

**PEMBANGKITAN ATURAN UNTUK MENGETAHUI
HUBUNGAN ANTARA INDEKS PRESTASI MAHASISWA
TAHUN PERTAMA DENGAN INDEKS PRESTASI
KELULUSAN MENGGUNAKAN *FUZZY ID3***

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana
Komputer



Disusun oleh :

FATAYATUL YATALATHAF

NIM. 0810963039

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

LEMBAR PERSETUJUAN
PEMBANGKITAN ATURAN UNTUK MENGETAHUI
HUBUNGAN ANTARA INDEKS PRESTASI MAHASISWA
TAHUN PERTAMA DENGAN INDEKS PRESTASI
KELULUSAN MENGGUNAKAN *FUZZY* ID3

SKRIPSI



Disusun oleh :

FATAYATUL YATALATHAF

NIM. 0810963039

Telah diperiksa dan disetujui oleh :

Pembimbing I,

Pembimbing II,

Candra Dewi S.Kom, M.Sc

NIP. 197711142003122001

Drs. Muh. Arif Rahman M.Kom

NIP. 196604231991111001

LEMBAR PENGESAHAN
PEMBANGKITAN ATURAN UNTUK MENGETAHUI
HUBUNGAN ANTARA INDEKS PRESTASI MAHASISWA
TAHUN PERTAMA DENGAN INDEKS PRESTASI
KELULUSAN MENGGUNAKAN *FUZZY ID3*

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

FATAYATUL YATALATHAF

NIM. 0810963039

Skripsi ini telah diuji dan dinyatakan lulus tanggal 14 Januari 2013

Penguji I

Penguji II

Drs. Marji, MT.
NIP. 196708011992031001

Suprpto, S.T., M.T.
NIP. 197107271966031001

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom.

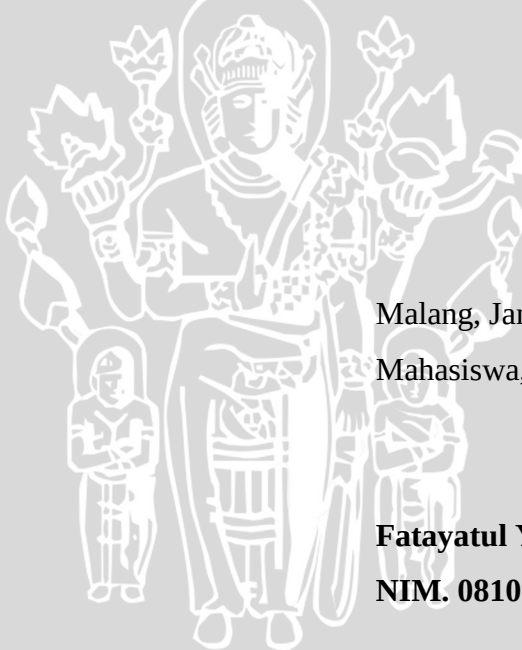
Mengetahui
Ketua Program Studi Ilmu Komputer

Drs. Marji, MT.
NIP. 196708011992031001

PERNYATAAN ORISINALITAS SKRIPSI

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

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



Malang, Januari 2013

Mahasiswa,

Fatayatul Yatalathaf

NIM. 0810963039

ABSTRAK

Fatayatul Yatalathaf. 2013. :Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama dengan Indeks Prestasi Kelulusan Menggunakan Fuzzy ID3

