses pembelajaran diawali dengan proses *clustering*, dari hasil *clustering* kemudian dilakukan analisa varian. Hasil *cluster* dengan nilai varian terkecil akan dijadikan bahan untuk proses ekstraksi aturan *fuzzy* menggunakan metode *least square estimator* (LSE). Setelah aturan terbentuk barulah dilakukan proses pengujian menggunakan *FIS* model Sugeno Orde-Satu. Dari hasil pengujian pembentukan aturan didapatkan jumlah aturan yang konvergen, yaitu 2 aturan. Sedangkan pengujian akurasi dipatkan hasil untuk akurasi sistem yang konvergen pula, yaitu sebesar 93.33% pada jumlah 2 aturan.

Kata kunci : beasiswa Dikti, *subtractive clustering*, ekstraksi aturan *fuzzy*, *fuzzy inference system* model sugeno

repository.ub.ac.id

IMPLEMENTATION OF SUBTRACTIVE CLUSTERING ALGORITHM FOR FUZZY RULES GENERATION ON RECOMMENDATIONS SCHOLARSHIP RECIPIENTS

ABSTRACT

Through the appropriate government authorities of *Directorate General of Higher Education (DIKTI)* to provide scholarship to the requirements set. Parameters set for scholarship applicants include Cumulative Grade Point (IPK), credits, achievement, and state of the economy. On the rules that apply to the determination of the scholarship recipients are elements of subjectivity to these parameters so as to complicate the decision-making for awardees. *Fuzzy inference system* is an algorithm that can provide a broad enough perspective in decision-making. Learning process to establish the rules can be generated automatically without needing the help of an expert, ie with the clustering algorithm. The basic concept of the *subtractive clustering* is determining the data points with the highest density as the *cluster* center, so that the algorithm will form a number of *cluster* or rules automatically without the need to be initialized at the beginning.

This study uses a data set of bids and awardees in 2012 were derived from PPTI-UB. The data set consist of 100 training data and 30 test data. The learning process begins with the process of *clustering*, from *clustering* results then performed analysis of variance. *Cluster* results with the smallest variance value will be used as material for the extraction of *fuzzy* rules using the methods of *least square estimation* (LSE). Once the rules are formed then do the testing process for test data using Sugeno's *fuzzy inference system* first-order. From the test results obtained formation rules convergent sum rules, namely 2 rules. While testing the accuracy of the results obtained for the system converging accuracy as well, amounting to 93,33 %.

Keyword: DIKTI's scholarship, *subtractive clustering*, extraction of *fuzzy* rules, Sugeno's *fuzzy inference system* models.

KATA PENGANTAR

Syukur *alhamdulillah* penulis panjatkan kehadiran Allah Yang Maha Esa karena berkat rahmat serta hidayah-Nya penulis dapat menyelesaikan skripsi dengan mengambil judul “IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING* UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA REKOMENDASI PENERIMA BEASISWA” sebagai salah satu syarat memperoleh gelar Sarjana Komputer pada program studi Ilmu Komputer/Teknik Informatika PTIIK Universitas Brawijaya - Malang .

Tidak dapat dipungkiri bahwa tidak mungkin penulis dapat menyelesaikan skripsi ini tanpa dukungan serta bantuan dari berbagai pihak. Penulis tidak lupa untuk menyampaikan penghargaan lewat rasa hormat dan terimakasih kepada semua pihak yang telah ikut andil dan berkontribusi, baik dalam bentuk bantuan tenaga, pikiran maupun dukungan moral selama penulisan skripsi sehingga dapat terselesaikannya skripsi ini, yang diantaranya kepada:

1. Ibu, Bapak, dan seluruh keluarga yang selalu memberikan motivasi dan doa kepada penulis.
2. Lailil Muflikhah, S.Kom., M.Sc., selaku pembimbing utama yang telah meluangkan waktu, tenaga, serta pikiran dalam mengarahkan dan membimbing kepada penulis.
3. Budi Darma S., S.Kom., M.Cs., selaku pembimbing pendamping yang telah meluangkan waktu, tenaga, serta pikiran dalam mengarahkan dan membimbing kepada penulis.
4. Drs. Mardji, MT., selaku dosen pembimbing akademik sekaligus Ketua Program Studi Teknik Informatika PTIIK Universitas Brawijaya – Malang.
5. Seluruh Bapak dan Ibu Dosen yang telah memberikan ilmu – ilmu dan pelajaran berharga kepada penulis.
6. Orang tercinta yang selama ini mengiringi langkah penulis dalam menuntut ilmu di Universitas Brawijaya – Malang.
7. Sahabat di Ilkom A 2009 dan HIMAMASTER – UB yang telah bersama – sama dengan penulis baik dikala suka maupun duka.

- repository.ub.ac.id
8. Sahabat, kakak serta adik tingkat, dan seluruh teman – teman di program studi Ilmu Komputer/Teknik Informatika ataupun selama di bangku kuliah yang telah selalu bersama dalam perjalanan mencari ilmu.
 9. Dan semua pihak yang telah membantu baik secara langsung maupun tidak langsung yang tidak bisa disebutkan satu – persatu.

Sebagai manusia biasa, penulis sadar bahwa begitu banyak kekurangan entah itu disengaja ataupun dari kebelummampuan penulis dalam menulis skripsi ini, oleh karena itu saran dan kritik dengan nuansa positif yang bertujuan membangunlah yang sangat dibutuhkan oleh penulis untuk memperbaiki skripsi ini agar kedepannya lebih baik. Penulis berharap skripsi ini dapat bermanfaat bagi penulis sendiri dan bagi semua pihak.



Malang, April 2013

Penulis

DAFTAR ISI

HALAMAN SAMBUTAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR SOURCE CODE	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 Beasiswa.....	7
2.1.1 Pengertian Beasiswa.....	7
2.1.2 Beasiswa Dikti.....	7
2.1.3 Beasiswa PPA.....	8
2.2 <i>Data Mining</i>	9
2.2.1 Pengertian <i>Data Mining</i>	9
2.2.2 Proses <i>Data Mining</i>	10
2.2.3 Fungsionalitas <i>Data Mining</i>	10
2.3 <i>Clustering</i>	13
2.3.1 Pengertian <i>Clustering</i>	13
2.3.2 Tipe <i>Clustering</i>	13



2.4 <i>Subtractive Clustering</i>	14
2.4.1 Pengertian <i>Subtractive Clustering</i>	14
2.4.2 Algoritma <i>Subtractive Clustering</i>	16
2.5 Analisa Cluster	19
2.5.1 Analisa Varian	19
2.6 Logika <i>Fuzzy</i>	20
2.6.1 Pengertian Logika <i>Fuzzy</i>	20
2.6.2 Himpunan <i>Fuzzy</i>	21
2.6.3 Fungsi Keanggotaan	22
2.6.4 Operator Dasar Zadeh	25
2.6.5 Fungsi Implikasi	26
2.6.6 <i>Fuzzy Inference System</i>	27
2.6.7 Teknik Pembangkitan Aturan <i>Fuzzy</i>	27
2.6.8 Ekstraksi Aturan <i>Fuzzy</i>	28
2.7 <i>LSE</i>	30
2.8 Akurasi	32
BAB III METODOLOGI DAN PERANCANGAN SISTEM	33
3.1 Studi Literatur	34
3.2 Data Penelitian	34
3.3 Analisa dan Perancangan Sistem.....	34
3.3.1 Deskripsi Umum Sistem	34
3.3.2 Perancangan Sistem	36
3.3.2.1 Proses <i>Clustering</i> dengan <i>Subtractive Clustering</i>	38
3.3.2.2 Proses Perhitungan Varian.....	53
3.3.2.3 Proses Pemilihan Jumlah <i>Cluster</i> dengan Varian Terkecil	54
3.3.2.4 Proses Ekstraksi Aturan <i>Fuzzy</i> dari <i>Cluster</i>	56
3.3.2.5 Proses Sistem Inferensi <i>Fuzzy</i>	60
3.3.3 Analisa dan Perancangan <i>Database</i>	62
3.3.4 Analisa dan Perancangan Antarmuka	63
3.4 Perhitungan Manual	65
3.4.1 Proses Pembelajaran	65

3.4.2	Proses Pengujian	89
3.5	Perancangan dan Pengujian Analisis	91
BAB IV IMPLEMENTASI		93
4.1	Lingkungan Implementasi	93
4.1.1	Lingkungan Implementasi Perangkat Keras	93
4.1.2	Lingkungan Implementasi Perangkat Lunak	93
4.2	Implementasi Program	93
4.2.1	Implementasi Proses <i>Subtractive Clustering</i>	94
4.2.2	Implementasi Proses Perhitungan Varian	100
4.2.3	Implementasi Proses Ekstraksi Aturan <i>Fuzzy</i>	103
4.2.4	Implementasi Proses Sistem Inferensi <i>Fuzzy</i>	105
4.3	Implementasi Antarmuka	109
4.3.1	Antarmuka Proses Pembelajaran	109
4.3.2	Antarmuka Hasil Pengujian Data Uji	110
BAB V ANALISA HASIL DAN PEMBAHASAN		111
5.1	Skenario Pengujian	111
5.2	Hasil Pengujian	112
5.2.1	Hasil Pengujian Tahap Satu	112
5.2.2	Hasil pengujian Tahap Dua	121
5.3	Analisa Hasil	124
5.3.1	Analisa Hasil Pengujian Tahap Satu	124
5.3.2	Analisa Hasil Pengujian Tahap Dua	127
BAB VI PENUTUP		129
6.1	Kesimpulan	129
6.2	Saran	130
DAFTAR PUSTAKA		131
LAMPIRAN		133

DAFTAR TABEL

Tabel 3.1 Hasil <i>Subtractive Clustering</i>	83
Tabel 3.2 Tabel pengujian pembentukan aturan	92
Tabel 3.3 Tabel uji akurasi.....	92
Tabel 4.1 Kelas – kelas pembentuk aplikasi rekomendasi beasiswa.....	94
Tabel 4.2 <i>Method – method</i> pembentuk kelas <i>SubtractiveClustering.java</i>	94
Tabel 4.3 <i>Method – method</i> pembentuk kelas <i>varianCluster.java</i>	100
Tabel 4.4 <i>Method – method</i> pembentuk kelas <i>SistemInferensiFuzzy.java</i>	105
Tabel 5.1 Tabel hasil pengujian pertama tahap satu (<i>reject ratio</i> 0,15 – 0,35).....	112
Tabel 5.2 Tabel hasil pengujian pertama tahap satu (<i>reject ratio</i> 0,4 – 0,5).....	113
Tabel 5.3 Tabel aturan pengujian pertama tahap satu.....	113
Tabel 5.4 Tabel hasil pengujian kedua tahap satu (<i>reject ratio</i> 0,15 – 0,35).....	115
Tabel 5.5 Tabel hasil pengujian kedua tahap satu (<i>reject ratio</i> 0,4 – 0,5).....	115
Tabel 5.6 Tabel aturan pengujian kedua tahap satu.....	115
Tabel 5.7 Tabel hasil pengujian ketiga tahap satu (<i>reject ratio</i> 0,15 – 0,35).....	117
Tabel 5.8 Tabel hasil pengujian ketiga tahap satu (<i>reject ratio</i> 0,4 – 0,5).....	117
Tabel 5.9 Tabel aturan pengujian ketiga tahap satu.....	118
Tabel 5.10 Tabel hasil pengujian keempat tahap satu (<i>reject ratio</i> 0,15 – 0,35).....	119
Tabel 5.11 Tabel hasil pengujian keempat tahap satu (<i>reject ratio</i> 0,4 – 0,5).....	120
Tabel 5.12 Tabel aturan pengujian keempat tahap satu.....	120

Tabel 5.13	Tabel akurasi hasil pengujian pertama.....	122
Tabel 5.14	Tabel akurasi hasil pengujian kedua.....	122
Tabel 5.15	Tabel akurasi hasil pengujian ketiga.....	123
Tabel 5.16	Tabel akurasi hasil pengujian keempat.....	123



DAFTAR GAMBAR

Gambar 2.1	Proses <i>Data Mining</i>	10
Gambar 2.2	Kurva Segitiga.....	23
Gambar 2.3	Kurva Trapesium.....	24
Gambar 2.4	Karakteristik Fungsi <i>Gauss</i>	24
Gambar 2.5	Karakteristik Fungsi <i>Generalized Bell</i>	25
Gambar 2.6	Kurva <i>Sigmoid</i>	25
Gambar 2.7	Fungsi Implikasi <i>MIN</i>	26
Gambar 2.8	Fungsi Implikasi <i>DOT</i>	27
Gambar 3.1	Langkah – langkah penelitian.....	33
Gambar 3.2	Gambaran umum perancangan sistem.....	37
Gambar 3.3	Tahapan proses <i>clustering</i>	38
Gambar 3.4	Alur proses normalisasi data	40
Gambar 3.5	Alur proses penentuan potensi awal tiap titik data.....	41
Gambar 3.6	Alur proses penentuan titik dengan potensi tertinggi	43
Gambar 3.7	Alur proses penentuan pusat <i>cluster</i>	45
Gambar 3.8	Alur proses <i>cluster</i> diterima.....	46
Gambar 3.9	Alur proses pertimbangan pusat <i>cluster</i>	48
Gambar 3.10	Alur proses pengembalian pusat <i>cluster</i>	50
Gambar 3.11	Alur proses perhitungan nilai sigma.....	51
Gambar 3.12	Alur proses pengelompokan data pada <i>cluster</i>	52
Gambar 3.13	Alur proses perhitungan varian.....	53
Gambar 3.14	Alur proses pemilihan jumlah <i>cluster</i> dengan varian terkecil	55
Gambar 3.15	Alur proses ekstraksi aturan <i>fuzzy</i> dari <i>cluster</i>	56
Gambar 3.16	Alur proses perhitungan koefisien <i>output</i>	57
Gambar 3.17	Alur proses normalisasi.....	58
Gambar 3.18	Alur proses pembentukan matriks <i>U</i>	59
Gambar 3.19	Alur proses perhitungan <i>LSE</i>	60
Gambar 3.20	Alur proses sistem inferensi <i>fuzzy</i>	61
Gambar 3.21	<i>Physical data model</i>	63

Gambar 3.22	Antarmuka proses pembelajaran data latih.....	63
Gambar 3.23	Antarmuka proses pengujian data uji.....	64
Gambar 4.1	Antarmuka proses pembelajaran.....	110
Gambar 4.2	Antarmuka proses pengujian.....	110
Gambar 5.1	Grafik hubungan nilai jari – jari terhadap jumlah <i>cluster</i> pada pengujian pertama tahap satu.....	125
Gambar 5.2	Grafik hubungan nilai jari – jari terhadap jumlah <i>cluster</i> pada pengujian kedua tahap satu.....	125
Gambar 5.3	Grafik hubungan nilai jari – jari terhadap jumlah <i>cluster</i> pada pengujian ketiga tahap satu.....	126
Gambar 5.4	Grafik hubungan nilai jari – jari terhadap jumlah <i>cluster</i> pada pengujian keempat tahap satu.....	126
Gambar 5.5	Grafik akurasi aturan pengujian tahap dua.....	128



DAFTAR SOURCE CODE

<i>Source code 4.1 Listing program proses masukan data latihan</i>	95
<i>Source code 4.2 Listing program proses normalisasi</i>	96
<i>Source code 4.3 Listing program proses penentuan potensi awal tiap titik data</i>	96
<i>Source code 4.4 Listing program proses pencarian titik dengan potensi tertinggi</i>	97
<i>Source code 4.5 Listing program proses penentuan pusat cluster</i>	98
<i>Source code 4.2 Listing program proses pengembalian pusat cluster</i>	99
<i>Source code 4.2 Listing program proses perhitungan nilai sigma cluster</i>	100
<i>Source code 4.2 Listing program proses perhitungan variap tiap cluster</i>	101
<i>Source code 4.2 Listing program proses perhitungan variance within cluster</i>	102
<i>Source code 4.2 Listing program proses perhitungan variance between cluster</i>	102
<i>Source code 4.2 Listing program proses perhitungan batasan varian</i>	103
<i>Source code 4.2 Listing program proses perhitungan nilai derajat keanggotaan</i>	103
<i>Source code 4.2 Listing program proses perhitungan koefisien output</i>	104
<i>Source code 4.2 Listing program proses pembacaan data uji</i>	106
<i>Source code 4.2 Listing program proses inialisasi pusat cluster</i>	107
<i>Source code 4.2 Listing program proses perhitungan derajat keanggotaan</i>	107
<i>Source code 4.2 Listing program proses perhitungan alpha predikat</i>	107
<i>Source code 4.2 Listing program proses perhitungan nilai z</i>	108
<i>Source code 4.2 Listing program proses defuzzy</i>	108



BAB I PENDAHULUAN

2.1. Latar Belakang

Peraturan Pemerintah Nomor 48 tahun 2008 tentang Pendanaan Pendidikan, Bagian Kelima, Pasal 27 ayat (1), menyebutkan bahwa Pemerintah dan pemerintah daerah sesuai kewenangannya memberi bantuan biaya pendidikan atau beasiswa kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikannya. Pasal 27 ayat (2), menyebutkan bahwa Pemerintah dan pemerintah daerah sesuai dengan kewenangannya dapat memberi beasiswa kepada peserta didik yang berprestasi. Mengacu kepada Undang-undang dan Peraturan Pemerintah tersebut, maka Pemerintah melalui Direktorat Jenderal Pendidikan Tinggi – Kementerian Pendidikan Nasional, mengupayakan pemberian bantuan biaya pendidikan bagi mahasiswa yang orang tua/walinya kurang mampu membiayai pendidikan, dalam bentuk Bantuan Biaya Mahasiswa (BBM) dan Beasiswa bagi mahasiswa berprestasi dalam bentuk Beasiswa Peningkatan Prestasi Akademik (PPA)[DIR-11]. Untuk itu setiap tahunnya Universitas Brawijaya mengadakan penjarangan penerima beasiswa dikti sesuai dengan ketentuan yang telah ditetapkan oleh dikti.

Dikti memberikan prioritas penerima beasiswa berdasarkan parameter yang telah ditentukan sebelumnya, seperti nilai IPK tertinggi, SKS paling banyak, prestasi, dan keadaan ekonomi[DIR-11]. Aturan yang diterapkan pada proses penetapan penerima beasiswa terdapat unsur subjektifitas terhadap parameter tersebut yang membuat rumit dalam pengambilan keputusan penerima beasiswa, karena pada dasarnya variabel pada data mahasiswa satu dengan lainnya juga berkaitan atau saling mempengaruhi pada proses penetapan penerima beasiswa. Sehingga tidak dapat dengan mudah menerapkan batasan penerima beasiswa berdasarkan parameter tersebut. Oleh karena itulah diperlukan suatu metode inferensi sebagai pendukung keputusan siapa – siapa penerima beasiswa yang lebih tepat berdasarkan kategori beasiswa, BBM atau PPA.

Fuzzy Inference System merupakan salah satu alternatif yang dapat digunakan untuk menyelesaikan persoalan penetapan penerima beasiswa. Keluwesan dari logika *fuzzy* bisa memberikan sudut pandang yang cukup luas

dalam membantu penentuan penerima beasiswa. Logika *Fuzzy* memodelkan fungsi - fungsi *input/output* yang sangat kompleks, sehingga alternatif hasil dari sistem bersifat luas. Logika *Fuzzy* merupakan logika yang memiliki nilai kekaburan atau kesamaran yang digunakan untuk melakukan penalaran [SKH-10:1]. Untuk inferensi penalaran diperlukan sebuah aturan - aturan. Biasanya aturan tersebut berasal dari seorang pakar bukan semerta - merta ada atau ditentukan sendiri. Namun seringkali seorang pakar tidak dapat mengungkapkan pengetahuannya secara eksplisit[CHS-97]. Selain itu terkadang aturan yang dihasilkan kurang optimal, bahkan untuk kasus – kasus tertentu belum terdapat pakar yang bisa membangun aturan sebagai bahan sistem inferensi *fuzzy*.

Penelitian terkait *fuzzy inference system* pada kasus rekomendasi penerima beasiswa sudah pernah dilakukan oleh Husnayadi Arief (2009). Pada penelitiannya aturan yang digunakan sebagai bahan inferensi *fuzzy* berasal dari pakar. Namun penelitian tersebut hanya menghasilkan tingkat akurasi sebesar 70%[HUA-09]. Berdasarkan kelemahan akan keakuratan aturan yang berasal dari pakar, maka pada tahun 2012 dilakukan penelitian lanjutan oleh Ida Nurlaili dengan menggunakan metode *k-means clustering* sebagai media pembelajaran dalam membangkitkan aturan. Aturan yang berasal dari pakar bisa digantikan dengan menggunakan pembangkitan aturan. Aturan tersebut dibangkitkan dengan menggunakan algoritma *clustering* seperti FCM, *k-means*, *subtractive clustering* dan *nearest neighborhood clustering* [AKH-10:4]. Jumlah aturan yang dibangkitkan nantinya sama dengan jumlah *cluster*[SKH-10:56].

K-means memiliki komputasi yang tidak terlalu tinggi jika dibandingkan metode lainnya, namun tetap handal untuk mengolah data dalam jumlah yang besar. Meski memiliki beberapa kehandalan, algoritma *k-means* juga memiliki kekurangan. Salah satu permasalahan yang ada pada *k-means* adalah penentuan jumlah *cluster* atau nilai *k* yang tepat [YUA-07:50]. Jika inisialisasi jumlah *cluster* tidak tepat, maka nantinya akan mempengaruhi hasil inferensi *fuzzy* dengan aturan yang telah dibangkitkan. Hal ini diperkuat oleh hasil penelitian Ida Nurlaili (2012) yang menyebutkan perlu adanya optimasi untuk pemilihan jumlah aturan yang akan dibangkitkan[IDN-12:103]. Oleh karena itu diperlukan sebuah algoritma

clustering lain sebagai alternatif pengganti algoritma *k-means* dengan harapan dapat mengatasi permasalahan terhadap pemilihan jumlah *cluster*.

Apabila jumlah *cluster* yang akan dibentuk belum diketahui sebelumnya, maka kita harus menggunakan algoritma *clustering* yang tidak terawasi. *Subtractive clustering* didasarkan atas ukuran densitas (potensi) titik – titik data dalam suatu ruang (variabel). Konsep dasar dari *subtractive clustering* adalah menentukan daerah – daerah dalam suatu variabel yang memiliki densitas tinggi terhadap titik – titik di sekitarnya. Titik dengan jumlah terbanyak akan dipilih sebagai pusat *cluster*. Titik yang sudah terpilih sebagai pusat *cluster* ini kemudian akan dikurangi densitasnya. Kemudian algoritma akan memilih titik lain yang memiliki tetangga terbanyak untuk dijadikan pusat *cluster* yang lain. Hal ini akan dilakukan berulang – ulang hingga semua titik diuji[SKH-96]. Selain itu pada penelitian yang dilakukan oleh Arifin (2009) menyebutkan bahwa *subtractive clustering* lebih baik dibandingkan dengan *k-means* berdasarkan kualitas pengelompokan. *Subtractive clustering* juga lebih handal dalam menghadapi data *outlier* dibandingkan dengan *k-means*[ARI-09:1].

Pembentukan aturan dari *clustering* dapat menggunakan *fuzzy inference system* metode sugeno orde satu. Pemilihan metode sugeno orde satu dikarenakan konsekuen(output) berupa kumpulan konstanta[SKH-10:46]. Sedangkan untuk menghitung nilai derajat keanggotaan pada masing – masing aturan dapat menggunakan fungsi *gauss* dengan bantuan nilai pusat *cluster* dan sigma[SKH-10:152].

Berdasarkan uraian yang telah dipaparkan, maka dilakukan sebuah penelitian dengan mengambil judul “IMPLEMENTASI ALGORITMA *SUBTRACTIVE CLUSTERING* UNTUK PEMBANGKITAN ATURAN *FUZZY* PADA REKOMENDASI PENERIMA BEASISWA”.

2.2. Rumusan Masalah

Adapun permasalahan yang dapat dirumuskan dari latar belakang di atas adalah sebagai berikut :

- 1) Bagaimana membangkitkan jumlah *cluster* untuk pembentukan aturan *fuzzy* menggunakan algoritma *subtractive clustering*.

- 2) Bagaimana menerapkan algoritma *subtractive clustering* untuk membangkitkan aturan *fuzzy* pada inferensi penerimaan beasiswa.
- 3) Bagaimana tingkat akurasi hasil inferensi *fuzzy* berdasarkan pembangkitan aturan pada inferensi penerima beasiswa menggunakan algoritma *subtractive clustering*.

2.3. Batasan Masalah

Berdasarkan rumusan masalah di atas, penulis perlu memberikan batasan masalah sebagai berikut :

- 1) Data yang digunakan pada penelitian ini berasal dari bagian kemahasiswaan Universitas Brawijaya dan Pusat Pengembangan Teknologi Informasi (PPTI) Universitas Brawijaya pada periode penawaran dan penerimaan beasiswa tahun 2012.
- 2) Data latih dan data uji yang akan digunakan diklasifikasikan sesuai angkatan data pemohon beasiswa, yaitu untuk mahasiswa angkatan 2009 Universitas Brawijaya.
- 3) Jenis beasiswa pada penelitian ini merupakan beasiswa Dikti, yaitu pada beasiswa PPA jenis pengembangan akademik.
- 4) *Fuzzy inference system* yang digunakan adalah model Sugeno orde satu dengan aturan yang dibangkitkan oleh algoritma *subtractive clustering*.
- 5) Tingkat akurasi dan kinerja sistem yang akan diimplementasikan, yaitu kesesuaian hasil inferensi *fuzzy* dengan aturan yang dibangkitkan oleh algoritma *subtractive clustering* terhadap data uji sesuai batasan penerima beasiswa.

2.4. Tujuan

Tujuan dari penelitian ini, antara lain:

- 1) Menerapkan algoritma *subtractive clustering* pada pembentukan jumlah *cluster* untuk pembangkitan aturan *fuzzy* pada kasus penerimaan beasiswa.

- 2) Mengetahui tingkat akurasi sistem yang akan diimplementasikan (pembentukan jumlah *cluster* sebagai pembangkitan aturan *fuzzy* menggunakan algoritma *subtractive clustering*) pada penerimaan beasiswa.

2.5. Manfaat

Manfaat dari penelitian ini adalah didapatkan sebuah metode baru dalam rekomendasi penerimaan beasiswa sehingga dapat membantu pemegang kebijakan di kemahasiswaan Universitas Brawijaya bagian beasiswa. Kemudian juga bisa menjadi acuan bagi Universitas Brawijaya dalam mengeluarkan kebijakan khususnya pada penerimaan beasiswa. Pada sisi akademisi, penelitian dapat bermanfaat sebagai tambahan referensi guna dikembangkan kearah penelitian yang lebih lanjut.

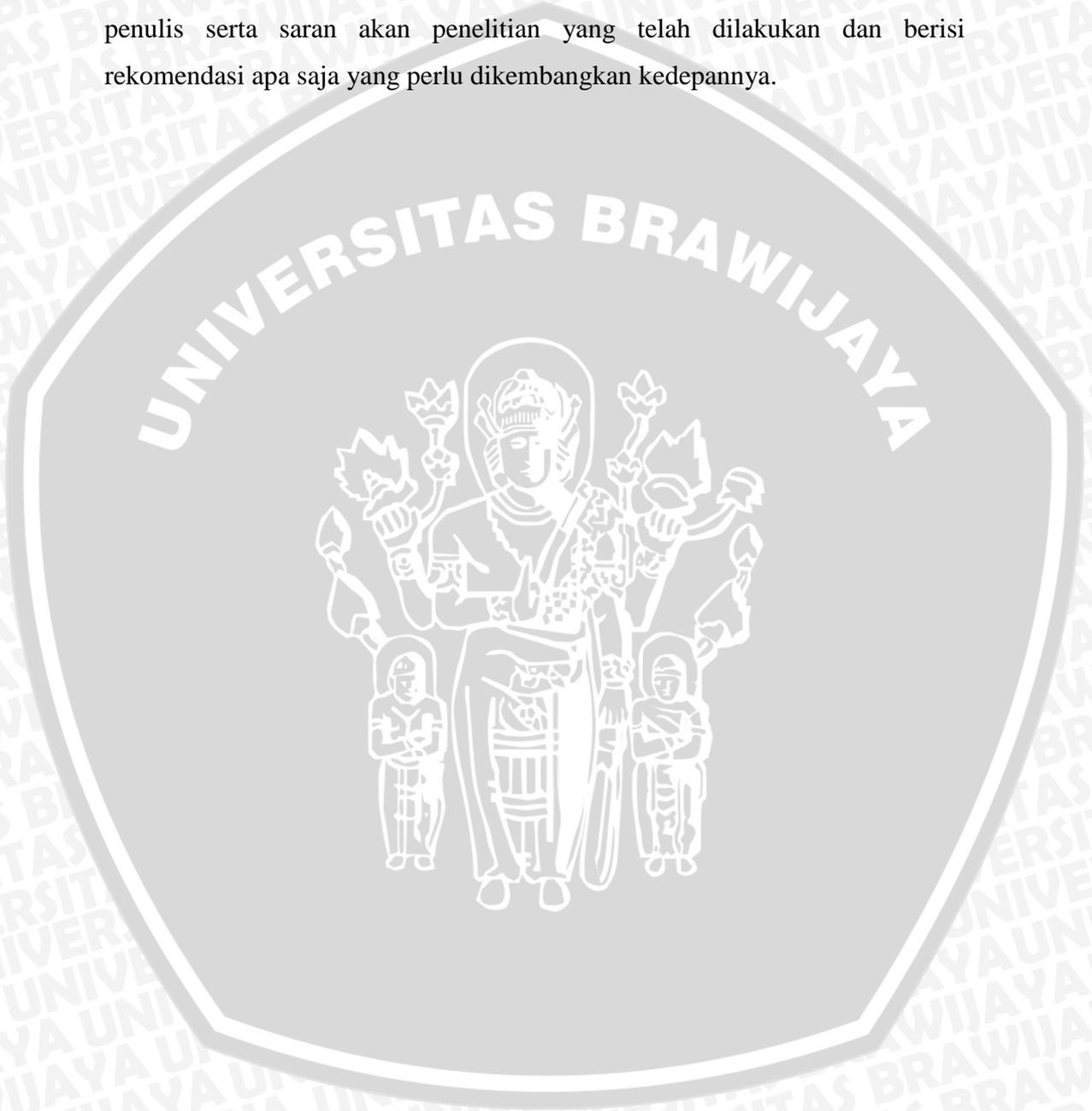
2.6. Sistematika Penulisan

Pembuatan hasil penelitian yang didokumentasikan dalam bentuk skripsi ini berdasarkan sistematika penulisan sebagai berikut:

- 1) BAB I PENDAHULUAN
Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan dari penelitian ini.
- 2) BAB II TINJAUAN PUSTAKA
Berisi hal – hal yang berkaitan dengan objek penelitian dimana didalamnya terdapat teori – teori sebagai landasan dasar dilakukannya penelitian dan penulisan penelitian ini.
- 3) BAB III METODOLOGI DAN PERANCANGAN SISTEM
Menjelaskan tentang metode – metode atau langkah – langkah serta perancangan sistem yang akan dilakukan dalam penelitian.
- 4) BAB IV IMPLEMENTASI
Bab ini berisi tahapan implementasi algoritma *subtractive clustering* untuk pembangkitan aturan *fuzzy* pada sistem penerimaan beasiswa.
- 5) BAB V ANALISA HASIL DAN PEMBAHASAN
Menjelaskan bagaimana pembahasan hasil pengujian dari implementasi sistem.

6) BAB VI PENUTUP

Bab terakhir sebagai penutup dimana didalamnya berisikan kesimpulan dari penulis serta saran akan penelitian yang telah dilakukan dan berisi rekomendasi apa saja yang perlu dikembangkan kedepannya.



BAB II TINJAUAN PUSTAKA

2.1. Beasiswa

2.1.1. Pengertian Beasiswa

Beasiswa adalah tunjangan uang yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar[KBI-97:103]. Biasanya beasiswa secara umum terbagi menjadi dua, yaitu bantuan pendidikan yang diberikan kepada pelajar atau mahasiswa yang mengalami kesulitan ekonomi dan/atau memiliki prestasi, baik dibidang akademik maupun non akademik.

2.1.2. Beasiswa DIKTI

Peraturan Pemerintah Nomor 48 tahun 2008 tentang Pendanaan Pendidikan, Bagian Kelima, Pasal 27 ayat (1), menyebutkan bahwa Pemerintah dan pemerintah daerah sesuai kewenangannya memberi bantuan biaya pendidikan atau beasiswa kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikannya. Pasal 27 ayat (2), menyebutkan bahwa Pemerintah dan pemerintah daerah sesuai dengan kewenangannya dapat memberi beasiswa kepada peserta didik yang berprestasi.

Mengacu kepada Undang-undang dan Peraturan Pemerintah tersebut, maka Pemerintah melalui Direktorat Jenderal Pendidikan Tinggi – Kementerian Pendidikan Nasional, mengupayakan pemberian bantuan biaya pendidikan bagi mahasiswa yang orang tua/walinya kurang mampu membiayai pendidikan, dalam bentuk Bantuan Biaya Mahasiswa (BBM) dan Beasiswa bagi mahasiswa berprestasi dalam bentuk Beasiswa Peningkatan Prestasi Akademik (PPA)[DIR-11].

Berdasarkan aturan dari dikti, bahwa penerima beasiswa harus memiliki syarat dan ketentuan umum bahwa mahasiswa yang berhak menerima beasiswa adalah mahasiswa :

- a. Jenjang S1/Diploma IV, paling rendah duduk pada semester II dan paling tinggi duduk pada semester VIII.
- b. Diploma III, paling rendah duduk pada semester II dan paling tinggi duduk pada semester VI.

2.1.3. Beasiswa PPA

Beasiswa PPA atau peningkatan prestasi akademik merupakan jenis beasiswa yang ditawarkan kepada mahasiswa yang memiliki prestasi di bidang akademik maupun non akademik[DIR-11]. Apabila calon penerima melebihi kuota yang telah ditetapkan, maka perguruan tinggi dapat menentukan mahasiswa penerima beasiswa sesuai dengan urutan berikut:

- 1) Mahasiswa yang mempunyai IPK paling tinggi.
- 2) Mahasiswa yang mempunyai SKS paling banyak (jumlah semester paling sedikit)
- 3) Mahasiswa yang memiliki prestasi dikegiatan ko/ekstrakurikuler (olahraga, teknologi, seni/budaya, tingkat internasional/dunia, regional/Asia/Asean dan Nasional).
- 4) Mahasiswa yang (orang tuanya) paling tidak mampu.

Nilai IPK dan jumlah SKS didapatkan dari nilai IPK dan jumlah SKS terakhir pemohon beasiswa pada saat pendaftaran beasiswa. Parameter prestasi berupa nilai numerik dari hasil konversi prestasi pemohon beasiswa pada 1 tahun terakhir. Universitas Brawijaya mendefinisikan prestasi menjadi dua kategori, yaitu untuk prestasi berupa kejuaraan atau perlombaan dan prestasi berupa keaktifan dalam organisasi. Proses konversi prestasi menjadi nilai numerik dilakukan dengan memberikan penilaian terhadap prestasi kejuaraan atau perlombaan dari pemohon seperti berikut:

- a. Tingkat Kabupaten: 5 point untuk juara 1, 3.5 point untuk juara 2, dan 2.5 point untuk juara 3.
- b. Tingkat Provinsi: 10 point untuk juara 1, 8.5 point untuk juara 2, dan 7.5 point untuk juara 3.
- c. Tingkat Nasional: 15 point untuk juara 1, 13.5 point untuk juara 2, dan 12.5 point untuk juara 3.
- d. Tingkat Internasional: 20 point untuk juara 1, 18.5 point untuk juara 2, dan 17.5 point untuk juara 3

Sedangkan penilaian untuk prestasi keaktifan dalam organisasi dari pemohon beasiswa adalah sebagai berikut:

- a. Tingkat program studi/jurusan: Ketua pelaksana diberikan point 5, sekretaris/bendahara/koordinator bidang diberikan point 3, sedangkan anggota sebesar 1.5 point.
- b. Tingkat fakultas: Ketua pelaksana diberikan point 7, sekretaris/bendahara/koordinator bidang diberikan point 5, sedangkan anggota sebesar 3 point.
- c. Tingkat universitas: Ketua pelaksana diberikan point 9, sekretaris/bendahara/koordinator bidang diberikan point 7, sedangkan anggota sebesar 5 point.
- d. Tingkat program studi/jurusan: Ketua organisasi diberikan point 10, sekretaris/bendahara/koordinator bidang organisasi diberikan point 8, sedangkan anggota organisasi sebesar 5 point.
- e. Tingkat fakultas: Ketua organisasi diberikan point 13, sekretaris/bendahara/koordinator bidang organisasi diberikan point 10, sedangkan anggota organisasi sebesar 7 point.
- f. Tingkat universitas: Ketua organisasi diberikan point 15, sekretaris/bendahara/koordinator bidang organisasi diberikan point 12, sedangkan anggota organisasi sebesar 9 point.

Keadaan ekonomi orang tua dilihat dari hasil pengurangan antara gaji orang tua dengan biaya listrik, PBB (dalam bulan), biaya air, dan telepon.

2.2. Data Mining

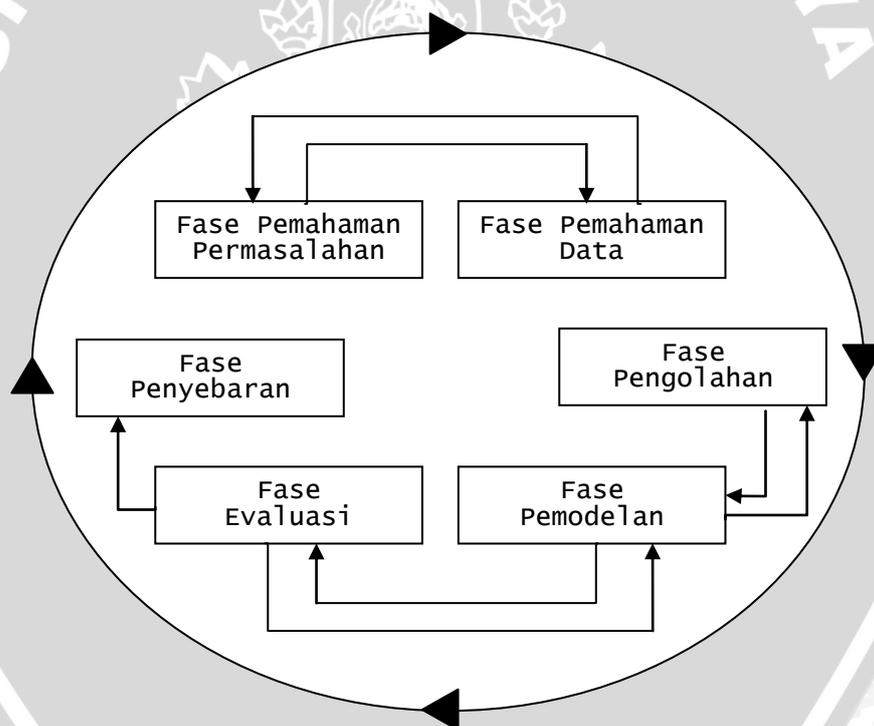
2.2.1. Pengertian Data Mining

Pada beberapa literatur dijelaskan dengan bahasa yang berbeda terkait *data mining*, namun tetap memiliki makna yang hampir sama dalam menggambarkan *data mining*. Menurut Dr. Daniel T. Larose (2005) menjelaskan bahwa data mining adalah sebuah proses pencarian hubungan yang lebih berarti, pengenalan pola dan mencari kecenderungan data yang disimpan pada repositori [DAT-05:2]. Sedangkan menurut Kusri dan Emha (2009) *data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu [KUE-09:7]. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses sistem secara keseluruhan. Untuk itu perlu ada kajian

mendalam terhadap permasalahan agar dapat memberikan gambaran yang cukup jelas terhadap permasalahan itu sendiri sehingga pemilihan metode atau algoritma yang akan digunakan relevan dengan topik permasalahan yang diangkat.

2.2.2. Proses *Data Mining*

Dalam CRISP-DM, sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase (Gambar 2.1). keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase *modeling*. Berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation* [KUE-09:8].



Gambar 2.1 Proses *data mining*

2.2.3. Fungsionalitas *Data Mining*

Data mining dapat dikelompokkan menjadi beberapa kelompok sesuai dengan tugas yang dilakukan, yaitu [KUE-09:10]:

a. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpul suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik daripada kearah kategori. Model dengan menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

c. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

Contoh prediksi dalam bisnis dan penelitian diantaranya adalah prediksi harga beras dalam tiga bulan yang akan datang, prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

d. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

Contoh lain dari klasifikasi dalam bisnis dan penelitian adalah:

- Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
- Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
- Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

e. Pengklasteran

Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek – objek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* – *record* dalam klaster lain.

Contoh pengklasteran dalam bisnis dan penelitian adalah:

- Mendapatkan kelompok – kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.
- Melakukan pengklasteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

f. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisa keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian diantaranya adalah:

- Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.
- Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.3. Clustering

2.3.1. Pengertian Clustering

Dalam bukunya yang berjudul *An Introduction to Data Mining*, Dr. Daniel T. Larose (2005) mendefinisikan *clustering* sebagai upaya mengelompokkan *record*, observasi, atau mengelompokkan kedalam kelas yang memiliki kesamaan objek [DAT-05:147].

Pengklasteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklasteran. Pengklasteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok – kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam suatu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Prinsip dasar untuk mendapatkan homogen atau heterogen dapat menggunakan konsep ukuran jarak. Jarak yang dimaksud bisa berarti ukuran jarak kedekatan atau kemiripan (*similarity measures*), bisa juga jarak yang berjauhan atau ketidakmiripan (*disimilarity measures*).

2.3.2. Tipe Clustering

Metode pengelompokan pada dasarnya ada dua, yaitu *Hierarchical Clustering Method* dan *Non Hierarchical Clustering Method*. *Hierarchical clustering method* digunakan apabila belum ada informasi jumlah *cluster*. Sedangkan *non hierarchical clustering method* bertujuan untuk mengelompokkan n objek kedalam k *cluster* ($k < n$) [BAD-05].

Salah satu contoh *hierarchical clustering method* atau bisa juga disebut dengan algoritma *clustering* tidak terawasi adalah algoritma *subtractive clustering*. Dalam *subtractive clustering* tiap isi data akan dievaluasi untuk mengetahui densitasnya, ketika terdapat titik data yang memiliki densitas tinggi maka dimungkinkan titik data tersebut menjadi pusat *cluster*. Sehingga tidak dapat diketahui berapa nantinya jumlah *cluster* yang akan terbentuk karena bergantung pada data yang akan di *cluster*.

2.4. Subtractive Clustering

2.4.1. Pengertian Subtractive Clustering

Subtractive clustering merupakan algoritma *clustering* tidak terawasi yang dapat membentuk jumlah dan pusat kluster sesuai dengan kondisi data. *Subtractive clustering* didasarkan atas ukuran densitas (potensi) titik – titik data dalam suatu ruang (variabel). Konsep dasar dari *subtractive clustering* adalah menentukan daerah – daerah dalam suatu variabel yang memiliki densitas tinggi terhadap titik – titik di sekitarnya. Titik dengan jumlah tetangga terbanyak akan dipilih sebagai pusat kluster. Titik yang dimaksud disini merupakan analogi untuk *record* dari data. Sehingga dengan kata lain, pusat *cluster* adalah salah satu *record* data yang termasuk dalam bagian sebuah data yang akan dikluster dan memiliki kedekatan dengan *record* data sekitarnya. Titik yang sudah terpilih sebagai pusat kluster ini kemudian akan dikurangi densitasnya. Kemudian algoritma akan memilih titik lain yang memiliki tetangga terbanyak untuk dijadikan pusat kluster yang lain. Hal ini akan dilakukan berulang – ulang hingga semua titik diuji [SKH-10:97].