Dosen Pembimbing : Candra Dewi S.Kom, M.Sc. dan Drs. Muh. Arif Rahman M.Kom

Prestasi mahasiswa biasanya diukur dari tingginya nilai indeks prestasi yang diraih. Pada pengamatan data akademik mahasiswa Fakultas MIPA Universitas Brawijaya, ada kecenderungan nilai indeks prestasi mahasiswa tahun pertama yang tidak sejalan dengan indeks prestasi kelulusannya. Pada penelitian ini akan dilakukan pembangkitan aturan untuk mengetahui hubungan antara indeks prestasi mahasiswa tahun pertama dengan indeks prestasi kelulusannya. Metode yang digunakan dalam penelitian ini yaitu metode *fuzzy decision tree* dengan algoritma ID3. Tahap awal dilakukan pembentukan himpunan *fuzzy* pada data latih, kemudian pembentukan *tree* dengan algoritma *Iterative Dicotomiser 3* (ID3) dan menghasilkan aturan-aturan. Pengujian dilakukan terhadap aturan yang telah terbentuk berdasarkan kombinasi nilai *Fuzziness Control Threshold* (FCT) dan *Leaf Decision Threshold* (LDT). Pengujian ini menggunakan metode inferensi Mamdani yang dilakukan pada setiap jurusan/program studi dengan proporsi data latih yang berbeda-beda yaitu 30%, 40%, 50% dan 60%. Dari hasil proses pengujian, aturan terbanyak dihasilkan pada data jurusan/program studi Fisika dengan 21 aturan dan aturan terendah yaitu 4, yang terdapat pada semua jurusan/program studi. Sedangkan akurasi tertinggi diperoleh yaitu 93,548 %.

Kata kunci : indeks prestasi tahun pertama, indeks prestasi kelulusan, *fuzzy* ID3, FCT, LDT

ABSTRACT

Fatayatul Yatalathaf. 2013. :Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama dengan Indeks Prestasi Kelulusan Menggunakan Fuzzy ID3

Dosen Pembimbing : Candra Dewi S.Kom, M.Sc. dan Drs. Muh. Arif Rahman M.Kom

Generally student achievement is measured by the grade points of the student. On the students' academic data observation in Mathematics and Science Faculty of Brawijaya University, it is found a tendency that students' first year grade points are not linear with their graduation grade point. This research will be conducted on finding the pattern to determine the relationship between between students' first year grade points and students' graduation grade point.. The method used in this research is fuzzy decision tree with ID3 algorithm. On the first step, the researcher form the sets of fuzzy on the accustomed data then form the tree with Iterative Dicotomiser 3 (ID3) algorithm and generate patterns. The test is done toward the generated pattern based on the combination of Fuzziness Control Threshold (FCT) and Leaf Decision Threshold (LDT) value. This test is using Mamdani inference method done on each department/study program with different accustomed data as follow: 30%, 40%, 50%, and 60%. From test process result, the most pattern found on data of Physics department/study program. While the highest accuracy found is 93.548%

Key words: first year grade points, graduation grade points, fuzzy Iterative Dicotomiser 3 (ID3), Fuzziness Control Threshold (FCT), Leaf Decision Threshold (LDT)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, hanya dengan rahmat dan karunia yang telah diberikan kepada penulis, sehingga dapat menyelesaikan skripsi yang berjudul “Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan Menggunakan *Fuzzy ID3*”.

Skripsi ini merupakan salah satu syarat untuk memenuhi persyaratan akademis untuk menyelesaikan studi di program Sarjana Ilmu Komputer Universitas Brawijaya.

Pada kesempatan ini penulis mengucapkan banyak terima kasih atas segala bantuan dan dedikasi moral maupun material dalam rangka penyusunan skripsi ini.

1. Candra Dewi, S.Kom., MSc., selaku dosen pembimbing I yang telah membimbing dengan bijaksana dan sabar dalam membimbing dengan baik penyusunan skripsi ini.
2. Drs.Muh Arif Rahman,M.Kom., selaku dosen pembimbing II yang telah membimbing dengan bijaksana dan sabar dalam membimbing dengan baik penyusunan skripsi ini.
3. Drs. Marji, M.T, selaku Ketua Program Studi Ilmu Komputer di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Drs. Achmad Ridok, M.Kom., selaku dosen pembimbing akademik atas nasehat, bimbingan, saran, dukungan yang diberikan selama penulis menuntut ilmu di Program Studi Ilmu Komputer.
5. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
6. Segenap staf dan karyawan di Program Teknik Informatika Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
7. Segenap staf dan karyawan Jurusan Matematika Universitas Brawijaya yang telah membantu penyusunan skripsi ini.

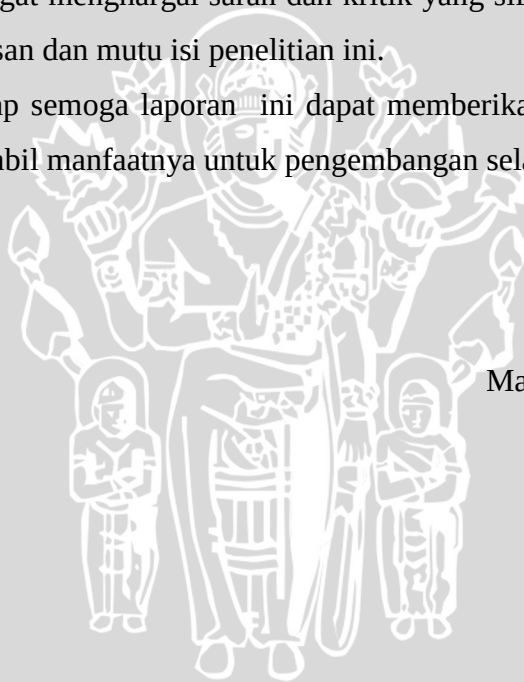
8. Ayah, Ibu, dan adik-adikku serta keluarga besarku yang tersayang, terima kasih atas dukungan dan doanya.
9. Radix A. Rafsanjani, terima kasih atas dukungan dan bantuannya.
10. Jihan, Mila dan ulli, yang senantiasa memotivasi penulis.
11. Teman-teman Program Studi Ilmu Komputer 2008 atas segala bantuan, motivasi dan doanya.
12. Seluruh pihak yang tidak dapat disebut secara langsung yang telah memberikan bantuan demi terselesaikannya skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan laporan ini yang disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu, Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi penelitian ini.

Penulis berharap semoga laporan ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan selanjutnya.

Malang, Januari 2013

Penulis



DAFTAR ISI

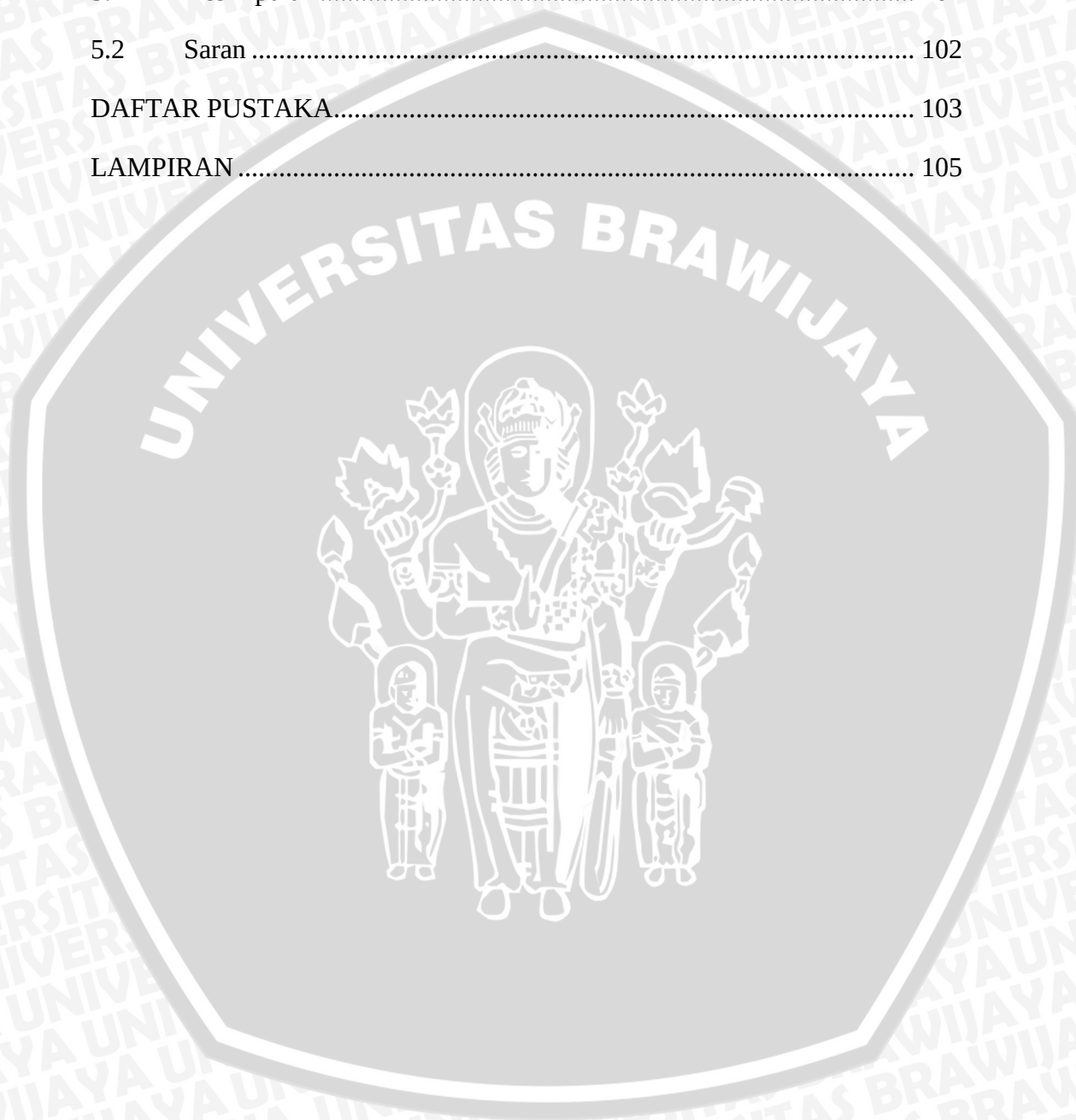
LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS SKRIPSI	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR SOURCECODE.....	xvi
BAB 1	1
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
BAB II	5
TINJAUAN PUSTAKA.....	5
2.1 Data.....	5
2.1.1 Data Akademik Mahasiswa	5
2.2 Evaluasi Keberhasilan Studi Program Sarjana (S1)	6
2.2.1 Evaluasi Keberhasilan Studi Akhir Semester	6