Apabila terdapat N buah data: X_1, X_2, \dots, X_n dan dengan menganggap bahwa data – data tersebut sudah dalam keadaan normal, maka densitas titik X_k dapat dihitung dengan menggunakan Persamaan (2-1).

$$D_k = \sum_{j=1}^N \left(- \frac{\|X_k - X_j\|}{(r/2)^2} \right) \quad (2-1)$$

dengan $\|X_k - X_j\|$ adalah jarak antara X_k dengan X_j dan r adalah konstanta positif yang kemudian akan dikenal dengan nama jari – jari. Jari – jari berupa vektor yang akan menentukan seberapa besar pengaruh pusat *cluster* pada tiap – tiap variabel. Dengan demikian, suatu titik data akan memiliki densitas yang besar jika dia memiliki banyak tetangga dekat.

Setelah menghitung densitas tiap – tiap titik, maka titik dengan densitas tertinggi akan dipilih sebagai pusat *cluster*. Misalkan X_{cl} adalah titik yang terpilih sebagai pusat *cluster*, sedangkan D_{cl} adalah ukuran densitasnya. Selanjutnya densitas dari titik – titik disekitarnya akan dikurangi seperti pada Persamaan (2-2).

$$D'_k = D_k - D_{c1} * \exp\left(-\frac{\|X_k - X_{c1}\|}{(r_b/2)^2}\right) \quad (2-2)$$

dengan r_b adalah konstanta positif. Hal ini berarti bahwa titik – titik yang berada dekat dengan pusat *cluster* u_{c1} akan mengalami pengurangan densitas besar – besaran. Hal ini akan berakibat titik tersebut akan sangat sulit untuk menjadi pusat *cluster* berikutnya. Nilai r_b menunjukkan suatu lingkungan yang mengakibatkan titik – titik berkurang ukuran densitasnya. Biasanya r_b bernilai lebih besar dibandingkan dengan r , $r_b = q * r$ (biasanya *squash_factor* (q) = 1,25).

Setelah densitas tiap – tiap titik diperbaiki, maka selanjutnya akan dicari pusat *cluster* yang kedua yaitu X_{c2} . Sesudah X_{c2} didapat, ukuran densitas setiap titik data akan diperbaiki kembali, demikian seterusnya.

Pada implementasinya, bisa digunakan 2 pecahan sebagai faktor pembandingan, yaitu *accept ratio* dan *reject ratio*. Baik *accept ratio* dan *reject ratio* keduanya merupakan suatu bilangan pecahan yang bernilai 0 sampai 1. *Accept ratio* merupakan batas bawah di mana suatu titik data yang menjadi kandidat (calon) pusat *cluster* diperbolehkan untuk menjadi pusat *cluster*. Sedangkan *reject ratio* merupakan batas atas di mana suatu titik data yang menjadi kandidat (calon) pusat *cluster* tidak diperbolehkan untuk menjadi pusat *cluster*. Pada suatu iterasi, apabila telah ditemukan suatu titik data dengan potensi tertinggi (misal: X_k dengan potensi D_k), kemudian akan dilanjutkan dengan mencari rasio potensi titik data tersebut dengan potensi tertinggi suatu titik data pada awal iterasi (misal: X_h dengan potensi D_h). Hasil bagi antara D_k dengan D_h ini kemudian disebut dengan rasio ($rasio = D_k/D_h$). Ada 3 kondisi yang bisa terjadi dalam suatu iterasi:

- a. Apabila $rasio > Accept\ ratio$, maka titik data tersebut diterima sebagai pusat *cluster* baru.
- b. Apabila $Reject\ rasio < rasio \leq Accept\ ratio$ maka titik data tersebut baru akan diterima sebagai pusat *cluster* baru hanya jika titik data tersebut terletak pada jarak yang cukup jauh dengan pusat *cluster* yang lainnya (hasil penjumlahan antara rasio dan jarak terdekat titik data tersebut dengan pusat *cluster* lainnya yang telah ada ≥ 1). Apabila hasil penjumlahan antara rasio dan jarak terpanjang titik data tersebut dengan pusat *cluster* lainnya yang telah ada < 1 ,

maka selain titik data tersebut tidak akan diterima sebagai pusat *cluster*, dia sudah tidak akan dipertimbangkan lagi untuk menjadi pusat *cluster* baru (potensinya diset sama dengan nol).

- c. Apabila rasio \leq *Reject ratio*, maka sudah tidak ada lagi titik data yang akan dipertimbangkan untuk menjadi kandidat pusat *cluster*, iterasi dihentikan.

2.4.2. Algoritma Subtractive Clustering

Algoritma *subtractive clustering*[SKH-10:100] :

1. Menentukan Matriks X yang merupakan data yang akan dicluster, berukuran $i \times j$, dengan i = jumlah data yang akan di-cluster dan j = jumlah variabel/atribut (kriteria)
2. Menentukan :
 - 2.1. r_j (jari-jari setiap atribut data); $j=1,2,\dots,m$.
 - 2.2. q (squash factor)
 - 2.3. Accept ratio
 - 2.4. Reject ratio
 - 2.5. $Xmin_j$ (min data yang diperbolehkan dalam setiap atribut data) ; $j=1,2,\dots,m$.
 - 2.6. $Xmax_j$ (max data yang diperbolehkan dalam setiap atribut data) ; $j=1,2,\dots,m$.

3. Normalisasi

$$x_{ij} = \frac{x_{ij} - xMin_j}{xMax_j - xMin_j} \quad ; i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (2-3)$$

4. Menentukan potensi awal tiap – tiap titik data :

- 4.1. $i = n$ dan kerjakan hingga $i = 1$,

- 4.1.1. $T_j = X_{ij} \quad j=1,2,\dots,m \quad (2-4)$

- 4.1.2. Hitung :

$$Dist_{kj} = \left[\frac{T_j - X_{kj}}{r} \right] \quad ; j=1,2..m ; k=1,2..n \quad (2-5)$$

- 4.1.3. Potensi awal

Jika $m = 1$, maka

$$D_i = \sum_{k=1}^n e^{-4 \left(Dist_{kl}^2 \right)} \quad (2-6)$$



Jika $m > 1$, maka

$$D_i = \sum_{k=1}^n e^{-4 \left(\sum_{j=1}^m Dist_{kl}^2 \right)} \quad (2-7)$$

4.1.4. $i=i+1$

5. Mencari titik dengan potensi tertinggi

5.1. $M = \max[D_i | i=1,2,..n]$

5.2. $h = i$, sedemikian hingga $D_i = M$

6. Tentukan pusat *cluster* dan kurangi potensinya terhadap titik – titik di sekitarnya

6.1. $Center = []$, $Center$ merupakan pusat cluster

6.2. $V_j = Xh_j; j=1,2,..,m$. V_j merupakan nilai normalisasi pada data dengan potensi tertinggi.

6.3. $C = 0$, C merupakan jumlah cluster.

6.4. $Kondisi = 1$

6.5. $Z = M$, Z merupakan potensi titik yang diacu sebagai pusat cluster.

6.6. Kerjakan selama $(Kondisi \neq 0) \& (Z \neq 0)$

6.6.1 $Kondisi = 0$ (sudah tidak ada calon pusat baru lagi).

6.6.2 $Rasio = Z/M$

6.6.3 Jika $Rasio > accept\ ratio$, maka $kondisi = 1$ Hal ini menandakan ada calon pusat baru.

6.6.4 Jika tidak

a. Jika $rasio > reject\ ratio$, (calon baru akan diterima sebagai pusat jika keberadaannya akan memberikan keseimbangan terhadap data – data yang letaknya cukup jauh dengan pusat *cluster* yang telah ada), maka kerjakan :

b. $Md = -1$

c. Kerjakan untuk $i=1$ sampai $i=C$:

$$i. \quad G_{ij} = \frac{v_j - Center_{ij}}{r}, j = 1, 2, \dots, m \quad (2-8)$$



$$\text{ii. } Sd_i = \sum_{j=1}^m (G_{ij})^2 \quad (2-9)$$

iii. Jika ($Md < 0$) atau ($Sd < Md$), maka $Md = Sd$;

$$- Smd = \sqrt{Md} \quad (2-10)$$

- Jika ($Rasio + Smd$) ≥ 1 , maka kondisi = 1 Hal ini berarti data diterima sebagai pusat *cluster*.

- Jika ($Rasio + Smd$) < 1 , maka kondisi = 2 Hal ini berarti data tidak akan dipertimbangkan kembali sebagai pusat *cluster*.

6.6.5 Jika kondisi = 1, maka kerjakan:

a. $C = C + 1$

b. $Center_c = V$

c. Kurangi potensi dari titik-titik di dekat pusat cluster :

$$\text{i. } S_{ij} = \frac{V_j - X_{ij}}{r_j * q}, j = 1, 2, \dots, m; i = 1, 2, \dots, n \quad (2-11)$$

$$\text{ii. } D_{ei} = M * e^{-4 \left(\sum_{j=1}^m (s_{ij})^2 \right)}, i = 1, 2, \dots, n \quad (2-12)$$

$$\text{iii. } D = D - D_c \quad (2-13)$$

iv. Jika $D_i \leq 0$, maka $D_i = 0, i = 1, 2, \dots, n$

$$\text{v. } Z = \max [D_i | i = 1, 2, \dots, n]$$

vi. Pilih $h = i$, sedemikian hingga $D_i = Z$

6.6.6 Jika kondisi = 2, maka :

a. $D_h = 0$ dan $Z = \max [D_i | i = 1, 2, \dots, n]$

b. Pilih $h = i$, sedemikian hingga $D_i = Z$

7. Kembalikan pusat *cluster* dari bentuk ternormalisasi ke bentuk semula

$$Center_{ij} = Center_{ij} * (XMax_j - XMin_j) + XMin_j \quad (2-14)$$

8. Hitung nilai sigma *cluster*

$$\sigma_j = \frac{r_j * (XMax_j - XMin_j)}{\sqrt{8}} \quad (2-15)$$



2.5. Analisa Cluster

Cluster atau ‘klaster’ dapat diartikan ‘kelompok’, dengan demikian, pada dasarnya analisa klaster akan menghasilkan sejumlah klaster (kelompok). Analisis ini diawali dengan pemahaman bahwa sejumlah data tertentu sebenarnya mempunyai kemiripan di antara anggotanya; karena itu, dimungkinkan untuk mengelompokkan anggota – anggota yang ‘mirip’ atau mempunyai karakteristik yang serupa tersebut dalam satu atau lebih dari satu klaster[SAS-10:111].

Seperti diketahui, analisis klaster akan membagi sejumlah data satu atau beberapa klaster tertentu. Pertanyaan yang kemudian timbul adalah ‘apa yang menjadi batas bahwa sejumlah data dapat disebut sebagai satu klaster?’ secara logika, sebuah klaster yang baik adalah klaster yang mempunyai[SAS-10:113]:

- Homogenitas (kesamaan) yang tinggi antara anggota dalam satu klaster (*within cluster*).
- Heterogenitas (perbedaan) yang tinggi antara klaster yang satu dengan klaster yang lainnya (*between cluster*).

Dari dua hal di atas dapat disimpulkan bahwa klaster yang baik adalah klaster yang mempunyai anggota – anggota yang semirip mungkin satu dengan yang lain, namun sangat tidak mirip dengan anggota – anggota klaster yang lain.

2.5.1. Analisa Varian

Karena ciri dari klaster yang baik memiliki homogenitas yang tinggi antara anggota dalam satu klaster dan heterogenitas yang tinggi antara klaster yang satu dengan klaster yang lainnya, maka bisa dilakukan pendekatan untuk mengetahui baik tidaknya suatu klaster berdasarkan nilai varian.

Varian merupakan nilai penyebaran dari data. Sehingga nilai varian dapat digunakan untuk mengetahui baik atau tidaknya suatu klaster. Varian dalam *clustering* menurut Dr. Daniel T. Larose (2005) ada dua, yaitu varian dalam klaster (*variance within cluster*) dan varian antar klaster (*variance between cluster*)[DAT-05:149].

Cluster yang ideal dapat dilihat dari nilai varian yang kecil. Berikut persamaan untuk *variance within cluster*, *variance cluster*, dan *variance between*

cluster seperti pada Persamaan (2-16), Persamaan (2-17), Persamaan (2-18), dan Persamaan (2-19)[BEI-11]:

$$V_w = \frac{1}{N-k} \sum_{i=1}^k (n_i - 1) \cdot V_i^2 \quad (2-16)$$

dimana: V_w = variance within cluster

N = jumlah semua data,

k = jumlah cluster

n_i = jumlah data pada cluster ke- i

V_i^2 = varian pada cluster ke- i

Untuk mencari V_i^2 digunakan persamaan berikut:

$$V_c^2 = \frac{1}{n_c - 1} \sum_{i=1}^{n_c} (d_i - \bar{d}_i)^2 \quad (2-17)$$

dimana: $V_c^2 = V_i^2$ atau varaian pada cluster

$c = 1 \dots k$, dimana k = jumlah cluster

n_c = jumlah data pada cluster c

d_i = data ke- i pada suatu cluster

\bar{d}_i = rata-rata dari data pada suatu cluster

$$V_b = \frac{1}{k-1} \sum_{i=1}^k n_i (d_i - \bar{d}_i)^2 \quad (2-18)$$

Sehingga untuk menilai cluster dikatakan baik atau tidak bisa dilihat dari kepadatan sebaran datanya menggunakan persamaan berikut:

$$V = \frac{V_w}{V_b} \quad (2-19)$$

2.6. Logika Fuzzy

2.6.1 Pengertian Logika Fuzzy

Konsep logika fuzzy dicetuskan oleh Lotfi Zadeh, seorang profesor University of California di Berkeley, dan dipresentasikan bukan sebagai metodologi kontrol, namun sebagai suatu cara pemrosesan data yang memperbolehkan anggota himpunan parsial daripada anggota himpunan kosong atau non-anggota. Pendekatan ini pada teori himpunan tidak diaplikasikan untuk mengontrol sistem sampai tahun 70-an karena kurangnya kemampuan komputer-mini pada saat itu. Profesor Zadeh beralasan bahwa masyarakat tidak butuh

ketepatan, input informasi numeris, dan mereka belum sanggup dengan kontrol adaptif yang tinggi. Jika kembalian dari kontroler dapat diprogram untuk menerima *noisy*, input yang tidak teliti, mereka akan lebih efektif dan lebih mudah diimplementasikan[KUS-08:136].

Namun sekarang logika *fuzzy* telah menyentuh keberbagai pengembangan algoritma. Konsep dasar dari logika *fuzzy* yang berupa perluasan dari algoritma *boolean* (1 atau 0/ya atau tidak) dimana himpunan *fuzzy* berisi derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting dan membuat logika *fuzzy* bisa ditanam dalam algoritma lainnya.

Dalam banyak hal, logika *fuzzy* digunakan sebagai suatu cara untuk memetakan permasalahan dari *input* menuju ke *output* yang diharapkan. Selain itu ada beberapa alasan yang membuat logika *fuzzy* banyak digunakan, diantaranya adalah logika *fuzzy* mudah dimengerti, sangat fleksibel, memiliki toleransi terhadap data yang tidak tepat, mampu memodelkan fungsi – fungsi nonlinear yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman pakar tanpa ada pelatihan sebelumnya, dapat bekerjasama dengan teknik – teknik kendali secara konvensional, dan logika *fuzzy* didasarkan pada bahasa alami[SKH-10:1].

2.6.2 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu 0 atau 1[SKH-10:3]. Logika klasik tersebut memiliki kekurangan dari segi keadilan dalam memasukkan suatu nilai dalam keanggotaan berdasarkan range yang telah ditentukan sebelumnya. Misalnya, ditentukan range untuk usia dimana umur 0 sampai 35 tahun dikategorikan muda dan 35 sampai 55 tahun dikategorikan parobaya. Apabila seseorang berusia 15 tahun, maka dia dikatakan muda. Lalu, bagaimana jika ada orang yang berusia 35 tahun kurang 1 hari, maka dia dikatakan tetap muda. Padahal usia orang tersebut mendekati kategori parobaya. Dari sini bisa dikatakan bahwa pemakaian himpunan *crisp* untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan[SKH-4].

Berbeda dengan himpunan *crisp*, logika *fuzzy* mengelompokkan himpunan *fuzzy* dari semesta U untuk dikelompokkan oleh fungsi keanggotaan yang berada pada nilai antara $[0,1]$. Fungsi keanggotaan dari himpunan *fuzzy* merupakan kontinu dengan range 0 sampai 1 [KUS-08:136].

Logika *fuzzy* digunakan untuk mengantisipasi ketidakadilan dari himpunan *crisp*. Lewat logika *fuzzy*, satu nilai dapat masuk dalam dua himpunan yang berbeda tergantung dari seberapa besar derajat keanggotaan nilai tersebut terhadap himpunan satu dengan lainnya. Derajat keanggotaan pada himpunan *fuzzy* dapat dihitung menggunakan fungsi keanggotaan.

Himpunan *fuzzy* memiliki dua atribut, yaitu:

1. Linguistik, yaitu penamaan grup yang mewakili suatu keadaan dan kondisi tertentu dengan menggunakan bahasa alami, seperti: muda, parobaya, tua.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

2.6.3 Fungsi Keanggotaan

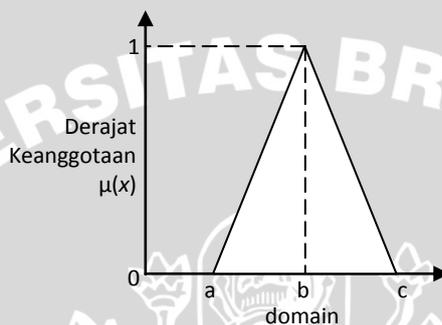
Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaan yang memiliki nilai interval antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan [SKH-10:8]. Diantaranya adalah representasi linear, Representasi Kurva Segitiga, Representasi Kurva Trapesium, Representasi Kurva Bentuk Bahu, Representasi Kurva-S, dan Representasi Kurva Bentuk Lonceng (*Bell Curve*) yang terbagi lagi menjadi Kurva PI, Kurva Beta, dan Kurva Gauss.

Menurut Jang dan Mirzutani (1997) menyebutkan bahwa fungsi fungsi keanggotaan *fuzzy* terparameterisasi satu dimensi yang umum digunakan diantaranya adalah [JAM-97].

1. Fungsi keanggotaan segitiga, disifati oleh parameter $\{a,b,c\}$ yang didefinisikan seperti pada Persmaan (2-20).

$$\text{segitiga}(x;a,b,c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (2-20)$$

Parameter $\{a,b,c\}$ (dengan $a < b < c$) yang menentukan koordinat x dari ketiga sudut segitiga tersebut, seperti terlihat pada gambar berikut:

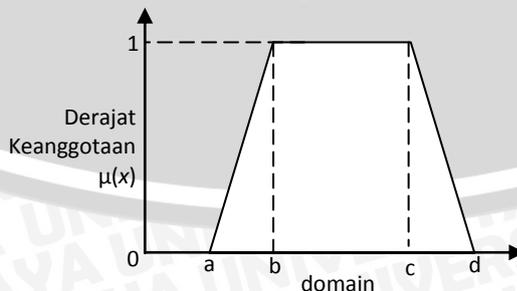


Gambar 2.2 Kurva segitiga

2. Fungsi keanggotaan trapesium, disifati oleh parameter $\{a,b,c,d\}$ yang didefinisikan pada Persmaan (2-21).

$$\text{trapesium}(x;a,b,c,d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (2-21)$$

Parameter $\{a,b,c,d\}$ pada kurva trapesium adalah sebagai berikut:



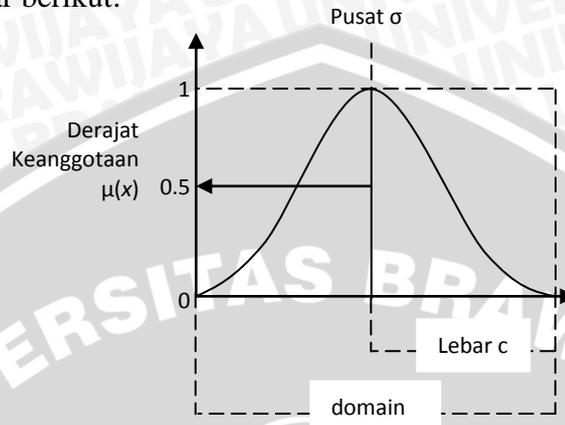
Gambar 2.3 Kurva trapesium

3. Fungsi keanggotaan gaussian, disifati oleh parameter $\{c,\sigma\}$ yang didefinisikan pada Persamaan (2-22).



$$gaussian(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (2-22)$$

Karakteristik fungsi keanggotaan *gauss* ditentukan oleh parameter c dan σ seperti terlihat pada gambar berikut:

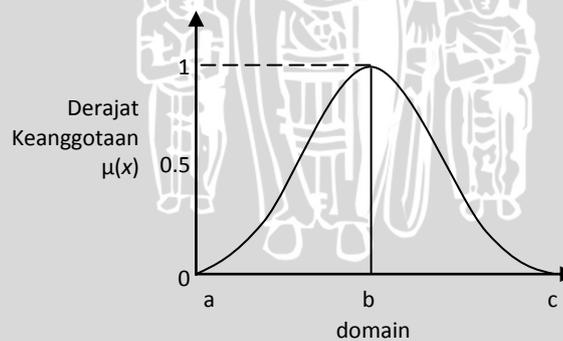


Gambar 2.4 Karakteristik fungsi *gauss*

4. Fungsi keanggotaan *generalized bell*, disifati oleh parameter $\{a, b, c\}$ yang didefinisikan seperti pada Persamaan (2-23).

$$bell(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2-23)$$

Parameter b selalu positif, supaya kurva menghadap kebawah, seperti terlihat pada gambar berikut:



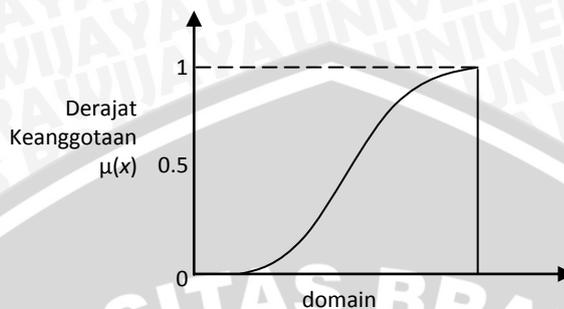
Gambar 2.5 Karakteristik fungsi *generalized bell*

5. Fungsi keanggotaan *sigmoid*, disifati oleh parameter $\{a, c\}$ yang didefinisikan seperti pada Persamaan (2-24).

$$sig(x; a, c) = \frac{1}{1 + \exp[-a(x-c)]} \quad (2-24)$$



Parameter a digunakan untuk menentukan kemiringan kurva pada saat $x = c$. Polaritas dari a akan menentukan kurva itu kanan atau kiri terbuka, seperti terlihat pada gambar berikut:



Gambar 2.6 Kurva *sigmoid*

2.6.4 Operator Dasar Zadeh

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu [SKH-10:23]:

a. Operator *AND*

Operator ini berhubungan dengan operasi interseksi pada himpunan α -predikat sebagai hasil operasi dengan operator *AND* diperoleh dengan mengambil nilai keanggotaan terkecil antarelemen pada himpunan – himpunan yang bersangkutan.

b. Operator *OR*

Operator ini berhubungan dengan operasi *union* pada himpunan α -predikat sebagai hasil operasi dengan operator *OR* diperoleh dengan mengambil nilai keanggotaan terbesar antarelemen pada himpunan – himpunan yang bersangkutan.

c. Operator *NOT*

Operator ini berhubungan dengan operasi komplemen pada himpunan α -predikat sebagai hasil operasi dengan operator *NOT* diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

2.6.5 Fungsi Implikasi

Kaidah *fuzzy if-then* (dikenal juga sebagai kaidah *fuzzy*, implikasi *fuzzy* atau pernyataan kondisi *fuzzy*) diasumsikan seperti pada Persmaan (2-25).

$$\text{Jika } x \text{ adalah } A \text{ maka } y \text{ adalah } B \tag{2-25}$$

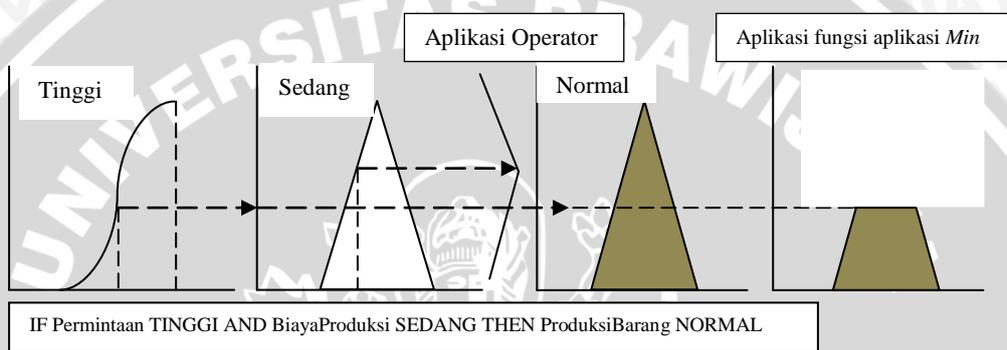


Dengan A dan B adalah nilai linguistik yang dinyatakan dengan himpunan fuzzy dalam semesta pembicaraan X dan Y . Seringkali “ x adalah A ” disebut sebagai *antecedent* atau *premise*, sedangkan “ y adalah B ” disebut *consequence* atau *conclusion*[JAM-97].

Secara umum, ada 2 fungsi implikasi yang dapat digunakan, yaitu[SKH-28]:

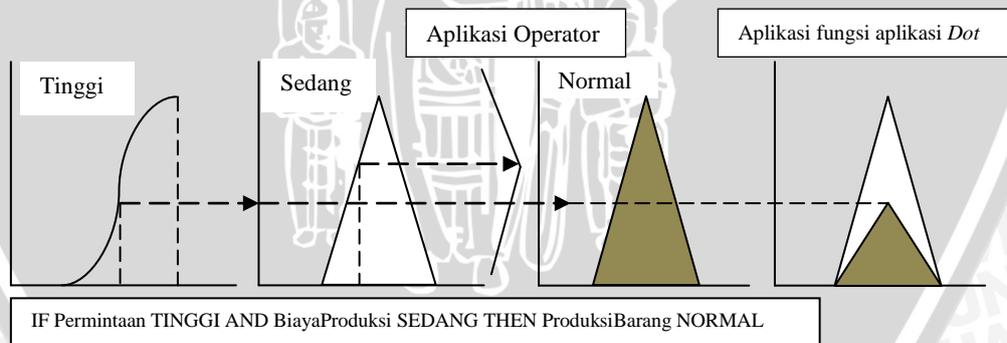
- a. *Min (minimum)*, fungsi ini akan memotong *output* himpunan fuzzy.

Gambar 2.7 menunjukkan salah satu contoh penggunaan fungsi *min*.



Gambar 2.7 Fungsi implikasi *MIN*

- b. *Dot (product)*, fungsi ini akan menskala *output* himpunan fuzzy. Gambar 2.8 menunjukkan salah satu contoh penggunaan fungsi *dot*.



Gambar 2.8 Fungsi implikasi *DOT*

2.6.6 Fuzzy Inference System

Dalam buku Sri Kusumadewi dan Hari Purnomo (2010) yang berjudul “Aplikasi Logika Fuzzy untuk Pendukung Keputusan” dijelaskan terdapat tiga metode dalam sistem inferensi fuzzy (*fuzzy inference system*). Diantaranya adalah *fuzzy inference system* metode *tsukamoto*, *mamdani* dan *sugeno*. Metode *tsukamoto* merupakan perluasan dari penalaran monoton. Pada metode *tsukamoto*,

setiap konsekuensi pada aturan yang berbentuk *IF-THEN* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap – tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata – rata terbobot [SKH-10:31].

Metode *mamdani* sering dikenal sebagai metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan yaitu pembentukan himpunan *fuzzy*, aplikasi fungsi implikasi, komposisi aturan dan penegasan (*defuzzy*) [SKH-10:37].

Metode yang terakhir adalah metode *sugeno*. Pada penelitian ini digunakan metode *sugeno* orde-satu. Penalaran dengan metode *sugeno* hampir sama dengan penalaran *mamdani*, hanya saja *output* (konsekuensi) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK. Metode TSK terdiri atas 2 jenis, yaitu [SKH-10:46]:

a. Model *fuzzy* *sugeno* orde-nol

Secara umum bentuk model *fuzzy* *sugeno* orde-nol adalah:

$$\text{IF } (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_n \text{ is } A_n) \text{ THEN } z = k \quad (2-26)$$

dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden dan k adalah suatu konstanta (tegas) sebagai konsekuensi.

b. Model *fuzzy* *sugeno* orde-satu

Secara umum bentuk model *fuzzy* *sugeno* orde-satu adalah:

$$\text{IF } (x_1 \text{ is } A_1) \circ \dots \circ (x_n \text{ is } A_n) \text{ THEN } z = p_1 * x_1 + \dots + p_n * x_n + q \quad (2-27)$$

dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden dan p_i adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuensi.

2.6.7 Teknik Pembangkitan Aturan *Fuzzy*

Menurut Arapoglou, Kostas, dan Stathes (2010) mesin pembelajaran adalah bagian dari kecerdasan buatan yang membuat keputusan berdasarkan data. Algoritma yang dapat digunakan untuk membangkitkan aturan *fuzzy* secara otomatis berdasarkan data adalah dengan metode *clustering* [AKH-10:4].

Misalkan kita memiliki n buah data di mana setiap data memiliki p variabel (*input*), maka kita dapat menyusun data – data tersebut menjadi sebuah matriks X yang berukuran $n \times p$. Dengan menggunakan *subtractive clustering* dengan: jari – jari (r), *accept ratio*, *reject ratio*, dan *squash factor* tertentu, kita akan mendapatkan pusat *cluster* c dan sigma [SKH-10:151].

2.6.8 Ekstraksi Aturan Fuzzy

Untuk membentuk *fuzzy inference system* dari hasil *clustering* ini, kita dapat menggunakan metode inferensi *fuzzy* sugeno orde-satu. Sebelumnya, data yang ada dipisahkan terlebih dahulu antara data pada variabel – variabel *input* dengan data pada variabel *output*. Misalkan jumlah variabel *input* adalah m , dan variabel *output* biasanya 1. Pada metode ini, akan diperoleh kumpulan aturan yang berbentuk[SKH-10:151]:

$$[R1] \text{ IF } (x_1 \text{ is } A_{11}) \circ (x_2 \text{ is } A_{12}) \circ \dots \circ (x_n \text{ is } A_{1m}) \tag{2-28}$$

$$\text{THEN } (z = k_{11}x_1 + \dots + k_{1m}x_m + k_{10});$$

$$[R2] \text{ IF } (x_1 \text{ is } A_{21}) \circ (x_2 \text{ is } A_{22}) \circ \dots \circ (x_n \text{ is } A_{2m})$$

$$\text{THEN } (z = k_{21}x_1 + \dots + k_{2m}x_m + k_{20});$$

...

$$[Rr] \text{ IF } (x_1 \text{ is } A_{r1}) \circ (x_2 \text{ is } A_{r2}) \circ \dots \circ (x_n \text{ is } A_{rm})$$

$$\text{THEN } (z = k_{r1}x_1 + \dots + k_{rm}x_m + k_{r0});$$

dengan:

- A_{ij} adalah himpunan *fuzzy* aturan ke- i variabel ke- j sebagai anteseden,
- k_{ij} adalah koefisien persamaan *output fuzzy* aturan ke- i variabel ke- j ($i = 1, 2, \dots, r; j = 1, 2, \dots, m$), dan k_{i0} adalah konstanta persamaan *output fuzzy* aturan ke- i ,
- tanda \circ menunjukkan operator yang digunakan

Hasil dari *clustering* ini nantinya adalah pusat *cluster* (c) dan sigma (σ). Dari nilai c dan σ nantinya digunakan untuk mengetahui derajat keanggotaan setiap titik data pada matriks k dengan menggunakan fungsi *gauss* seperti pada Persmaan (2-29).

$$\mu_{ki} = e^{-\sum_{j=1}^m \frac{(x_{ij}-C_{kj})^2}{2\sigma_j^2}} \tag{2-29}$$



Matriks k itu sendiri merupakan matriks pusat *cluster* yang juga membentuk aturan sesuai dengan jumlah *cluster* yang terbentuk. Ukuran dari matriks k adalah $r + (m + 1)$ yang kemudian disusun menjadi vektor dengan ukuran $r * (m+1)$.

Kemudian derajat keanggotaan setiap data i dalam *cluster* k ini kita akan kalikan dengan setiap atribut j dari data i . Misalkan dinotasikan sebagai d_{ij}^k dengan Persamaan (2-30).

$$d_{ij}^k = X_{ij} * \mu_{ki} \text{ dan } d_{i(m+1)}^k = \mu_{ki} \tag{2-30}$$

Proses normalisasi dilakukan dengan cara membagi d_{ij}^k dan $d_{i(m+1)}^k$ dengan jumlah derajat keanggotaan setiap titik data i pada *cluster* k , diperoleh:

$$d_{ij}^k = \frac{d_{ij}^k}{\sum_{k=1}^r \mu_{ki}} \tag{2-31}$$

$$d_{i(m+1)}^k = \frac{d_{i(m+1)}^k}{\sum_{k=1}^r \mu_{ki}} \tag{2-32}$$

Langkah selanjutnya adalah membentuk matriks U yang berukuran $n \times (r * (m + 1))$ dengan:

$$u_{i1} = d_{i1}^1; \quad u_{i(2m+1)} = d_{i(m+1)}^2; \tag{2-33}$$

$$u_{i2} = d_{i2}^1; \quad u_{i(r*(m+1)-m)} = d_{i1}^r;$$

$$u_{im} = d_{im}^1; \quad u_{i(r*(m+1)-m+1)} = d_{i2}^r;$$

$$u_{i(m+1)} = d_{i(m+1)}^1; \quad u_{i(r*(m+1)-1)} = d_{im}^r;$$

$$u_{i(m+2)} = d_{i1}^2; \quad u_{i(r*(m+1))} = d_{i(m+1)}^r;$$

$$u_{i(m+3)} = d_{i2}^2; \quad \dots$$

$$u_{i(2m)} = d_{im}^2; \quad \text{dan seterusnya.}$$

Vektor z , merupakan vektor *output* berbentuk:

$$z = [z_1 \ z_2 \ \dots \ z_n]^T \tag{2-34}$$

Dari vektor k , matriks U , dan vektor z ini dapat dibentuk suatu sistem persamaan linier yang berbentuk:

$$U * k = z \tag{2-35}$$

untuk mencari nilai koefisien *output* tiap – tiap aturan pada setiap variabel ($k_{ij}, i=1,2,\dots,r$; dan $j=1,2,\dots,m+1$). Matriks U bukan matriks bujursangkar, sehingga untuk menyelesaikan persamaan ini digunakan metode kuadrat terkecil. Untuk membentuk anteseden, setiap variabel *input* juga akan terbagi menjadi r himpunan *fuzzy*, dengan setiap himpunan memiliki fungsi keanggotaan *gauss*, dengan derajat keanggotaan data X_i , variabel ke- j , himpunan ke- k dirumuskan dengan Persamaan (2-36).

$$\mu_{Var - j; Himp - k} [X_i] = e^{-\frac{(X_{ij} - C_{kj})^2}{2\sigma_j^2}} \tag{2-36}$$

Dengan aturan sebagai berikut:

$$[R1] : \text{IF } (X_{i1} \text{ is } V1H1) \circ (X_{i2} \text{ is } V2H1) \circ \dots \circ (X_{im} \text{ is } VmH1) \tag{2-37}$$

$$\text{THEN } Y = Z_1$$

$$[R2] : \text{IF } (X_{i1} \text{ is } V1H2) \circ (X_{i2} \text{ is } V2H2) \circ \dots \circ (X_{im} \text{ is } VmH2)$$

$$\text{THEN } Y = Z_2$$

$$[R3] : \text{IF } (X_{i1} \text{ is } V1H3) \circ (X_{i2} \text{ is } V2H3) \circ \dots \circ (X_{im} \text{ is } VmH3)$$

$$\text{THEN } Y = Z_3$$

...

$$[Rr] : \text{IF } (X_{i1} \text{ is } V1Hr) \circ (X_{i2} \text{ is } V2Hr) \circ \dots \circ (X_{im} \text{ is } VmHr)$$

$$\text{THEN } Y = Z_r$$

dengan $V_p H_q$ adalah variabel ke- p himpunan ke- q .

2.7. LSE

Menurut Jang, Chiu, dan Mizutani (1997), salah satu metode yang dapat menentukan metode kuadrat terkecil adalah dengan menggunakan metode *least square estimator*. Namun sebelumnya diperlukan identifikasi struktur dalam langkah ini. Tujuannya agar dapat menerapkan pengetahuan tentang target sistem



untuk dapat menentukan kelas yang paling cocok dari model yang dicari [JAM-97:95].

Jika parameter konsekuen k dinotasikan seperti pada Persamaan (2-38).

$$k^t = [k_0^t, k_1^t, k_2^t, \dots, k_n^t]^T = \begin{bmatrix} k_0^t \\ k_1^t \\ k_2^t \\ \dots \\ k_n^t \end{bmatrix}, \quad (2-38)$$

maka TS inferensi untuk 1 sampai n data latih dapat ditulis seperti pada Persamaan (2-39).

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_N \end{bmatrix} = \begin{bmatrix} U_1 & U_1 & \dots & U_1 \\ U_2 & U_2 & \dots & U_2 \\ \dots & \dots & \dots & \dots \\ U_n & U_n & \dots & U_n \end{bmatrix} \bullet \begin{bmatrix} k^1 \\ k^2 \\ \dots \\ k^m \end{bmatrix} \quad (2-39)$$

Dimana dimensi untuk masing – masing matriks tersebut adalah $[Y] = N \times 1$, $[U] = N \times ((n+1).m)$, $[k] = ((n+1).m) \times 1$

Untuk menghitung koefisien k dapat digunakan metode *least square estimator* (dengan menggunakan *pseudo matrix*), sehingga persamaan TS inferensi menjadi seperti pada Persamaan (2-40).

$$[U]^T \bullet [U][k] = [U]^T \bullet [Y] \quad (2-40)$$

sehingga nilai dapat ditunjukkan seperti pada Persamaan (2-41).

$$[k] = ([U]^T \bullet [U])^{-1} \bullet [U]^T \bullet [Y] \quad (2-41)$$

$$[k] = [k_0^t, k_1^t, k_2^t, \dots, k_n^t]^T \quad (2-42)$$

Dimensi dari matriks parameter k adalah

$[k] = (m \cdot (n+1) \times N) \cdot (N \times 1) = (m \cdot (n+1)) \times 1$, dimana m adalah jumlah aturan, N adalah jumlah data latih dan n adalah jumlah *fuzzy input* [JAM-97:106].



2.8. Akurasi

Salah satu cara untuk mengetahui hasil penelitian adalah melihat akurasi. Akurasi merupakan kedekatan suatu angka atau hasil pengujian terhadap angka ataupun data sebenarnya (*true value* atau *reference value*) [NUG-06]. Dalam penelitian ini perhitungan akurasi akan melibatkan hasil penelitian dan data nyata yang didapatkan dari sumber, berikut persamaan untuk perhitungan akurasi:

$$\text{TingkatAkurasi} = \frac{\sum \text{dataUjiBenar}}{\sum \text{totalDataUji}} \quad (2-43)$$

$$\text{Akurasi}(\%) = \frac{\sum \text{dataUjiBenar}}{\sum \text{totalDataUji}} \times 100\% \quad (2-44)$$

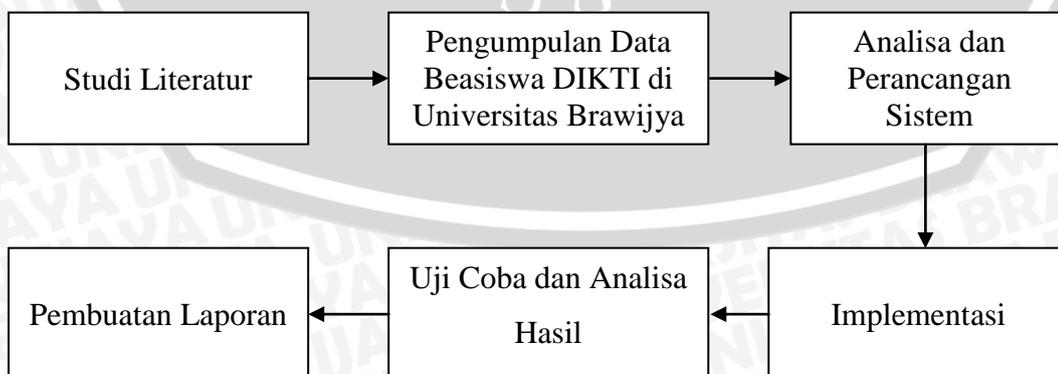


BAB III METODOLOGI DAN PERANCANGAN SISTEM

Dalam bab ini dijelaskan mengenai langkah – langkah pengerjaan penelitian untuk mendapatkan hasil inferensi *fuzzy* model sugeno orde-satu pada rekomendasi penerima beasiswa yang aturannya dibangkitkan menggunakan algoritma *subtractive clustering*.

Pengerjaan penelitian ini dilakukan dengan tahapan – tahapan sebagai berikut:

1. Mempelajari literatur yang berkaitan dengan pengerjaan penelitian, diantaranya mengenai beasiswa DIKTI, *data mining*, *fuzzy inference system*, teknik pembangkitan aturan *fuzzy*, dan *subtractive clustering*.
2. Mengumpulkan data pemohon beasiswa DIKTI 2012 di Universitas Brawijaya sesuai dengan parameter yang telah ditetapkan oleh DIKTI. Data set nantinya terdiri dari data latih dan data uji.
3. Menganalisa dan melakukan perancangan sistem dengan mengimplementasikan algoritma *subtractive clustering* sebagai metode pembentuk aturan pada sistem inferensi *fuzzy*. Kemudian hasil *subtractive clustering* yang diekstrak menjadi aturan *fuzzy* diterapkan pada *fuzzy inference system* model sugeno orde-satu.
4. Mengimplementasikan sistem yang telah dibangun.
5. Melakukan uji coba sistem serta pembahasan terhadap hasil penelitian.
6. Membuat laporan tertulis dengan format sesuai ketentuan.



Gambar 3.1 Langkah – langkah penelitian

3.1 Studi Literatur

Studi literatur adalah proses mempelajari dan memahami secara mendalam terkait teori – teori dan dasar keilmuan yang akan menjadi bahan untuk penelitian yang akan dilakukan. Teori tersebut dapat berasal dari buku, jurnal, *ebook*, penelitian sebelumnya, *browsing* dari internet, dan dari sumber pustaka lain yang terkait dengan penelitian serta dapat dipertanggungjawabkan.

Teori – teori yang akan dipelajari diantaranya adalah terkait beasiswa DIKTI di Universitas Brawijaya, *data mining*, *fuzzy inference system*, *subtractive clustering*, dan pembangkitan aturan *fuzzy*.

3.2 Data Penelitian

Data yang akan digunakan pada penelitian ini merupakan data beasiswa DIKTI jenis PPA (Pengembangan Potensi Akademik) di Universitas Brawijaya pada tahun 2012. Data tersebut berasal dari PPTI (Pusat Pengembangan Teknologi Informasi) Universitas Brawijaya. Atribut data yang diperlukan dalam penelitian ini terdiri dari nilai IPK, jumlah SKS, prestasi, keadaan ekonomi, dan nilai kelayakan. Untuk atribut keadaan ekonomi didapat dari pengurangan antara gaji orang tua dengan biaya listrik, PBB, biaya air, dan telepon.

Pada penelitian ini akan digunakan data latih sebanyak 60, 80, dan 100. Untuk data uji akan digunakan sebanyak 30 *record*. Data latih digunakan untuk bahan pembelajaran bagi algoritma *clustering* dalam membangkitkan aturan *fuzzy*. Sedangkan data uji merupakan data yang akan diuji coba pada sistem untuk mengetahui tingkat akurasi dari hasil penelitian.