2.2.2	Evaluasi Keberhasilan Studi pada Akhir Studi Program Sarjana	6
2.3	Logika <i>Fuzzy</i>	7
2.4	Himpunan <i>Fuzzy</i>	8
2.5	Fungsi Keanggotaan	10
2.6	Fungsi Implikasi	12
2.7	Pengertian <i>Decision Tree</i>	13
2.8	Algoritma <i>Iterative Dichotomiser 3</i> (ID3).....	13
2.8.1	<i>Entropy</i> dan <i>Information Gain</i> pada Algoritma ID3	14
2.8.2	Pembangunan <i>Decision Tree</i> dengan Algoritma ID3	17
2.9	<i>Fuzzy Decision Tree</i> dengan Algoritma ID3	17
2.9.1	<i>Threshold</i> dalam <i>Fuzzy Decision Tree</i> dengan Algoritma ID3	18
2.9.2	<i>Fuzzy Entropy</i> dan <i>Information Gain</i> dalam <i>Fuzzy Decision Tree</i> dengan Algoritma ID3.....	19
2.9.3	Pembangunan <i>Fuzzy Decision Tree</i> dengan Algoritma ID3.....	19
2.10	Sistem Inferensi <i>Fuzzy</i> Metode Mamdani	20
2.11	Akurasi.....	22
BAB III		23
METODOLOGI DAN PERANCANGAN		23
3.1	Studi Literatur	24
3.2	Data yang digunakan	24
3.3	Deskripsi Umum Sistem	25
3.4	Perancangan Sistem	26
3.4.1	Pembangkitan Aturan	26
3.4.1.1	Proses Fuzzifikasi	27
3.4.1.2	Proses Pembelajaran	27
3.4.1.3	Proses ID3.....	28

3.4.1.4	Konversi <i>Tree</i> menjadi Aturan	30
3.4.2	Pengujian Aturan	31
3.4.2.1	Proses Inferensi Mamdani	32
3.4.2.2	Fungsi Implikasi	33
3.4.2.3	Proses Pilih Aturan	34
3.4.2.4	Fungsi <i>Min</i>	35
3.4.2.5	Komposisi Aturan	36
3.4.2.6	Proses Defuzzifikasi	37
3.5	Contoh Perhitungan Manual	38
3.5.1	Pelatihan.....	38
3.5.2	Inferensi Mamdani.....	63
3.5.3	Pengujian Tingkat Akurasi	65
3.6	Perancangan Antar Muka.....	65
3.7	Perancangan Uji Coba	68
3.7.1	Pengujian Jumlah Aturan yang Terbentuk.....	68
3.7.2	Pengujian Tingkat Akurasi	69
BAB IV		71
IMPLEMENTASI DAN PEMBAHASAN		71
4.1	Lingkungan Implementasi	71
4.1.1	Lingkungan Perangkat Keras.....	71
4.1.2	Lingkungan Perangkat Lunak.....	71
4.2	Implementasi Program.....	72
4.2.1	Struktur Data.....	72
4.2.2	Implementasi Pelatihan.....	73
4.2.2.1	Implementasi Mengambil Data.....	73
4.2.2.2	Implementasi Perhitungan Derajat Keanggotaan	74

4.2.2.3	Implementasi Perhitungan <i>Fuzzy Entropy</i>	76
4.2.2.4	Implementasi Perhitungan <i>Information Gain</i>	76
4.2.2.5	Implementasi Pembentukan <i>Tree</i>	78
4.2.2.6	Implementasi Perhitungan Proporsi.....	78
4.2.2.7	Implementasi Pembentukan Aturan.....	79
4.2.3	Implementasi Pengujian.....	79
4.2.3.1	Implementasi Pengujian <i>Record</i>	79
4.2.3.2	Implementasi Pengujian Data Tabel	81
4.3	Implementasi Antarmuka.....	83
4.3.1	Tab Pelatihan	83
4.3.2	Tab Pengujian	86
4.3.3	Tab Pengujian <i>Record</i>	87
4.4	Sistematika Pengujian.....	88
4.4.1	Sistematika Uji Jumlah Aturan yang Terbentuk.....	89
4.4.2	Sistematika Uji Tingkat Akurasi Kelas Kategori <i>Output</i>	90
4.5	Uji Coba Jurusan/Program Studi Ilmu Komputer.....	90
4.5.1	Pengujian Jumlah Aturan Data Latih 30%	90
4.5.2	Pengujian Jumlah Aturan Data Latih 40%	91
4.5.3	Pengujian Jumlah Aturan Data Latih 50%	92
4.5.4	Pengujian Jumlah Aturan Data Latih 60%	93
4.5.5	Pengujian Tingkat Akurasi Data Latih 30%.....	94
4.5.6	Pengujian Tingkat Akurasi Data Latih 40%.....	95
4.5.7	Pengujian Tingkat Akurasi Data Latih 50%.....	96
4.5.8	Pengujian Tingkat Akurasi Data Latih 60%.....	97
4.6	Analisa Hasil Jurusan/Program Studi Ilmu Komputer	98
4.6.1	Analisa Hasil Jumlah Aturan yang Terbentuk.....	99

4.6.2	Analisa Hasil Tingkat Akurasi Kelas Kategori <i>Output</i>	100
BAB V		101
PENUTUP		101
5.1	Kesimpulan	101
5.2	Saran	102
DAFTAR PUSTAKA		103
LAMPIRAN		105



DAFTAR GAMBAR

Gambar 2. 1 Kurva Segitiga	10
Gambar 2. 2 Kurva Trapesium	11
Gambar 2. 3 Kurva Bentuk Bahu	11
Gambar 2. 4 Fungsi Implikasi <i>Min</i>	12
Gambar 2. 5 Fungsi Implikasi <i>DOT</i>	12
Gambar 2. 6 Proses Pembentukan <i>Tree</i> Tahap Awal.....	17
Gambar 2. 7 <i>Decision Tree</i>	17
 Gambar 3. 1 Diagram Alir Penelitian.....	 23
Gambar 3. 2 Nilai <i>Membership</i> Indeks Prestasi	25
Gambar 3. 3 Pembangkitan Aturan	26
Gambar 3. 4 Proses Fuzzifikasi.....	27
Gambar 3. 5 Proses Pembelajaran.....	27
Gambar 3. 6 Proses ID3	29
Gambar 3. 7 Konversi <i>Tree</i> menjadi Aturan	30
Gambar 3. 8 Pengujian Aturan.....	31
Gambar 3. 9 Proses Inferensi Mamdani	32
Gambar 3. 10 Fungsi Implikasi	33
Gambar 3. 11 Proses Pilih Aturan.....	34
Gambar 3. 12 Fungsi <i>Min</i>	35
Gambar 3. 13 Komposisi Aturan.....	36
Gambar 3. 14 Proses Defuzzifikasi	37
Gambar 3. 15 <i>Tree</i> Tahap 1.....	44
Gambar 3. 16 <i>Tree</i> Tahap 2.....	47
Gambar 3. 17 <i>Tree</i> Tahap 3.....	49
Gambar 3. 18 <i>Tree</i> Tahap 4.....	50
Gambar 3. 19 <i>Tree</i> Tahap 5.....	52
Gambar 3. 20 <i>Tree</i> Tahap 6.....	54
Gambar 3. 21 <i>Tree</i> Tahap 7.....	56
Gambar 3. 22 <i>Tree</i> Tahap 8.....	58
Gambar 3. 23 <i>Tree</i> Tahap 9.....	60