3.3 Analisa dan Perancangan Sistem

3.3.1. Deskripsi Umum Sistem

Secara umum sistem yang akan dibangun merupakan sistem inferensi *fuzzy* model sugeno orde-satu untuk memberikan rekomendasi pada sistem penerimaan beasiswa. Aturan yang semula berasal dari pakar, kali ini dibangkitkan dengan teknik *clustering* menggunakan metode *subtractive clustering* sebagai media pembelajaran terhadap data latih. *Output* dari sistem ini merupakan nilai kelayakan dari pemohon beasiswa. Semakin besar nilai kelayakan pemohon beasiswa, maka semakin besar pula peluang untuk

permohonan beasiswanya diterima. Sistem yang akan dibangun hanya memberikan rekomendasi untuk penerimaan beasiswa berdasarkan nilai kelayakan, sehingga keputusan akhir tetap pada pemegang kebijakan pada bagian terkait untuk menentukan siapa – siapa penerima beasiswa.

Proses pertama dari sistem ini adalah mengklaster data latih menggunakan algoritma *subtractive clustering* dengan beberapa parameter yang ditentukan sebelumnya. Tujuan dari pengklasteran adalah sebagai media pembelajaran pengganti pakar untuk membangkitkan aturan *fuzzy*. Hasil dari *clustering* adalah pusat *cluster* dan sigma yang akan digunakan pada proses ekstraksi aturan *fuzzy*. Inisialisasi awal dari parameter untuk proses pengklasteran akan menentukan hasil *cluster* (jumlah *cluster*). Oleh karena itu uji coba akan dilakukan beberapa kali terhadap inisialisasi awal parameter pada algoritma *subtractive clustering* untuk melihat perubahan hasil *cluster*. Dengan inisialisasi berbeda pada parameter dimungkinkan akan menghasilkan jumlah *cluster* yang berbeda. Dengan variasi jumlah *cluster* inilah akan dihitung nilai varian sebagai proses analisa *cluster*. Jumlah *cluster* dengan varian terkecil yang akan diambil untuk dijadikan bahan pembangkitan aturan *fuzzy*.

Langkah berikutnya adalah mengekstraksi aturan *fuzzy*. Ekstraksi aturan *fuzzy* menggunakan pusat *cluster* dan sigma dari hasil *clustering*. Nantinya jumlah aturan sama dengan jumlah *cluster* yang terbentuk. Derajat keanggotaan data latih terhadap *cluster* dihitung menggunakan pusat *cluster* dan sigma. Derajat keanggotaan dan matriks U nantinya digunakan untuk mencari koefisien *output* menggunakan metode kuadrat terkecil. Metode kuadrat terkecil yang dipakai adalah LSE (*least square estimator*).

Aturan yang terbentuk diterapkan pada *fuzzy inference system* model sugeno orde-satu. Pemilihan model sugeno orde-satu dikarenakan konsekuennya berupa persamaan linier. Proses dari metode sugeno orde-satu dimulai dengan mencari derajat keanggotaan tiap parameter pada data uji menggunakan fungsi *gauss*. Kemudian menghitung *fire strength* dan nilai Z (proses *defuzzy*). Hasil akhir dari proses inferensi merupakan nilai kelayakan yang dijadikan bahan pertimbangan untuk diterima atau tidaknya permohonan beasiswa.

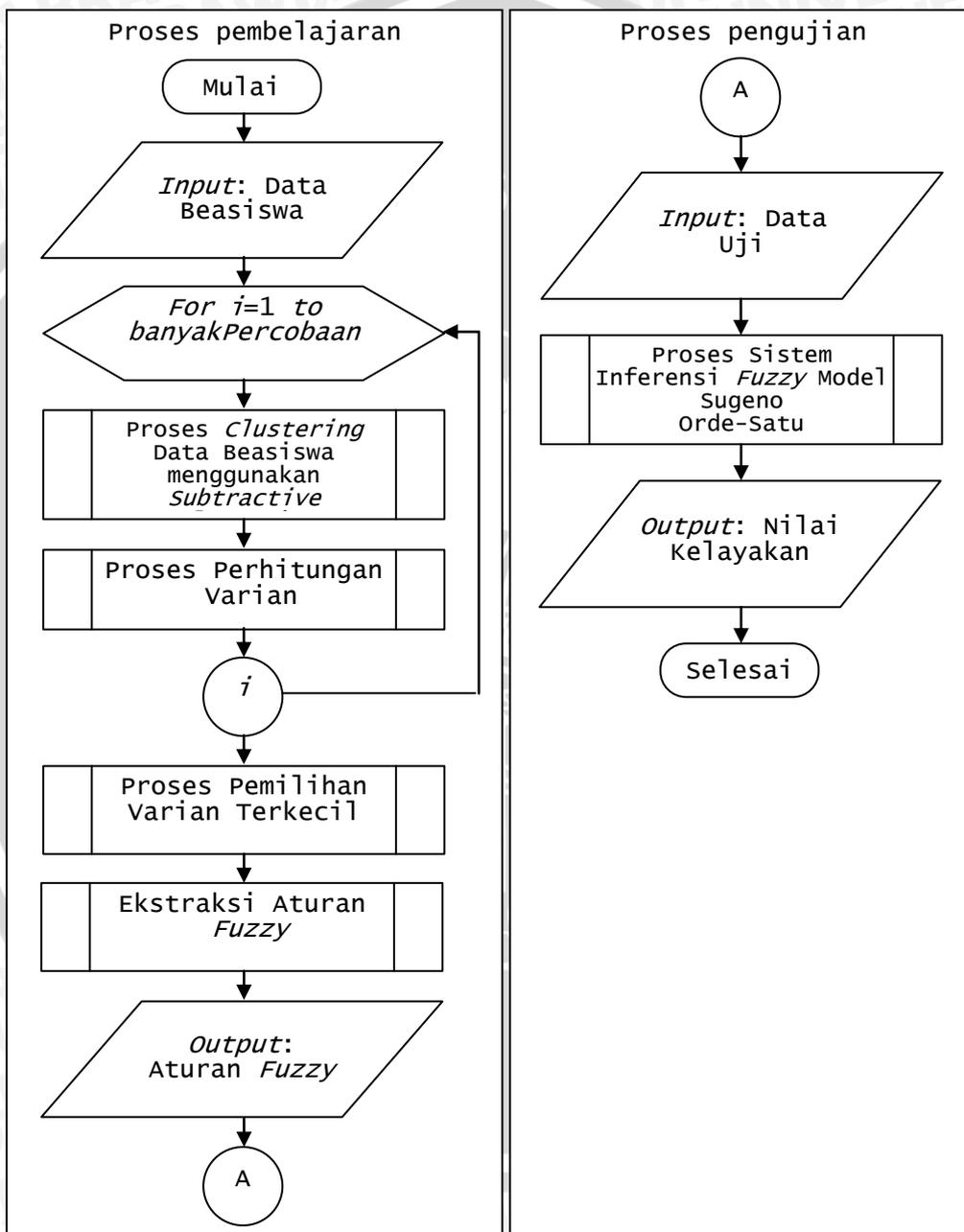
3.3.2. Perancangan Sistem

Tahapan pada perancangan sistem ini ialah sebagai berikut:

1. Input data beasiswa DIKTI.
2. Proses pembelajaran dari data beasiswa sebagai upaya menggantikan pakar untuk membangkitkan aturan *fuzzy* dengan menggunakan metode *subtractive clustering*. Jumlah *cluster* tidak diinisialisasi diawal, namun terbentuk dengan sendirinya berdasarkan densitas dari tiap – tiap titik data dan parameter awal yang ditentukan diawal proses. Inisialisasi awal digunakan pada parameter jari – jari, *squash factor*, *accept ratio*, dan *reject ratio*. Hasil dari algoritma *subtractive clustering* adalah pusat *cluster* dan sigma yang nantinya digunakan untuk menghitung derajat keanggotaan menggunakan fungsi *gauss*. Proses *clustering* ini akan dilakukan beberapa kali dengan inisialisasi parameter awal yang berbeda untuk melihat perubahan hasil *clustering*.
3. Proses analisa varian adalah upaya untuk menganalisa hasil *cluster* yang terbentuk pada proses sebelumnya. Proses analisa *cluster* dengan melihat nilai varian digunakan untuk mengetahui berapa jumlah *cluster* yang tepat guna diterapkan pada proses selanjutnya.
4. Hasil *cluster* yang akan dipilih untuk digunakan pada proses selanjutnya merupakan hasil *cluster* dengan nilai varian terkecil.
5. Proses pembangkitan aturan *fuzzy* bisa memanfaatkan jumlah *cluster* yang didapatkan pada proses sebelumnya. Jumlah aturan yang terbentuk sama dengan jumlah *cluster*.
6. Tahap selanjutnya adalah proses pengujian, dimana pada proses ini akan diinputkan data pemohon beasiswa untuk dicari nilai kelayakannya.
7. Proses penentuan nilai kelayakan dilakukan dengan sistem inferensi *fuzzy* model sugeno orde-satu. Aturan yang digunakan berasal dari pembangkitan aturan *fuzzy* pada proses sebelumnya. *Output* dari sistem inferensi *fuzzy* merupakan nilai *crisp* dari nilai kelayakan pemohon beasiswa.

8. Proses yang terakhir adalah mengurutkan nilai kelayakan pemohon beasiswa dari yang tertinggi hingga terendah untuk memberikan rekomendasi siapa yang layak mendapatkan beasiswa.

Alur proses dari perancangan sistem tersaji dalam bagan alir pada Gambar 3.2.

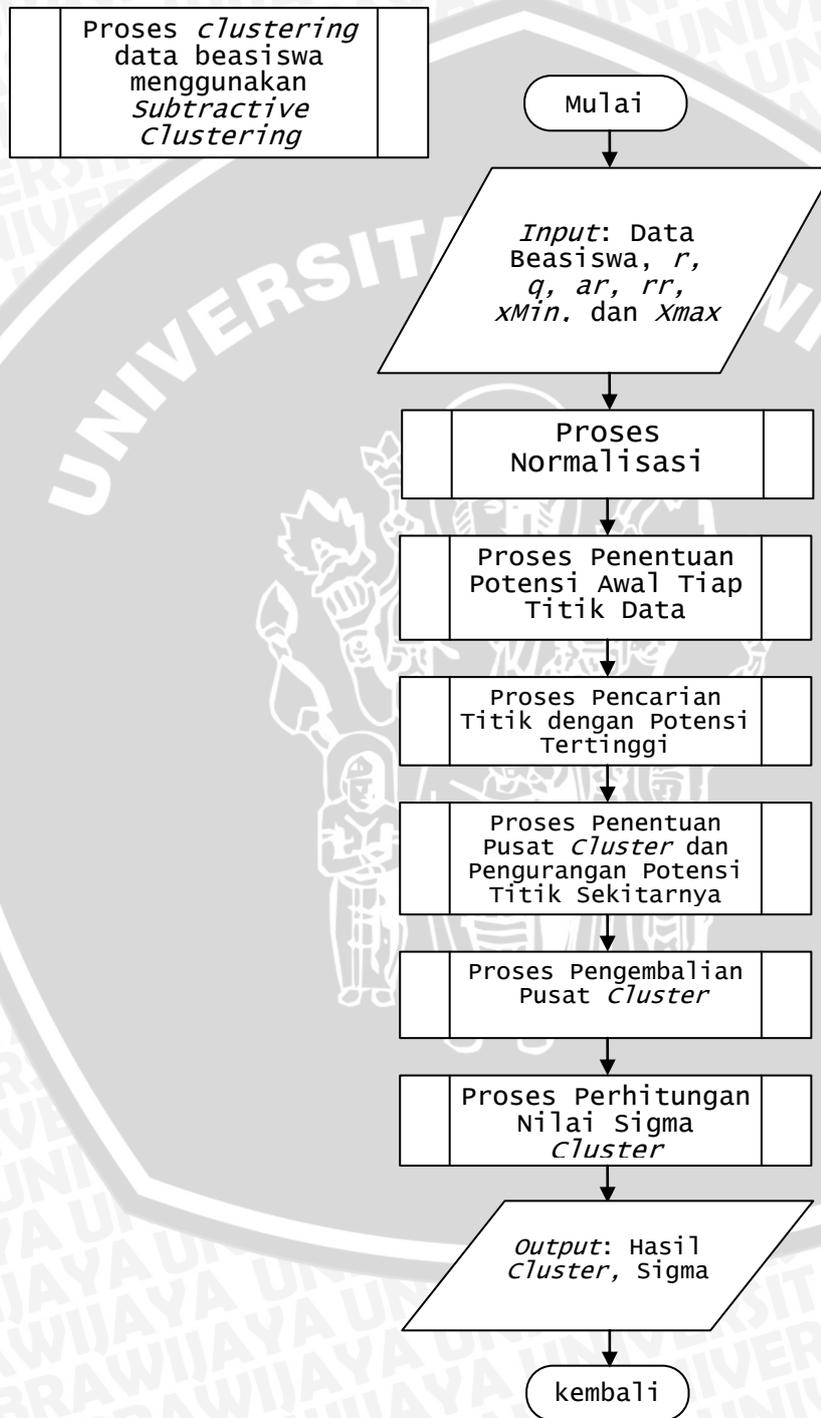


Gambar 3.2 Gambaran umum perancangan sistem



3.3.2.1 Proses *Clustering* dengan *Subtractive Clustering*

Proses pembelajaran terhadap data latih untuk membangkitkan aturan menggunakan algoritma *subtractive clustering*. Langkah – langkah dalam proses ini digambarkan oleh bagan alir pada Gambar 3.3.



Gambar 3.3 Tahapan proses *clustering*



Penjelasan langkah demi langkah proses *clustering* data latih menggunakan algoritma *subtractive clustering* seperti pada Gambar 3.3 adalah sebagai berikut:

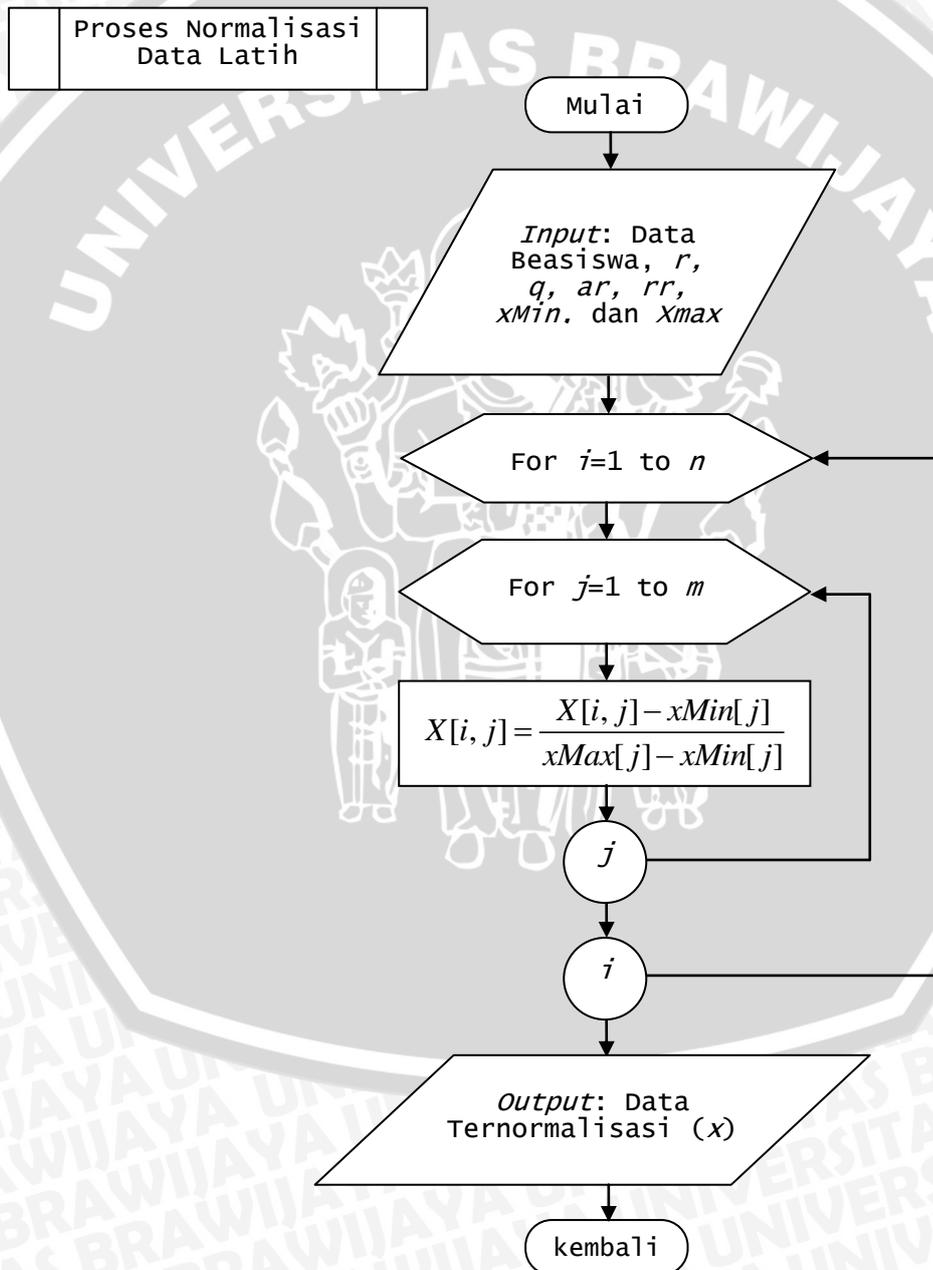
1. Memasukkan data beasiswa sebagai data latih yang akan di *clustering*.
2. Menentukan inisialisasi awal untuk variabel jari – jari setiap atribut data (r), nilai *squash factor* (q), nilai *accept ratio* (ar), nilai *reject ratio* (rr), nilai minimum pada setiap parameter data ($xMin$), dan nilai maksimum pada setiap parameter data ($xMax$).
3. Menghitung data ternormalisasi untuk data latih. Alur proses perhitungan normalisasi data terdapat pada Gambar 3.4.
4. Menentukan potensi awal untuk tiap titik data. Alur proses perhitungan nilai potensi awal untuk tiap titik data terdapat pada Gambar 3.5.
5. Mencari titik data dengan potensi tertinggi. Alur proses pencarian titik data dengan potensi tertinggi terdapat pada Gambar 3.6.
6. Menentukan pusat *cluster* dan mengurangi potensinya terhadap titik – titik data disekitarnya. Alur proses penentuan pusat *cluster* dan pengurangan potensinya terdapat pada Gambar 3.7.
7. Mengembalikan pusat *cluster* dari bentuk ternormalisasi ke bentuk semula. Alur proses pengembalian pusat *cluster* terdapat pada Gambar 3.10.
8. Menghitung nilai sigma *cluster*. Alur proses penghitungan nilai sigma *cluster* terdapat pada Gambar 3.11.

1) Proses Normalisasi

Proses normalisasi data dilakukan pertama kali pada proses *clustering*. Langkah ini dilakukan untuk menormalkan bentuk data untuk proses selanjutnya. Alur proses dari normalisasi data ditunjukkan pada Gambar 3.4. Langkah – langkah pada proses normalisasi data ialah sebagai berikut:

1. Memasukkan data latih (X) yang akan dinormalisasi, nilai $xMin$, dan $xMax$.

2. Inisialisasi awal untuk variabel m sebagai jumlah atribut data latih dan variabel n sebagai jumlah data latih yang menjadi masukan pada proses sebelumnya.
3. Iterasi dari $i = 1$ sampai n , dilakukan langkah berikut:
 - a. Iterasi dari $j = 1$ sampai m , dilakukan proses perhitungan nilai X_{ij} dengan Persamaan (2-3).
4. Hasil dari proses ini adalah data ternormalisasi (x).



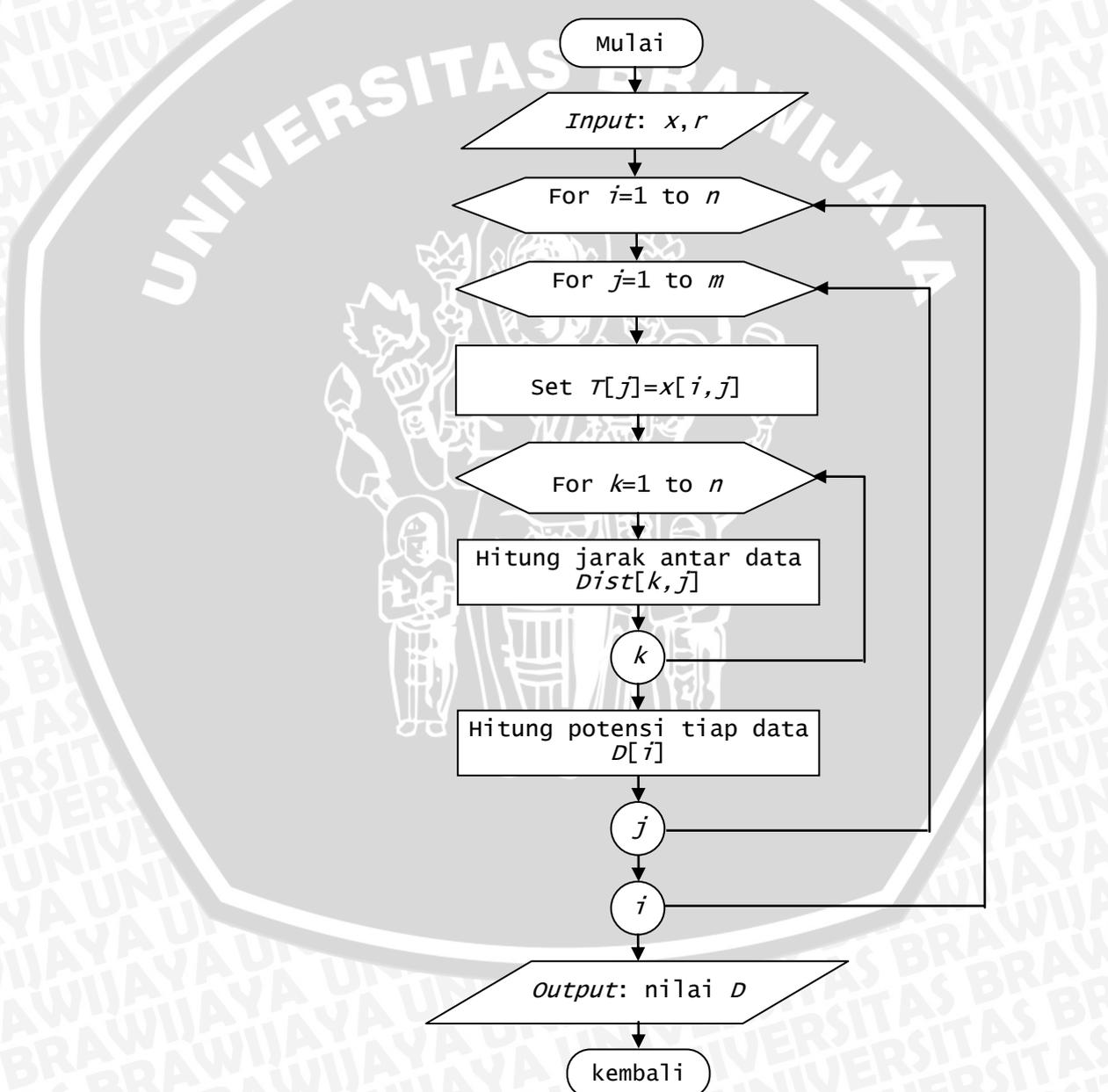
Gambar 3.4 Alur proses normalisasi data latih



2) Proses Penentuan Potensi Awal untuk Tiap Titik Data

Proses penentuan potensi awal dilakukan terhadap masing - masing data ternormalisasi. Tujuan dari proses ini adalah menilai titik data untuk menjadi kandidat pusat *cluster*. Alur dari proses penentuan potensi awal untuk tiap titik data ditunjukkan pada Gambar 3.5.

Proses Penentuan Potensi Awal untuk Tiap Titik Data



Gambar 3.5 Alur proses penentuan potensi awal tiap titik data



Penjelasan Gambar 3.5 untuk proses penentuan potensi awal tiap titik data ialah sebagai berikut:

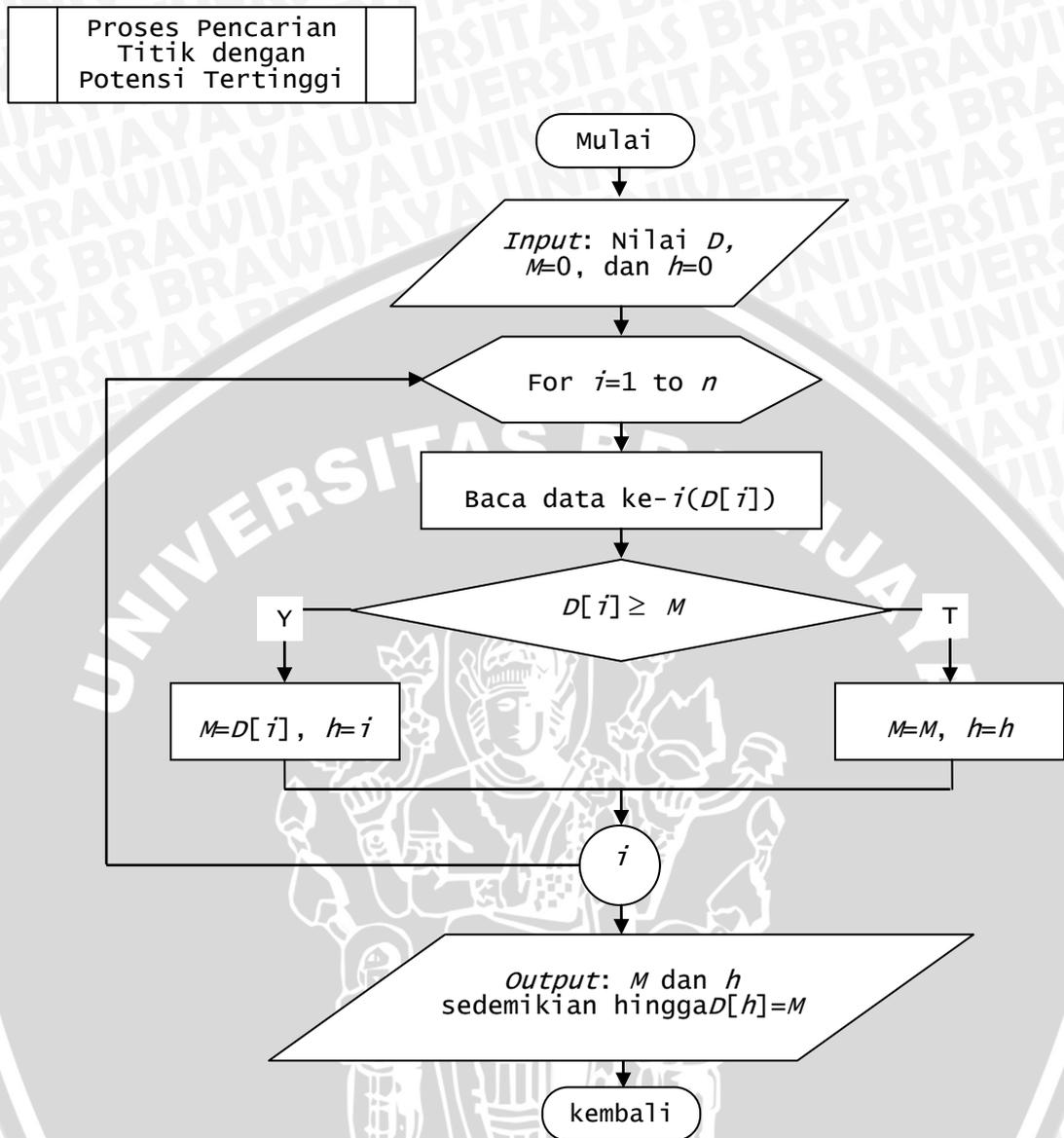
1. Data yang diolah pada proses ini adalah data yang sudah ternormalisasi (x) pada proses sebelumnya. Selain itu perlu memberi masukan juga untuk nilai jari – jari (r).
2. Inisialisasi awal untuk variabel m sebagai jumlah atribut data latih dan variabel n sebagai jumlah data latih.
3. Iterasi dari $i = 1$ sampai n , dilakukan langkah berikut:
 - a. Iterasi dari $j = 1$ sampai m , dilakukan langkah berikut:
 - i. Nilai T_j diset sama dengan X_{ij} seperti pada Persamaan (2-4).
 - ii. Iterasi dari $k=1$ sampai n , dilakukan langkah berikut:
 - Menghitung jarak antar data dengan Persamaan (2-5).
 - iii. Menghitung potensi tiap titik data dengan Persamaan (2-6) atau Persamaan (2-7).
4. Hasil akhir pada proses ini berupa nilai data D yang merupakan potensi tiap titik data.

3) Proses Pencarian Titik Data dengan Potensi Tertinggi

Proses pencarian titik data dengan potensi tertinggi adalah proses pencarian nilai tertinggi dari data untuk menjadi pusat *cluster*. Tahapan dari proses ini dijelaskan sebagai berikut:

1. Menyiapkan nilai data D sebagai nilai potensi awal tiap titik data.
2. Inisialisasi awal untuk variabel n sebagai jumlah data D , $M=0$, dan $h=0$. Dimana M adalah nilai data tertinggi dan h untuk indeks data tertinggi.
3. Iterasi $i=1$ sampai n , dilakukan langkah berikut:
 - a. Lakukan pengecekan: jika $D[i] \geq M$, sehingga $M=D_i$ dan $h=i$.
 - b. Lakukan pengecekan: jika $D[i] < M$, sehingga $M=M$ dan $h=h$.
4. Hasil akhir dari proses ini adalah nilai M sebagai nilai data dengan potensi tertinggi dan h adalah indeks posisi data tertinggi.

Alur proses pencarian titik data dengan potensi tertinggi ditunjukkan pada Gambar 3.6.



Gambar 3.6 Alur proses penentuan titik dengan potensi tertinggi

4) Proses Penentuan Pusat Cluster dan Pengurangan Potensinya

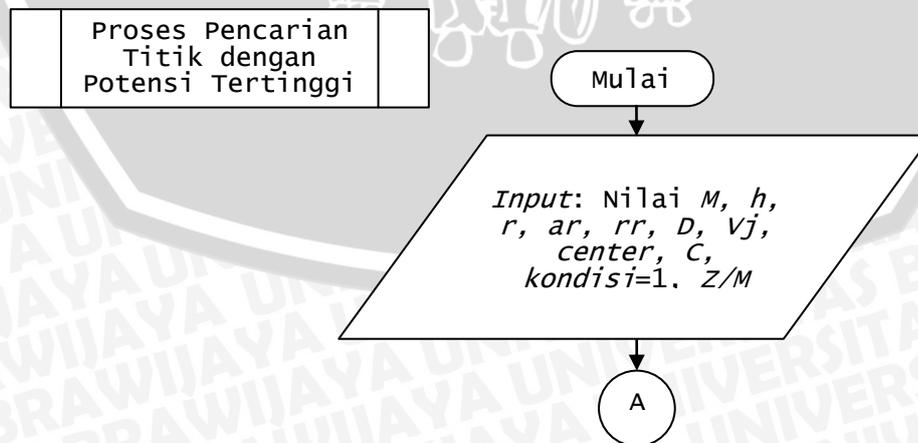
Pada proses penentuan pusat *cluster* dan pengurangan potensinya merupakan proses untuk memilih titik data manakah yang akan dipilih sebagai pusat *cluster*. Alur proses penentuan pusat *cluster* dan pengurangan potensinya ditunjukkan pada Gambar 3.7. Penjelasan dari tiap langkah untuk proses ini dijelaskan sebagai berikut:

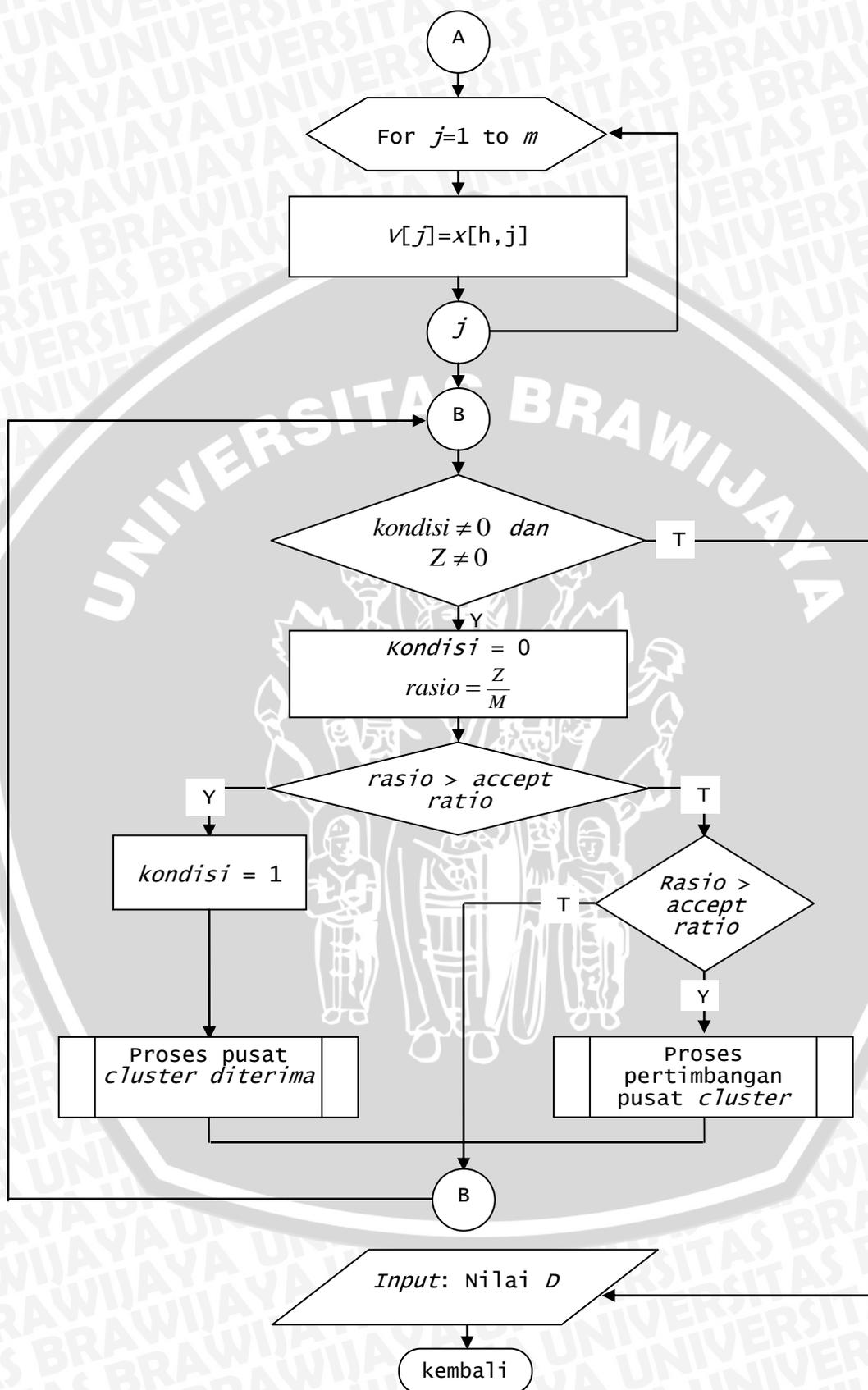
1. Masukan untuk proses ini adalah nilai M sebagai data ternormalisasi dengan potensi awal tertinggi, h sebagai indeks posisi dari data



ternormalisasi dengan potensi awal tertinggi, nilai jari – jari (r), q sebagai *squash factor*, *accept ratio* (ar), *reject ratio* (rr), dan nilai D sebagai data ternormalisasi.

2. Inisialisasi V_j yang merupakan nilai potensi awal tertinggi pada data ternormalisasi sehingga $V_j = x_{hj}$, *center* sebagai pusat *cluster*, C jumlah *cluster*, *kondisi*=1, dan Z yang merupakan potensi titik yang diacu sebagai pusat *cluster* bernilai sama dengan M .
3. Selama *kondisi* $\neq 0$ dan $Z \neq 0$, maka kerjakan proses berikut:
 - a. *kondisi* = 0, sudah tidak ada lagi calon pusat *cluster* yang baru.
 - b. $rasio = \frac{Z}{M}$
 - c. Lakukan pengecekan: jika $rasio > accept\ ratio(ar)$, maka *kondisi* = 1. Hal ini menandakan ada calon pusat baru. Alur proses pusat *cluster* diterima ditunjukkan seperti pada Gambar 3.8.
 - d. Lakukan pengecekan: jika $rasio > reject\ ratio(rr)$, maka calon pusat *cluster* baru akan diterima sebagai pusat *cluster* jika keberadaannya akan memberikan keseimbangan terhadap data – data yang letaknya cukup jauh dengan pusat *cluster* yang telah ada. Alur proses pertimbangan untuk calon pusat *cluster* ditunjukkan seperti pada Gambar 3.9.
4. Hasil akhir dari proses ini adalah hasil *cluster* yang berupa jumlah *cluster* yang terbentuk dan pusat *cluster* yang terpilih (*medoid*).

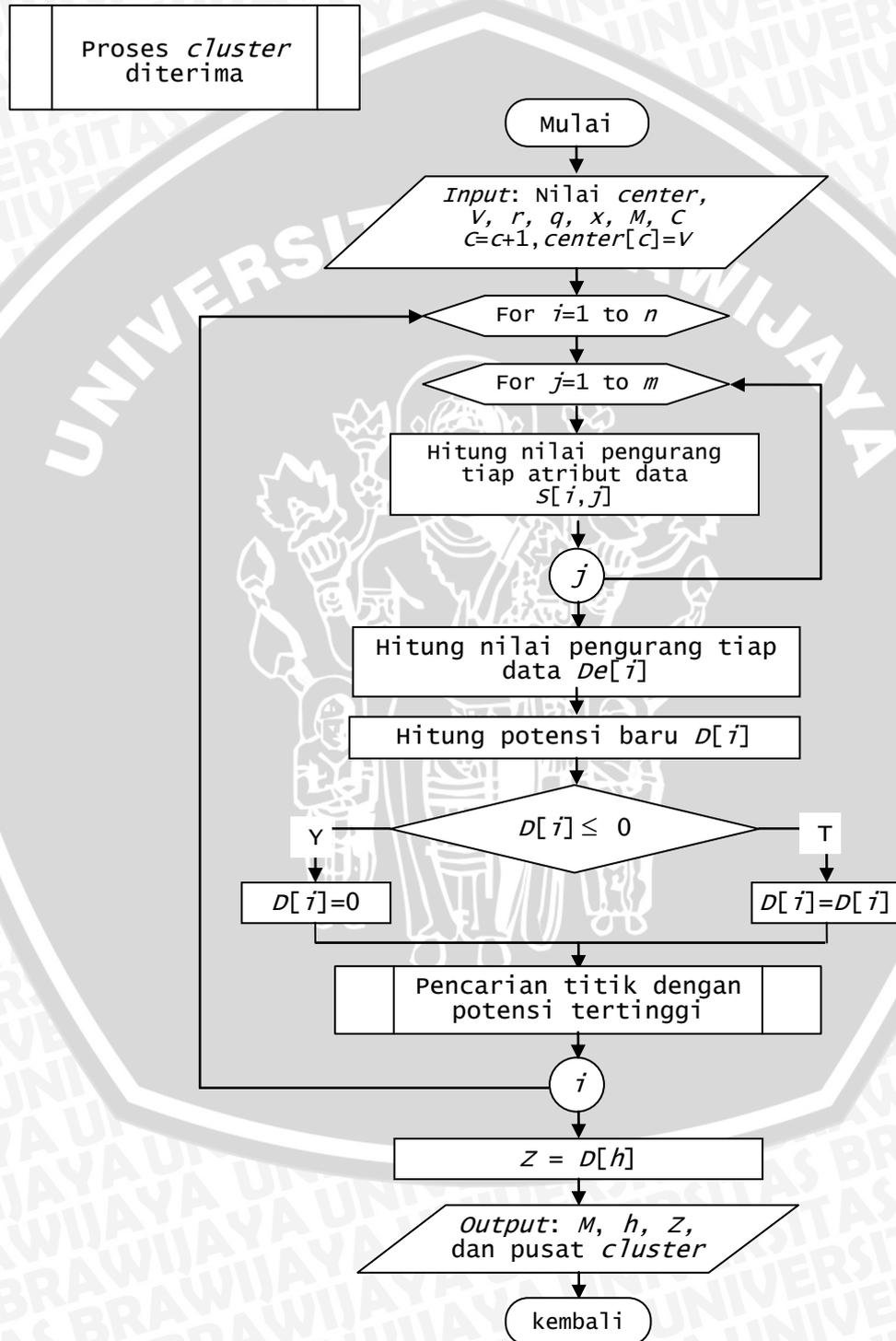




Gambar 3.7 Alur proses penentuan pusat cluster dan pengurangan potensinya

5) Proses Cluster Diterima

Pada proses *cluster* diterima merupakan proses diterimanya suatu titik data untuk menjadi pusat *cluster* berdasarkan suatu kondisi. Alur proses *cluster* diterima ditunjukkan pada Gambar 3.8.



Gambar 3.8 Alur proses *cluster* diterima



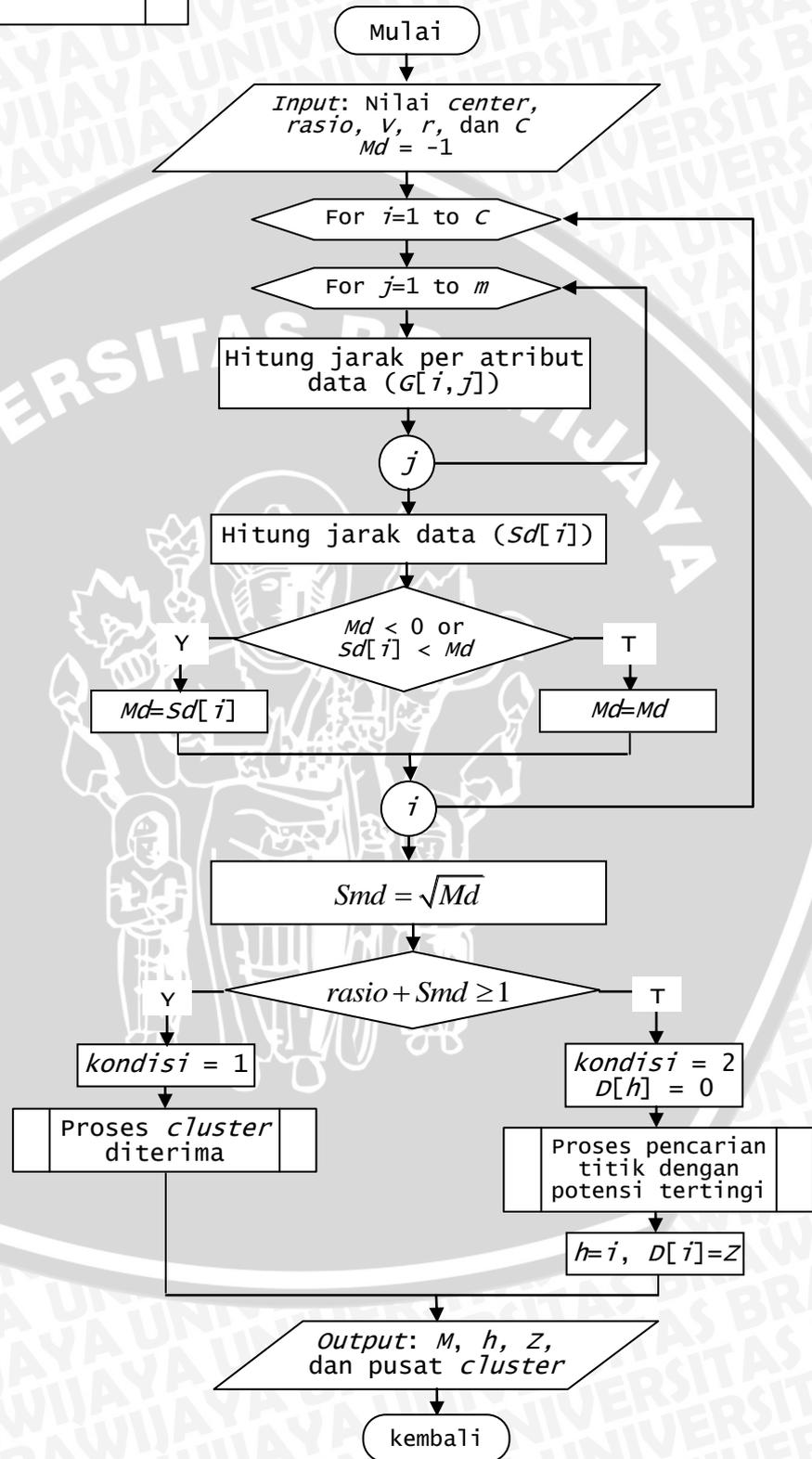
Penjelasan untuk bagan alir pada Gambar 3.8 adalah sebagai berikut:

1. Menyiapkan data untuk proses ini, diantaranya adalah *center* yang merupakan nilai dari pusat *cluster* yang telah terbentuk, V merupakan nilai normalisasi pada data dengan potensi tertinggi, jumlah *cluster* (C), jari – jari (r), data ternormalisasi (x), *squash factor* (q), dan M sebagai nilai potensi data tertinggi.
2. Inisialisasi $center[C] = V$.
3. Kerjakan $C = C + 1$.
4. Mengurangi potensi dari titik – titik di dekat pusat *cluster*, dengan langkah sebagai berikut:
 - a. Menghitung nilai pengurang tiap atribut data (S_{ij}) dengan menggunakan Persamaan (2-11).
 - b. Mencari pengurang potensi lama setiap titik data dengan menggunakan Persamaan (2-12).
 - c. Mengurangi potensi lama titik data dengan potensi baru titik data menggunakan Persamaan (2-13).
 - d. Melakukan pengecekan: jika potensi data ke- i (D_i) ≤ 0 , maka $D_i=0$.
 - e. Nilai Z diset menjadi nilai tertinggi pada D_i , dimana $i = 1,2,3,\dots,n$.
 - f. Pilih $h = i$, sedemikian hingga $D_i = Z$.
5. Hasil akhir dari proses ini adalah nilai M , h , Z , dan pusat *cluster*.

6) Proses Pertimbangan Pusat Cluster

Pada proses ini akan dilakukan langkah – langkah dalam mempertimbangkan bagaimana calon pusat *cluster* dapat diterima sebagai pusat *cluster* atau tidak. Calon pusat *cluster* baru akan diterima jika keberadaannya mampu memberikan keseimbangan terhadap data – data yang letaknya cukup jauh dengan pusat *cluster* yang telah ada. Parameter yang dapat dijadikan acuan untuk dapat menilai apakah sebuah calon pusat *cluster* dikatakan cukup jauh dengan pusat *cluster* yang telah ada adalah ketika nilai variabel $Smd + rasio \geq 1$. Untuk lebih jelas, alur proses pertimbangan *cluster* ditunjukkan pada Gambar 3.9.

Proses pertimbangan pusat cluster



Gambar 3.9 Alur proses pertimbangan pusat cluster



Penjelasan langkah demi langkah pada proses pertimbangan pusat *cluster* adalah sebagai berikut:

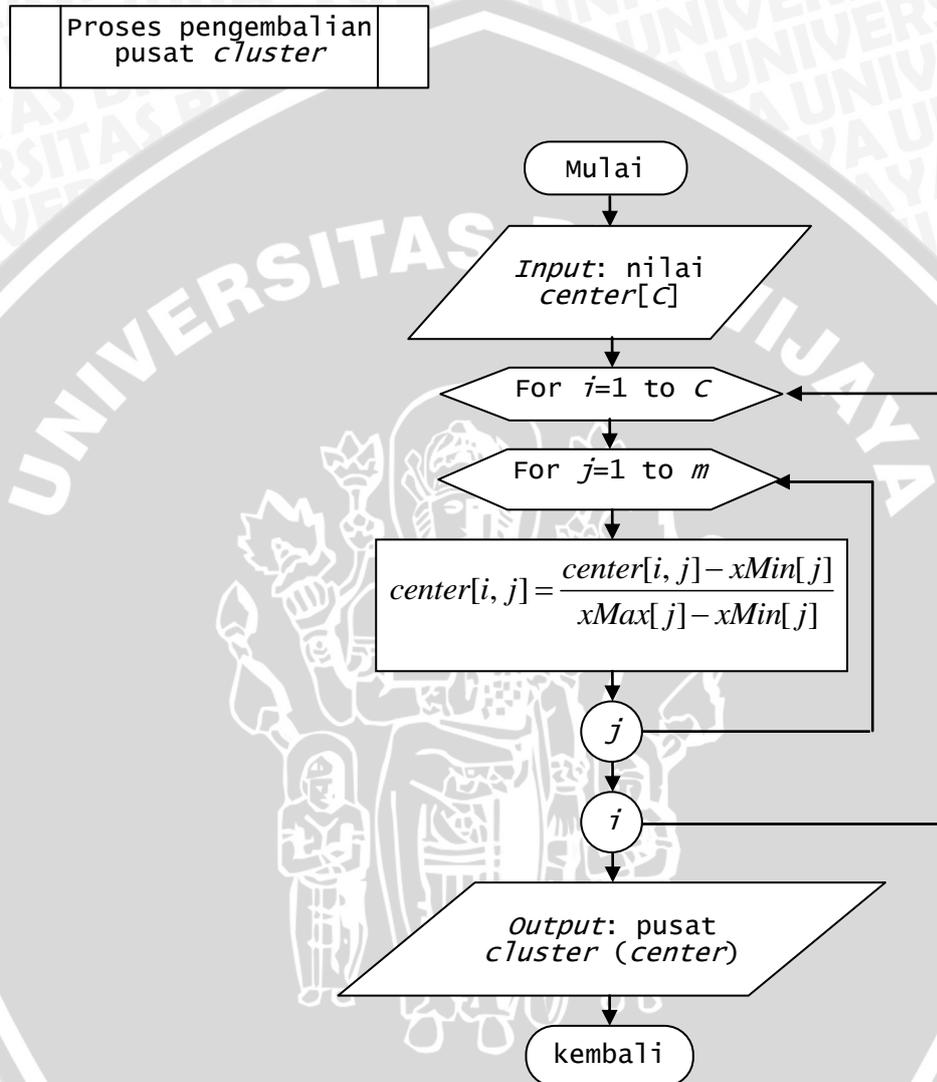
1. Menyiapkan variabel – variabel yang dibutuhkan pada proses ini, diantaranya adalah *center*, calon pusat *cluster* (V), jari – jari (r), jumlah *cluster* (C), dan *rasio*.
2. Inisialisasi awal untuk nilai $Md = -1$.
3. Iterasi $i=1$ sampai C , dilakukan langkah berikut:
4. Menghitung nilai Smd sebagai jarak terdekat dengan pusat *cluster* sebelumnya dengan Persamaan (2-10).
5. Lakukan pengecekan: jika $(rasio + Smd) \geq 1$, maka $kondisi = 1$ yang artinya data diterima sebagai pusat *cluster*. Proses *cluster* diterima dapat dilihat pada Gambar 3.8.
6. Lakukan pengecekan: jika $(rasio + Smd) < 1$, maka $kondisi = 2$ yang menandakan bahwa data tidak akan dipertimbangkan kembali sebagai pusat *cluster*.
7. Hasil akhir dari proses ini adalah nilai M , h , dan pusat *cluster* baru (*center*)

7) Proses Pengembalian Pusat Cluster

Proses pengembalian pusat *cluster* adalah proses dimana nilai dari pusat *cluster* dalam bentuk ternormalisasi dikembalikan ke bentuk semula sebelum dilakukan proses normalisasi. Jadi nantinya pusat *cluster* yang dihasilkan merupakan nilai sesungguhnya dari titik data yang menjadi pusat *cluster*. Alur proses pengembalian pusat *cluster* ditunjukkan pada Gambar 3.10 dan penjelasan untuk tiap langkah pada proses pengembalian pusat *cluster* dijelaskan sebagai berikut:

1. Menyiapkan nilai *center* sebagai kumpulan dari pusat *cluster* yang akan didenormalisasi, nilai $xMin$, dan $xMax$.
2. Inisialisasi awal untuk variabel m sebagai jumlah atribut pusat *cluster* dan variabel C sebagai jumlah *cluster*.
3. Iterasi dari $i = 1$ sampai C , dilakukan langkah berikut:

- a. Iterasi dari $j = 1$ sampai m , dilakukan proses perhitungan nilai $center_{ij}$ dengan Persamaan (2-14).
- 4. Hasil dari proses ini adalah pusat *cluster* (*cluster*) dalam bentuk data semula.



Gambar 3.10 Alur proses pengembalian pusat *cluster*

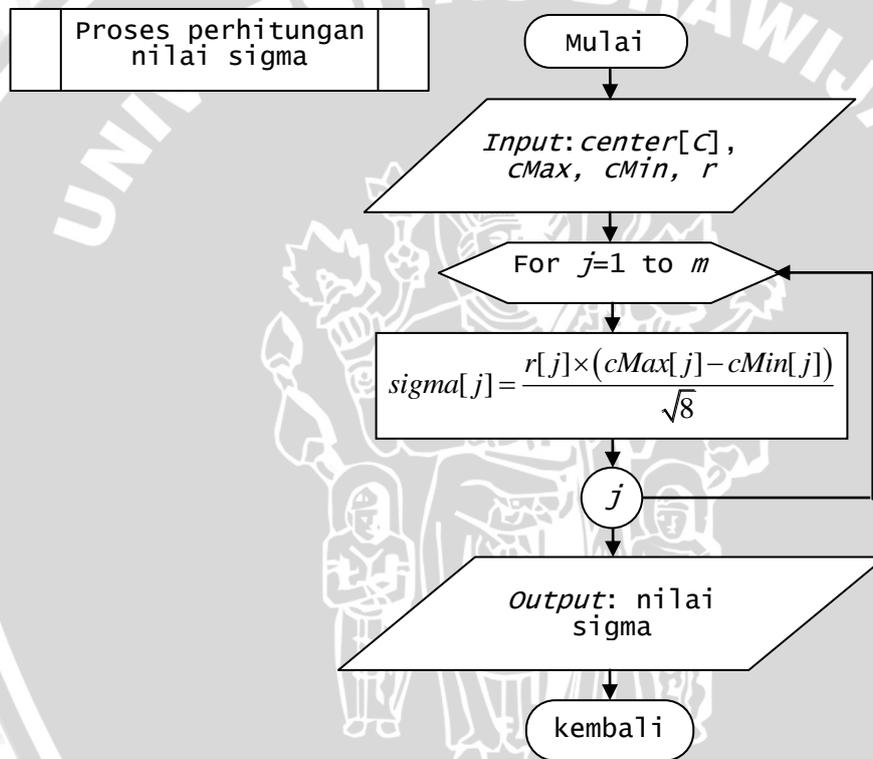
8) Proses Perhitungan Nilai Sigma

Proses perhitungan nilai sigma merupakan proses untuk mendapatkan nilai sigma pada masing – masing atribut pusat *cluster*. Nilai sigma dan pusat *cluster* nantinya akan digunakan untuk mencari nilai derajat keanggotaan masing – masing titik data. Alur proses perhitungan nilai sigma ditunjukkan



pada Gambar 3.11. Penjelasan untuk tiap langkah pada proses perhitungan nilai sigma ialah sebagai berikut:

1. Nilai masukan pada proses ini diantaranya adalah nilai terbesar (*cMax*) dan terkecil (*cMin*) pada masing – masing atribut pusat *cluster*, nilai *center* sebagai pusat *cluster*, dan nilai jari – jari (*r*).
2. Iterasi dari *j* = 1 sampai *m*, dilakukan langkah berikut:
 - a. Dilakukan perhitungan nilai sigma pada masing – masing atribut pusat *cluster* dengan menggunakan Persamaan (2-15).
3. Hasil akhir dari proses ini adalah nilai sigma.



Gambar 3.11 Alur proses perhitungan nilai sigma

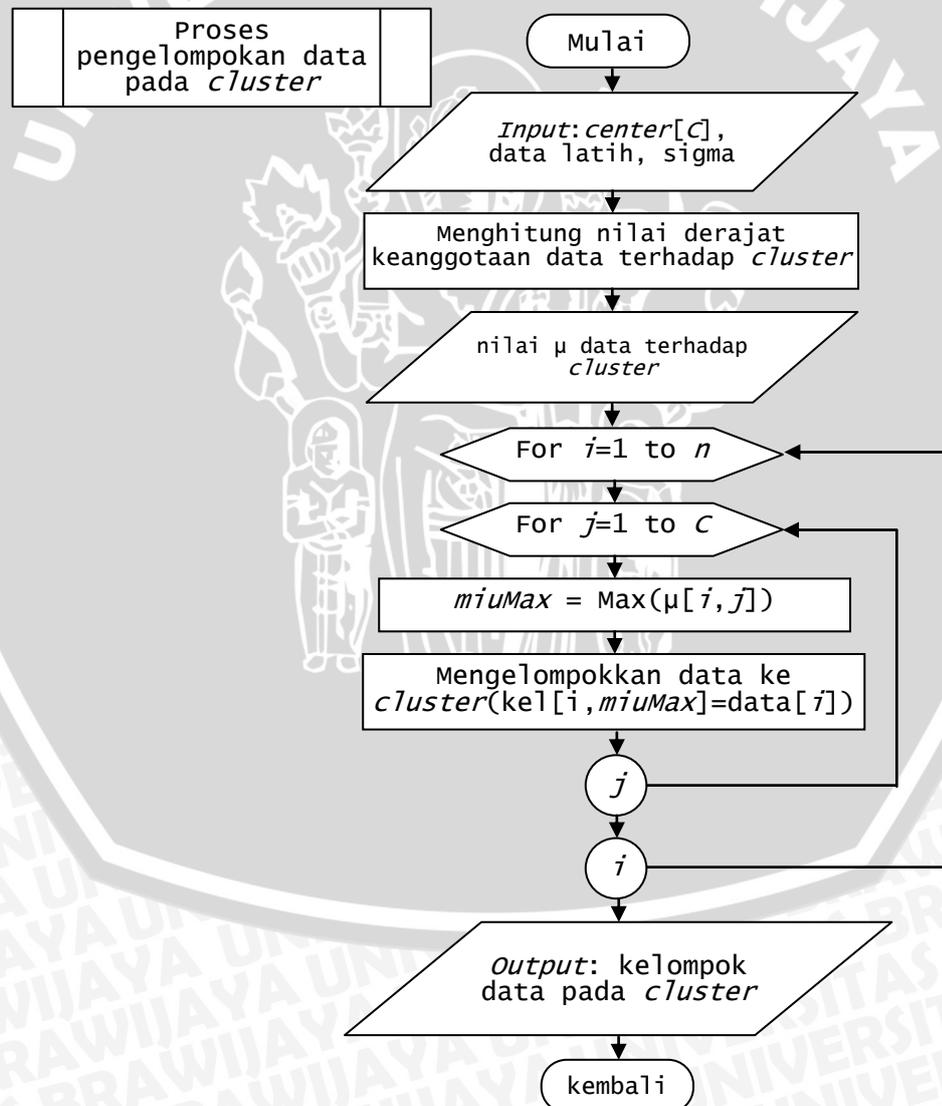
9) Pengelompokan Data pada Cluster

Proses pengelompokan data pada *cluster* merupakan proses untuk mengelompokkan data latih kedalam *cluster* yang terbentuk berdasarkan derajat keanggotaan masing – masing titik data terhadap pusat *cluster*. Alur proses pengelompokan data pada *cluster* ditunjukkan pada Gambar 3.12 dan dijelaskan sebagai berikut:

1. Masukan pada proses ini adalah nilai *center*, sigma dan data latih.



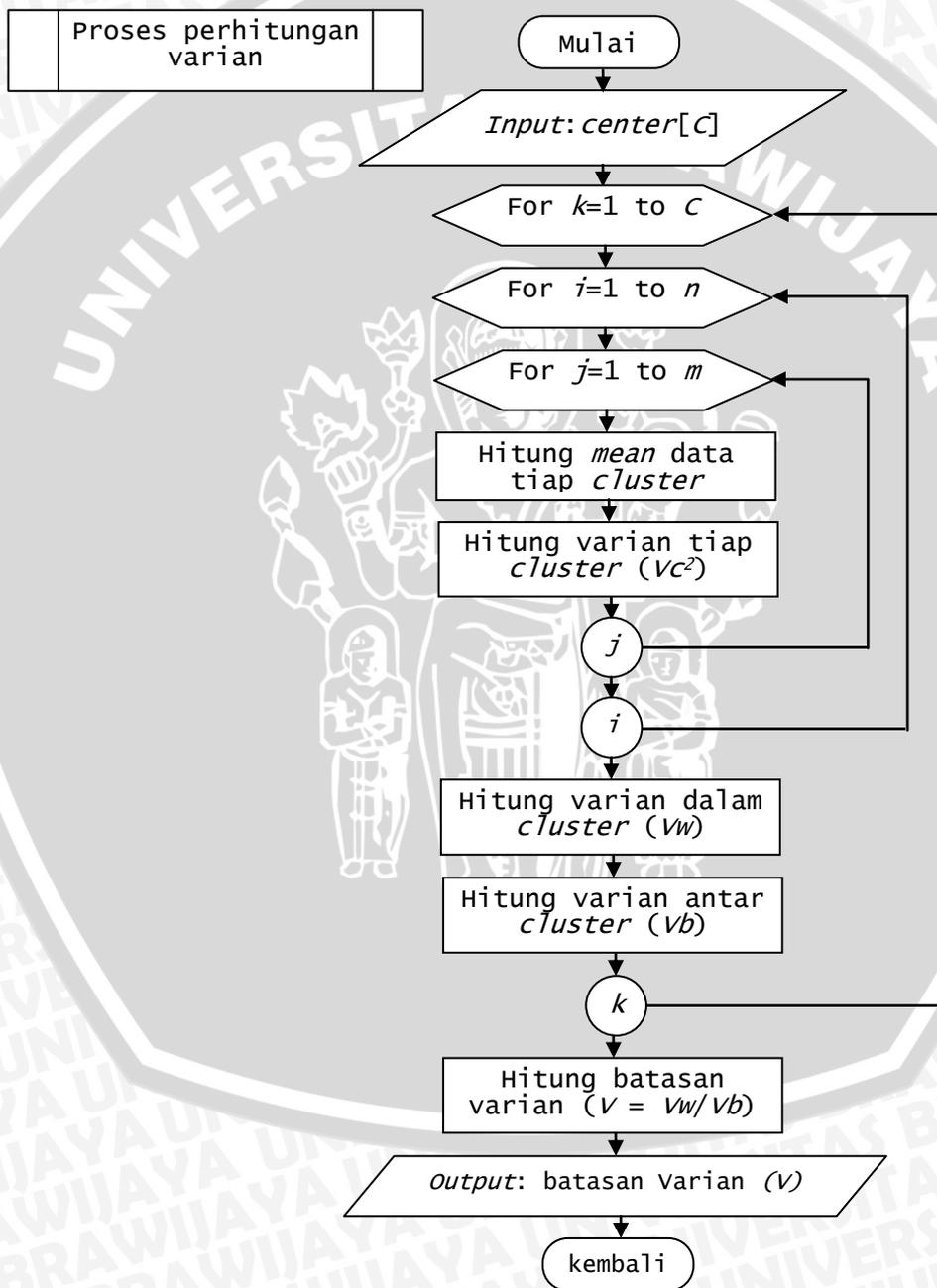
2. Menghitung derajat keanggotaan masing – masing titik data terhadap pusat *cluster* menggunakan fungsi *gauss*.
3. Iterasi dari $i = 1$ sampai n , dilakukan langkah berikut:
 - a. Iterasi dari $j = 1$ sampai C , dilakukan perhitungan untuk mendapatkan nilai derajat keanggotaan titik data terhadap *cluster* yang terbentuk. Setelah itu dicari nilai derajat keanggotaan tertinggi terletak pada *cluster* mana, sehingga dapat disimpulkan bahwa titik data menempati *cluster* dengan derajat keanggotaan tertinggi.
4. Hasil akhir dari proses ini adalah kelompok data yang masuk sesuai dengan *cluster*-nya masing – masing.



Gambar 3.12 Alur proses pengelompokan data pada *cluster*

3.3.2.2 Proses Perhitungan Varian

Pada proses ini akan dihitung nilai varian pada tiap hasil *cluster* yang terbentuk sebagai langkah untuk menganalisa *cluster*. Analisa *cluster* digunakan untuk mengetahui hasil *cluster* yang ideal untuk proses pembangkitan aturan *fuzzy*. Langkah – langkah perhitungan varian dapat dilihat di bagan alir pada Gambar 3.13.



Gambar 3.13 Alur proses perhitungan varian

Berikut langkah – langkah perhitungan varian pada proses analisa *cluster*:

1. Mengelompokkan data latih berdasarkan hasil *cluster* yang terbentuk.
2. Menghitung nilai varian pada tiap *cluster* seperti pada Persamaan (2-17). Nilai dari varian ini akan digunakan untuk menghitung nilai *variance within cluster*.
3. Menghitung nilai *variance within cluster* seperti pada Persamaan (2-16) untuk mengetahui sebaran data dalam sebuah *cluster*.
4. Menghitung nilai *variance between cluster* seperti pada Persamaan (2-18) untuk mengetahui sebaran data antar *cluster*.
5. Menghitung batasan varian dengan Persamaan (2-19). Hasil dari perhitungan inilah yang nantinya dijadikan bahan pertimbangan untuk dapat menentukan jumlah *cluster* mana yang akan diambil untuk dijadikan bahan pada proses selanjutnya.

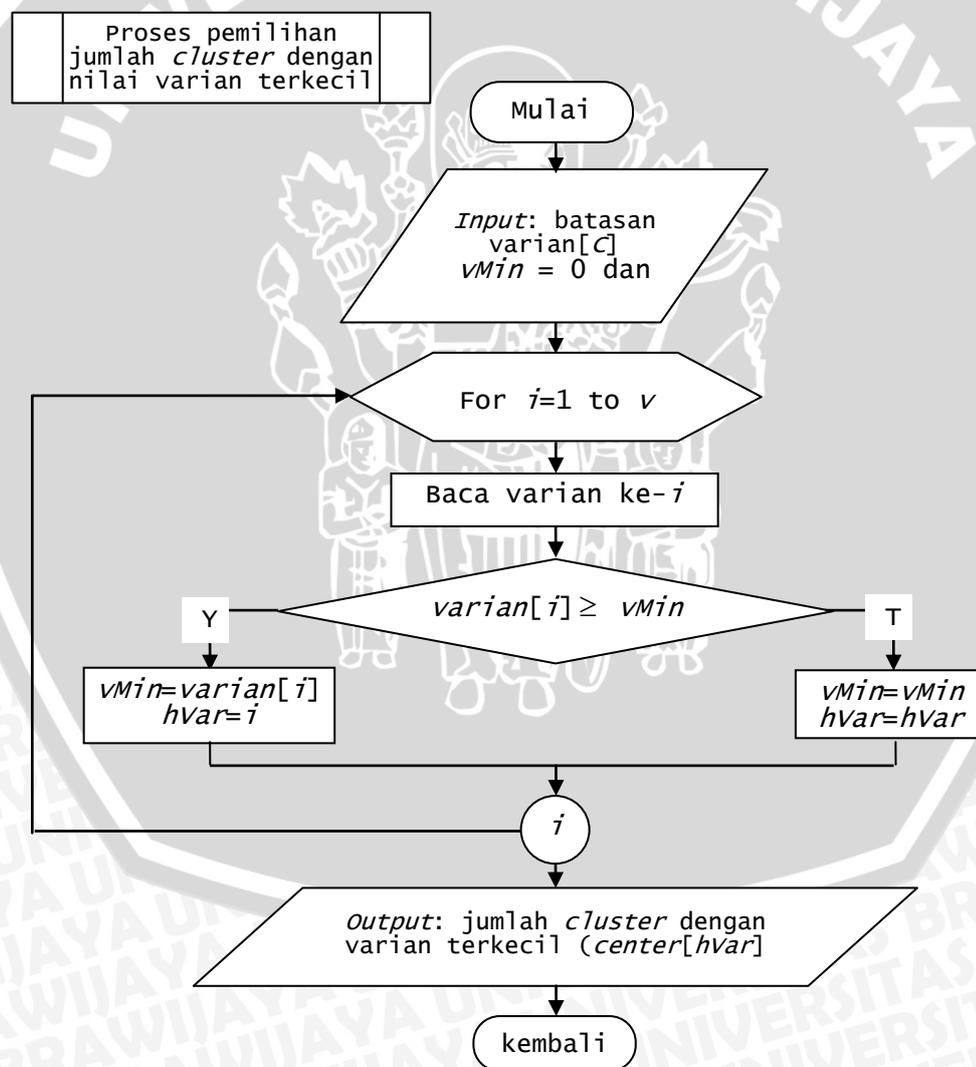
3.3.2.3 Proses Pemilihan Jumlah *Cluster* dengan Varian Terkecil

Proses pemilihan jumlah *cluster* dengan varian terkecil adalah proses untuk mendapatkan jumlah *cluster* yang dijadikan bahan pada proses pembangkitan aturan *fuzzy*. Berdasarkan literatur pada bab dua, *cluster* yang baik adalah *cluster* yang memiliki varian kecil. Hal ini dapat diasumsikan bahwa sebaran data pada *cluster* tidak memiliki variasi yang tinggi. Langkah – langkah pemilihan jumlah *cluster* dengan varian terkecil ialah sebagai berikut:

1. Masukan pada proses ini adalah nilai batasan varian (*varian*) dari C *cluster* (*center*) yang terbentuk dari beberapa inisialisasi awal yang berbeda pada algoritma *subtractive clustering*.
2. Inisialisasi awal untuk variabel v sebagai jumlah varian, $vMin = 0$ dimana $vMin$ adalah variabel untuk menampung nilai varian terkecil, dan nilai $hVar$ sebagai variabel untuk menampung indeks posisi pusat *cluster* yang nilai variannya terkecil.
3. Iterasi $i=1$ sampai v , dilakukan langkah berikut:

- a. Lakukan pengecekan: jika $varian \geq vMin$, maka nilai $vMin = varian[i]$ dan nilai $hVar=i$.
 - b. Lakukan pengecekan: jika $varian[i] < vMin$, maka nilai $vMin = vMin$ dan nilai $hVar = hVar$.
4. Hasil akhir dari proses ini adalah jumlah *cluster* yang memiliki nilai varian terkecil. *Cluster* dengan varian terkecil ditandai dengan pusat *cluster* ($center[hVar]$) memiliki varian terkecil, dimana $hVar$ adalah indeks posisi dari pusat *cluster*.

Alur proses untuk pemilihan hasil *cluster* dengan varian terkecil ditunjukkan pada bagan alir seperti yang terlihat pada Gambar 3.14.



Gambar 3.14 Alur proses pemilihan jumlah *cluster* dengan varian terkecil

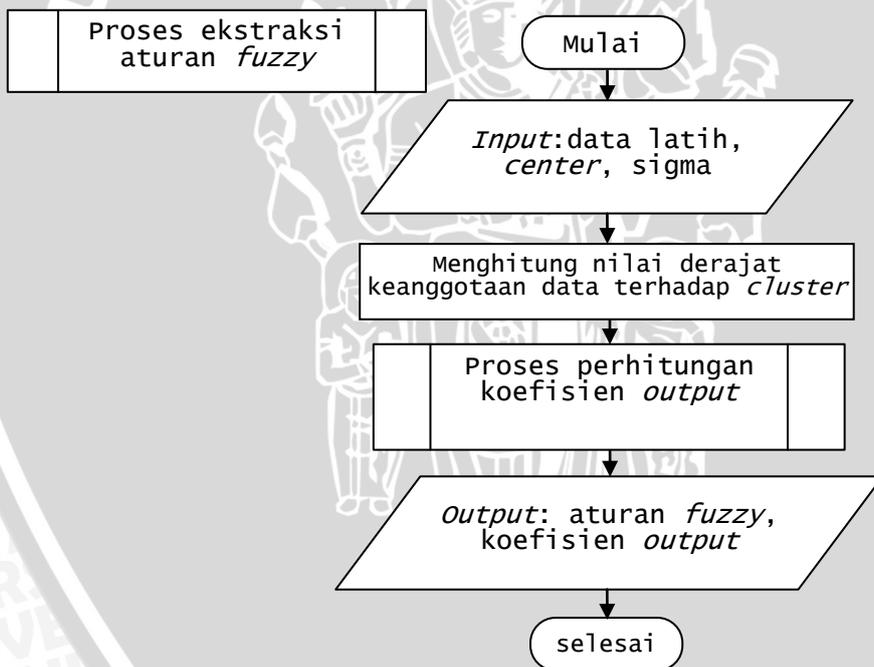


3.3.2.4 Proses Ekstraksi Aturan *Fuzzy* dari *Cluster*

Proses ekstraksi aturan *fuzzy* merupakan proses untuk mengubah *cluster* yang telah terbentuk menjadi kumpulan aturan yang nantinya akan diterapkan pada siste inferensi *fuzzy* model sugeno orde-satu. Tahapan dari proses ini dapat dilihat pada penjelasan sebagai berikut:

1. Menyiapkan data latih, sigma, dan hasil *cluster* (*center*).
2. Menghitung nilai derajat keanggotaan data terhadap masing – masing *cluster* untuk mengetahui kelompok data pada suatu *cluster* dengan menggunakan Persamaan (2-29).
3. Menghitung nilai koefisien *output*.
4. Hasil akhir dari proses ini adalah aturan *fuzzy* dan koefisien *output*.

Alur proses ekstraksi aturan *fuzzy* dari *cluster* ditunjukkan pada Gambar 3.15.



Gambar 3.15 Alur proses ekstraksi aturan *fuzzy* dari *cluster*

1) Proses Perhitungan Koefisien *Output*

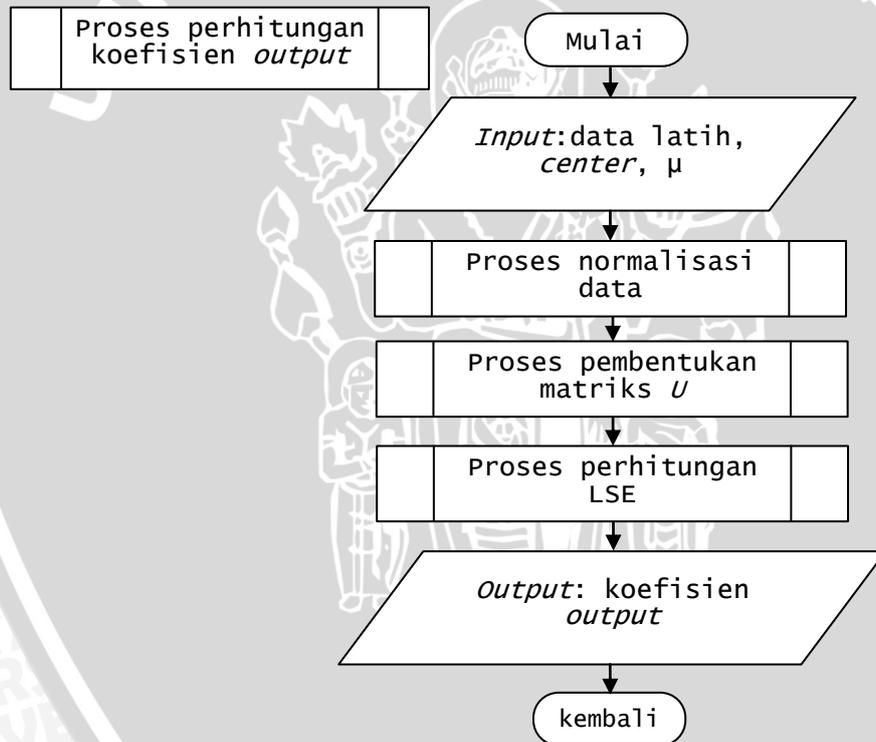
Proses perhitungan koefisien *output* adalah proses untuk mendapatkan nilai koefisien *output* pada sistem inferensi *fuzzy*. Koefisien *output* adalah suatu konstanta yang mempengaruhi variabel dalam menentukan target *output* dari sistem inferensi *fuzzy*.



Langkah - langkah proses perhitungan koefisien *output* dijelaskan lebih rinci sebagai berikut:

1. Masukan dari proses ini terdiri dari data latih, pusat *cluster* (*center*), dan derajat keanggotaan (μ).
2. Proses normalisasi data yang dijabarkan lebih rinci pada Gambar 3.17.
3. Pembentukan matriks U yang dijabarkan lebih rinci pada Gambar 3.18.
4. Proses perhitungan *LSE* yang dijabarkan pada Gambar 3.19.
5. Hasil akhir dari proses ini adalah koefisien *output*.

Tahapan dari proses perhitungan koefisien *output* secara umum ditunjukkan lewat bagan alir pada Gambar 3.16. Sedangkan untuk sub-proses pada proses ini dijelaskan pada sub-bab lainnya agar lebih mudah dimengerti.

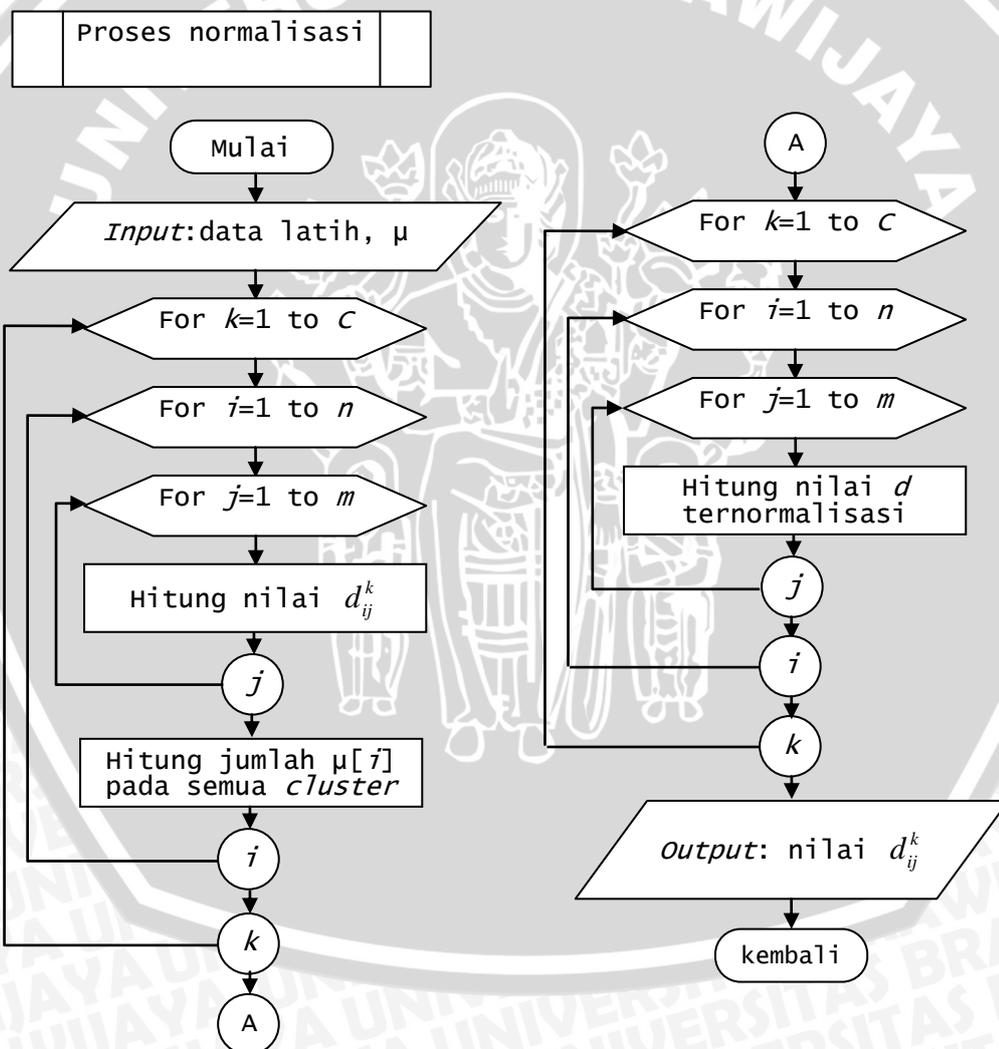


Gambar 3.16 Alur proses perhitungan koefisien *output*

2) Proses Normalisasi

Proses normalisasi adalah proses untuk mendapatkan nilai d_{ij}^k yang nantinya digunakan untuk pembentukan matriks U . Gambar 3.17 menunjukkan bagan alir pada proses normalisasi. Berikut penjelasan tahapan untuk proses normalisasi:

1. Masukan untuk proses normalisasi adalah data latih dan derajat keanggotaan masing – masing titik data (μ).
2. Menghitung nilai d_{ij}^k dengan menggunakan Persamaan (2-30).
3. Menjumlahkan derajat keanggotaan (μ_{ik}) semua *cluster*.
4. Menghitung nilai d_{ij}^k ternormalisasi dengan cara membaginya dengan $d_{i(m+1)}^k$.
5. Hasil akhir dari proses ini adalah nilai d_{ij}^k yang nantinya digunakan pada proses pembentukan matriks U .

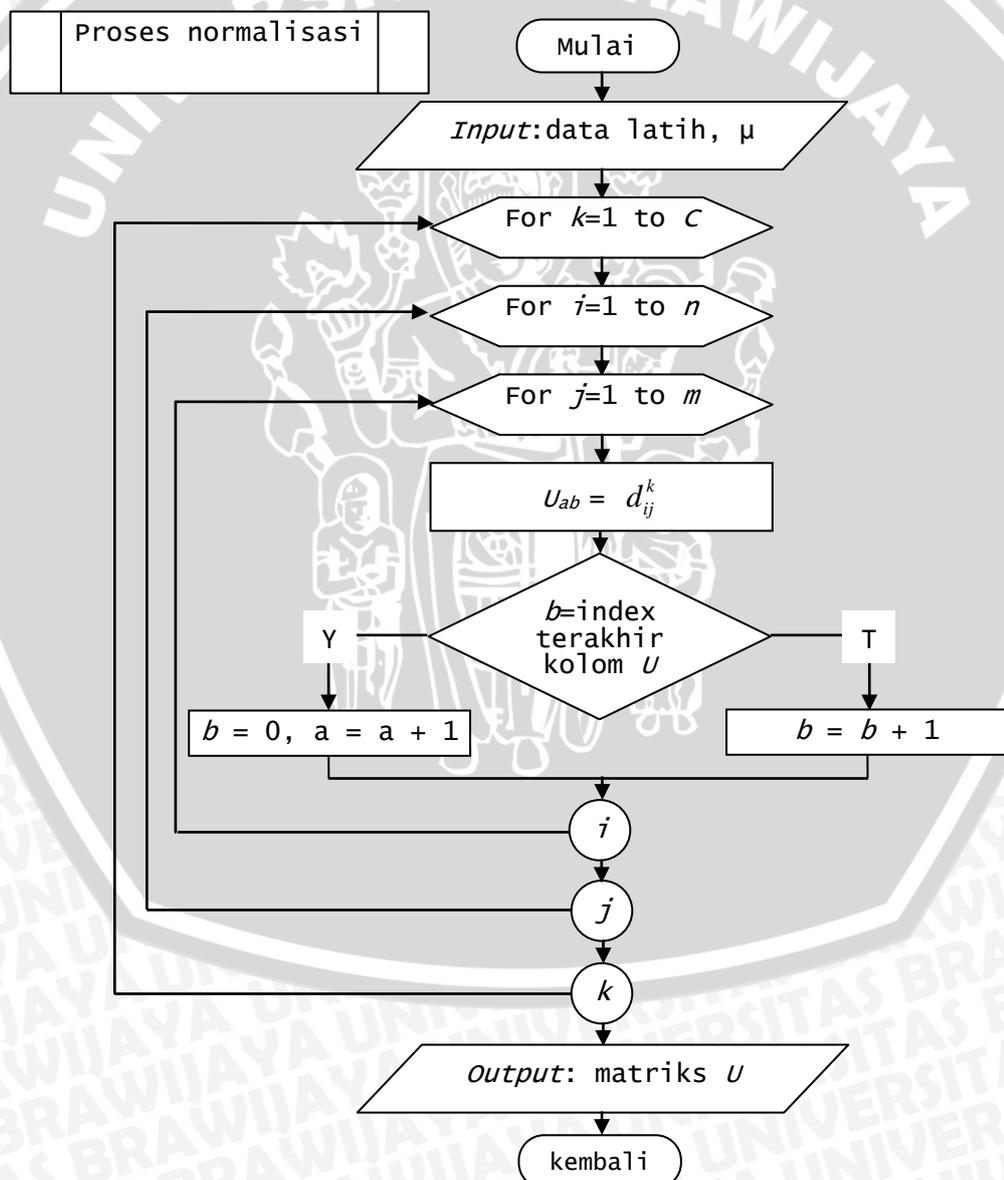


Gambar 3.17 Alur proses normalisasi

3) Proses Pembentukan Matriks U

Proses pembentukan matriks U adalah proses untuk mendapatkan sebuah matriks yang berisi normalisasi derajat keanggotaan data dikalikan dengan data latih pada tiap *cluster*.

Matrik U nantinya berperan pada proses perhitungan LSE untuk mendapatkan nilai koefisien *output*. Masukan dari matriks U adalah matriks d^k_{ij} dan menghasilkan keluaran berupa matriks U yang berdimensi jumlah data (n) \times (jumlah *cluster* * (jumlah atribut + 1)). Proses pembentukan matriks U ditunjukkan pada Gambar 3.18.

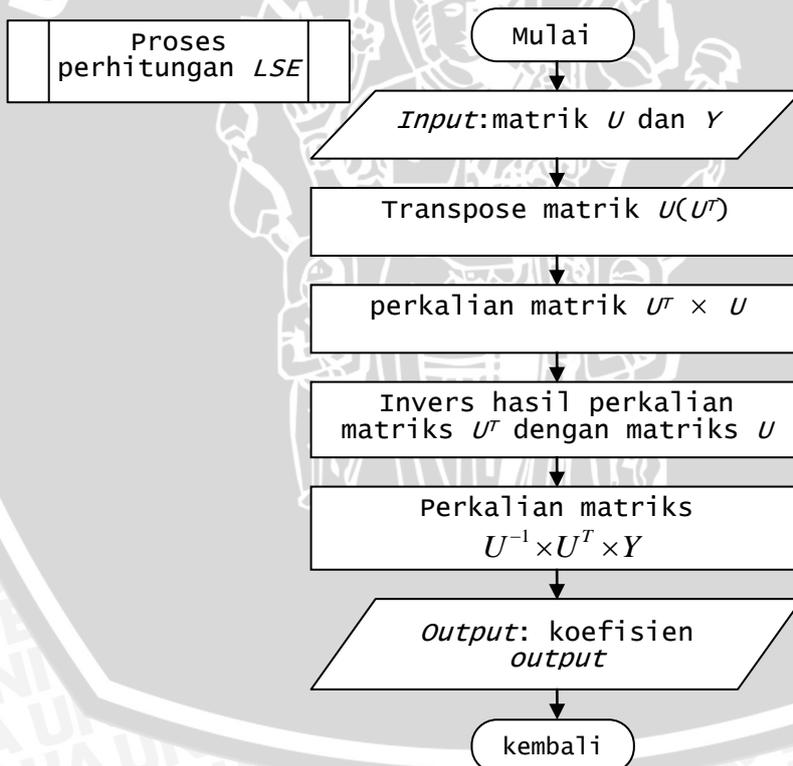


Gambar 3.18 Alur proses pembentukan matriks U

4) Proses Perhitungan *LSE*

Proses perhitungan *LSE* adalah proses untuk mendapatkan koefisien *output* dengan metode kuadrat terkecil karena matriks U sebagai variabel pembentuk koefisien *output* berbentuk bukan matriks bujur sangkar. Alur proses perhitungan *LSE* ditunjukkan pada Gambar 3.19 dan dijelaskan sebagai berikut:

1. Masukan dari proses ini adalah matriks U dan nilai kelayakan (Y).
2. Melakukan proses transpose matriks U (U^T).
3. Melakukan perkalian matriks $U^T \times U$.
4. Melakukan proses invers matriks hasil perkalian (U^{-1}).
5. Melakukan proses perkalian matriks $U^{-1} \times U^T \times Y$, dimana Y adalah matriks nilai kelayakan dari data latih.
6. Hasil akhir dari proses ini adalah koefisien *output*.



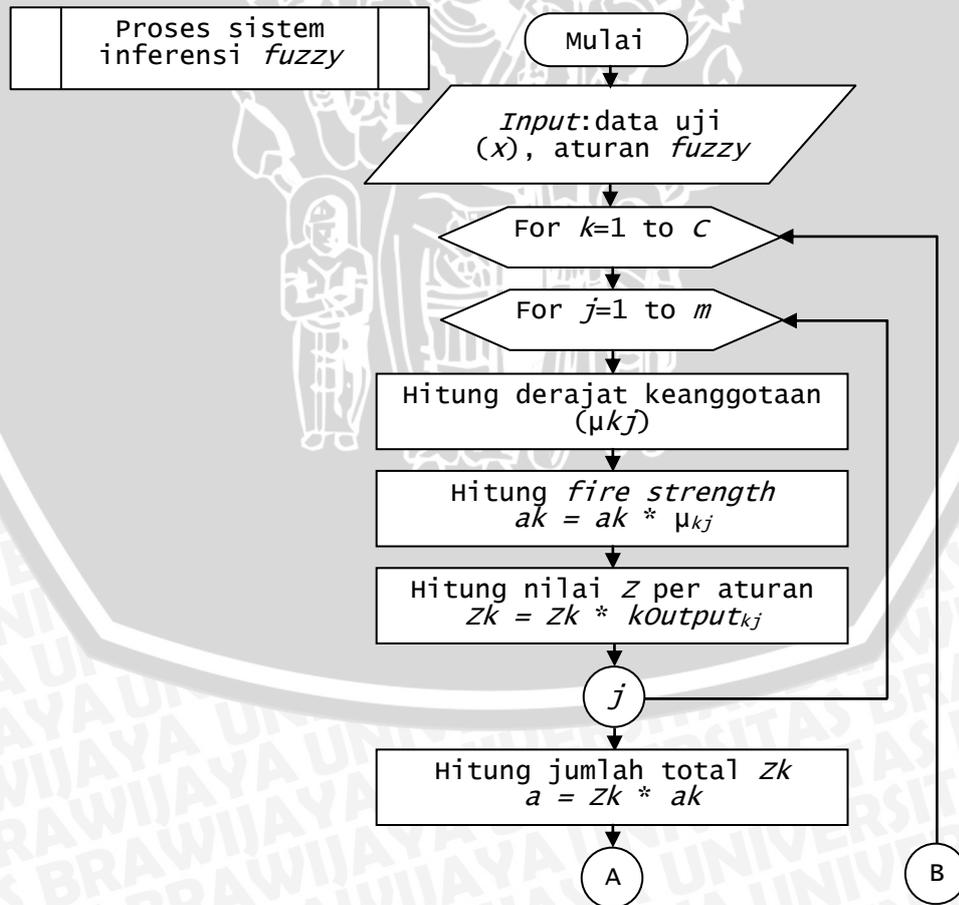
Gambar 3.19 Alur proses perhitungan *LSE*

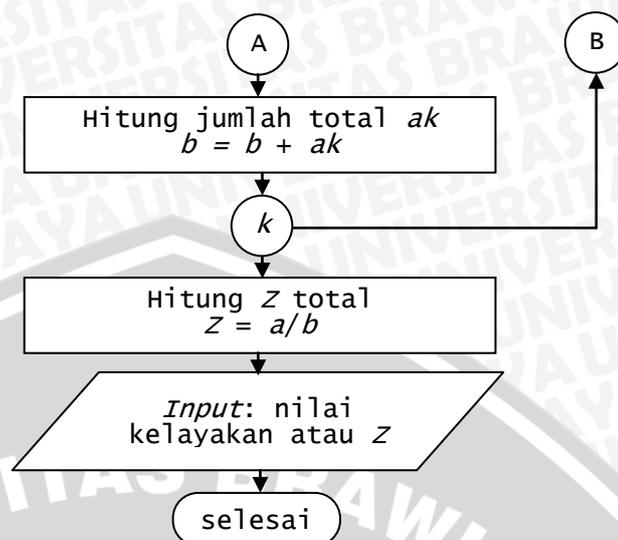
3.3.2.5 Proses Sistem Inferensi *Fuzzy*

Proses sistem inferensi *fuzzy* merupakan proses untuk mendapatkan nilai kelayakan. Pada proses ini akan dilakukan pengujian terhadap data uji

untuk diketahui nilai kelayakannya. Alur proses ini ditunjukkan pada Gambar 3.20 dan dijelaskan sebagai berikut:

1. Masukan dari proses ini adalah data uji (x) dan aturan *fuzzy*.
2. Iterasi $k=1$ sampai C , dilakukan langkah berikut:
 - a. Iterasi $j=1$ sampai m , dilakukan langkah berikut:
 - i. menghitung derajat keanggotaan menggunakan fungsi *Gauss*.
 - ii. Menghitung *fire strength* masing – masing aturan (ak).
 - iii. Menghitung nilai Z masing – masing aturan (Zk).
 - b. Hitung nilai a , dimana a adalah perkalian Zk dengan ak .
 - c. Hitung nilai b , dimana b adalah penjumlahan dari tiap ak pada *cluster*.
3. Hitung Z dengan cara membagi antara Zk dengan ak , dimana hasil akhir dari pembagian tersebut dinamakan dengan nilai kelayakan.
4. Hasil akhir dari prose ini adalah nilai Z atau nilai kelayakan.





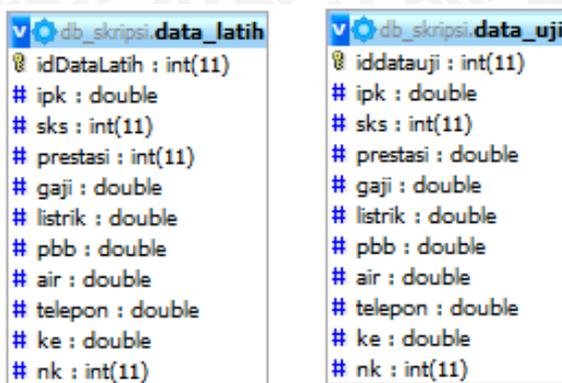
Gambar 3.20 Alur proses sistem inferensi *fuzzy*

3.3.3. Analisa dan Perancangan *Database*

Pada penelitian ini terdapat beberapa data yang perlu disimpan untuk mempermudah dalam perancangan dan implementasi sistem. Oleh karena itulah pertimbangan kebutuhan sistem menjadi acuan pada proses analisa dan perancangan basis data. Pada subbab perancangan sistem telah dijelaskan data apa saja yang dibutuhkan, mulai dari data latih, data uji, data *cluster*, dan data aturan *fuzzy*. Berdasarkan pertimbangan kebutuhan pada perancangan sistem, maka dibuat tabel basis data untuk menampung nilai – nilai yang dibutuhkan pada penelitian, diantaranya adalah tabel datauji dan tabel datalatih.

Penjelasan untuk tabel – tabel yang digunakan adalah sebagai berikut:

1. Tabel datalatih digunakan untuk menyimpan data latih pada penelitian. Atribut pada tabel datalatih terdiri dari idDataLatih, ipk, sks, prestasi, gaji, listrik, pbb, air, telepon, dan nk.
2. Tabel datauji digunakan untuk menyimpan data yang akan diuji coba pada sistem. Atribut pada tabel datauji terdiri dari idDataUji, ipk, sks, prestasi, gaji, listrik, pbb, air, telepon, dan nk.

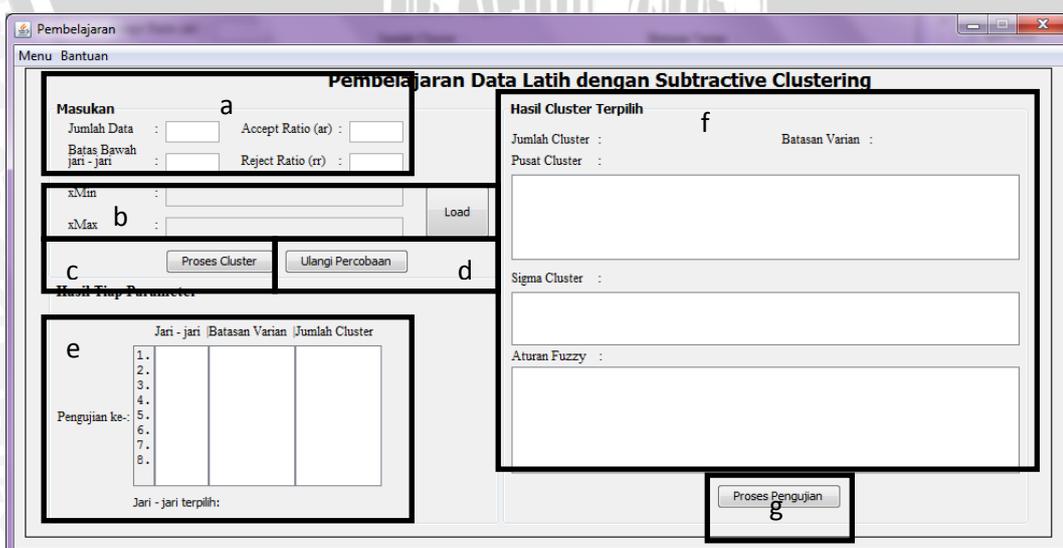


Gambar 3.21 Physical data model

3.3.4. Analisa dan Perancangan Antarmuka

Pada perancangan sistem telah dijelaskan bagaimana proses berjalannya sistem. Mulai dari pembelajaran terhadap data latih, pemilihan hasil *cluster* dengan varian terkecil, proses ekstraksi aturan *fuzzy* dari *cluster* terpilih, dan pengujian terhadap data uji menggunakan sistem inferensi *fuzzy*. Berdasarkan alur proses dari sistem yang akan dibangun, maka perancangan antarmuka juga disesuaikan dengan kebutuhan. Sehingga nantinya terdapat 2 tampilan utama antarmuka, diantaranya adalah antarmuka untuk proses pembelajaran terhadap data latih dan proses pengujian data uji menggunakan aturan yang dipilih. Tampilan antarmuka sistem dapat dilihat pada Gambar 3.22 dan Gambar 3.23.

1. Antarmuka Proses Pembelajaran Data Latih

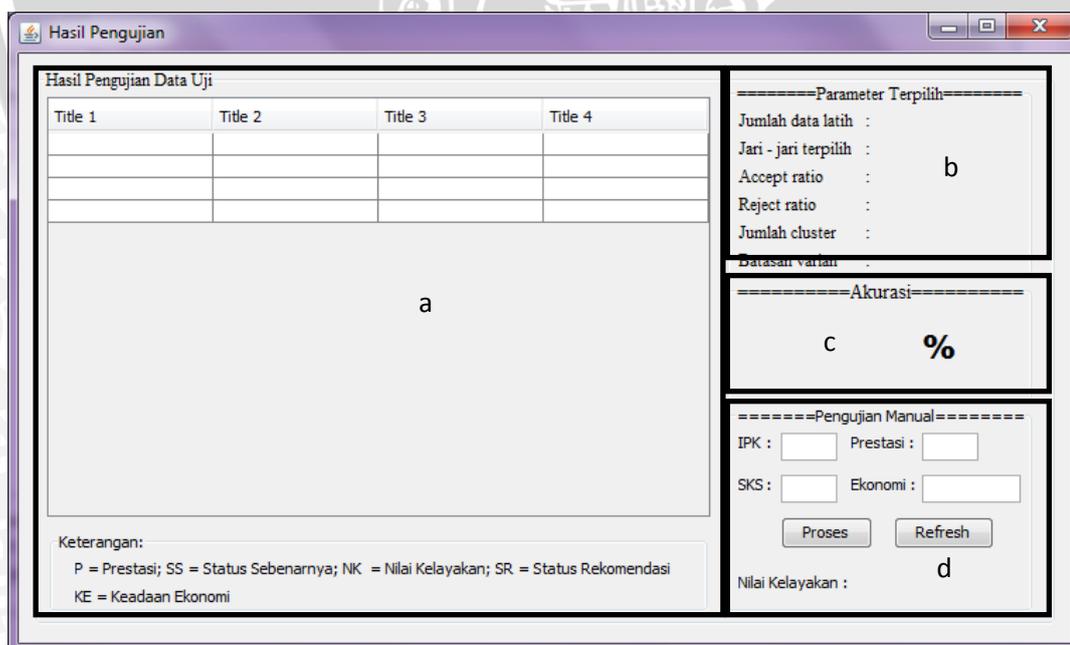


Gambar 3.22 Antarmuka proses pembelajaran data latih



Penjelasan untuk Gambar 3.22 adalah sebagai berikut:

- a. *Field* untuk memasukkan jumlah data latih, batas bawah jari – jari, *accept ratio*, dan *reject ratio*.
 - b. Tombol untuk menginisialisasi secara otomatis variabel *xMin* dan *xMax* yang kemudian akan ditampilkan pada *field xMin* dan *xMax*.
 - c. Tombol untuk melakukan proses pembelajaran menggunakan *subtractive clustering* yang kemudian sekaligus dianalisa menggunakan analisa varian untuk menentukan hasil *cluster* terpilih.
 - d. Tombol untuk mengulangi percobaan dengan mengosongkan semua *fields* dan nilai variabel pada program.
 - e. Tampilan nilai jari – jari, jumlah *cluster*, dan nilai batasan varian untuk tiap hasil percobaan serta menampilkan jari – jari terpilih berdasarkan analisa varian.
 - f. Tampilan untuk nilai batasan varian, jumlah *cluster*, pusat *cluster*, sigma *cluster*, dan koefisien *output* berdasarkan jari – jari terpilih dari hasil analisa varian.
 - g. Tombol untuk melakukan proses pengujian.
2. Antarmuka Proses Pengujian Data Uji



Gambar 3.23 Antarmuka proses pengujian data uji

Penjelasan untuk Gambar 3.23 adalah sebagai berikut:

- Tabel yang berisi tampilan data uji beserta status rekomendasi sistem berdasarkan nilai kelayakan dari hasil pengujian sistem.
- Tampilan informasi tambahan untuk jumlah *cluster*, jumlah data latih, jari – jari terpilih, *accept ratio* terpilih, *reject ratio* terpilih, dan nilai batasan varian.
- Tampilan akurasi dari sistem.
- Field* untuk melakukan proses pengujian secara manual berdasarkan aturan terpilih sebelumnya.

3.4 Perhitungan Manual

Perhitungan manual dilakukan untuk mengimplementasikan sistem secara matematis lewat perhitungan langkah demi langkah pada data latih dan data uji. Proses perhitungan manual dibedakan menjadi dua bagian sesuai dengan analisa perancangan sistem, yaitu proses perhitungan manual proses pembelajaran dan perhitungan manual proses pengujian.

3.4.1. Proses Pembelajaran

1. Proses *Subtractive Clustering*

Langkah pertama pada proses *subtractive clustering* adalah memasukkan data latih yang akan diklaster. Jumlah atribut yang digunakan sebanyak lima parameter yang terdiri dari IPK (IPK), jumlah SKS (SKS), prestasi (P), keadaan ekonomi (KE), dan nilai kelayakan (NK). Berikut data latih yang digunakan pada proses *clustering*:

IPK	SKS	P	KE	NK
3,58	104	7	3503752,667	1
3,062	104	18	1440496,667	0
3,56	107	0	7427433,333	1
3,226	104	16	3886919,167	0
3,548	104	0	4576998,75	1
3,697	104	20	4382001,917	1
3,18	104	0	2344500	0
3,23	104	0	4545230,5	0
3,66	109	3	1498516,667	1
3,05	101	9	2891906,667	0

Langkah selanjutnya adalah menentukan parameter awal sebagai berikut:

$$r = 1,3$$

$$\text{accept_ratio}(ar) = 0,5$$

$$\text{reject_ratio}(rr) = 0,15$$

$$\text{squash_factor}(q) = 1,25$$

$$\text{batas_bawah}(xMin) = (3,05; 101; 0; 1440496,667; 0)$$

$$\text{batas_atas}(xMax) = (3,697; 109; 20; 7427433,333; 1)$$

Langkah ketiga adalah menghitung data ternormalisasi menggunakan Persamaan (2-3). Sebagai contoh, untuk menghitung data ternormalisasi data latih pertama (X) adalah sebagai berikut:

$$X_{11} = \frac{x_{11} - xMin_1}{xMax_1 - xMin_1} = \frac{3,58 - 3,05}{3,697 - 3,05} = 0,8192$$

$$X_{12} = \frac{x_{12} - xMin_2}{xMax_2 - xMin_2} = \frac{104 - 101}{109 - 101} = 0,375$$

$$X_{13} = \frac{x_{13} - xMin_3}{xMax_3 - xMin_3} = \frac{7 - 0}{20 - 0} = 0,35$$

$$X_{14} = \frac{x_{14} - xMin_4}{xMax_4 - xMin_4} = \frac{3503752,667 - 1440496,667}{7427433,333 - 1440496,667} = 0,3446$$

$$X_{15} = \frac{x_{15} - xMin_5}{xMax_5 - xMin_5} = \frac{1 - 0}{1 - 0} = 1$$

Hasil selengkapnya untuk data latih ternormalisasi adalah sebagai berikut:

IPK	SKS	P	KE	NK
0,8192	0,3750	0,3500	0,3446	1
0,0185	0,3750	0,9000	0,0000	0
0,7883	0,7500	0,0000	1,0000	1
0,2720	0,3750	0,8000	0,4086	0
0,7697	0,3750	0,0000	0,5239	1
1,0000	0,3750	1,0000	0,4913	1
0,2009	0,3750	0,0000	0,1510	0
0,2782	0,3750	0,0000	0,5186	0
0,9428	1,0000	0,1500	0,0097	1
0,0000	0,0000	0,4500	0,2424	0

Setelah data ternormalisasi terbentuk, langkah selanjutnya adalah menghitung potensi awal tiap titik data (D), mulai dari $i = 1$ sampai $i = 10$. Sebagai contoh pada perhitungan potensi awal pada data pertama (D_1):

$$T = X_i, \text{ yaitu } T_1 = 0,8192; T_2 = 0,3750; T_3 = 0,3500; T_4 = 0,3446; T_5 = 1.$$

Kemudian hitung jarak setiap titik data terhadap T . Karena nilai $m > 1$, maka menggunakan Persamaan (2-7). Misal, kita hitung jarak T terhadap data pertama:

$$Dist_{11} = \left(\frac{T_1 - X_{11}}{r} \right) = \left(\frac{0,8192 - 0,8192}{1,3} \right) = 0$$

$$Dist_{12} = \left(\frac{T_2 - X_{12}}{r} \right) = \left(\frac{0,3750 - 0,3750}{1,3} \right) = 0$$

$$Dist_{13} = \left(\frac{T_3 - X_{13}}{r} \right) = \left(\frac{0,3500 - 0,3500}{1,3} \right) = 0$$

$$Dist_{14} = \left(\frac{T_4 - X_{14}}{r} \right) = \left(\frac{0,3446 - 0,3446}{1,3} \right) = 0$$

$$Dist_{15} = \left(\frac{T_5 - X_{15}}{r} \right) = \left(\frac{1 - 1}{1,3} \right) = 0$$

Selanjutnya setelah jarak titik data terhadap T diketahui, kemudian nilai $Dist_{ij}$ dijumlahkan. Untuk mempermudah dinotasikan hasil penjumlahan tersebut menjadi DS_k . Sebagai contoh untuk jumlah total jarak setiap atribut pada data pertama terhadap T adalah sebagai berikut:

$$DS_1 = Dist_{11} + Dist_{12} + Dist_{13} + Dist_{14} + Dist_{15}$$

$$DS_1 = 0 + 0 + 0 + 0 + 0 = 0$$

Proses perhitungan jarak terhadap T dilakukan untuk setiap data, yaitu $i = 1$ sampai $i = 10$, sehingga didapatkan hasil sebagai berikut:

DS
0,0000
1,2203
0,4104
0,8911
0,0929
0,2821
0,9126
0,8553
0,3302
1,0841

Untuk mempermudah kembali perhitungan dalam mendapatkan nilai potensi awal pada data pertama, dapat dihitung dengan menggunakan persamaan berikut yang sebenarnya adalah bentuk sederhana dari Persamaan (2-7):

$$D_1 = \sum_{k=1}^{10} e^{-4(DS_k)}$$

sehingga,

$$D_1 = e^{-4(0)} + e^{-4(1,2203)} + e^{-4(0,4104)} + \dots + e^{-4(1,0841)} = 2,5813$$

Dengan melakukan proses perhitungan yang sama untuk seluruh data dalam

menentukan potensi awal tiap titik data, maka nilai potensi awal masing – masing titik data adalah sebagai berikut:

D
2,5813
2,1623
1,7521
2,4561
2,4841
1,5159
1,9287
2,4205
1,5924
2,5377

Setelah potensi awal tiap titik data diketahui, langkah selanjutnya adalah mencari potensi titik tertinggi (M) dan letak data dengan potensi tertinggi tersebut (h). Sehingga didapatkan hasil sebagai berikut:

$$M = 2,5813$$

$$h = 1$$

Hal ini menandakan bahwa data ke-1 berpotensi menjadi pusat *cluster* jika memenuhi kriteria yang ditentukan. Selanjutnya langkah awal untuk menentukan pusat *cluster* dan mengurangi potensinya terhadap titik – titik disekitarnya adalah dengan menginisialisasi variabel $V_j = X_{hj}$, $C = 0$, *kondisi* = 1, dan $Z = M$, sehingga didapatkan hasil sebagai berikut:

$$V_j = (0,8192; 0,3750; 0,3500; 0,3446; 1)$$

$$C = 0$$

$$kondisi = 1$$

$$Z = 2,5813$$

Langkah selanjutnya adalah melakukan perulangan, dimana ketika *kondisi* bernilai tidak sama dengan nol dan nilai Z juga tidak bernilai nol. Proses awal pada iterasi adalah nilai *kondisi* diset menjadi nol dan mencari nilai *rasio* dengan cara membagi Z dengan M , sehingga didapatkan hasil sebagai berikut:

*Iterasi ke-1

$$kondisi = 0$$

$$rasio = Z / M = 2,5813 / 2,5813 = 1$$

$$rasio > ar = YA$$



Karena nilai $rasio > ar$, maka masuk $kondisi = 1$ yang artinya calon pusat $cluster$ diterima sebagai pusat $cluster$. Sehingga didapatkan hasil sebagai berikut:

$$C = C + 1 = 0 + 1 = 1$$

$$Center_c = V_j = (0,8192; 0,3750; 0,3500; 0,3446; 1)$$

Kemudian kurangi potensi dari titik – titik di dekat pusat $cluster$ menggunakan Persamaan (2-11), Persamaan (2-12), dan Persamaan (2-13).

Sebagai contoh, untuk pengurangan pada data pertama:

$$S_{11} = \frac{V_1 - X_{11}}{r \times q} = \frac{0,8192 - 0,8192}{1,3 \times 1,25} = 0$$

$$S_{12} = \frac{V_2 - X_{12}}{r \times q} = \frac{0,3750 - 0,3750}{1,3 \times 1,25} = 0$$

$$S_{13} = \frac{V_3 - X_{13}}{r \times q} = \frac{0,3500 - 0,3500}{1,3 \times 1,25} = 0$$

$$S_{14} = \frac{V_4 - X_{14}}{r \times q} = \frac{0,3446 - 0,3446}{1,3 \times 1,25} = 0$$

$$S_{15} = \frac{V_5 - X_{15}}{r \times q} = \frac{1 - 1}{1,3 \times 1,25} = 0$$

Untuk mempermudah perhitungan, dimisalkan

$$ST_i = \sum_{j=1}^5 (S_{ij})^2; i = 1, 2, 3, \dots, 10$$

sehingga:

$$ST_1 = (S_{11})^2 + (S_{12})^2 + (S_{13})^2 + (S_{14})^2 + (S_{15})^2$$

$$ST_1 = (0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 = 0$$

Demikian seterusnya perhitungan tersebut dilakukan untuk $i = 2$ sampai $i = 10$. Sehingga hasil akhir ST_i untuk $i = 1$ sampai $i = 10$ adalah:

ST
0
0,7810
0,2627
0,5703
0,0595
0,1805
0,5840
0,5474
0,2113
0,6938
69

Kemudian dihitung nilai DC_i sebagai nilai pengurang potensial setiap titik data. Sebagai contoh, untuk data pertama:

$$DC_1 = M \times e^{-4(ST_1)} = 2,5813 \times e^{-4(0)} = 2,5812$$

Sehingga hasil akhir untuk DC_i untuk $i = 1$ sampai $i = 10$ adalah:

DC
2,5813
0,1135
0,9027
0,2637
2,0347
1,2538
0,2496
0,2890
1,1084
0,1609

Setelah itu dihitung potensi baru dari titik data dengan Persamaan (2-13). Potensi baru merupakan pengurangan antara potensi lama dengan pengurang potensi (DC_i). Sebagai contoh untuk potensi baru dari data pertama adalah:

$$D_1 = D_1 - DC_1 = 2,5813 - 2,5813 = 0$$

Sehingga hasil akhir untuk potensi baru (D) semua titik data adalah:

D
0,0000
2,0488
0,8494
2,1924
0,4494
0,2621
1,6791
2,1314
0,4841
2,3768

Ketika potensi titik data ada yang bernilai kurang dari nol, maka nilai potensi titik data tersebut diset menjadi nol. Karena untuk potensi baru titik data tidak ada yang bernilai kurang dari nol, maka tidak ada perubahan dan potensi tersebut dijadikan acuan pada proses selanjutnya.

Tentukan potensi titik tertinggi (Z) dan letak titik data tersebut (h) berdasarkan potensi baru dari titik data. Sehingga didapatkan hasil sebagai berikut:

$$Z = 2,3768$$

$$h = 10$$

Hal ini menandakan bahwa data ke-10 berpotensi menjadi pusat *cluster* baru dan akhir dari iterasi pertama. Karena nilai Z tidak samadengan nol dan *kondisi* samadengan satu yang artinya juga tidak samadengan nol, maka perulangan dilanjutkan ke iterasi ke-2, yaitu:

*Iterasi ke-2

$$kondisi = 0$$

$$rasio = Z / M = 2,3768 / 2,5813 = 0,9208$$

$$rasio > ar = YA$$

Karena nilai $rasio > ar$, maka masuk *kondisi* = 1 yang artinya calon pusat *cluster* diterima sebagai pusat *cluster*. Sehingga didapatkan hasil sebagai berikut:

$$C = C + 1 = 1 + 1 = 2$$

$$Center_c = V_j = (0; 0; 0,4500; 0,2424; 0)$$

Kemudian kurangi potensi dari titik – titik di dekat pusat *cluster* dengan cara yang sama pada iterasi ke-1. Sebagai contoh, untuk pengurangan pada data pertama:

$$S_{11} = \frac{V_1 - X_{11}}{r \times q} = \frac{0 - 0,8912}{1,3 \times 1,25} = -0,5041$$

$$S_{12} = \frac{V_2 - X_{12}}{r \times q} = \frac{0 - 0,3750}{1,3 \times 1,25} = -0,2307$$

$$S_{13} = \frac{V_3 - X_{13}}{r \times q} = \frac{0 - 0,3500}{1,3 \times 1,25} = 0,0615$$

$$S_{14} = \frac{V_4 - X_{14}}{r \times q} = \frac{0 - 0,3446}{1,3 \times 1,25} = -0,0629$$

$$S_{15} = \frac{V_5 - X_{15}}{r \times q} = \frac{0 - 1}{1,3 \times 1,25} = -0,6154$$



Untuk mempermudah perhitungan, dimisalkan

$$ST_i = \sum_{j=1}^5 (S_{ij})^2; i = 1, 2, 3, \dots, 10$$

sehingga:

$$\begin{aligned} ST_1 &= (S_{11})^2 + (S_{12})^2 + (S_{13})^2 + (S_{14})^2 + (S_{15})^2 \\ &= (-0,5041)^2 + (-0,2307)^2 + (0,0615)^2 + (-0,0629)^2 + (-0,6154)^2 \\ &= 0,6938 \end{aligned}$$

Demikian seterusnya perhitungan tersebut dilakukan untuk $i = 2$ sampai $i = 10$. Sehingga hasil akhir ST_i untuk $i = 1$ sampai $i = 10$ adalah:

ST
0,6938
0,1523
1,1210
0,1381
0,7630
0,9487
0,1484
0,1881
1,1486
0

Kemudian dihitung nilai DC_i sebagai nilai pengurang potensial setiap titik data. Sebagai contoh, untuk data pertama:

$$DC_1 = M \times e^{-4(ST_1)} = 2,5813 \times e^{-4(0,6938)} = 0,1609$$

Sehingga hasil akhir untuk DC_i untuk $i = 1$ sampai $i = 10$ adalah:

ST
0,1609
1,4035
0,0291
1,4855
0,1220
0,0581
1,4258
1,2162
0,0261
2,5813

Setelah itu dihitung potensi baru dari titik data dengan Persamaan (2-13). Potensi baru merupakan pengurangan antara potensi lama dengan pengurang potensi (DC_i). Sebagai contoh untuk potensi baru dari data pertama adalah:



$$D_1 = D_1 - DC_1 = 0 - 0,1609 = -0,1609$$

Sehingga hasil akhir untuk potensi baru (D) semua titik data adalah:

D
-0,1609
0,6453
0,8203
0,7069
0,3274
0,2041
0,8241
0,9152
0,4580
-0,2044

Ketika potensi titik data ada yang bernilai kurang dari nol, maka nilai potensi titik data tersebut diset menjadi nol. Sehingga potensi baru menjadi sebagai berikut:

D
0
0,6453
0,8203
0,7069
0,3274
0,2041
0,8241
0,9152
0,4580
0

Tentukan potensi titik tertinggi (Z) dan letak titik data tersebut (h) berdasarkan potensi baru dari titik data. Sehingga didapatkan hasil sebagai berikut:

$$Z = 0,9152$$

$$h = 8$$

Hal ini menandakan bahwa data ke-8 berpotensi menjadi pusat *cluster* baru dan akhir dari iterasi kedua. Karena nilai Z tidak samadengan nol dan *kondisi* samadengan satu yang artinya juga tidak samadengan nol, maka perulangan dilanjutkan ke iterasi ke-3, yaitu:



*Iterasi ke-3

$$kondisi = 0$$

$$rasio = Z / M = 0,9152 / 2,5813 = 0,3546$$

$$rasio > ar = \text{TIDAK}$$

$$rasio > rr = \text{YA}$$

Karena nilai $rasio < ar$ dan $rasio > rr$, maka calon pusat *cluster* baru akan diterima sebagai pusat *cluster* jika keberadaannya cukup jauh dengan pusat *cluster* yang telah terpilih sebelumnya. Untuk itu dikerjakan langkah sebagai berikut:

$$Md = -1$$

$$Vj = Xhj = (0,2782; 0,3750; 0; 0,5186; 0)$$

Kerjakan langkah selanjutnya menggunakan Persamaan (2-8) dan Persamaan (2-9) untuk $i = 1$ sampai $i = C$, dimana nilai C adalah 2. Misalkan akan dihitung jarak calon pusat *cluster* (*center*) terhadap pusat *cluster* pertama ($center_1$) adalah sebagai berikut:

$$G_{11} = \frac{V_1 - center_{11}}{r} = \frac{0,2782 - 0,8192}{1,3} = -0,4161$$

$$G_{12} = \frac{V_2 - center_{12}}{r} = \frac{0,3750 - 0,3750}{1,3} = 0$$

$$G_{13} = \frac{V_3 - center_{13}}{r} = \frac{0 - 0,3500}{1,3} = -0,2692$$

$$G_{14} = \frac{V_4 - center_{14}}{r} = \frac{0,5186 - 0,3466}{1,3} = 0,1338$$

$$G_{15} = \frac{V_5 - center_{15}}{r} = \frac{0 - 1}{1,3} = -0,7692$$

$$Sd_1 = (-0,4161)^2 + (0)^2 + (-0,2692)^2 + (0,1338)^2 + (-0,7692)^2$$

$$Sd_1 = 0,8553$$

Jika $Md < 0$ atau $Sd < Md$, maka $Md = Sd$ sehingga nilai $Md = Sd = 0,8533$. Untuk $i = 2$ juga dilakukan perhitungan dengan cara yang sama sehingga didapatkan nilai $Sd_2 = 0,2940$. Karena nilai $Sd_2 < Md$, maka $Md = Sd = 0,2940$.

Setelah itu dicari nilai Smd menggunakan Persamaan (2-10), sehingga didapatkan hasil sebagai berikut:

$$Smd = \sqrt{Md} = \sqrt{0,2940} = 0,5422$$

Hal ini menandakan bahwa jarak terdekat data ke-8 dengan pusat *cluster* adalah 0,5422. Untuk menentukan apakah data ke-8 diterima sebagai pusat *cluster* adalah dengan cara melihat hasil penjumlahan antara *rasio* dengan *Smd*. Jika nilainya lebih besar samadengan 1, maka data tersebut diterima sebagai pusat *cluster* (*kondisi* = 1). Namun jika nilainya kurang dari 1, maka data tidak akan diterima sebagai pusat *cluster* dan potensinya diset menjadi nol (*kondisi* = 2).

$$rasio + Smd = 0,3546 + 0,5422 = 0,8968$$

Hasil ini menunjukkan bahwa masuk *kondisi* = 2, sehingga calon pusat *cluster* (data ke-8) tidak diterima sebagai pusat *cluster* dan potensinya diset menjadi nol. Berikut hasil potensi baru setelah potensi data ke-8 diset 0:

0
0,6453
0,8203
0,7069
0,3274
0,2041
0,8241
0
0,4580
0

Setelah itu tentukan potensi titik tertinggi dan letak titik data tersebut. Sehingga didapatkan hasil sebagai berikut:

$$Z = 0,9152$$

$$h = 7$$

Hal ini menandakan bahwa data ke-7 berpotensi menjadi pusat *cluster* baru dan akhir dari iterasi ketiga. Karena nilai *Z* tidak samadengan nol dan *kondisi* samadengan satu yang artinya juga tidak samadengan nol, maka perulangan dilanjutkan ke iterasi ke-4, yaitu:

*Iterasi ke-4

$$kondisi = 0$$

$$rasio = Z / M = 0,8241 / 2,5813 = 0,3192$$

$$rasio > ar = \text{TIDAK}$$

$$rasio > rr = \text{YA}$$

$$Smd = 0,4815$$



$rasio + Smd = 0,8008 < 1, kondisi = 2$

Data ke-7 ditolak sebagai pusat cluster

Potensi baru:

0
0,6453
0,8203
0,7069
0,3274
0,2041
0
0
0,4580
0

$Z = 0,8203$

$h = 3$

Untuk iterasi selanjutnya dilakukan perhitungan dengan langkah – langkah yang sama pada proses perhitungan sebelumnya sehingga didapatkan hasil sebagai berikut:

*Iterasi ke-5

 $kondisi = 0$

$rasio = Z / M = 0,8203 / 2,5813 = 0,3178$

$rasio > ar = \text{TIDAK}$

$rasio > rr = \text{YA}$

$Smd = 0,4815$

$rasio + Smd = 0,9584 < 1, kondisi = 2$

Data ke-3 ditolak sebagai pusat cluster

Potensi baru:

0
0,6453
0
0,7069
0,3274
0,2041
0
0
0,4580
0

$Z = 0,7069$



$h = 4$

*Iterasi ke-6

kondisi = 0

rasio = $Z / M = 0,7069 / 2,5813 = 0,2738$

rasio > *ar* = TIDAK

rasio > *rr* = YA

Smd = 0,4646

rasio + *Smd* = 0,7384 < 1, *kondisi* = 2

Data ke-4 ditolak sebagai pusat *cluster*

Potensi baru:

0
 0,6453
 0
 0
 0,3274
 0,2041
 0
 0
 0,4580
 0

$Z = 0,6453$

$h = 2$

*Iterasi ke-7

kondisi = 0

rasio = $Z / M = 0,6453 / 2,5813 = 0,2499$

rasio > *ar* = TIDAK

rasio > *rr* = YA

Smd = 0,4879

rasio + *Smd* = 0,7378 < 1, *kondisi* = 2

Data ke-2 ditolak sebagai pusat *cluster*

Potensi baru:

0
 0
 0
 0
 0,3274
 0,2041



0
0
0,4580
0

$Z = 0,4580$

$h = 9$

*Iterasi ke-8

 $kondisi = 0$

$rasio = Z / M = 0,4580 / 2,5813 = 0,1774$

$rasio > ar =$ TIDAK

$rasio > rr =$ YA

$Smd = 0,5747$

$rasio + Smd = 0,7521 < 1$, $kondisi = 2$

Data ke-9 ditolak sebagai pusat cluster

Potensi baru:

0
0
0
0
0,3274
0,2041

$Z = 0,3274$

$h = 5$

*Iterasi ke-9

 $kondisi = 0$

$rasio = Z / M = 0,3274 / 2,5813 = 0,1268$

$rasio > ar =$ TIDAK

$rasio > rr =$ TIDAK

$Z = 0,3274$

$h = 5$

$kondisi = 0$ dan $Z = 0,3274$ maka iterasi dihentikan

Dari proses perhitungan data latih menggunakan *subtractive clustering*, iterasi terakhir adalah iterasi ke-9. Proses perhitungan ini menghasilkan 2 buah pusat *cluster* (*center*), yaitu:

IPK	SKS	P	KE	NK
0,8192	0,375	0,35	0,3446	1
0	0	0,45	0,2424	0

Pusat *cluster* dalam bentuk ternormalisasi tersebut harus dikembalikan pada kondisi semula, yaitu sebelum dilakukan proses normalisasi. Sebagai contoh untuk mengembalikan pusat *cluster* pertama adalah sebagai berikut:

$$center_{11} = center_{11} \times (xMax_1 - xMin_1) + xMin_1$$

$$center_{11} = 0,8192 \times (3,697 - 3,05) + 3,05 = 3,58$$

$$center_{12} = center_{12} \times (xMax_2 - xMin_2) + xMin_2$$

$$center_{11} = 0,375 \times (109 - 101) + 101 = 104$$

$$center_{13} = center_{13} \times (xMax_3 - xMin_3) + xMin_3$$

$$center_{11} = 0,35 \times (20 - 0) + 0 = 7$$

$$center_{14} = center_{14} \times (xMax_4 - xMin_4) + xMin_4$$

$$center_{11} = 0,3446 \times (7427433,333 - 1440496,667)$$

$$center_{11} = +1440496,667$$

$$center_{11} = 3503752,667$$

$$center_{15} = center_{15} \times (xMax_5 - xMin_5) + xMin_5$$

$$center_{11} = 1 \times (1 - 0) + 0 = 1$$

Berikut hasil proses pengembalian pusat *cluster* selengkapnya:

$$3,58 \quad 104 \quad 7 \quad 3503752,667 \quad 1$$

$$3,05 \quad 101 \quad 9 \quad 2891906,667 \quad 0$$

Langkah selanjutnya adalah menghitung nilai sigma menggunakan Persamaan (2-15). Contoh perhitungan untuk menghitung nilai sigma *cluster* adalah sebagai berikut:

$$\sigma_1 = \frac{r \times (xMax_1 - xMin_1)}{\sqrt{8}} = \frac{1,3 \times (3,697 - 3,05)}{\sqrt{8}} = 0,2974$$

$$\sigma_2 = \frac{r \times (xMax_2 - xMin_2)}{\sqrt{8}} = \frac{1,3 \times (109 - 101)}{\sqrt{8}} = 3,6769$$

$$\sigma_3 = \frac{r \times (xMax_3 - xMin_3)}{\sqrt{8}} = \frac{1,3 \times (20 - 0)}{\sqrt{8}} = 9,1923$$

$$\begin{aligned} \sigma_4 &= \frac{r \times (xMax_4 - xMin_4)}{\sqrt{8}} \\ &= \frac{1,3 \times (7427433,333 - 1440496,667)}{\sqrt{8}} = 2751712,285 \end{aligned}$$

$$\sigma_5 = \frac{r \times (xMax_5 - xMin_5)}{\sqrt{8}} = \frac{1,3 \times (31 - 0)}{\sqrt{8}} = 0,4596$$

Sigma cluster:

0,2974 3,6770 9,1924 2751712,2851 0,4596

Langkah terakhir adalah menentukan masing – masing titik data masuk ke dalam *cluster* mana berdasarkan derajat keanggotaannya menggunakan fungsi *gaussian* pada Persamaan (2-22). Sebagai contoh untuk perhitungan derajat keanggotaan pada data pertama terhadap pusat *cluster* pertama adalah sebagai berikut:

$$\mu_{11} = e^{-\left(\left(\frac{x_{11}-center_{11}}{\sqrt{2} \times \sigma_{11}} \right)^2 + \left(\frac{x_{12}-center_{12}}{\sqrt{2} \times \sigma_{11}} \right)^2 + \left(\frac{x_{13}-center_{13}}{\sqrt{2} \times \sigma_{11}} \right)^2 + \left(\frac{x_{14}-center_{14}}{\sqrt{2} \times \sigma_{14}} \right)^2 + \left(\frac{x_{15}-center_{15}}{\sqrt{2} \times \sigma_{15}} \right)^2 \right)}$$

$$\mu_{11} = e^{-\left(\left(\frac{3,58-3,58}{\sqrt{2} \times 0,30} \right)^2 + \left(\frac{104-104}{\sqrt{2} \times 3,68} \right)^2 + \left(\frac{7-7}{\sqrt{2} \times 9,19} \right)^2 + \left(\frac{3503752,667-3503752,667}{\sqrt{2} \times 2751712,29} \right)^2 + \left(\frac{1-1}{\sqrt{2} \times 0,46} \right)^2 \right)}$$

$$\mu_{11} = e^{-\left((0)^2 + (0)^2 + (0)^2 + (0)^2 + (0)^2 \right)} = e^{-(0)} = 1$$

Sehingga hasil akhir untuk derajat keanggotaan pada *cluster* pertama (μ_{C1}) adalah sebagai berikut:

μ_{C1}
1,0000
0,0076
0,1937
0,0283
0,6895
0,3236
0,0260
0,0327
0,2669
0,0131

Dengan langkah pengerjaan yang sama dalam mendapatkan μ_{C1} , maka hasil akhir untuk derajat keanggotaan pada *cluster* kedua (μ_{C2}) adalah sebagai berikut:



$\mu C2$
 0,0131
 0,3859
 0,0009
 0,4218
 0,0085
 0,0027
 0,3956
 0,3086
 0,0008
 1,0000

Penentuan suatu titik data masuk ke *cluster* mana dapat dilihat dari besarnya nilai derajat keanggotaan. Sehingga suatu titik data dikatakan masuk ke dalam *cluster* yang derajat keanggotaan titik data pada *cluster* tersebut lebih besar dari nilai derajat keanggotaan pada *cluster* lainnya. Hasil selengkapnya untuk proses pengklasteran tersaji pada Tabel 3.1 sebagai berikut:

Tabel 3.1 Hasil *subtractive clustering*

Data	Atribut					$\mu C1$	$\mu C2$	C1	C2
	IPK	S	P	KE	NK				
1	3,58	104	7	3503752,667	1	1,0000	0,0131	v	
2	3,062	104	18	1440496,667	0	0,0076	0,3859		v
3	3,56	107	0	7427433,333	1	0,1937	0,0009	v	
4	3,226	104	16	3886919,167	0	0,0283	0,4218		v
5	3,548	104	0	4576998,75	1	0,6895	0,0085	v	
6	3,697	104	20	4382001,917	1	0,3236	0,0027	v	
7	3,18	104	0	2344500	0	0,0260	0,3956		v
8	3,23	104	0	4545230,5	0	0,0327	0,3086		v
9	3,66	109	3	1498516,667	1	0,2669	0,0008	v	
10	3,05	101	9	2891906,667	0	0,0131	1,0000		v

2. Proses Perhitungan Varian

Proses perhitungan varian digunakan untuk mengetahui baik tidaknya hasil proses *clustering* berdasarkan kepadatan sebaran datanya. Langkah pertama adalah menghitung nilai varian tiap *cluster* menggunakan Persamaan (2-17). Sebagai contoh untuk menghitung nilai varian pada *cluster* pertama adalah sebagai berikut:



$$\begin{aligned}
 V_1^2 &= \frac{1}{nc - 1} \sum_{i=1}^{nc} (di - \bar{d})^2 \\
 &= \frac{1}{5 - 1} \left((3,58 - 3,609)^2 + \dots + (3,66 - 3,609)^2 + \right. \\
 &\quad (104 - 105,6)^2 + \dots + (109 - 105,6)^2 + \\
 &\quad (7 - 6)^2 + \dots + (3 - 6)^2 + (3503752,7 - 4277740,7)^2 \\
 &\quad + \dots + (1498516,7 - 4277740,7)^2 + \\
 &\quad \left. (1 - 1)^2 + \dots + (1 - 1)^2 \right) \\
 &= 4586033292441,22
 \end{aligned}$$

Dengan langkah perhitungan yang sama pada *cluster* kedua, maka didapatkan hasil untuk nilai varian pada kedua *cluster* adalah sebagai berikut:

$$\begin{aligned}
 V_1^2 &= 4586033292441,22 \\
 V_2^2 &= 1511349865164,12
 \end{aligned}$$

Setelah varian pada masing – masing *cluster* diketahui, maka nilai varian *within cluster* dapat dihitung dengan menggunakan Persamaan (2-16).

$$\begin{aligned}
 V_w &= \frac{1}{N - k} \sum_{i=1}^k (ni - 1) \times V_i^2 \\
 V_w &= \frac{1}{10 - 2} \times \left(((5 - 1) \times 4586033292441,22) \right. \\
 &\quad \left. + ((5 - 1) \times 1511349865164,12) \right) \\
 V_w &= 3048691578802,67
 \end{aligned}$$

Perhitungan nilai varian *between cluster* menggunakan Persamaan (2-18) sebagai berikut:

$$\begin{aligned}
 V_b &= \frac{1}{k - 1} \sum_{i=1}^k ni (di - \bar{d})^2 \\
 V_b &= \frac{1}{2 - 1} \times \left(\left((5 \times (3,58 - 3,6)^2) + \dots + (5 \times (1 - 1)^2) \right) + \right. \\
 &\quad \left. \left((5 \times (3,06 - 3,15)^2) + \dots + (5 \times (0 - 0)^2) \right) \right) \\
 V_b &= 1,21948E + 14
 \end{aligned}$$

Batasan varian dapat diketahui dengan menggunakan Persamaan (2-19), sehingga didapatkan proses perhitungan sebagai berikut:

$$\begin{aligned}
 V &= \frac{V_w}{V_b} \\
 &= \frac{3048691578802,670}{1,21948E + 14} \\
 &= 0,025
 \end{aligned}$$



Nilai batasan varian tersebut nantinya digunakan sebagai bahan pertimbangan untuk menentukan hasil proses *clustering* mana yang akan dipakai untuk proses ekstraksi aturan *fuzzy*.

3. Proses Ekstraksi Aturan *Fuzzy*

Proses ekstraksi aturan *fuzzy* merupakan proses untuk membentuk aturan beserta koefisien *outputnya* yang nantinya akan digunakan pada *fuzzy inference system* model sugeno orde-satu. Langkah pertama adalah mencari nilai d_{ij}^k yang terbentuk dari derajat keanggotaan data pada masing – masing *cluster* menggunakan Persamaan (2-30). Sebagai contoh untuk menghitung d_{ij}^k pertama adalah sebagai berikut:

$$d_{11}^1 = X_{11} \times \mu_{11} = 3,58 \times 1 = 3,58$$

$$d_{12}^1 = X_{12} \times \mu_{12} = 104 \times 1 = 104$$

$$d_{13}^1 = X_{13} \times \mu_{13} = 7 \times 1 = 7$$

$$d_{14}^1 = X_{14} \times \mu_{14} = 3503752,667 \times 1 = 3503752,667$$

$$d_{15}^1 = X_{15} \times \mu_{15} = 1 \times 1 = 1$$

Hasil dari d_{ij}^k adalah sebagai berikut:

d_{ij}^k untuk aturan pertama

3,5800	104,0000	7,0000	3503752,6667	1,0000
0,0233	0,7904	0,1368	10947,7747	0,0076
0,6896	20,7259	0,0000	1438693,8367	0,1937
0,0913	2,9432	0,4528	109999,8124	0,0283
2,4463	71,7080	0,0000	3155840,6381	0,6895
1,1963	33,6544	6,4720	1418015,8202	0,3236
0,0827	2,7040	0,0000	60957,0000	0,0260
0,1056	3,4008	0,0000	148629,0374	0,0327
0,9769	29,0921	0,8007	399954,0983	0,2669
0,0400	1,3231	0,1179	37883,9773	0,0131

d_{ij}^k untuk aturan kedua

0,0469	1,3624	0,0917	45899,1599	0,0131
1,1816	40,1336	6,9462	555887,6637	0,3859
0,0032	0,0963	0,0000	6684,6900	0,0009
1,3607	43,8672	6,7488	1639502,5045	0,4218
0,0302	0,8840	0,0000	38904,4894	0,0085
0,0100	0,2808	0,0540	11831,4052	0,0027

1,2580	41,1424	0,0000	927484,2000	0,3956
0,9968	32,0944	0,0000	1402658,1323	0,3086
0,0029	0,0872	0,0024	1198,8133	0,0008
3,0500	101,0000	9,0000	2891906,6667	1,0000

Setelah d_{ij}^k diketahui, langkah selanjutnya adalah menormalisasi d_{ij}^k menggunakan Persamaan (2-31) dan Persamaan (2-32). Sebagai contoh untuk menghitung d_{ij}^k ternormalisasi pertama:

$$d_{11}^1 = \frac{d_{11}^1}{\sum_{k=1}^2 \mu_{1i}} = \frac{3,58}{(1 + 0,0131)} = 3,5337$$

$$d_{12}^1 = \frac{d_{12}^1}{\sum_{k=1}^2 \mu_{1i}} = \frac{104}{(1 + 0,0131)} = 102,6552$$

$$d_{13}^1 = \frac{d_{13}^1}{\sum_{k=1}^2 \mu_{1i}} = \frac{7}{(1 + 0,0131)} = 6,9095$$

$$d_{14}^1 = \frac{d_{14}^1}{\sum_{k=1}^2 \mu_{1i}} = \frac{3503752,667}{(1 + 0,0131)} = 3458447,011$$

$$d_{15}^1 = \frac{d_{15}^1}{\sum_{k=1}^2 \mu_{1i}} = \frac{1}{(1 + 0,0131)} = 0,9781$$

Setelah proses normalisasi d_{ij}^k selesai, langkah selanjutnya adalah membentuk matriks U menggunakan Persamaan (2-33). Hasil pembentukan matriks U adalah sebagai berikut:

Matriks U , kolom 1 sampai 5

3,5337	102,6552	6,9095	3458447,0108	0,9871
0,0591	2,0086	0,3476	27821,5366	0,0193
3,5435	106,5051	0,0000	7393082,4084	0,9954
0,2028	6,5390	1,0060	244389,7188	0,0629
3,5048	102,7335	0,0000	4521261,6592	0,9878
3,6664	103,1394	19,8345	4345742,6302	0,9917
0,1961	6,4137	0,0000	144584,9146	0,0617
0,3095	9,9643	0,0000	435479,1601	0,0958
3,6491	108,6743	2,9910	1494038,4697	0,9970
0,0394	1,3060	0,1164	37394,1144	0,0129



Matriks U , kolom 6 sampai 10

0,0463	1,3448	0,0905	45305,6558	0,0129
3,0029	101,9914	17,6524	1412675,1300	0,9807
0,0165	0,4949	0,0000	34350,9250	0,0046
3,0232	97,4610	14,9940	3642529,4479	0,9371
0,0432	1,2665	0,0000	55737,0908	0,0122
0,0306	0,8606	0,1655	36259,2865	0,0083
2,9839	97,5863	0,0000	2199915,0854	0,9383
2,9205	94,0357	0,0000	4109751,3399	0,9042
0,0109	0,3257	0,0090	4478,1970	0,0030
3,0106	99,6940	8,8836	2854512,5522	0,9871

Setelah pembentukan matriks U selesai, maka tahapan selanjutnya adalah membentuk matriks U^T seperti berikut:

Matriks U^T , kolom 1 sampai 5

3,5337	0,0591	3,5435	0,2028	3,5048
102,6552	2,0086	106,5051	6,5390	102,7335
6,9095	0,3476	0,0000	1,0060	0,0000
3458447,0108	27821,5366	7393082,4084	244389,7188	4521261,6592
0,9871	0,0193	0,9954	0,0629	0,9878
0,0463	3,0029	0,0165	3,0232	0,0432
1,3448	101,9914	0,4949	97,4610	1,2665
0,0905	17,6524	0,0000	14,9940	0,0000
45305,6558	1412675,1300	34350,9250	3642529,4479	55737,0908
0,0129	0,9807	0,0046	0,9371	0,0122

Matriks U^T , kolom 6 sampai 10

3,6664	0,1961	0,3095	3,6491	0,0394
103,1394	6,4137	9,9643	108,6743	1,3060
19,8345	0,0000	0,0000	2,9910	0,1164
4345742,6302	144584,9146	435479,1601	1494038,4697	37394,1144
0,9917	0,0617	0,0958	0,9970	0,0129
0,0306	2,9839	2,9205	0,0109	3,0106
0,8606	97,5863	94,0357	0,3257	99,6940
0,1655	0,0000	0,0000	0,0090	8,8836
36259,2865	2199915,0854	4109751,3399	4478,1970	2854512,5522
0,0083	0,9383	0,9042	0,0030	0,9871

Setelah itu dilakukan operasi perkalian matriks antara matriks U^T dengan U , sehingga didapatkan hasil sebagai berikut:



Hasil operasi perkalian U^T dengan U , kolom 1 sampai 5

64,2660	1880,7667	108,2812	75865802,6878	17,8077
1880,7667	55072,3750	3087,4919	2224464365,0783	521,2754
108,2812	3087,4919	451,2414	114820358,3783	29,5441
75865802,68	2224464365,1	114820358,38	108450405760032,0	21105204,58
17,8077	521,2754	29,5441	21105204,5801	4,9359
2,9239	93,2586	5,3949	3264659,1970	0,8968
93,2586	2980,2626	172,4396	103706020,9270	28,6637
5,3949	172,4396	26,1893	5533308,9001	1,6610
3264659,19	103706020,9	5533308,90	3970939632154,7	997037,042
0,8968	28,6637	1,6610	997037,0427	0,2757

Hasil operasi perkalian U^T dengan U , kolom 6 sampai 10

2,9239	93,2586	5,3949	3264659,1970	0,8968
93,2586	2980,2626	172,4396	103706020,9270	28,6637
5,3949	172,4396	26,1893	5533308,9001	1,6610
3264659,197	103706020,927	5533308,9001	3970939632154,71	997037,0427
0,8968	28,6637	1,6610	997037,0427	0,2757
44,6586	1467,0180	125,0910	42420921,6557	14,1917
1467,0180	48210,0997	4147,6289	1384989415,067	466,3971
125,0910	4147,6289	615,3800	104921687,4882	40,1340
42420921,6	1384989415,	104921687,4	45149270173112	13398477,6
557	0678	882	,9000	678
14,1917	466,3971	40,1340	13398477,6678	4,5127

Dari hasil perkalian tersebut dilakukan operasi inverse sehingga didapatkan hasil sebagai berikut:

Hasil operasi inverse, kolom 1 sampai 5

2115,372861	-33,05663377	-16,36001896	1,42033E-05	-4107,307259
-33,05663377	0,583019336	0,264460942	-1,99987E-07	57,01861435
-16,36001896	0,264460942	0,131466415	-1,04575E-07	30,78334338
1,42033E-05	-1,99987E-07	-1,04575E-07	1,59166E-13	-3,02028E-05
-4107,307259	57,01861435	30,78334338	-3,02028E-05	8749,182409
166,2752766	-2,249766108	-1,321046161	1,36677E-06	-361,7970639
5,493245085	-0,09839415	-0,040851066	2,82057E-08	-9,268607219
-0,161358695	0,002328084	0,00082018	-1,25711E-09	0,335158006
3,58502E-06	-7,16815E-08	-2,64788E-08	9,23392E-15	-5,19728E-06
-1056,471155	16,7661778	8,113392757	-6,93351E-06	2023,08816



Hasil operasi inverse, kolom 6 sampai 10

166,2752766	5,493245085	-0,161358695	3,58502E-06	-1056,471155
-2,249766108	-0,09839415	0,002328084	-7,16815E-08	16,7661778
-1,321046161	-0,040851066	0,00082018	-2,64788E-08	8,113392757
1,36677E-06	2,82057E-08	-1,25711E-09	9,23392E-15	-6,93351E-06
-361,7970639	-9,268607218	0,335158006	-5,19728E-06	2023,08816
474,3662957	-17,38750471	0,778625532	-2,13126E-05	365,1233482
-17,38750471	0,848380169	-0,031206322	8,36939E-07	-35,10032
0,778625532	-0,031206322	0,005930168	-2,52617E-08	0,795641857
-2,13126E-05	8,36939E-07	-2,52617E-08	1,23582E-12	-2,28515E-05
365,1233482	-35,10032	0,795641857	-2,28515E-05	2518,85053

Hasil operasi inverse sebelumnya selanjutnya dikalikan dengan U^T , sehingga didapatkan hasil sebagai berikut:

Hasil perkalian $(U^T \times U)^{-1} \times U^T$, kolom 1 sampai 5

-34,868987	-0,20358679	-7,415046	0,328294	26,323442
0,43174124	-0,00513184	0,2229389	0,004349062	-0,562622
0,25644841	0,001421618	0,0566306	-0,002885521	-0,245312
-3,12E-07	-4,7665E-09	1,485E-07	6,05687E-09	1,296E-07
80,2665262	1,314384018	2,4476082	-1,687895483	-34,48194
-2,4986281	-9,30552031	-0,26609	10,96558203	2,1556549
-0,1053993	0,4590441	-0,031219	-0,340186674	0,0585132
0,00469747	0,006455864	0,0006756	0,059115836	0,0014367
-7,403E-08	2,10182E-07	-3,57E-08	-2,47323E-07	3,815E-08
18,1900016	-18,6498271	4,0189411	1,100372893	-12,43947

Hasil perkalian $(U^T \times U)^{-1} \times U^T$, kolom 6 sampai 10

11,464043	0,36922951	-0,426687	4,441629482	-0,012331
-0,160652	0,00350192	-0,002574	0,067066465	0,001383
-0,033682	-0,0024897	0,0036968	-0,033921883	9,42E-05
1,219E-07	8,3532E-09	-8,91E-09	-8,89238E-08	2,47E-10
-24,53098	-1,7533849	1,8561673	-22,32928744	-0,101204
0,4533134	10,4589576	-12,12222	0,159388305	-0,000442
0,0454828	-0,3078835	0,5433412	0,01607247	-0,337765
-0,004687	-0,0272017	-0,040089	-0,000404688	1,12E-06
3,142E-08	-7,201E-07	7,761E-07	2,14103E-08	-5,94E-11
-5,949835	1,52720241	-19,7922	-2,134047314	35,12886

Langkah terakhir adalah mengalikan hasil dari perkalian matriks sebelumnya dengan Y , dimana Y adalah target *output* dari data latih seperti berikut:

1
0
1
0
1
1
0
0
1
0

Dengan demikian didapatkan koefisien *output* sebagai berikut:

KoefisienOutput

-0,0549188
-0,0015282
0,00016264
-9,801E-10
1,37193322
0,00363881
-0,01655
0,00171826
-1,879E-08
1,68558903

Dari hasil tersebut, barulah dibentuk menjadi matriks yang berukuran $r \times m$, dimana r merupakan jumlah aturan dan m panjang atribut. Sebagai hasil akhir untuk koefisien *output* yang siap digunakan pada proses inferensi *fuzzy* adalah sebagai berikut:

-0,054918764	-0,001528	0,0001626	-9,801E-10	1,3719332
0,003638812	-0,01655	0,0017183	-1,879E-08	1,685589

Dari matriks koefisien *output* ini diperoleh parameter – parameter *output* untuk setiap aturan sebagai berikut:

$$Z_1 = (k_{11} \times X_1) + (k_{12} \times X_2) + (k_{13} \times X_3) + (k_{14} \times X_4) + k_{15}$$

$$Z_1 = (-0,0549 \times X_1) + (-0,0015 \times X_2) + (0,0001 \times X_3) + (-9,801E-10 \times X_4) + 1,3719$$

$$Z_2 = (k_{21} \times X_1) + (k_{22} \times X_2) + (k_{23} \times X_3) + (k_{24} \times X_4) + k_{25}$$

$$Z_2 = (0,0036 \times X_1) + (-0,0165 \times X_2) + (0,0017 \times X_3) + (-1,879E-08 \times X_4) + 1,6855$$

Setiap variabel *input* juga akan terbagi menjadi 2 himpunan *fuzzy*.

Aturan – aturan yang terbentuk adalah sebagai berikut:



[R1] : IF (*IPK* is *center*₁₁) and (*SKS* is *center*₁₂) and (*PRESTASI* is *center*₁₃) and (*KE* is *center*₁₄) THEN *NK* = *Z*₁

[R2] : IF (*IPK* is *center*₂₁) and (*SKS* is *center*₂₂) and (*PRESTASI* is *Center*₂₃) and (*KE* is *center*₂₄) THEN *NK* = *Z*₂

3.4.2. Proses Pengujian

Proses pengujian dilakukan terhadap data uji dengan menggunakan aturan yang telah terbentuk pada proses pembangkitan aturan *fuzzy* menggunakan *subtractive clustering*. Langkah awal untuk proses pengujian adalah memasukkan data uji dan aturan yang terbentuk sebagai berikut:

Data uji

IPK = 3,71; *SKS* = 107; *PRESTASI* = 0; *KE* = 1142850

Aturan

[R1] : IF (*IPK* is *center*₁₁) and (*SKS* is *center*₁₂) and (*PRESTASI* is *center*₁₃) and (*KE* is *center*₁₄) THEN *NK* = *Z*₁

[R2] : IF (*IPK* is *center*₂₁) and (*SKS* is *center*₂₂) and (*PRESTASI* is *Center*₂₃) and (*KE* is *center*₂₄) THEN *NK* = *Z*₂

Pusat *cluster* (*center*)

3,58	104	7	3503752,667	1
3,05	101	9	2891906,667	0

Sigma *cluster*

0,2974	3,6770	9,1924	2751712,2851	0,4596
--------	--------	--------	--------------	--------

Parameter koefisien *output*

$$\begin{aligned}
 Z_1 &= (k_{11} \times X_1) + (k_{12} \times X_2) + (k_{13} \times X_3) + (k_{14} \times X_4) + k_{15} \\
 &= (-0,0549 \times X_1) + (-0,0015 \times X_2) + (0,0001 \times X_3) + \\
 &= (-9,801E-10 \times X_4) + 1,3719 \\
 Z_2 &= (k_{21} \times X_1) + (k_{22} \times X_2) + (k_{23} \times X_3) + (k_{24} \times X_4) + k_{25} \\
 &= (0,0036 \times X_1) + (-0,0165 \times X_2) + (0,0017 \times X_3) + \\
 &= (-1,879E-08 \times X_4) + 1,6855
 \end{aligned}$$

Selanjutnya adalah mencari derajat keanggotaan tiap atribut data uji terhadap aturan. Sebagai contoh untuk menghitung derajat keanggotaan pada aturan pertama adalah sebagai berikut:



$$\mu_{11} = e^{-\left(\frac{(X_{11}-center_{11})^2}{\sqrt{2} \times \sigma_1}\right)^2} = e^{-\left(\frac{(3,71-3,58)^2}{\sqrt{2} \times 0,3}\right)^2} = 0,91$$

$$\mu_{12} = e^{-\left(\frac{(X_{12}-center_{12})^2}{\sqrt{2} \times \sigma_2}\right)^2} = e^{-\left(\frac{(107-104)^2}{\sqrt{2} \times 3,68}\right)^2} = 0,71$$

$$\mu_{13} = e^{-\left(\frac{(X_{13}-center_{13})^2}{\sqrt{2} \times \sigma_3}\right)^2} = e^{-\left(\frac{(0-7)^2}{\sqrt{2} \times 9,19}\right)^2} = 0,75$$

$$\mu_{14} = e^{-\left(\frac{(X_{14}-center_{14})^2}{\sqrt{2} \times \sigma_4}\right)^2} = e^{-\left(\frac{(1142850-3503752,667)^2}{\sqrt{2} \times 2751712,28}\right)^2} = 0,70$$

Hasil selengkapnya untuk nilai derajat keanggotaan masing – masing aturan adalah sebagai berikut:

$$R1: \quad 0,91 \quad 0,72 \quad 0,75 \quad 0,69$$

$$R2: \quad 0,09 \quad 0,26 \quad 0,62 \quad 0,81$$

Kemudian tahapan selanjutnya adalah mencari *fire strength* (α -predikat) untuk setiap aturan. Apabila digunakan operator *product* sebagai operator pada anteseden, diperoleh:

$$[R1] : \alpha_1 = \mu_{11} \cdot \mu_{12} \cdot \mu_{13} \cdot \mu_{14} = 0,91 \cdot 0,72 \cdot 0,75 \cdot 0,69 = 0,34$$

$$[R2] : \alpha_2 = \mu_{21} \cdot \mu_{22} \cdot \mu_{23} \cdot \mu_{24} = 0,09 \cdot 0,26 \cdot 0,62 \cdot 0,81 = 0,01$$

Menentukan nilai Z untuk setiap aturan merupakan langkah selanjutnya pada proses pengujian. Sebagai contoh untuk menghitung nilai Z pada aturan pertama adalah sebagai berikut:

$$Z_1 = (k_{11} \times X_{11}) + (k_{12} \times X_{12}) + (k_{13} \times X_{13}) + (k_{14} \times X_{14}) + k_{15}$$

$$Z_1 = (-0,0549 \times 3,71) + (-0,0015 \times 107) + (0,0001 \times 0) +$$

$$Z_1 = (-9,801E-10 \times 1142850) + 1,3719$$

$$= 1,0036$$

$$Z_2 = (k_{21} \times X_{21}) + (k_{22} \times X_{22}) + (k_{23} \times X_{23}) + (k_{24} \times X_{14}) + k_{25}$$

$$= (0,0036 \times 3,71) + (-0,0165 \times 107) + (0,0017 \times 0) +$$

$$(-1,879E-08 \times 1142850) + 1,6855$$

$$= -0,0932$$

Langkah terakhir, yaitu menghitung nilai Z total (*defuzzy*) dengan metode *weighed average*, sebagai berikut:

$$\begin{aligned}
 Z &= \frac{(\alpha_1 \times Z_1) + (\alpha_2 \times Z_2)}{\alpha_1 + \alpha_2} \\
 &= \frac{(0,34 \times 1,0035) + (0,01 \times -0,0932)}{0,34 + 0,01} \\
 &= 0,9663
 \end{aligned}$$

Dengan demikian hasil akhir untuk proses pengujian dalam mendapatkan nilai kelayakan dari parameter yang diberikan pada data uji adalah sebesar 0,9663. Hasil ini menyatakan besarnya nilai kelayakan yang menjadi bahan pertimbangan apakah pemohon beasiswa bisa diterima atau tidak pengajuan permohonan beasiswanya. Dalam implementasinya nilai kelayakan nantinya akan diurutkan dari yang terbesar sampai terkecil, sehingga penentuan penerima beasiswa juga ditentukan oleh batasan penerima beasiswa. Ketika batasan penerima telah ditetapkan, maka nilai kelayakan pada urutan teratas sesuai dengan jumlah batasan penerima beasiswa akan berubah menjadi 1 yang artinya pemohon dengan nilai 1 tersebut berhak mendapatkan beasiswa.

3.5 Perancangan Pengujian dan Analisis

Pengujian dan analisis dilakukan untuk mengetahui bagaimana akurasi dari hasil *cluster* yang terbentuk guna membangkitkan aturan *fuzzy* untuk proses sistem inferensi *fuzzy* pada rekomendasi penerima beasiswa. Ada beberapa skenario pengujian yang dilakukan, diantaranya adalah sebagai berikut:

1. Evaluasi pembentukan aturan dari hasil proses *clustering*

Pada skenario pengujian ini akan dilakukan percobaan dengan menggunakan beberapa jumlah data latih. Nantinya nilai jari – jari dan nilai *reject ratio* yang digunakan adalah bervariasi, yaitu antara 0,5 sampai 3 untuk nilai jari – jari serta antara 0,15 sampai 0,5 untuk nilai *reject ratio* dan antara 0,5 sampai 0,9 untuk nilai *accept ratio*. Dengan skenario pengujian ini akan diketahui jumlah *cluster* yang terbentuk beserta nilai batasan variannya. Ketika ditemukan nilai batasan varian dengan nilai yang sama – sama terkecil pada beberapa jari – jari sebagai bahan pengujian parameter, maka hasil *cluster* dengan nilai jari – jari terkecil yang akan diambil sebagai bahan ekstraksi aturan *fuzzy*. Meski secara pengelompokan data akan menghasilkan

cluster yang sama, namun nilai jari – jari akan berpengaruh terhadap sigma *cluster* dan nantinya akan berpengaruh juga pada nilai derajat keanggotaan. Sehingga dengan pemilihan jari – jari terkecil pada nilai batasan varian terkecil diharapkan dapat memaksimalkan proses inferensi *fuzzy*. Perancangan tabel pengujian untuk proses pembentukan *cluster* ditunjukkan pada Tabel 3.2.

Tabel 3.2 Tabel pengujian pembentukan aturan

Jumlah Data Latih = x						
Jenis Aturan	Jari – jari	$ar = 0,5$ dan $rr = 0,15$...	$ar = 0,9$ dan $rr = 0,5$	
		Nilai varian	Jumlah <i>cluster</i>	...	Nilai varian	Jumlah <i>cluster</i>
Aturan 1	0,5					
Aturan 2	0,6					
...	...					
Aturan n	3					

2. Evaluasi data uji terhadap aturan terpilih

Pada skenario pengujian ini akan dicari tingkat akurasi sistem dengan membandingkan hasil sistem terhadap data uji menggunakan beberapa aturan yang terpilih sebelumnya. Untuk mengetahui akurasi dari sistem digunakan Persamaan (2-44). Perancangan tabel pengujian untuk mengetahui hasil pengujian data uji terhadap aturan terpilih ditunjukkan pada Tabel 3.3.

Tabel 3.3 Tabel uji akurasi

Jenis Aturan	Jumlah Batasan Penerima	Nilai Kelayakan		Akurasi
		Benar	Salah	
Aturan 1				
Aturan 2				
...				
Aturan n				



BAB IV IMPLEMENTASI

Pada bab implementasi ini berisi mengenai penerapan dari sistem yang telah dikembangkan pada bab metodologi dan perancangan sistem.

3.4 Lingkungan Implementasi

3.3.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah sebuah *notebook* dengan spesifikasi sebagai berikut:

1. *Processor* Intel(R) Core(TM) i5-2450M CPU @ 2,50GHz (4 CPUs), ~2,5GHz
2. *Memory* 4096MB RAM
3. *Harddisk* 750GB
4. *Monitor* 14"
5. *Keyboard*
6. *Mouse pad*

3.3.2. Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem diantaranya adalah sebagai berikut:

1. Sistem operasi Windows 7 Home Premium 64-bit (6.1, Build 7601)
2. Java(TM) SE Development Kit 6 Update 16
3. NetBeans IDE 6.7.1
4. XAMPP 1.7.1

3.5 Implementasi Program

Berdasarkan analisa dan perancangan sistem yang telah dibangun pada subbab 3.3, maka pada subbab implementasi program ini akan dijelaskan bagaimana menerapkan sistem tersebut ke dalam sebuah program yang di dalamnya berisikan *source code* yang membentuk aplikasi sistem.

Pada implementasi program, sistem diimplementasikan ke dalam sebuah *package* yang didalamnya berisi kelas. Secara umum kelas yang dibentuk merupakan kelas untuk mengimplementasikan tampilan antarmuka aplikasi sistem

dan kelas untuk mengoperasikan proses jalannya sistem. Kelas – kelas di dalam *package* yang membentuk program ditunjukkan pada Tabel 4.1.

Tabel 4.1 Kelas – kelas pembentuk aplikasi rekomendasi beasiswa

Nama Kelas	Deskripsi
FormPembelajaran.java	Sebuah kelas yang diimplementasikan untuk tampilan antarmuka proses pembelajaran data latih.
FormHasilPenguajian.java	Merupakan kelas yang diimplementasikan untuk tampilan form perekomendasi data uji yang direkomendasikan mendapatkan beasiswa.
SubtractiveClustering.java	Merupakan kelas yang berisi proses pembelajaran data latih menggunakan <i>Subtractive Clustering</i> .
VarianCluster.java	Kelas yang digunakan untuk memproses nilai batasan varian dari setiap hasil <i>clustering</i> pada proses sebelumnya.
EkstraksiAturanFuzzy.java	Sebuah kelas yang digunakan untuk mencari nilai koefisien <i>output</i> pada setiap aturan yang terbentuk.
SistemInferensiFuzzy.java	Kelas yang digunakan untuk proses pengujian data uji menggunakan <i>fuzzy inference system</i> model Sugeno

4.2.1. Implementasi Proses *Subtractive Clustering*

Proses *subtractive clustering* diterapkan pada *class* *SubtractiveClustering.java*, dimana pada *class* ini terdapat 7 proses utama yang diterapkan pada beberapa *method*. Selain itu ada tambahan *method* lain lagi dalam *class* ini untuk mempermudah implementasi proses *subtractive clustering*. Penjelasan *method* pembentuk *class* *SubtractiveClustering.java* dijelaskan pada Tabel 4.2.

Tabel 4.2 *Method – method* pembentuk kelas *SubtractiveClustering.java*

Nama <i>method</i>	Deskripsi
BacaDataLatih()	<i>Method</i> yang digunakan untuk membaca dan menampung data latih dari <i>database</i> .

Normalisasi ()	Sebuah <i>method</i> yang digunakan untuk proses normalisasi data latih.
potensiAwal ()	Merupakan <i>method</i> yang digunakan untuk mencari potensi awal setiap titik data latih.
findIdxMax ()	<i>Method</i> yang digunakan untuk proses penentuan potensi tertinggi titik data.
tentukanPusatCluster ()	<i>Method</i> yang digunakan untuk menyeleksi apakah suatu titik data dengan potensi tertinggi bisa menjadi pusat <i>cluster</i> atau tidak.
denormalisasi ()	Sebuah <i>method</i> yang digunakan pada proses pengembalian pusat <i>cluster</i> .
hitungSigmaCluster ()	<i>Method</i> untuk menghitung nilai sigma <i>cluster</i> .

1. Implementasi Masukan Data Latih

Proses pertama pada *subtractive clustering* adalah memasukkan data latih sebanyak n buah data. Proses masukan data latih ini diterapkan kedalam sebuah *method* `BacaDataLatih()`. Untuk dapat membaca data latih yang telah disimpan sebelumnya pada sebuah *database*, maka diperlukan sebuah *method* `KoneksiDB()` untuk mengoneksikan antara program dengan *database*. Jumlah data latih yang digunakan pada proses *subtractive clustering* berupa inputan dari *user* yang diinisialisasikan pada variabel `jumlahData`. Penerapan untuk proses masukan data latih dapat dilihat pada *Source code* 4.1.

```
private static void BacaDataLatih(){
    KoneksiDB();
    try {
        String sql =
            "select * from data_latih limit 0, "+jumlahData+"";
        statement.executeQuery(sql);
        ResultSet rs = statement.executeQuery(sql);
        matriksData = new double[jumlahData][jumlahAtribut];
        double ipk=0, sks=0, prestasi=0, ke=0, nk=0;
        int cek=0;
        while(rs.next()){
            ipk=rs.getDouble("ipk");
            sks=rs.getDouble("sks");
            prestasi=rs.getDouble("prestasi");
            ke=rs.getDouble("ke");
            nk=rs.getDouble("nk");
            matriksData[cek][0]=ipk;
            matriksData[cek][1]=sks;
            matriksData[cek][2]=prestasi;
        }
    }
}
```

```

        matriksData[cek][3]=ke;
        matriksData[cek][4]=nk;
        cek++;
    }
    rs.close();
    statement.close();
} catch(Exception e){
    JOptionPane.showMessageDialog(null,e,"Error",
    JOptionPane.ERROR_MESSAGE);
}
}
}

```

Source code 4.1 Listing program proses masukan data latihan

2. Implementasi Normalisasi Data

Proses normalisasi data digunakan agar data latihan berada pada proses normal yang artinya bahwa *range* antara data latihan satu dengan lainnya sama - sama berada pada kisaran 0 sampai 1. Kode program untuk proses normalisasi merujuk pada Persamaan (2-3). Penerapan untuk proses normalisasi data adalah dengan membuat *method* `normalisasi()` seperti yang terlihat pada *Source code 4.2*.

```

public static void normalisasi(){
    for(int i=0;i<jumlahData;i++){
        for(int j=0;j<jumlahAtribut;j++){
            matriksData[i][j]=
            (matriksData[i][j]-Xmin[j])/(Xmax[j]-Xmin[j]);
        }
    }
}

```

Source code 4.2 Listing program proses normalisasi

3. Implementasi Penentuan Potensi Awal Tiap Titik Data

Pada proses penentuan potensi awal tiap titik data akan dicari nilai potensi awal (D) sebagai bahan pertimbangan pertama untuk menentukan apakah suatu titik data bisa menjadi pusat *cluster* atau tidak. Untuk itu, prosesnya akan diterapkan kedalam sebuah *method* `potensiAwal()` yang alur algoritmanya merujuk pada Persamaan (2-5) sampai Persamaan (2-7). Struktur penulisan kode program untuk proses perhitungan potensi awal tiap titik data ditunjukkan pada *Source code 4.3*.

```

public static void potensiAwal(){
    double[] T=new double[jumlahAtribut];
    double[][] dist=new double[jumlahData][jumlahAtribut];
    D=new double[jumlahData];
    for(int i=0;i<jumlahData;i++){
        double[] DS=new double[jumlahData];
        for(int j=0;j<jumlahAtribut;j++)

```

```

        T[j]=matriksData[i][j];
        for(int k=0;k<jumlahData;k++)
            for(int j=0;j<jumlahAtribut;j++)
                dist[k][j]=(T[j]-matriksData[k][j])/r[j];
        for(int k=0;k<jumlahData;k++){
            for(int j=0;j<jumlahAtribut;j++)

            DS[k]=DS[k]+Math.pow(dist[k][j],2);
        }
        for(int k=0;k<jumlahData;k++)
            D[i]=D[i]+Math.exp((-4)*DS[k]);
    }
    M=D[findIdxMax(D)];
}

```

Source code 4.3 Listing program proses penentuan potensi awal tiap titik data

4. Implementasi Pencarian Titik dengan Potensi Tertinggi

Proses pencarian titik dengan potensi tertinggi merupakan proses untuk menentukan titik data dengan nilai potensi tertinggi yang nantinya digunakan sebagai acuan untuk penentuan pusat *cluster*. Dalam implementasinya, proses ini diterapkan ke dalam sebuah *method* `findIdxMax()` dimana proses ini adalah menampung nilai potensi tertinggi kedalam variabel *max* pada tiap kali iterasi sesuai jumlah data latih. Implementasi untuk proses pencarian titik dengan potensi tertinggi dapat dilihat pada *Source code 4.4*.

```

public static int findIdxMax(double[] arr){
    int idx=0;
    double max=0;
    for(int i=0;i<arr.length;i++){
        if(arr[i]>max){
            max=arr[i];
            idx=i;
        }
    }
    return idx;
}

```

Source code 4.4 Listing program proses pencarian titik dengan potensi tertinggi

5. Implementasi Penentuan Pusat Cluster

Proses penentuan pusat *cluster* merupakan proses untuk menentukan apakah suatu titik data dengan potensi tertinggi pada suatu iterasi dapat dijadikan sebagai pusat *cluster* atau tidak. Setelah titik data tersebut bisa dijadikan sebagai pusat *cluster* dengan kondisi tertentu, maka akan dilakukan pengurangan potensi titik data tersebut sehingga pada iterasi selanjutnya potensi titik data tersebut tidak diperhitungkan pada proses pencarian titik dengan potensi tertinggi.

Secara umum proses penentuan pusat *cluster* yang telah dibangun pada subbab 3.3 sebelumnya akan diterapkan kedalam sebuah *method* tentukanPusatCluster(), dimana dalam *method* ini berisi runtutan algoritma sesuai dengan proses yang telah dibangun sebelumnya. Untuk lebih jelasnya, implementasi pada proses penentuan pusat *cluster* dapat dilihat pada *Source code* 4.5.

```

public static void tentukanPusatCluster() {
    double[] V=new double[jumlahAtribut];
    int iterasi=1;
    Z=M;C=0;kondisi=1;
    while(kondisi!=0 && Z!=0){
        for(int j=0;j<jumlahAtribut;j++){
            V[j]=matriksData[idxTitikPotensiTertinggi][j];
        }
        double rasio=Z/M;
        kondisi=0;
        if(rasio>acceptRatio){
            kondisi=1;
        }else{
            if(rasio>rejectRatio){
                double Md=-1;
                double[][] G=new double[C][jumlahAtribut];
                double[] Sd=new double[C];
                for(int i=0;i<C;i++){
                    for(int j=0;j<jumlahAtribut;j++){
                        G[i][j]=(V[j]-((double[])
                            (pusatCluster.get(i)))[j])/r[j];
                    }
                    for(int j=0;j<jumlahAtribut;j++){
                        Sd[i]=Sd[i]+Math.pow(G[i][j],2);
                    }
                    if(Md<0 || Sd[i]<Md)
                        Md=Sd[i];
                }
                double Smd=Math.sqrt(Md);
                if((rasio+Smd)>=1){
                    kondisi=1;
                } else{
                    kondisi=2;
                }
            }else{
                tampil=tampil+"\nRasio < Reject Ratio";
                tampil=tampil+"\nIterasi dihentikan!";
            }
        }
    }
    if(kondisi==1){
        C++;
        pusatCluster.add(Arrays.copyOf(V,V.length));
        double[][] S=new
        double[jumlahData][jumlahAtribut];
        double[] ST=new double[jumlahData];
        double[] Dc=new double[jumlahData];
        for(int i=0;i<jumlahData;i++){
            double sigma=0;

```

```

        for(int j=0;j<jumlahAtribut;j++){
            S[i][j]=(V[j]-matriksData[i][j])
                /(r[j]*q);
            sigma=sigma+Math.pow(S[i][j],2);
        }
        ST[i]=sigma;
    }
    for(int i=0;i<jumlahData;i++){
        Dc[i]=Z*Math.exp((-4)*ST[i]);
    }
    for(int i=0;i<jumlahData;i++){
        D[i]=D[i]-Dc[i];
        if(D[i]<0)
            D[i]=0;
    }
    cariTitikPotTertinggi();
}
if(kondisi==2){
    D[idxTitikPotensiTertinggi]=0;
    cariTitikPotTertinggi();
}
iterasi++;
}
}

```

Source code 4.5 Listing program proses penentuan pusat cluster

6. Implementasi Pengembalian Pusat Cluster

Proses pengembalian pusat *cluster* atau bisa dikatakan sebagai proses denormalisasi merupakan kebalikan dari proses normalisasi data, yaitu mengembalikan pusat *cluster* yang awalnya ada pada keadaan normal berubah menjadi kondisi semula atau nilai sebenarnya. Penerapan untuk proses denormalisasi data adalah dengan membuat *method* `denormalisasi()` yang alur proses programnya sesuai dengan Persamaan (2-14). Kode program untuk proses denormalisasi ditunjukkan pada *Source code 4.6*.

```

public static void denormalisasi(){
    for(int i=0;i<C;i++){
        for (int j=0;j<jumlahAtribut;j++){
            double[] v=(double[]) (pusatCluster.get(i));
            v[j]=(v[j]*(Xmax[j]-Xmin[j]))+Xmin[j];
        }
    }
}

```

Source code 4.6 Listing program proses pengembalian pusat cluster

7. Implementasi Perhitungan Nilai Sigma Cluster

Implementasi untuk proses perhitungan nilai sigma mengacu pada Persamaan (2-15). Pada implementasinya, proses perhitungan nilai sigma *cluster*

diterapkan kedalam sebuah *method* `hitungSigmaCluster()`. Untuk lebih jelas terkait *method* `hitungSigmaCluster()` dapat dilihat pada *Source code* 4.7.

```
public static void hitungSigmaCluster(){
    sigma=new double[jumlahAtribut];
    for(int j=0;j<jumlahAtribut;j++){
        sigma[j]=r[j]*((Xmax[j]-Xmin[j])/Math.sqrt(8));
    }
}
```

Source code 4.7 Listing program proses perhitungan nilai sigma cluster

4.2.2. Implementasi Proses Perhitungan Varian

Proses perhitungan varian terbagi menjadi 3 jenis, yaitu proses perhitungan varian tiap *cluster*, proses perhitungan *varian within cluster*, dan proses perhitungan *varian between cluster*. Proses dari perhitungan varian adalah untuk mendapatkan nilai batasan varian dari hasil *cluster* yang terbentuk. Lewat nilai batasan varian inilah yang nantinya bisa dijadikan acuan untuk memilih hasil *cluster* mana yang efektif guna dijadikan bahan bagi proses ekstraksi aturan *fuzzy*. Pada implementasinya, proses perhitungan varian dibagi menjadi 4 *method* utama, yaitu `varianTiapCluster()`, `varianWtihinCluster()`, `varianBetweenCluster()`, dan `batasanVarian()`. Sedangkan untuk lebih lengkapnya terkait *method* pembentuk kelas `varianCluster.java` ditunjukkan pada Tabel 4.3.

Tabel 4.3 *Method – method* pembentuk kelas `varianCluster.java`

Nama <i>method</i>	Deskripsi
<code>jumlahDataTiapCluster(int indexCluster)</code>	<i>Method</i> yang digunakan untuk menghitung jumlah data pada tiap <i>cluster</i> yang terbentuk.
<code>rataRataAtribut(int indexCluster, int indexAtribut)</code>	Sebuah <i>method</i> yang digunakan menghitung nilai rata – rata pada tiap atribut data pada suatu <i>cluster</i> yang terbentuk.
<code>varianTiapAtribut(int x, int y)</code>	<i>Method</i> yang digunakan untuk mencari nilai varian tiap atribut pada suatu <i>cluster</i> .
<code>varianTiapCluster(int x)</code>	<i>Method</i> yang digunakan untuk proses perhitungan varian tiap <i>cluster</i> .
<code>varianWithinCluster()</code>	Sebuah <i>method</i> yang digunakan untuk proses perhitungan <i>variance within cluster</i> .

varianAntarAtribut(int x, int y)	Sebuah <i>method</i> yang digunakan untuk menghitung nilai varian antar atribut pada suatu <i>cluster</i> .
varianAntarCluster(int x)	<i>Method</i> untuk menghitung nilai varian antar <i>cluster</i> .
varianBetweenCluster()	Sebuah <i>method</i> yang digunakan untuk menghitung nilai <i>variance between cluster</i> .
batasanVarian()	<i>Method</i> yang diperuntukan untuk menghitung nilai batasan varian.

1. Implementasi Proses Perhitungan Varian Tiap Cluster

Proses perhitungan varian tiap *cluster* diterapkan dengan membuat *method* `varianTiapCluster()`, dimana dalam *method* tersebut perlu untuk memanggil *method* tambahan lainnya untuk membantu proses perhitungan, diantaranya adalah *method* `jumlahDataTiapCluster()`, `rataRataAtribut()`, dan `varianTiapAtribut()`. Alur proses untuk *method* `varianTiapCluster()` mengacu pada Persamaan (2-17). Implementasi untuk proses perhitungan varian tiap *cluster* ditunjukkan pada *Source code* 4.8.

```
public static double varianTiapAtribut(int x, int y){
    double varianAtribut=0,jumDataCluster=0,rataAtribut=0;
    int[] idxHasilCluster=sc.tentukanIdxCluster(sc.tabelMyu);
    rataAtribut=rataRataAtribut(x,y);
    for(int k=0;k<sc.jumlahData;k++){
        if(idxHasilCluster[k]==x){
            varianAtribut=varianAtribut+Math.pow(sc.matriksData[k][y]-
            rataAtribut,2);
        }
    }
    varianAtribut=varianAtribut*1/((jumlahDataTiapCluster(x))-1);
    return varianAtribut;
}
public static double varianTiapCluster(int x){
    double varian=0, temp=0;
    int[] idxHasilCluster=sc.tentukanIdxCluster(sc.tabelMyu);
    for(int j=0;j<sc.jumlahAtribut;j++){
        temp=varianTiapAtribut(x,j);
        varian=varian+temp;
    }return varian;
}
```

Source code 4.8 Listing program proses perhitungan varian tiap *cluster*

2. Implementasi Proses Perhitungan Varian Within Cluster

Proses perhitungan *varian within cluster* nantinya melibatkan hasil dari *method* `varianTiapCluster()`. Dalam implementasinya, proses perhitungan *varian within cluster* diterapkan dengan membuat *method*

`varianWithinCluster()`. Proses pada *method* ini mengacu pada Persmaan (2-16). Variabel *temp* digunakan untuk menampung jumlah total nilai varian tiap cluster yang terbentuk. Implementasi untuk proses perhitungan *varian within cluster* ditunjukkan pada *Source code 4.9*.

```
public static double varianWithinCluster() {
    double VW=0,temp=0;
    for(int i=0;i<sc.C;i++){
        temp=temp+
            ((jumlahDataTiapCluster(i)-1)*varianTiapCluster(i));
    }
    VW=(1/ sc.jumlahData-sc.C)*temp;
    return VW;
}
```

Source code 4.9 Listing program proses perhitungan variance within cluster

3. Implementasi Proses Perhitungan Varian Between Cluster

Proses perhitungan varian tiap *cluster* diterapkan dengan membuat *method* `varianBetweenCluster()`, dimana dalam *method* tersebut perlu untuk memanggil *method* tambahan lainnya untuk membantu proses perhitungan, diantaranya adalah *method* `jumlahDataTiapCluster()`, `rataRataAtribut()`, dan `varianAntarAtribut()`. Implementasi untuk proses perhitungan varian tiap *cluster* ditunjukkan pada *Source code 4.10*.

```
public static double varianAntarAtribut(int x, int y){
    double varianAntarAtribut=0,jumDataCluster=0,rataAtribut=0;
    int[] idxHasilCluster=sc.tentukanIdxCluster(sc.tabelMyu);
    rataAtribut=rataRataAtribut(x,y);
    for(int k=0;k<sc.jumlahData;k++){
        if(idxHasilCluster[k]==x){
            varianAntarAtribut=
                varianAntarAtribut+(Math.pow(sc.matriksData[k][y]
                    -rataAtribut,2))*jumlahDataTiapCluster(x);
        }
    }return varianAntarAtribut;
}
public static double varianAntarCluster(int x){
    double varian=0, temp=0;
    int[] idxHasilCluster=sc.tentukanIdxCluster(sc.tabelMyu);
    for(int j=0;j<sc.jumlahAtribut;j++){
        temp=varianAntarAtribut(x,j);
        varian=varian+temp;
    }return varian;
}
public static double varianBetweenCluster(){
    double VB=0,temp=0;
    for(int i=0;i<sc.C;i++){
        temp=temp+varianAntarCluster(i);
    }double temp2=sc.C-1;double temp3=1/temp2;
    VB=temp3*temp;return VB;
}
```

Source code 4.10 Listing program proses perhitungan varian between cluster

4. Implementasi Proses Perhitungan Batasan Varian

Perhitungan nilai batasan varian merupakan proses terakhir pada proses perhitungan varian. Nilai batasan varian didapatkan dari hasil bagi antara nilai *varian within cluster* dengan *varian between cluster* sesuai dengan Persamaan (2-19). Pada implementasinya, proses perhitungan batasan varian diterapkan pada sebuah *method* `batasanVarian()` sama seperti pada *Source code* 4.11.

```
public static double batasanVarian(){
    double batasanVarian=0;
    batasanVarian=varianWithinCluster()/varianBetweenCluster();
    return batasanVarian;
}
```

Source code 4.11 Listing program proses perhitungan batasan varian

4.2.3. Implementasi Proses Ekstraksi Aturan Fuzzy

Proses ekstraksi aturan *fuzzy* merupakan proses untuk untuk mendapatkan nilai koefisien *output* pada konsekuen sistem inferensi *fuzzy* model sugeno. Implementasi ekstraksi aturan *fuzzy* pada kelas `ekstraksiAturanFuzzy.java` mengacu pada sub-subbab 3.3.2.4. Pada kelas `ekstraksiAturanFuzzy()` terdapat dua *method* utama, yaitu *method* `hitungDerajatKeanggotaan()` dan `perhitunganKoefisienOutput()`.

4.2.3.1 Implementasi Perhitungan Nilai Derajat Keanggotaan

Proses perhitungan nilai derajat keanggotaan untuk tiap data terhadap pusat *cluster* dan sigma dengan menggunakan fungsi *gauss* sama seperti pada Persamaan (2-29). Pada implementasinya, proses ini diterapkan ke dalam *method* `hitungDerajatKeanggotaan()`. Struktur kode program ini ditunjukkan pada *Source code* 4.12.

```
public static void hitungDerajatKeanggotaan() throws
FileNotFoundException{
    tabelMyu=new double[jumlahData][C];
    matriksData = new double[jumlahData][jumlahAtribut];
    BacaDataLatih();
    for(int i=0;i<jumlahData;i++){
        for(int k=0;k<C;k++){
            double signal=0;
            double[] v =((double[])
            (pusatCluster.get(k)));
            for(int j=0;j<jumlahAtribut;j++)
                signal = signal +
                ((Math.pow((matriksData[i][j]-v[j]),2))
                / (2*(Math.pow(sigma[j],2)))));
            tabelMyu[i][k]=Math.exp((-1)*signal);
        }
    }
}
```

```

    }
}
}

```

Source code 4.12 Listing program proses perhitungan nilai derajat keanggotan

4.2.3.2 Implementasi Perhitungan Koefisien Output

Implementasi proses perhitungan koefisien *output* diterapkan pada *method* `perhitunganKoefisienOutput()` yang proses jalannya algoritma sesuai dengan sub-subbab 3.3.2.4 point 1. Proses perhitungan koefisien *output* memerlukan operasi matriks yang cukup kompleks, diantaranya seperti perkalian matriks, *transpose*, dan *inverse*. Oleh karena itulah perlu adanya sebuah *library* tambahan, yaitu `org.ejml.simple.SimpleMatrix` untuk memudahkan operasi matriks dengan dimensi yang cukup panjang. Struktur penulisan kode program untuk proses perhitungan koefisien *output* ditunjukkan pada *Source code* 4.13.

```

public static double [][] perhitunganKoefisienOutput() {
    SimpleMatrix u = new SimpleMatrix(hitungMatriksU());
    SimpleMatrix uT = u.transpose();
    double uTranspose[][] = new
    double[uT.numRows()][uT.numCols()];
    for(int i=0;i<uT.numRows();i++){
        for(int j=0;j<uT.numCols();j++){
            uTranspose[i][j]=uT.get(i, j);
        }
    }
    SimpleMatrix uTU = uT.mult(u);
    double uTransposeU[][] = new
    double[uTU.numRows()][uTU.numCols()];
    for(int i=0;i<uTU.numRows();i++){
        for(int j=0;j<uTU.numCols();j++){
            uTransposeU[i][j]=uTU.get(i, j);
        }
    }
    SimpleMatrix ui = new SimpleMatrix(uTransposeU);
    SimpleMatrix uTUIinverse = ui.invert();
    double inverseU[][] = new
    double[uTUIinverse.numRows()][uTUIinverse.numCols()];
    for(int i=0;i<uTUIinverse.numRows();i++){
        for(int j=0;j<uTUIinverse.numCols();j++){
            inverseU[i][j]=uTUIinverse.get(i, j);
        }
    }
    SimpleMatrix inverseUxUT = uTUIinverse.mult(uT);
    double nk[][] = new double[sc.jumlahData][1];
    for(int i=0;i<sc.jumlahData;i++){
        nk[i][0]=sc.matriksData[i][sc.jumlahAtribut-1];
    }
    SimpleMatrix targetOutput = new SimpleMatrix(nk);
    SimpleMatrix ko = inverseUxUT.mult(targetOutput);
    double koefisienOutput[][] = new double

```

```

[sc.C][sc.jumlahAtribut];
int k=0;
for(int i=0;i<sc.C;i++){
    for(int j=0;j<sc.jumlahAtribut;j++){
        koefisienOutput[i][j]=ko.get(k,0);
        k++;
    }
}
return koefisienOutput;
}
    
```

Source code 4.13 Listing program proses perhitungan koefisien output

4.2.4. Implementasi Proses Sistem Inferensi Fuzzy Model Sugeno

Proses sistem inferensi fuzzy merupakan bagian pengujian terhadap data uji menggunakan fuzzy inference system model sugeno dengan aturan yang telah dibangkitkan pada proses clustering dan ekstraksi aturan fuzzy. Implementasi pada proses ini diterapkan ke dalam kelas SistemInferensiFuzzy.java, dimana di dalamnya terdapat 6 buah method diantaranya adalah BacaDataUji(), getCenter(), hitungMiu(), hitungAlphaPredikat(), hitungZ(), dan defuzzy(). Deskripsi untuk masing – masing method yang ada pada kelas SistemInferensiFuzzy.java ditunjukkan pada Tabel 4.4.

Tabel 4.4 Method – method pembentuk kelas SistemInferensiFuzzy.java

Nama method	Deskripsi
KoneksiDB ()	Method yang digunakan sebagai koneksi antara aplikasi dengan database MySql.
BacaDataUji ()	Sebuah method yang digunakan membaca dan menampung data uji dari database.
getCenter ()	Method yang digunakan untuk menginisialisasi center yang telah terbentuk pada proses clustering.
hitungMiu(double [][] dataUji, int x)	Sebuah method yang digunakan untuk proses perhitungan nilai derajat keanggotaan.
hitungAlphaPredikat(double [][] dataUji, int x)	Sebuah method yang digunakan untuk proses perhitungan alpha predikat.
hitungZ(double [][] dataUji, int x)	Sebuah method yang digunakan untuk menghitung nilai Z pada masing – masing aturan.
defuzzy(double [][] dataUji, int x)	Method yang digunakan pada proses defuzzyfikasi.



pengujian()	Sebuah <i>method</i> yang digunakan untuk menghitung nilai kelayakan data uji.
-------------	--

4.2.4.1 Implementasi Proses Pembacaan Data Uji

Langkah pertama dalam proses pengujian adalah memasukkan data uji. Dalam implementasinya, data uji disimpan ke dalam sebuah database yang nantinya dipanggil menggunakan *method* BacaDataUji(). Di dalam *method* BacaDataUji() terdapat *method* KoneksiDB() yang fungsinya sebagai konektor antara aplikasi dengan *database*. Struktur kode program untuk proses baca data uji ditunjukkan pada *Source code* 4.14.

```
private static void BacaDataUji(){
    KoneksiDB();
    matriksDataUji = new double[3][sc.jumlahAtribut-1];
    try {
        String sql = "select * from data_uji";
        statement.executeQuery(sql);
        ResultSet rs = statement.executeQuery(sql);
        double ipk=0, sks=0, prestasi=0, ke=0;
        int cek=0;
        while(rs.next()){
            ipk=rs.getDouble("ipk");
            sks=rs.getDouble("sks");
            prestasi=rs.getDouble("prestasi");
            ke=rs.getDouble("ke");
            matriksDataUji[cek][0]=ipk;
            matriksDataUji[cek][1]=sks;
            matriksDataUji[cek][2]=prestasi;
            matriksDataUji[cek][3]=ke;
            cek++;
        }
        rs.close();
        statement.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e,
        "Kesalahan", JOptionPane.ERROR_MESSAGE);
    }
}
```

Source code 4.14 Listing program proses pembacaan data uji

4.2.4.2 Implementasi Proses Inisialisasi Pusat Cluster

Proses inisialisasi pusat *cluster* merupakan proses untuk mendapatkan nilai pusat *cluster* yang sebelumnya telah dihitung pada proses *clustering*. Proses ini diimplementasikan dengan membuat *method* getCenter(). struktur program untuk proses inisialisasi pusat *cluster* ditunjukkan pada *Source code* 4.15.

```

private static double[][] getCenter(){
    double [][] pusatCluster = new
    double[sc.C][sc.jumlahAtribut];
    for(int i=0;i<sc.pusatCluster.size();i++){
        double [] temp=(double[])sc.pusatCluster.get(i);
        for(int j=0;j<sc.jumlahAtribut;j++){
            pusatCluster[i][j]=temp[j];
        }
    }
    return pusatCluster;
}

```

Source code 4.15 Listing program proses inialisasi pusat cluster

4.2.4.3 Implementasi Proses Perhitungan Derajat Keanggotaan

Proses perhitungan derajat keanggotaan dilakukan terhadap data uji yang memiliki 4 parameter. Nantinya 4 parameter tersebut dicari derajat keanggotaannya menggunakan fungsi *gauss*. Implementasi untuk proses perhitungan derajat keanggotaan diterapkan pada *method* *hitungMiu()* yang struktur programnya ditunjukkan pada *Source code 4.16*.

```

private static double[][] hitungMiu(double [][] dataUji, int
x){
    double [][] miuDataUji = new double[sc.C][sc.jumlahAtribut-1];
    double [][] center = getCenter();
    for(int i=0;i<sc.C;i++){
        double tempSigma=0;
        for(int j=0;j<sc.jumlahAtribut-1;j++){
            tempSigma =((Math.pow((dataUji[x][j]-
            center[i][j]),2))/(2*(Math.pow(sc.sigma[j],2)))));
            miuDataUji[i][j]=Math.exp((-1)*tempSigma);
        }
    }
    return miuDataUji;
}

```

Source code 4.16 Listing program proses perhitungan derajat keanggotaan

4.2.4.4 Implementasi Proses Perhitungan Alpha Predikat

Implementasi proses perhitungan *alpha predikat* diterapkan pada sebuah *method* *hitungAlphaPredikat()*. Struktur program untuk proses perhitungan *alpha predikat* ditunjukkan pada *Source code 4.17*.

```

private static double [] hitungAlphaPredikat(double [][]
dataUji, int x){
    double []alphaPredikat = new double [sc.C];
    double [][]miuDataUji = hitungMiu(dataUji, x);
    for(int i=0;i<sc.C;i++){
        double temp=1;
        for(int j=0;j<sc.jumlahAtribut-1;j++){
            temp=temp*miuDataUji[i][j];
        }
    }
}

```

```

        alphaPredikat[i]=temp;
    }
    return alphaPredikat;
}

```

Source code 4.17 Listing program proses perhitungan *alpha predikat*

4.2.4.5 Implementasi Proses Perhitungan Nilai z

Implementasi proses perhitungan nilai z untuk tiap aturan diterapkan pada sebuah *method* `hitungZ()` yang struktur programnya terlihat seperti pada *Source code* 4.18.

```

private static double [] hitungZ(double [][] dataUji, int x){
    double []z = new double [sc.C];
    double [][]ko = aturan.perhitunganKoefisienOutput();
    for(int i=0;i<sc.C;i++){
        double temp=0;
        for(int j=0;j<sc.jumlahAtribut-1;j++){
            double temp2=0;
            temp2=dataUji[x][j]*ko[i][j];
            temp=temp+temp2;
        }
        z[i]=temp+ko[i][sc.jumlahAtribut-1];
    }
    return z;
}

```

Source code 4.18 Listing program proses perhitungan nilai z

4.2.4.6 Implementasi Proses Defuzzy

Proses *defuzzy* merupakan proses terakhir pada fase pengujian. Proses *defuzzy* merupakan proses merubah nilai *fuzzy* menjadi bentuk *crisp*. Pada implementasinya, proses *defuzzy* diterapkan ke dalam *method* `defuzzy()`. Untuk memudahkan alur penulisan kode program, maka diperlukan variabel *temp* untuk menampung jumlah total antara perkalian *alpha predikat* dengan nilai z, sedangkan variabel *temp2* digunakan untuk menampung nilai total dari hasil penjumlahan setiap *alpha predikat* pada masing – masing aturan. Struktur program untuk *method* `defuzzy()` ditunjukkan pada *Source code* 4.19.

```

private static double defuzzy(double [][] dataUji, int x){
    double z=0;
    double []a=hitungAlphaPredikat(dataUji,x);
    double []b=hitungZ(dataUji,x);
    double temp=0, temp2=0;
    for(int i=0;i<sc.C;i++){
        temp=temp+(a[i]*b[i]);
        temp2=temp2+a[i];
    }z=temp/temp2;return z;
}

```

Source code 4.19 Listing program proses *defuzzy*

4.3. Implementasi Antarmuka

Implementasi antarmuka mengacu berdasarkan subbab 3.3.4 yang telah dibangun sebelumnya. Tampilan antarmuka pada sistem terbagi menjadi 2 form, diantaranya adalah form pembelajaran sebagai form utama sekaligus antarmuka untuk proses *clustering* dan pengujian pembentukan aturan serta form hasil pengujian data uji atau rekomendasi beasiswa yang merupakan antarmuka untuk menampilkan hasil pengujian terhadap data uji berdasarkan aturan yang telah didapatkan pada proses sebelumnya.

4.3.1. Antarmuka Proses Pembelajaran

Form pembelajaran merupakan antarmuka yang muncul pada saat pertama kali program dijalankan. Fungsi dari antarmuka ini adalah untuk menjembatani *user* dengan sistem dalam melakukan proses pembelajaran menggunakan *subtractive clustering* terhadap data latih. Ada beberapa *field* dan tombol yang digunakan, diantaranya adalah *field* untuk memasukkan jumlah data latih yang sebelumnya telah disimpan pada *database*, *field* untuk memasukkan batas bawah jari – jari, *accept ratio*, dan *reject ratio*. Nilai batas bawah jari – jari merupakan nilai awal untuk proses pengujian pembentukan aturan yang kemudian akan bertambah 0,1 pada setiap iterasi. Untuk parameter nilai batas atas dan batas bawah tiap atribut data didapatkan secara otomatis dengan menekan tombol disebelah *field* *xMin* dan *xMax*. Tombol “Proses Cluster” merupakan tombol untuk memproses data latih berdasarkan parameter yang ditentukan sebelumnya. Sedangkan tombol “Ulangi Percobaan” merupakan tombol untuk mengosongkan seluruh *field* masukan agar dapat memulai proses *clustering* lagi.

Pada form ini disediakan informasi hasil dari proses setiap pengujian sebanyak 8 kali, diantaranya adalah nilai batasan varian dan jumlah *cluster*. Sistem juga akan memilih secara otomatis hasil *cluster* mana yang akan dipakai pada proses ekstraksi aturan *fuzzy* berdasarkan analisa varian. Dari proses analisa varian, diberikan informasi terkait jari – jari terpilih, jumlah *cluster* terpilih, batasan varian terpilih, pusat *cluster* terpilih, sigma *cluster* terpilih, dan aturan *fuzzy* beserta koefisien *output* yang terbentuk. Secara keseluruhan antarmuka proses pembelajaran dapat dilihat pada Gambar 4.1.

Gambar 4.1 Antarmuka proses pembelajaran

4.3.2. Antarmuka Hasil Pengujian Data Uji

Form hasil pengujian data uji merupakan antarmuka untuk menampilkan rekomendasi penerima beasiswa berdasarkan ekstraksi aturan *fuzzy* pada proses pembelajaran sebelumnya. 10 nilai kelayakan tertinggi akan direkomendasikan sebagai penerima beasiswa. Dalam form ini juga ditampilkan parameter – parameter terpilih dari hasil proses pembelajaran dengan *subtractive clustering* berdasarkan analisa varian. Tampilan antarmuka untuk form pengujian pada Gambar 4.2.

ID	IPK	SKS	P	KE	SS	NK Sistem	SR
9	3.63	114	10	4643798.33333	Diterima	0,8993	Diterima
5	3.652	114	9	1248416.66667	Diterima	0,9135	Diterima
1	3.761	114	10	2876085	Diterima	0,9742	Diterima
12	3.832	110	0	730416.66667	Diterima	0,9857	Diterima
13	3.698	110	0	778535	Diterima	0,9378	Diterima
24	3.726	102	0	3629464	Diterima	0,9422	Diterima
36	3.877	84	0	2361633.33333	Diterima	0,9776	Diterima
40	3.866	103	12.5	2763148.5	Diterima	0,9878	Diterima
42	3.628	101	0	926900	Diterima	0,8285	Tidak
44	3.705	110	0	4708556.83333	Diterima	0,9472	Diterima
61	3.004	112	20	458183.75	Tidak	0,0061	Tidak
62	3.118	112	0	3212660	Tidak	0,0235	Tidak
63	3.441	112	0	4095604.16667	Tidak	0,4892	Tidak
67	3.61	112	0	1963370	Tidak	0,8607	Diterima
68	3.452	112	0	3358625	Tidak	0,5159	Tidak
69	3.529	112	2.5	4396000	Tidak	0,7235	Tidak
70	3.588	112	5	650446.16667	Tidak	0,8234	Tidak

Gambar 4.2 Antarmuka proses pengujian

BAB V ANALISA HASIL DAN PEMBAHASAN

Pada BAB V ini dibahas mengenai analisa hasil dan pembahasan dari skenario pengujian yang telah dijelaskan pada subbab 3.5. Subbab – subbab pada BAB V ini berisi bagaimana skenario pengujian, hasil pengujian, dan analisa hasil.

3.6 Skenario Pengujian

Skenario pengujian yang dilakukan mengacu pada penjabaran dari subbab 3.5, yaitu dibagi menjadi dua tahap. Tahap pertama disebut dengan “Pengujian Tahap Satu”, sedangkan tahap kedua disebut dengan “Pengujian Tahap Dua”.

Pada skenario pengujian satu, pengujian dilakukan terhadap proses pembelajaran data latih menggunakan *subtractive clustering* dimana bahan pengujian berupa jumlah data latih yang bervariasi antara 60 sampai 100, parameter jari – jari, parameter *accept ratio*, dan parameter *reject ratio*. Dengan jumlah data latih dan parameter – parameter *subtractive clustering* yang berbeda nantinya akan menghasilkan jumlah *cluster* dan nilai varian yang berbeda pula. Tujuan dari pengujian ini adalah untuk menentukan jumlah aturan atau jumlah *cluster* terpilih berdasarkan analisa varian. Skenario pengujian tahap satu dibedakan menjadi tiga, yaitu pengujian pertama menggunakan 60 data latih, pengujian kedua menggunakan 80 data latih, dan pengujian ketiga menggunakan 100 data latih. Masing – masing tahapan pengujian akan menggunakan nilai jari – jari, *accept ratio*, dan *reject ratio* yang berbeda, yaitu berkisar antara 0,3 sampai 1 untuk parameter jari – jari, selang antara 0,5 sampai 0,9 untuk parameter *accept ratio*, dan berkisar antara 0.15 sampai 0,5 untuk parameter *reject ratio*. Analisa varian digunakan untuk menentukan hasil *clustering* yang bagus sebagai bahan acuan untuk ekstraksi aturan.

Skenario pengujian dua merupakan tahapan pengujian untuk mengetahui akurasi hasil inferensi berdasarkan aturan terpilih sebelumnya. Data yang digunakan untuk skenario pengujian tiga sebanyak 30 data uji yang memiliki empat atribut, yaitu nilai IPK, SKS, prestasi dalam angka, dan keadaan ekonomi.

3.7 Hasil Pengujian

3.3.1. Hasil Pengujian Tahap Satu

Pengujian tahap satu terbagi menjadi tiga berdasarkan pada jumlah data latih yang berbeda. Nilai parameter jari – jari, *accept ratio*, dan *reject ratio* bervariasi pada tiap pengujian berdasarkan parameter. Pada pengujian tahap satu ini menghasilkan 192 macam jenis hasil *cluster* berdasarkan parameter jari – jari, *accept ratio*, dan *reject ratio* yang berbeda. Nantinya dari 192 akan diambil 24 hasil *cluster* terbaik berdasarkan analisa varian untuk dijadikan aturan.

5.2.1.1. Hasil Pengujian Pertama (60 Data Latih)

Pada pengujian pertama, proses pembentukan aturan dilakukan sebanyak 48 kali yang kemudian dipecah menjadi 6 jenis kumpulan aturan berdasarkan parameter jari – jari untuk analisa varian. Sehingga nantinya tiap 1 kumpulan aturan berisi 8 jenis aturan yang nilai jari – jarinya berbeda namun nilai *accept ratio* dan *reject rationya* sama. Dari tiap kumpulan aturan akan dilakukan analisa varian untuk menentukan aturan terpilih, sehingga hasil akhir pada pengujian pertama pada pengujian tahap satu akan menghasilkan 6 aturan terpilih. Hasil selengkapnya untuk pengujian pertama ditunjukkan pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1 Tabel hasil pengujian pertama tahap satu
(*reject ratio* 0,15 – 0,35)

Jumlah Data Latih = 60						
Jari – jari	<i>ar</i> = 0,5 dan <i>rr</i> = 0,15		<i>ar</i> = 0,6 dan <i>rr</i> = 0,25		<i>ar</i> = 0,7 dan <i>rr</i> = 0,35	
	Nilai varian	Jumlah <i>cluster</i>	Nilai varian	Jumlah <i>cluster</i>	Nilai varian	Jumlah <i>cluster</i>
0,3	0,0504	15	0,0135	7	0,0054	5
0,4	0,0077	7	0,0023	4	0,0014	3
0,5	0,0029	4	0,0006	2	0,0006	2
0,6	0,0006	2	0,0006	2	0,0006	2
0,7	0,0006	2	0,0006	2	0,0006	2
0,8	0,0006	2	0,0006	2	0,0006	2
0,9	0,0006	2	0,0006	2	0,0006	2
1	0,0006	2	0,0006	2	0,0006	2

Tabel 5.2 Tabel hasil pengujian pertama tahap satu
(*reject ratio* 0,4 – 0,5)

Jumlah Data Latih = 60						
Jari – jari	$ar = 0,8$ dan $rr = 0,4$		$ar = 0,85$ dan $rr = 0,45$		$ar = 0,9$ dan $rr = 0,5$	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0054	5	0,0029	4	0,0014	3
0,4	0,0014	3	0,0006	2	0,0006	2
0,5	0,0006	2	0,0006	2	0,0006	2
0,6	0,0006	2	0,0006	2	0,0006	2
0,7	0,0006	2	0,0006	2	0,0006	2
0,8	0,0006	2	0,0006	2	0,0006	2
0,9	0,0006	2	0,0006	2	0,0006	2
1	0,0006	2	0,0006	2	0,0006	2

Dari Tabel 5.1 dan Tabel 5.2 dapat dilihat bahwa terdapat 6 jenis aturan terpilih dengan parameter – parameter yang berbeda yang dihasilkan berdasarkan nilai batasan varian terkecil pada masing – masing *accept ratio* dan *reject ratio*. Aturan – aturan yang dihasilkan ditunjukkan pada Tabel 5.3.

Tabel 5.3 Tabel aturan pengujian pertama tahap satu

Jenis Aturan (r, ar, rr)	Batasan Varian	Jumlah cluster
Aturan 1 (0,6; 0,5; 0,15)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = ($Z1 = (-4.013230436631992E-5 \times IPK) + (-6.126135070963363E-7 \times SKS) + (-8.28407225122959E-8 \times Prestasi) + (7.107408227333627E-14 \times X14) + 1.9025197395200434E-4$) [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = ($Z2 = (-5.923460065559993E-6 \times IPK) + (-2.0076252105205583E-8 \times SKS) + (1.9649781274333478E-8 \times Prestasi) + (-5.792594846086005E-15 \times X14) + 1.0000250644021662$)		
Aturan 2 (0,5; 0,6; 0,25)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = ($Z1 = (-3.6785878037315457E-5 \times IPK) + (-1.5576050545010134E-5 \times SKS) + (-7.148540790475435E-7 \times Prestasi) + (-5.932328854682879E-13 \times X14) + 0.001829403748094169$) [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = ($Z2 = (-1.6740692387884337E-8 \times IPK) + (-5.4887427936023414E-11 \times SKS) + (7.301576010476651E-11 \times Prestasi) + (-1.5293973956064444E-18 \times X14) + 1.0000000701236331$)		
Aturan 3 (0,5; 0,7; 0,35)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = ($Z1 = (-3.6785878037315457E-5 \times IPK) + (-1.5576050545010134E-5 \times SKS) + (-7.148540790475435E-7 \times Prestasi) + (-5.932328854682879E-13 \times X14) +$		

0.001829403748094169)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-1.6740692387884337E-8 x IPK) + (-5.4887427936023414E-11 x SKS) + (7.301576010476651E-11 x Prestasi) + (-1.5293973956064444E-18 x X14) + 1.0000000701236331)		
Aturan 4 (0,5; 0,8; 0,4)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-3.6785878037315457E-5 x IPK) + (-1.5576050545010134E-5 x SKS) + (-7.148540790475435E-7 x Prestasi) + (-5.932328854682879E-13 x X14) + 0.001829403748094169)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-1.6740692387884337E-8 x IPK) + (-5.4887427936023414E-11 x SKS) + (7.301576010476651E-11 x Prestasi) + (-1.5293973956064444E-18 x X14) + 1.0000000701236331)		
Aturan 5 (0,4; 0,85; 0,45)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-8.994401822632802E-11 x IPK) + (-1.142296213397587E-12 x SKS) + (-3.138528370083045E-13 x Prestasi) + (6.805483658093156E-19 x X14) + 4.012822628468887E-10)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-1.2221335055073723E-12 x IPK) + (2.4424906541753444E-15 x SKS) + (6.716849298982197E-15 x Prestasi) + (1.9058241313221758E-21 x X14) + 1.0000000000040075)		
Aturan 6 (0,4; 0,9; 0,5)	0,0006	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-8.994401822632802E-11 x IPK) + (-1.142296213397587E-12 x SKS) + (-3.138528370083045E-13 x Prestasi) + (6.805483658093156E-19 x X14) + 4.012822628468887E-10)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-1.2221335055073723E-12 x IPK) + (2.4424906541753444E-15 x SKS) + (6.716849298982197E-15 x Prestasi) + (1.9058241313221758E-21 x X14) + 1.0000000000040075)		

5.2.1.2. Hasil Pengujian Kedua (80 Data Latih)

Pada pengujian kedua, proses pembentukan aturan dilakukan sebanyak 48 kali yang kemudian dipecah menjadi 6 jenis kumpulan aturan berdasarkan parameter jari – jari untuk analisa varian. Sehingga nantinya tiap 1 kumpulan aturan berisi 8 jenis aturan yang nilai jari – jarinya berbeda namun nilai *accept ratio* dan *reject rationya* sama. Dari tiap kumpulan aturan akan dilakukan analisa varian untuk menentukan aturan terpilih, sehingga hasil akhir pada pengujian kedua pada pengujian tahap satu akan menghasilkan 6 aturan terpilih. Hasil selengkapnya untuk pengujian kedua ditunjukkan pada Tabel 5.4 dan Tabel 5.5.

Tabel 5.4 Tabel hasil pengujian kedua tahap satu
(*reject ratio* 0,15 – 0,35)

Jumlah Data Latih = 80						
Jari – jari	<i>ar</i> = 0,5 dan <i>rr</i> = 0,15		<i>ar</i> = 0,6 dan <i>rr</i> = 0,25		<i>ar</i> = 0,7 dan <i>rr</i> = 0,35	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0104	9	0,0021	5	0,0014	4
0,4	0,0018	4	0,0009	3	0,0003	2
0,5	0,0008	3	0,0003	2	0,0003	2
0,6	0,0003	2	0,0003	2	0,0003	2
0,7	0,0003	2	0,0003	2	0,0003	2
0,8	0,0003	2	0,0003	2	0,0003	2
0,9	0,0003	2	0,0003	2	0,0003	2
1	0,0003	2	0,0003	2	0,0003	2

Tabel 5.5 Tabel hasil pengujian kedua tahap satu
(*reject ratio* 0,4 – 0,5)

Jumlah Data Latih = 80						
Jari – jari	<i>ar</i> = 0,8 dan <i>rr</i> = 0,4		<i>ar</i> = 0,85 dan <i>rr</i> = 0,45		<i>ar</i> = 0,9 dan <i>rr</i> = 0,5	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0014	4	0,0003	2	0,0003	2
0,4	0,0003	2	0,0003	2	0,0003	2
0,5	0,0003	2	0,0003	2	0,0003	2
0,6	0,0003	2	0,0003	2	0,0003	2
0,7	0,0003	2	0,0003	2	0,0003	2
0,8	0,0003	2	0,0003	2	0,0003	2
0,9	0,0003	2	0,0003	2	0,0003	2
1	0,0003	2	0,0003	2	0,0003	2

Dari Tabel 5.4 dan Tabel 5.5 dapat dilihat bahwa terdapat 6 jenis aturan terpilih dengan parameter – parameter yang berbeda yang dihasilkan berdasarkan nilai batasan varian terkecil pada masing – masing *accept ratio* dan *reject ratio*. Aturan – aturan yang dihasilkan ditunjukkan pada Tabel 5.6.

Tabel 5.6 Tabel aturan pengujian kedua tahap satu

Jenis Aturan (<i>r, ar, rr</i>)	Batasan Varian	Jumlah cluster
Aturan 7 (0,6; 0,5; 0,15)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-3.2448642358255913E-5 x IPK) + (1.1658452992436321E-7 x SKS) + (-1.4665289490652454E-7 x Prestasi) + (5.313678431966809E-13 x X14) + 8.470965857499714E-5)		

[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-8.09371725551955E-6 x IPK) + (1.0985835574572889E-9 x SKS) + (1.538163965641992E-8 x Prestasi) + (2.3926655340044466E-13 x X14) + 1.0000305710091908)		
Aturan 8 (0,5; 0,6; 0,25)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-3.017019977924484E-5 x IPK) + (-1.5203018884186398E-5 x SKS) + (-1.9074026980372864E-6 x Prestasi) + (1.413788891866705E-12 x X14) + 0.0017702919220385818)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-3.048731400667748E-8 x IPK) + (-2.3570978502363005E-11 x SKS) + (6.073683223029036E-11 x Prestasi) + (1.0863239370788173E-15 x X14) + 1.000000116641118)		
Aturan 9 (0,4; 0,7; 0,35)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-6.357472554142913E-11 x IPK) + (2.8094714517840813E-13 x SKS) + (-4.4032351175498885E-13 x Prestasi) + (1.0919150094328774E-18 x X14) + 1.6223585979600694E-10)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-8.917311333789257E-13 x IPK) + (-6.328271240363392E-15 x SKS) + (3.5110803153770576E-15 x Prestasi) + (4.605741650695258E-20 x X14) + 1.0000000000037943)		
Aturan 10 (0,4; 0,8; 0,4)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-6.357472554142913E-11 x IPK) + (2.8094714517840813E-13 x SKS) + (-4.4032351175498885E-13 x Prestasi) + (1.0919150094328774E-18 x X14) + 1.6223585979600694E-10)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-8.917311333789257E-13 x IPK) + (-6.328271240363392E-15 x SKS) + (3.5110803153770576E-15 x Prestasi) + (4.605741650695258E-20 x X14) + 1.0000000000037943)		
Aturan 11 (0,3; 0,85; 0,45)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-9.055700293463121E-19 x IPK) + (6.069430285923119E-21 x SKS) + (-6.701997205250351E-21 x Prestasi) + (1.477854854974975E-26 x X14) + 2.0911277439375526E-18)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-3.659295089164516E-13 x IPK) + (-1.1657341758564144E-14 x SKS) + (-1.942890293094024E-15 x Prestasi) + (6.008639969585193E-21 x X14) + 1.0000000000026432)		
Aturan 12 (0,3; 0,9; 0,5)	0,0003	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-9.055700293463121E-19 x IPK) + (6.069430285923119E-21 x SKS) + (-6.701997205250351E-21 x Prestasi) + (1.477854854974975E-26 x X14) + 2.0911277439375526E-18)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-3.659295089164516E-13 x IPK) + (-1.1657341758564144E-14 x SKS) + (-1.942890293094024E-15 x Prestasi) + (6.008639969585193E-21 x X14) + 1.0000000000026432)		

5.2.1.3. Hasil Pengujian Ketiga (100 Data Latih)

Pada pengujian ketiga, proses pembentukan aturan dilakukan sebanyak 48 kali yang kemudian dipecah menjadi 6 jenis kumpulan aturan berdasarkan parameter jari – jari untuk analisa varian. Sehingga nantinya tiap 1 kumpulan aturan berisi 8 jenis aturan yang nilai jari – jarinya berbeda namun nilai *accept ratio* dan *reject rationya* sama. Dari tiap kumpulan aturan akan dilakukan analisa varian untuk menentukan aturan terpilih, sehingga hasil akhir pada pengujian ketiga pada pengujian tahap satu akan menghasilkan 6 aturan terpilih. Hasil selengkapnya untuk pengujian ketiga ditunjukkan pada Tabel 5.7 dan Tabel 5.8.

Tabel 5.7 Tabel hasil pengujian ketiga tahap satu
(*reject ratio* 0,15 – 0,35)

Jumlah Data Latih = 100						
Jari – jari	$ar = 0,5$ dan $rr = 0,15$		$ar = 0,6$ dan $rr = 0,25$		$ar = 0,7$ dan $rr = 0,35$	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0094	10	0,0013	5	0,0005	3
0,4	0,0019	5	0,0006	3	0,0002	2
0,5	0,0009	4	0,0002	2	0,0002	2
0,6	0,0002	2	0,0002	2	0,0002	2
0,7	0,0002	2	0,0002	2	0,0002	2
0,8	0,0002	2	0,0002	2	0,0002	2
0,9	0,0002	2	0,0002	2	0,0002	2
1	0,0002	2	0,0002	2	0,0002	2

Tabel 5.8 Tabel hasil pengujian ketiga tahap satu
(*reject ratio* 0,4 – 0,5)

Jumlah Data Latih = 100						
Jari – jari	$ar = 0,8$ dan $rr = 0,4$		$ar = 0,85$ dan $rr = 0,45$		$ar = 0,9$ dan $rr = 0,5$	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0005	3	0,0005	3	0,0002	2
0,4	0,0002	2	0,0002	2	0,0002	2
0,5	0,0002	2	0,0002	2	0,0002	2
0,6	0,0002	2	0,0002	2	0,0002	2
0,7	0,0002	2	0,0002	2	0,0002	2
0,8	0,0002	2	0,0002	2	0,0002	2
0,9	0,0002	2	0,0002	2	0,0002	2
1	0,0002	2	0,0002	2	0,0002	2

Dari Tabel 5.7 dan Tabel 5.8 dapat dilihat bahwa terdapat 6 jenis aturan terpilih dengan parameter – parameter yang berbeda yang dihasilkan berdasarkan nilai batasan varian terkecil pada masing – masing *accept ratio* dan *reject ratio*. Aturan – aturan yang dihasilkan ditunjukkan pada Tabel 5.9.

Tabel 5.9 Tabel aturan pengujian ketiga tahap satu

Jenis Aturan (<i>r, ar, rr</i>)	Batasan Varian	Jumlah <i>cluster</i>
Aturan 13 (0,6; 0,5; 0,15)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and $NK = (Z1 = (-3.373153135453582E-5 \times IPK) + (6.92185026735851E-7 \times SKS) + (2.041865618714789E-7 \times Prestasi) + (3.4086572611794903E-13 \times X14) + 2.578003769580412E-5$ [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and $NK = (Z2 = (-8.014024569291678E-6 \times IPK) + (4.47038683848433E-9 \times SKS) + (1.7868814511379227E-8 \times Prestasi) + (2.3002468062154947E-13 \times X14) + 1.000029913210824)$		
Aturan 14 (0,5; 0,6; 0,25)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and $NK = (Z1 = (-8.276801192989939E-5 \times IPK) + (1.8470352934191052E-6 \times SKS) + (2.958859977614535E-8 \times Prestasi) + (2.6921752093460083E-13 \times X14) + 6.120949261943407E-5)$ [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and $NK = (Z2 = (-2.9696273884383118E-8 \times IPK) + (-2.27856622458944E-12 \times SKS) + (7.894997156032701E-11 \times Prestasi) + (1.0249634410116172E-15 \times X14) + 1.000000111387834)$		
Aturan 15 (0,4; 0,7; 0,35)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and $NK = (Z1 = (-6.962945273212336E-11 \times IPK) + (2.2173917500308075E-12 \times SKS) + (7.234972498928174E-13 \times Prestasi) + (3.9380114004873646E-19 \times X14) + -3.021951498464984E-11)$ [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and $NK = (Z2 = (-1.2647660696529783E-12 \times IPK) + (8.326672684688674E-16 \times SKS) + (1.0949574580365606E-14 \times Prestasi) + (4.309280119156253E-20 \times X14) + 1.000000000004576)$		
Aturan 16 (0,4; 0,8; 0,4)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and $NK = (Z1 = (-6.962945273212336E-11 \times IPK) + (2.2173917500308075E-12 \times SKS) + (7.234972498928174E-13 \times Prestasi) + (3.9380114004873646E-19 \times X14) + -3.021951498464984E-11)$ [R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and $NK = (Z2 = (-1.2647660696529783E-12 \times IPK) + (8.326672684688674E-16 \times SKS) + (1.0949574580365606E-14 \times Prestasi) + (4.309280119156253E-20 \times X14) + 1.000000000004576)$		
Aturan 17 (0,4; 0,85; 0,45)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and		

$NK = (Z1 = (-6.962945273212336E-11 \times IPK) + (2.2173917500308075E-12 \times SKS) + (7.234972498928174E-13 \times Prestasi) + (3.9380114004873646E-19 \times X14) + -3.021951498464984E-11)$		
<p>[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and</p> $NK = (Z2 = (-1.2647660696529783E-12 \times IPK) + (8.326672684688674E-16 \times SKS) + (1.0949574580365606E-14 \times Prestasi) + (4.309280119156253E-20 \times X14) + 1.000000000004576)$		
Aturan 18 (0,3; 0,9; 0,5)	0,0002	2
<p>[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and</p> $NK = (Z1 = (-1.0895383999960306E-18 \times IPK) + (4.5695650916665136E-20 \times SKS) + (1.3325893340195784E-20 \times Prestasi) + (2.354945632186075E-27 \times X14) + -1.6664234993279883E-18)$		
<p>[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and</p> $NK = (Z2 = (5.115907697472721E-13 \times IPK) + (-1.6042722705833512E-14 \times SKS) + (-2.8171909249863347E-15 \times Prestasi) + (1.5617169965001163E-21 \times X14) + 1.0000000000002842)$		

5.2.1.4 Hasil Pengujian Keempat

Pada pengujian keempat, proses pembentukan aturan dilakukan sebanyak 48 kali yang kemudian dipecah menjadi 6 jenis kumpulan aturan berdasarkan parameter jari – jari untuk analisa varian. Sehingga nantinya tiap 1 kumpulan aturan berisi 8 jenis aturan yang nilai jari – jarinya berbeda. Berbeda dengan pengujian pertama, kedua, dan ketiga pengujian keempat mencoba mengetahui pengaruh parameter *accept ratio* dan *reject ratio* terhadap pembentukan aturan. Data latih yang akan dipilih adalah jumlah data latih dengan nilai varian terkecil. Dari hasil pengujian pertama sampai ketiga didapatkan varian terkecil pada 100 jumlah data latih. Hasil selengkapnya untuk pengujian ketiga ditunjukkan pada Tabel 5.10 dan Tabel 5.11.

Tabel 5.10 Tabel hasil pengujian keempat tahap satu
(*accept ratio* = 0,5; *reject ratio* 0,15 – 0,5)

Jumlah Data Latih = 100						
Jari – jari	<i>ar</i> = 0,5 dan <i>rr</i> = 0,15		<i>ar</i> = 0,5 dan <i>rr</i> = 0,3		<i>ar</i> = 0,5 dan <i>rr</i> = 0,5	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0094	10	0,0010	4	0,0002	2
0,4	0,0019	5	0,0002	2	0,0002	2
0,5	0,0009	4	0,0002	2	0,0002	2
0,6	0,0002	2	0,0002	2	0,0002	2
0,7	0,0002	2	0,0002	2	0,0002	2
0,8	0,0002	2	0,0002	2	0,0002	2
0,9	0,0002	2	0,0002	2	0,0002	2
1	0,0002	2	0,0002	2	0,0002	2

Tabel 5.11 Tabel hasil pengujian keempat tahap satu
(*reject ratio* = 0,25; *accept ratio* 0,5 – 0,9)

Jumlah Data Latih = 100						
Jari – jari	<i>ar</i> = 0,5 dan <i>rr</i> = 0,25		<i>ar</i> = 0,7 dan <i>rr</i> = 0,25		<i>ar</i> = 0,9 dan <i>rr</i> = 0,25	
	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster	Nilai varian	Jumlah cluster
0,3	0,0013	5	0,0013	5	0,0013	5
0,4	0,0006	3	0,0006	3	0,0006	3
0,5	0,0002	2	0,0002	2	0,0002	2
0,6	0,0002	2	0,0002	2	0,0002	2
0,7	0,0002	2	0,0002	2	0,0002	2
0,8	0,0002	2	0,0002	2	0,0002	2
0,9	0,0002	2	0,0002	2	0,0002	2
1	0,0002	2	0,0002	2	0,0002	2

Dari Tabel 5.9 dan Tabel 5.10 dapat dilihat bahwa terdapat 6 jenis aturan terpilih dengan parameter – parameter yang berbeda yang dihasilkan berdasarkan nilai batasan varian terkecil pada masing – masing *accept ratio* dan *reject ratio*. Aturan – aturan yang dihasilkan ditunjukkan pada Tabel 5.12.

Tabel 5.12 Tabel aturan pengujian keempat tahap satu

Jenis Aturan (<i>r, ar, rr</i>)	Batasan Varian	Jumlah cluster
Aturan 19 (0,6; 0,5; 0,15)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-3.373153135453582E-5 x IPK) + (6.92185026735851E-7 x SKS) + (2.041865618714789E-7 x Prestasi) + (3.4086572611794903E-13 x X14) + 2.578003769580412E-5)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-8.014024569291678E-6 x IPK) + (4.47038683848433E-9 x SKS) + (1.7868814511379227E-8 x Prestasi) + (2.3002468062154947E-13 x X14) + 1.000029913210824)		
Aturan 20 (0,4; 0,5; 0,3)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-6.962945273212336E-11 x IPK) + (2.2173917500308075E-12 x SKS) + (7.234972498928174E-13 x Prestasi) + (3.9380114004873646E-19 x X14) + -3.021951498464984E-11)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-1.2647660696529783E-12 x IPK) + (8.326672684688674E-16 x SKS) + (1.0949574580365606E-14 x Prestasi) + (4.309280119156253E-20 x X14) + 1.000000000004576)		
Aturan 21 (0,3; 0,5; 0,5)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-1.0895383999960306E-18 x IPK) + (4.5695650916665136E-20 x SKS) + (1.3325893340195784E-20 x Prestasi) + (2.354945632186075E-27 x X14) +		

-1.6664234993279883E-18)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (5.115907697472721E-13 x IPK) + (-1.6042722705833512E-14 x SKS) + (-2.8171909249863347E-15 x Prestasi) + (1.5617169965001163E-21 x X14) + 1.0000000000002842)		
Aturan 22 (0,5; 0,5; 0,25)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-8.276801192989939E-5 x IPK) + (1.8470352934191052E-6 x SKS) + (2.958859977614535E-8 x Prestasi) + (2.6921752093460083E-13 x X14) + 6.120949261943407E-5)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-2.9696273884383118E-8 x IPK) + (-2.27856622458944E-12 x SKS) + (7.894997156032701E-11 x Prestasi) + (1.0249634410116172E-15 x X14) + 1.000000111387834)		
Aturan 23 (0,5; 0,7; 0,25)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-8.276801192989939E-5 x IPK) + (1.8470352934191052E-6 x SKS) + (2.958859977614535E-8 x Prestasi) + (2.6921752093460083E-13 x X14) + 6.120949261943407E-5)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-2.9696273884383118E-8 x IPK) + (-2.27856622458944E-12 x SKS) + (7.894997156032701E-11 x Prestasi) + (1.0249634410116172E-15 x X14) + 1.000000111387834)		
Aturan 24 (0,4; 0,9; 0,25)	0,0002	2
[R1] IF IPK is center11 and SKS is center12 and Prestasi is center13 and KE is center14 and NK = (Z1 = (-8.276801192989939E-5 x IPK) + (1.8470352934191052E-6 x SKS) + (2.958859977614535E-8 x Prestasi) + (2.6921752093460083E-13 x X14) + 6.120949261943407E-5)		
[R2] IF IPK is center21 and SKS is center22 and Prestasi is center23 and KE is center24 and NK = (Z2 = (-2.9696273884383118E-8 x IPK) + (-2.27856622458944E-12 x SKS) + (7.894997156032701E-11 x Prestasi) + (1.0249634410116172E-15 x X14) + 1.000000111387834)		

5.2.1. Hasil Pengujian Tahap Dua

Dari masing – masing hasil pengujian pada pengujian tahap satu menghasilkan 6 aturan terpilih. Dari aturan terpilih inilah akan dijadikan aturan pada proses inferensi menggunakan sistem inferensi *fuzzy* model Sugeno orde – satu. Dari hasil inferensi inilah yang akan dijadikan acuan untuk mengetahui tingkat akurasi dari masing – masing aturan terpilih dengan menguji 30 data uji.

5.2.2.1. Hasil Pengujian Pertama (60 Data Latih)

Untuk pengujian pertama pada pengujian tahap satu menghasilkan 6 jenis aturan. Hasil akurasi untuk masing – masing jenis aturan pada pengujian pertama ditunjukkan pada Tabel 5.13.

Tabel 5.13 Tabel akurasi hasil pengujian pertama

Jenis Aturan (r, ar, rr)	Jumlah Aturan	Nilai Batasan Penerima	Nilai Kelayakan		Akurasi
			Benar	Salah	
Aturan 1 (0,6; 0,5; 0,15)	2	10	28	2	93,33%
Aturan 2 (0,5; 0,6; 0,25)	2	10	28	2	93,33%
Aturan 3 (0,5; 0,7; 0,35)	2	10	28	2	93,33%
Aturan 4 (0,5; 0,8; 0,4)	2	10	28	2	93,33%
Aturan 5 (0,4; 0,85; 0,45)	2	10	28	2	93,33%
Aturan 6 (0,4; 0,9; 0,5)	2	10	28	2	93,33%

Berdasarkan Tabel 5.13 dapat dilihat bahwa tingkat akurasi yang dihasilkan konvergen, yaitu 93,33% pada jumlah 2 aturan. Dengan 1 data uji salah yaitu sama – sama pada indeks ke-42. Jumlah aturan yang dihasilkan untuk ke-6 aturan adalah sama, yaitu pada jumlah *cluster* 2.

5.2.2.2. Hasil Pengujian Kedua (80 Data Latih)

Untuk pengujian kedua pada pengujian tahap satu menghasilkan 6 jenis aturan. Hasil akurasi untuk masing – masing jenis aturan pada pengujian kedua ditunjukkan pada Tabel 5.14.

Tabel 5.14 Tabel akurasi hasil pengujian kedua

Jenis Aturan (r, ar, rr)	Jumlah Aturan	Nilai Batasan Penerima	Nilai Kelayakan		Akurasi
			Benar	Salah	
Aturan 7 (0,6; 0,5; 0,15)	2	10	28	2	93,33%
Aturan 8 (0,5; 0,6; 0,25)	2	10	28	2	93,33%
Aturan 9 (0,4; 0,7; 0,35)	2	10	28	2	93,33%
Aturan 10 (0,4; 0,8; 0,4)	2	10	28	2	93,33%
Aturan 11 (0,3; 0,85; 0,45)	2	10	28	2	93,33%
Aturan 12 (0,3; 0,9; 0,5)	2	10	28	2	93,33%

Berdasarkan Tabel 5.14 dapat dilihat bahwa tingkat akurasi yang dihasilkan konvergen, yaitu 93,33% pada jumlah 2 aturan. Dengan 1 data uji salah yaitu sama – sama pada indeks ke-9. Jumlah aturan yang dihasilkan untuk ke-6 aturan adalah sama, yaitu pada jumlah *cluster* 2.

5.2.2.3. Hasil Pengujian Ketiga (100 Data Latih)

Untuk pengujian ketiga pada pengujian tahap satu menghasilkan 6 jenis aturan. Hasil akurasi untuk masing – masing jenis aturan pada pengujian ketiga ditunjukkan pada Tabel 5.15.

Tabel 5.15 Tabel akurasi hasil pengujian ketiga

Jenis Aturan (r, ar, rr)	Jumlah Aturan	Nilai Batasan Penerima	Nilai Kelayakan		Akurasi
			Benar	Salah	
Aturan 7 (0,6; 0,5; 0,15)	2	10	28	2	93,33%
Aturan 8 (0,5; 0,6; 0,25)	2	10	28	2	93,33%
Aturan 9 (0,4; 0,7; 0,35)	2	10	28	2	93,33%
Aturan 10 (0,4; 0,8; 0,4)	2	10	28	2	93,33%
Aturan 11 (0,3; 0,85; 0,45)	2	10	28	2	93,33%
Aturan 12 (0,3; 0,9; 0,5)	2	10	28	2	93,33%

Berdasarkan Tabel 5.15 dapat dilihat bahwa tingkat akurasi yang dihasilkan konvergen, yaitu 93,33% pada jumlah 2 aturan. Dengan 1 data uji salah yaitu sama – sama pada indeks ke-9. Jumlah aturan yang dihasilkan untuk ke-6 aturan adalah sama, yaitu pada jumlah *cluster* 2.

5.2.2.4. Hasil Pengujian Keempat

Untuk pengujian keempat pada pengujian tahap satu menghasilkan 6 jenis aturan. Hasil akurasi untuk masing – masing jenis aturan pada pengujian keempat ditunjukkan pada Tabel 5.16.

Tabel 5.16 Tabel akurasi hasil pengujian keempat

Jenis Aturan (r, ar, rr)	Jumlah Aturan	Nilai Batasan Penerima	Nilai Kelayakan		Akurasi
			Benar	Salah	
Aturan 7 (0,6; 0,5; 0,15)	2	10	28	2	93,33%
Aturan 8 (0,4; 0,5; 0,3)	2	10	28	2	93,33%

Aturan 9 (0,3 0,5; 0,5)	2	10	28	2	93,33%
Aturan 10 (0,5; 0,5; 0,25)	2	10	28	2	93,33%
Aturan 11 (0,5; 0,7; 0,25)	2	10	28	2	93,33%
Aturan 12 (0,4; 0,9; 0,25)	2	10	28	2	93,33%

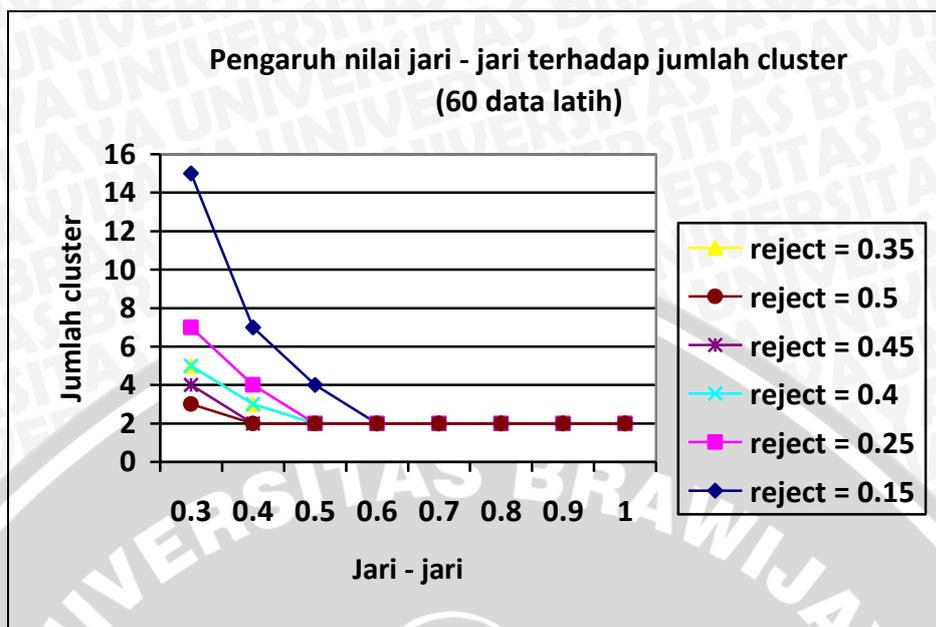
Berdasarkan Tabel 5.12 dapat dilihat bahwa tingkat akurasi yang dihasilkan konvergen, yaitu 93,33% pada jumlah 2 aturan. Dengan 1 data uji salah yaitu sama – sama pada indeks ke-9. Jumlah aturan yang dihasilkan untuk ke-6 aturan adalah sama, yaitu pada jumlah *cluster* 2.

5.3. Analisa Hasil

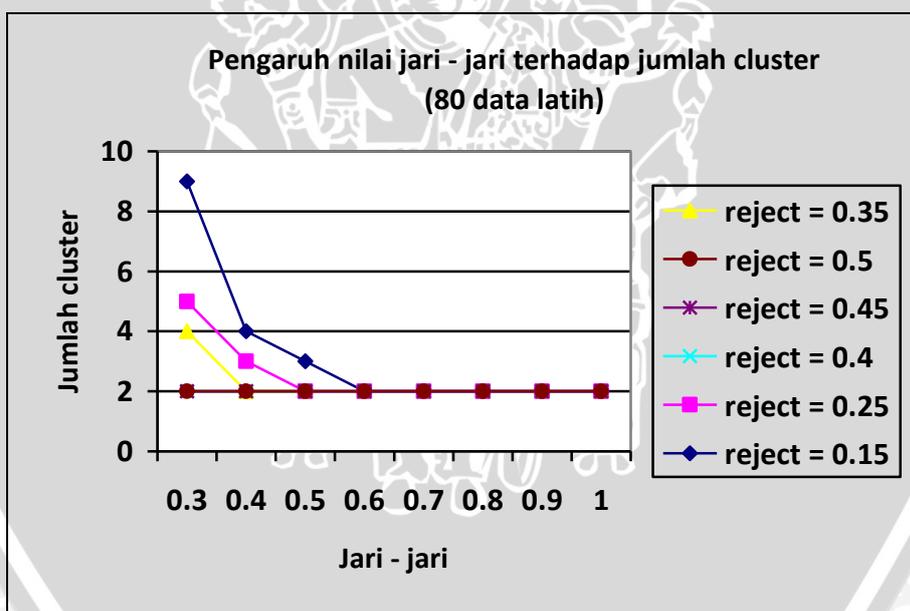
Hasil pengujian tahap satu dan pengujian tahap dua akan dianalisis lebih lanjut untuk melihat hasil pengujian agar dapat diambil kesimpulan apakah tujuan penelitian telah tercapai atau belum.

5.3.1. Analisa Hasil Pengujian Tahap Satu

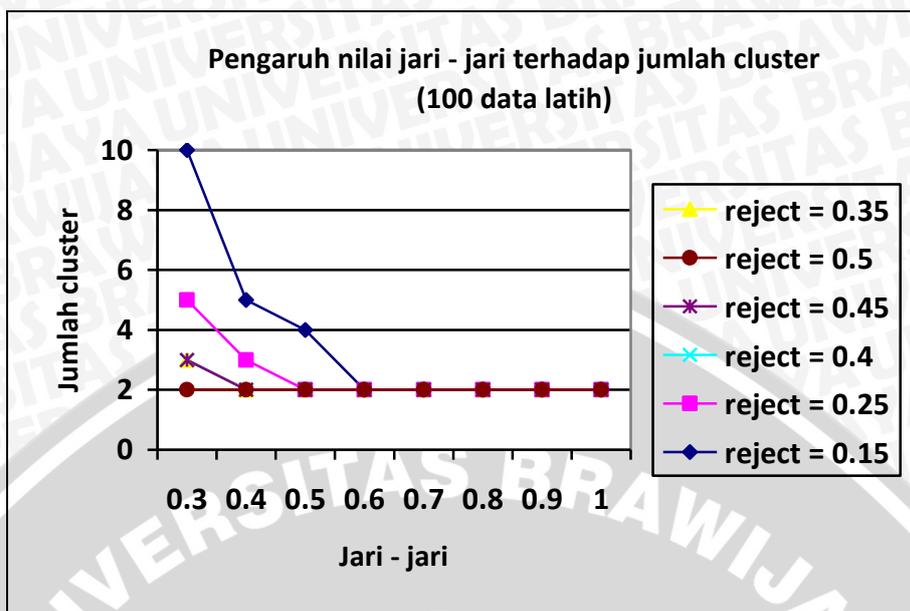
Dalam pengujian tahap satu terdapat empat tahapan pengujian dengan jumlah data latih berbeda. Parameter masukan berupa jari – jari, *accept ratio*, dan *reject ratio* dijadikan parameter untuk mengetahui pengaruh dari parameter tersebut terhadap aturan yang terbentuk. Pada pengujian keempat tahap satu menunjukkan bahwa parameter *accept ratio* tidak memberikan pengaruh terhadap pembentukan hasil *cluster*. Sedangkan parameter jari – jari dan *reject ratio* berbanding terbalik dengan nilai batasan varian dan jumlah *cluster*. Semakin besar nilai jari – jari atau *reject ratio*, maka semakin kecil nilai batasan varian atau jumlah *cluster* yang dihasilkan. Pengaruh jari – jari terhadap jumlah *cluster* yang terbentuk pada pengujian pertama sampai ketiga pada pengujian tahap satu ditunjukkan pada Gambar 5.1, Gambar 5.2, dan Gambar 5.3. Sedangkan hasil pengujian keempat pada pengujian tahap satu yang membuktikan bahwa parameter *accept ratio* tidak memberikan pengaruh terhadap jumlah *cluster* ditunjukkan pada Gambar 5.4.



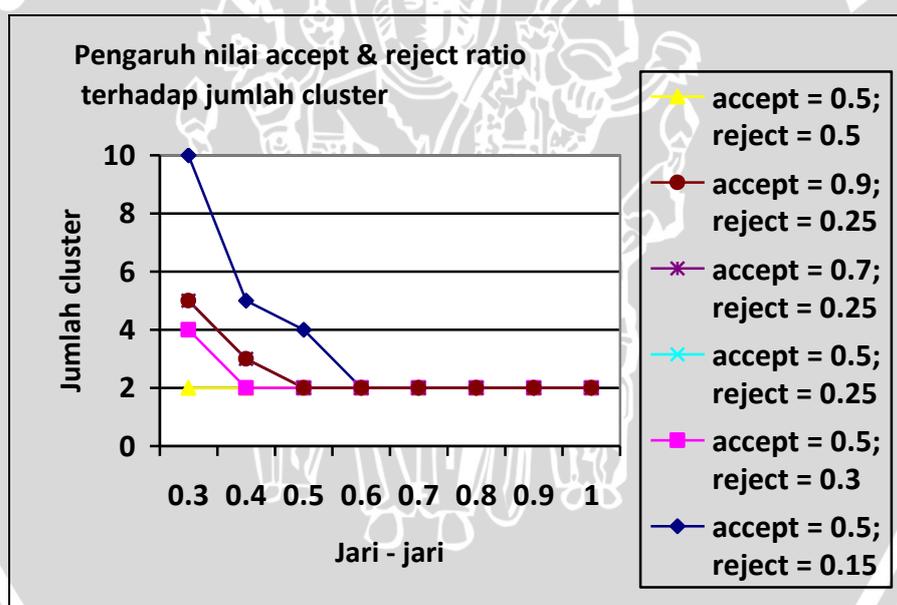
Gambar 5.1 Grafik hubungan nilai jari – jari terhadap jumlah *cluster* pada pengujian pertama tahap satu



Gambar 5.2 Grafik hubungan nilai jari – jari terhadap jumlah *cluster* pada pengujian kedua tahap satu



Gambar 5.3 Grafik hubungan nilai jari – jari terhadap jumlah *cluster* pada pengujian ketiga tahap satu



Gambar 5.4 Grafik pengaruh *accept ratio* dan *reject ratio* terhadap jumlah *cluster* pada pengujian keempat tahap satu

Berdasarkan Gambar 4.1, Gambar 4.2, dan Gambar 4.3 dapat dilihat bahwa nilai jari – jari berpengaruh pada jumlah *cluster* yang dihasilkan dan cenderung pergerakan jumlah *cluster* berbanding terbalik dengan nilai jari – jari dan *reject ratio*, walaupun untuk tiap pengujian menghasilkan jumlah *cluster* yang bervariasi.

Variasi jumlah *cluster* yang dihasilkan berdasarkan jari – jari dan dipengaruhi juga oleh jumlah data latih dan nilai *reject ratio*. Jumlah data latih memberikan pengaruh terhadap hasil *cluster* dikarenakan pada proses *clustering* diperlukan batas bawah dan batas atas tiap parameter yang berpengaruh pada nilai normalisasi data. Dengan jumlah data latih berbeda, dapat diasumsikan bahwa nilai normal data setelah proses normalisasi dimungkinkan bervariasi. Sedangkan nilai *accept ratio* tidak memberi pengaruh terhadap hasil *cluster* dan hal tersebut ditunjukkan pada grafik pengaruh nilai *accept ratio* dan *reject ratio* pada Gambar 4.4.

Jumlah *cluster* yang terbentuk konvergen mulai dari jari – jari 0,6 yaitu 2 *cluster*. Dalam setiap pengujian, nilai batasan varian terkecil cenderung stabil dan berada pada kisaran 0,0002 sampai 0,0006. Sama halnya seperti jumlah *cluster*, nilai batasan varian juga berbanding terbalik dengan nilai jari – jari dan *accept ratio* dan berbanding lurus dengan jumlah *cluster*. Selain itu nilai batasan varian dipengaruhi oleh nilai *variance between cluster* dan nilai *variance within cluster* pada tiap *cluster* yang dihasilkan. Sedangkan nilai *variance between cluster* dan nilai *variance within cluster* dipengaruhi oleh keanggotaan titik data terhadap *cluster* yang terbentuk.

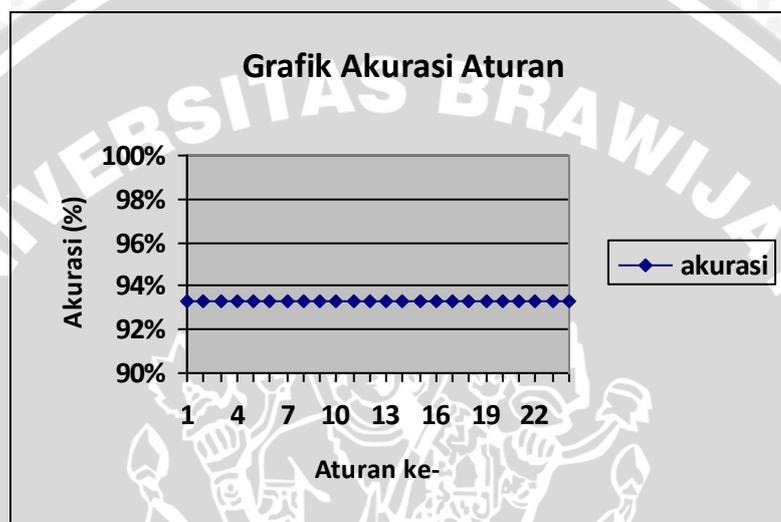
Jumlah *cluster* yang dihasilkan pada pengujian tahap satu akan dilanjutkan pada pengujian tahap dua, dimana batasan jumlah *cluster* yang digunakan pada pengujian tahap dua minimal adalah dua aturan. Dikarenakan pada hasil pengujian tahap satu tidak ada jumlah *cluster* sama dengan 1, maka semua hasil *cluster* pada pengujian tahap satu dapat digunakan pada pengujian tahap dua.

5.3.2. Analisa Hasil Pengujian Tahap Dua

Pada pengujian tahap dua terdapat empat tahap pengujian yang menghasilkan uji akurasi untuk 24 jenis aturan. Tidak ada perubahan akurasi pada pengujian tahap dua, yaitu sama – sama pada nilai akurasi sebesar 93,33% pada aturan yang berjumlah 2. Pada tiap pengujian terdapat variasi indeks data uji yang salah. Pada pengujian pertama dan kedua untuk tahap dua, data uji yang salah masing – masing berada pada indeks 9 dan 42. Sedangkan untuk data uji yang salah pada pengujian ketiga dan keempat pada tahap dua adalah sama, yaitu pada

indeks 42. Hal ini dipengaruhi oleh pusat *cluster* yang dihasilkan pada proses *clustering*. Pusat *cluster* berperan besar pada keanggotaan suatu titik data terhadap *cluster* berdasarkan derajat keanggotaan. Sehingga dimungkinkan data uji sama memiliki derajat keanggotaan berbeda tergantung pusat *cluster* yang terbentuk.

Proses uji akurasi dilakukan dengan memperhatikan data uji yang benar kemudian dibagi banyaknya data uji. Grafik uji akurasi pada pengujian tahap dua ditunjukkan pada Gambar 5.5.



Gambar 5.5 Grafik akurasi aturan pengujian tahap dua

Dari Gambar 5.5 dapat dilihat bahwa akurasi konvergen pada nilai 93,33% pada aturan yang berjumlah 2. Hasil tersebut berasal dari proses pembobotan nilai kelayakan yang dicari berdasarkan parameter data uji menggunakan sistem inferensi *fuzzy* model Sugeno. Nilai kelayakan yang dihasilkan oleh sistem kemudian dirangking. Nilai kelayakan yang berada pada nilai batasan penerima tertinggi akan direkomendasikan untuk diterima, sedangkan nilai kelayakan diluar rangking batasan penerima akan direkomendasikan tidak. Hasil inilah yang disesuaikan dengan data uji asli untuk dapat menentukan tingkat akurasi sistem.

Hasil penelitian ini membuktikan bahwa pembangkitan jumlah aturan (*cluster*) yang tepat dapat memberikan akurasi tinggi, terbukti dari pembangkitan aturan dengan jumlah *cluster* yang tidak optimal hasil akurasinya pun dibawah dari akurasi dengan jumlah *cluster* yang optimal, sehingga dengan demikian hasil penelitian ini juga menjawab kekurangan dari penelitian sebelumnya seperti yang termuat dalam latar belakang skripsi.

BAB VI PENUTUP

2.7. Kesimpulan

Kesimpulan yang didapat dari hasil penelitian ini diantaranya adalah sebagai berikut:

1. Metode *subtractive clustering* dapat membangkitkan jumlah *cluster* secara otomatis berdasarkan data latih dan inisialisasi nilai jari – jari dan *reject ratio*, tanpa perlu menginisialisasi awal jumlah *cluster* yang akan dibentuk. Hal ini dikarenakan algoritma *subtractive clustering* didasarkan atas ukuran densitas (potensi) titik – titik data dalam suatu ruang (variabel). Konsep dasar dari *subtractive clustering* adalah menentukan daerah – daerah dalam suatu variabel yang memiliki densitas tinggi terhadap titik – titik di sekitarnya. Titik dengan jumlah tetangga terbanyak akan dipilih sebagai pusat kluster. Titik yang sudah terpilih sebagai pusat kluster ini kemudian akan dikurangi densitasnya. Kemudian algoritma akan memilih titik lain yang memiliki tetangga terbanyak untuk dijadikan pusat kluster yang lain. Hal ini akan dilakukan berulang – ulang hingga semua titik diuji. Sedangkan parameter *accept ratio* tidak berpengaruh pada jumlah *cluster* yang dihasilkan.
2. Metode *subtractive clustering* dapat diimplementasikan dalam pembentukan aturan *fuzzy* pada rekomendasi beasiswa. Tahapan dalam pembentukan aturan dimulai dengan proses *clustering* dimana diperlukan inisialisasi awal untuk nilai jari – jari, *accept ratio*, *reject ratio*, batas atas serta batas bawah tiap atribut data, dan *squash factor* = 1,25. Dari hasil proses *clustering* akan dilakukan analisa varian untuk mengetahui nilai batasan varian pada tiap hasil *clustering*. Hasil *clustering* yang memiliki nilai batasan varian terkecil akan dijadikan bahan untuk proses ekstraksi aturan *fuzzy* menggunakan metode *Least Square Estimator*(LSE).
3. Hasil aturan *fuzzy* yang terbentuk dari serangkaian proses pembelajaran, yang terdiri dari proses *clustering*, analisa varian, dan ekstraksi aturan *fuzzy*, kemudian aturan tersebut diterapkan pada sistem inferensi *fuzzy* model Sugeno untuk mengetahui nilai kelayakan pemohon beasiswa berdasarkan atribut IPK,SKS, prestasi, dan keadaan ekonomi. Setelah nilai kelayakan diperoleh,

akan diketahui akurasi sistem. Berdasarkan percobaan pengujian pembentukan aturan diperoleh jumlah aturan yang konvergen, yaitu 2 aturan. Untuk pengujian tahap dua diperoleh akurasi sistem yang konvergen, yaitu sebesar 93,33% pada jumlah 2 aturan.

6.2. Saran

Untuk pengembangan sistem lebih lanjut dari hasil penelitian ini, maka penulis memberikan saran – saran sebagai berikut :

- 4) Data set yang digunakan dibatasi berdasarkan fakultas bukan universitas.
- 5) Dilakukan percobaan dalam mencari nilai derajat keanggotaan dengan menggunakan metode lainnya.
- 6) Digunakan metode analisa selain metode analisa varian, agar dapat mengestimasi hasil *cluster* yang optimal dengan waktu eksekusi yang lebih efektif.



DAFTAR PUSTAKA

- [AKH-10] Arapoglou, Roi.Kolomvatos, Kostas dan Hadjiefthymiades, Stathes.*Buye Agent Decision Process Based on Automatic Fuzzy Rules Generation Mehods*. http://p-comp.di.uoa.gr/pubs/WCCI_f427.pdf. [17 November 2012].
- [ARI-09] Arifin.2009.Perbandingan Metode *Fuzzy Clustering* dan *Fuzzy Subtractive Clustering* pada Pengelompokan Siswa Kelas IX SMP Al Falah Deltasari Sidoarjo. <http://digilib.its.ac.id/perbandingan-metode-fuzzy-kmeans-clustering-dan-fuzzy-subtractive-clustering-pada-pengelompokan-siswa-kelas-ix-smp-al-falah-deltasari-sidoarjo-4853.html>. [29 November 2012].
- [BEI-11] Ilham, B.Priyambodo.2011.Impelementasi Metode Single Linkage untuk Menentukan Kinerja Agen pada *Call Centre* Berbasis *Asterisk for Java*. <http://www.eepis-its.edu/uploadta/downloadmk.php?id=1374>. [5 Januari 2013].
- [BAD-05] Badriyah, T.2005.*Clustering Analysis*. <http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/5Clustering.pdf>. [3 Januari 2013].
- [CHS-97] Chiu, Stephen I.*An Efficient Method for Extracting Fuzzy Classification Rules from High Dimensional Data*. <http://chius.homestead.com/files/fuzzyclassif.pdf>. [31 Desember 2012].
- [DAT-05] Larose, Daniel T.2005.*Discovering Knowledge in Data: an Introduction to Data Mining*.John & Wiley & Sons, Inc.New Jersey.
- [DIR-11] Direktorat Pembelajaran dan Kemahasiswaa Direktorat Jenderal Pendidikan Tinggi.2011.*Beasiswa PPA dan Bantuan Belajar Mahasiswa (BBM)*. http://www.dikti.go.id/?page_id=397&lang=id. [26 November 2012].
- [HUA-09] Husnayadi, Arief.2009.Penentuan Nilai Kelayakan Penerima Beasiswa dengan Menggunakan Metode *Fuzzy Mamdani*.Skripsi.Universitas Brawijaya.Malang.
- [IDN-12] Nurlaili, Ida. 2012.Pembangkitan Aturan *Fuzzy* Menggunakan Algoritma *K-Means*.Skripsi.Universitas Brawijaya.Malang.

- [JAM-97] Jang, J.S.R., Sun, C.T., Mizutani,E., (1997), *Neuro-Fuzzy and Soft Computing*, Prentice-Hall International, New Jersey.
- [KUE-09] Kusrini dan TL, Emha.2009.Algoritma Data Mining.Edisi I.Andi Offset.Yogyakarta.
- [NUG-06] Nugraha, Dany.dkk.2006.Diagnosis Gangguan Sistem Urinari pada Anjing dan Kucing Menggunakan VFI 5.Institut Pertanian Bogor.Bogor.
- [PPP-94] Pusat Pembinaan dan Pengembangan Bahasa.1994.Kamus Besar Bahasa Indonesia.Edisi Kedua.Departemen Pendidikan dan Kebudayaan Balai Pustaka.Jakarta.
- [SAS-10] Santoso, Singgih.2010.Statistika Multivariat.PT Gramedia.Jakarta.
- [SKH-10] Kusumadewi, Sri dan Purnomo, Hari.Aplikasi logika *fuzzy* untuk pendukung keputusan.2010.Edisi kedua.Yogyakarta:Graha Ilmu
- [YUA-07] Agusta, Yudi.2007,"*K-Means* – Penerapan, Permasalahan dan Metode Terkait",[http:// yudiagusta.files.wordpress.com/2008/03/k-means](http://yudiagusta.files.wordpress.com/2008/03/k-means). [26 November 2012].



LAMPIRAN

Lampiran 1: Data Latih, Halaman 133-135

IPK	SKS	Pres tasi	Pendapatan		Pengeluaran				KE	NK
			Gaji Ayah	Gaji Ibu	Listrik	PBB	PDAM	Telepon		
3,728	114	3	1000000	0	39000	5034,8333	7800	0	948165,1667	1
3,728	114	0	3000000	0	70000	11390,0000	0	100000	2818610	1
3,761	114	0	2000000	1000000	100000	1032,0833	0	85125	2813842,917	1
3,393	112	0	2434600	2475600	26685	5808,3333	24000	121540	4732166,667	0
3,239	112	0	400000	500000	20000	5057,0000	0	0	874943	0
3,257	112	0	2852100	0	80000	3102,0833	0	0	2768997,917	0
3,489	112	0	1000000	500000	180465	3003,3333	0	0	1316531,667	0
3,710	107	0	1500000	0	175000	2150,0000	0	0	1142850,000	1
3,674	114	0	1000000	0	68025	10362,5000	0	77521	844091,5	1
3,761	114	0	1700000	0	40000	0,0000	0	0	1660000	1
3,117	113	0	1100000	0	70000	0,0000	0	0	1030000	0
3,824	112	3,5	600000	0	20000	0,0000	0	0	580000	1
3,775	114	0	7288563	6626242	515370	12135,8333	0	67560	13319739,17	1
3,904	112	3,5	3042500	2501000	159405	8583,3333	0	36340	5339171,667	1
3,966	90	17,5	3000000	0	268380	6712,5000	93900	60071	2570936,5	1
3,007	112	3	1000000	2838000	94200	3883,9166	0	60000	3679916,083	0
2,934	112	10	1000000	0	20000	539,0000	0	0	979461	0
2,868	112	0	1000000	0	13490	0,0000	0	38060	948450	0
3,180	104	0	750000	0	47000	3453,7500	0	0	699546,25	0
3,230	104	0	2705700	0	96800	375,0000	7000	0	2601525	0
3,854	110	0	2000000	0	76675	13683,3333	0	60003	1849638,667	1
3,854	110	0	2000000	0	76675	13683,3333	0	60003	1849638,667	1
3,743	110	0	5100000	0	155000	41666,6666	0	0	4903333,333	1
3,792	113	0	700000	2269700	50000	15973,5000	0	0	2903726,5	1
3,704	113	7,5	326500	3054500	47950	4819,6666	31150	99700	3197380,333	1
3,075	109	0	1250000	0	43870	18624,1666	0	0	1187505,833	0
3,150	109	0	2179400	0	109410	4168,3333	15000	0	2050821,667	0
3,316	109	0	3000000	0	90000	1750,0000	0	100000	2808250	0
3,102	109	15	2979800	2510400	91115	2231,0000	54000	232620	5110234	0
3,233	109	0	1050000	0	40000	0,0000	0	0	1010000	0
3,064	108	0	2000000	0	0	0,0000	0	0	2000000	0
3,098	108	0	4082700	2920200	223065	11303,4166	0	50000	6718531,583	0
3,239	108	0	3230600	2663600	40000	1250,0000	0	0	5852950	0
3,295	108	0	1300000	0	0	0,0000	0	0	1300000	0
3,155	108	10	2705700	0	96800	375,0000	7000	0	2601525	0
3,004	108	0	5000000	0	105000	0,0000	142000	47000	4706000	0



2,955	108	0	750000	2000000	0	1916,6666	0	0	2748083,333	0
3,027	108	3	2033300	2935700	135105	632,5000	0	63529	4769733,5	0
2,871	108	0	5340000	1740000	149000	1916,6666	21000	107000	6801083,333	0
3,068	108	5	0	1000000	60000	666,6666	0	0	939333,3333	0
3,765	110	0	2500000	0	152370	5410,0000	46164	64016	2232040	1
3,679	110	0	900000	0	49115	3741,6666	25000	35435	786708,3333	1
3,696	111	10	2197400	1804000	25000	166,6666	0	0	3976233,333	1
3,724	110	0	2000000	0	169995	6978,3333	0	173964	1649062,667	1
3,625	112	6	2764810	3512402	70725	500,0000	20000	37745	6148242	1
3,433	111	0	8000000	0	250000	14333,3333	100000	250000	7385666,667	0
3,222	111	0	4000000	0	59000	9750,0000	0	96000	3835250	0
3,133	111	0	9666987	0	182600	9858,3333	74000	90506	9310022,667	0
3,456	111	0	2413700	0	55740	5120,1666	0	0	2352839,833	0
3,544	111	0	2500000	0	0	0,0000	0	0	2500000	0
3,033	111	7	3670700	0	188050	7380,8333	97100	114080	3264089,167	0
3,189	111	0	4129238	0	260335	21319,1666	0	85635	3761948,833	0
3,511	111	0	2353200	1085000	125930	5116,3333	0	32985	3274168,667	0
3,200	111	0	4000000	0	120000	0,0000	0	0	3880000	0
3,089	111	0	2100000	1500000	130200	8479,2500	0	0	3461320,75	0
3,653	114	5	0	500000	50000	416,6666	0	0	449583,3333	1
3,772	112	0	4612100	0	420760	2255,0000	41500	124389	4023196	1
3,812	112	5	1448000	0	51140	11822,9166	0	97166	1287871,083	1
3,788	101	0	600000	0	0	0,0000	0	0	600000	1
3,775	114	0	7500000	0	201000	1690,0000	45100	427641	6824569	1
3,100	111	0	2500000	0	200000	23750,0000	0	100000	2176250	0
3,044	111	5	11170823	0	206185	5638,4166	0	126377	10832622,58	0
3,156	111	0	3000000	0	185250	11651,3333	0	0	2803098,667	0
2,940	110	0	0	350000	26135	1891,6666	0	0	321973,3333	0
2,830	108	5	2000000	2000000	100000	9750,0000	0	150000	3740250	0
3,767	107	0	3232900	0	66900	4747,8333	0	36972	3124280,167	1
3,788	113	9	3414500	0	120785	6272,9167	50880	34870	3201692,083	1
3,717	114	0	2000000	0	70000	7385,6667	120000	0	1802614,333	1
3,697	104	20	4460000	0	0	52665,0000	22700	0	4382001,917	1
3,660	109	3	1600000	0	0	100000,0000	0	0	1498516,667	1
3,130	114	0	1323500	0	186999	9289,8333	22640	40860	1063711,167	0
3,424	114	0	0	13406000	177800	12725,6667	18000	70615	13126859,33	0
3,337	114	0	3000000	0	100000	0,0000	0	0	2900000	0
3,326	114	0	4567063	1425400	67405	4064,3333	0	50000	5870993,667	0
3,370	114	0	1000000	0	95490	2034,9167	0	45976	856499,0833	0
3,683	110	3	0	500000	18425	985,0000	0	0	480590	1
3,659	111	5	1500000	0	114105	4447,9167	80000	76028	1225419,083	1
3,656	101	5	1000000	0	90000	291,6667	0	35000	874708,3333	1

3,656	104	0	0	600000	33000	833,3333	23000	0	543166,6667	1
3,673	109	0	300000	700000	50000	625,0000	0	0	949375	1
3,391	113	7,5	1500000	0	75000	0,0000	0	55000	1370000	0
3,272	110	3	1500000	0	58485	38207,8333	0	165279	1238028,167	0
3,127	110	0	4200000	750000	165000	21431,2500	25000	75000	4663568,75	0
3,168	110	9	1200000	0	41000	416,6667	0	0	1158583,333	0
3,153	110	0	1500000	0	11055	3983,3333	84500	44220	1356241,667	0
3,295	110	10	900000	0	40000	3200,0000	0	120000	736800	0
3,384	110	10	1600000	0	74540	6166,6667	0	0	1519293,333	0
3,381	110	7	3115500	2624100	100935	3036,6667	31300	89932	5514396,333	0
3,705	110	0	2908631	1978200	118375	13019,1667	0	46880	4708556,833	1
3,683	110	0	500000	0	32670	4000,0000	0	0	463330	1
3,754	110	0	1500000	0	61000	3743,3333	0	0	1435256,667	1
3,709	110	0	3206000	2586000	100000	2326,0000	0	0	5689674	1
3,680	114	0	600000	0	40000	504,0000	0	0	559496	1
3,422	110	0	2151800	2813400	141975	5506,5000	94250	39705	4683763,5	0
3,336	110	0	750000	0	47000	3453,7500	0	0	699546,25	0
3,328	110	0	1831886	500000	27470	7155,4167	0	0	2297260,583	0
3,399	110	17,5	2800500	2688150	79690	10534,3333	59500	64357	5274568,667	0
3,062	104	18	1500000	0	0	4854,1667	0	0	1440496,667	0
3,226	104	16	4000000	0	0	3436,0833	36150	29587	3886919,167	0
3,548	104	0	4693410	0	0	9426,8333	0	0	4576998,75	0



Lampiran 2: Data Uji, Halaman 136

IPK	SKS	Pres tasi	Pendapatan		Pengeluaran				KE	NK
			Gaji Ayah	Gaji Ibu	Listrik	PBB	PDAM	Telepon		
3,761	114	10	0	3000000	120000	3915	0	0	2876085	1
3,652	114	9	1300000	0	50000	1583,333	0	0	1248417	1
3,63	114	10	5000000	0	162135	39586,67	26000	128480	4643798	1
3,832	110	0	800000	0	65000	4583,333	0	0	730416,7	1
3,698	110	0	850000	0	62245	9220	0	0	778535	1
3,726	102	0	1500000	2500000	0	28536	72000	270000	3629464	1
3,877	84	0	2700000	0	170000	25366,67	90000	53000	2361633	1
3,866	103	12,5	3000000	0	146680	9137,5	15500	65534	2763149	1
3,628	101	0	1000000	0	70000	3100	0	0	926900	1
3,705	110	0	2908631	1978200	118375	13019,17	0	46880	4708557	1
3,004	112	20	800000	0	78630	7463,25	202000	53723	458183,8	0
3,118	112	0	2674100	785000	129100	16470	0	100870	3212660	0
3,441	112	0	4282000	0	178660	7735,833	0	0	4095604	0
3,61	112	0	1988500	0	23900	1230	0	0	1963370	0
3,452	112	0	3200000	250000	90000	1375	0	0	3358625	0
3,529	112	2,5	2500000	2000000	54000	0	0	50000	4396000	0
3,588	112	5	953500	0	55090	1455,833	189400	57108	650446,2	0
3,268	112	0	3853600	0	145000	446,9167	0	40676	3667477	0
3,603	112	5	2613113	0	95175	3106	70000	0	2444832	0
2,944	111	0	0	4500000	42520	11307,25	0	0	4446173	0
2,887	109	0	0	650000	40000	933,75	0	55000	554066,3	0
3,177	109	0	1150000	0	60620	5263,333	71800	46020	966296,7	0
2,996	110	10	1500000	0	50000	854,1667	0	0	1449146	0
3,146	110	0	1304000	0	38390	6499,167	27000	53020	1179091	0
3,101	110	0	1500000	0	140000	15333,33	45000	0	1299667	0
3,575	110	0	1000000	0	80000	0	0	0	920000	0
3,474	113	0	0	1547000	67660	884,1667	0	0	1478456	0
3,026	113	0	0	750000	0	21252,08	0	0	728747,9	0
3,565	114	0	3000000	2000000	363567	10233,33	0	63224	4562976	0
3,05	101	9	3000000	0	0	3860	30440	30000	2891907	0