Gambar 3. 24 <i>Tree</i> Tahap 10.....	61
Gambar 3. 25 Daerah Hasil Komposisi.....	64
Gambar 3. 26 Tampilan Rancangan Antarmuka Pelatihan	66
Gambar 3. 27 Tampilan Rancangan Antarmuka Pengujian	67
Gambar 3. 28 Tampilan Rancangan Antarmuka Pengujian Record	68
Gambar 4. 1 Tab Pelatihan.....	83
Gambar 4. 2 Halaman <i>Membership</i>	84
Gambar 4. 3 Halaman <i>Tree</i>	85
Gambar 4. 4 Halaman Aturan yang Terbentuk	85
Gambar 4. 5 Tab Pengujian.....	86
Gambar 4. 6 Tab Pengujian <i>Record</i>	87
Gambar 4. 7 Grafik Jumlah Aturan yang Terbentuk.....	99
Gambar 4. 8 Grafik Tingkat Akurasi Kelas Kategori <i>Output</i>	100



DAFTAR TABEL

Tabel 2. 1 Indeks Prestasi dalam Predikat Kelulusan	7
Tabel 2. 2 Dataset <i>Auto Driving</i>	15
Tabel 3. 1 Data Pelatihan	38
Tabel 3. 2 Nilai <i>Membership</i> IP Semester 1.....	40
Tabel 3. 3 Nilai <i>Membership</i> IP Semester 2.....	41
Tabel 3. 4 Level 1 Iterasi ke-1.....	45
Tabel 3. 5 Level 2 Iterasi ke-1.....	48
Tabel 3. 6 Level 2 Iterasi ke-2.....	49
Tabel 3. 7 Level 1 Iterasi ke-2.....	50
Tabel 3. 8 Level 2 Iterasi ke-3.....	53
Tabel 3. 9 Level 2 Iterasi ke-4.....	55
Tabel 3. 10 Level 1 Iterasi ke-3.....	56
Tabel 3. 11 Level 2 Iterasi ke-5.....	59
Tabel 3. 12 Level 2 Iterasi ke-6.....	60
Tabel 3. 13 Rancangan Pengujian Jumlah Aturan	69
Tabel 3. 14 Rancangan Pengujian Akurasi	70
Tabel 4. 1 Tabel Proporsi Data Latih dan Data Uji.....	88
Tabel 4. 2 Pengujian Jumlah Aturan dengan Data Latih 30%.....	90
Tabel 4. 3 Pengujian Jumlah Aturan dengan Data Latih 40%	92
Tabel 4. 4 Pengujian Jumlah Aturan dengan Data Latih 50%	93
Tabel 4. 5 Pengujian Jumlah Aturan dengan Data Latih 60%	94
Tabel 4. 6 Pengujian Tingkat Akurasi dengan Data Latih 30%.....	95
Tabel 4. 7 Pengujian Tingkat Akurasi dengan Data Latih 40%.....	96
Tabel 4. 8 Pengujian Tingkat Akurasi dengan Data Latih 50%.....	97
Tabel 4. 9 Pengujian Tingkat Akurasi dengan Data Latih 60%.....	98

DAFTAR SOURCECODE

<i>Sourcecode 4. 1</i> Struktur data koneksi basis data	72
<i>Sourcecode 4. 2</i> Struktur data pelatihan	72
<i>Sourcecode 4. 3</i> Struktur data pengujian	73
<i>Sourcecode 4. 4</i> Implementasi mengambil data	74
<i>Sourcecode 4. 5</i> Implementasi perhitungan derajat keanggotaan	75
<i>Sourcecode 4. 6</i> Implementasi perhitungan <i>fuzzy entropy</i>	76
<i>Sourcecode 4. 7</i> Implementasi perhitungan <i>information gain</i> atribut jenis kelamin	78
<i>Sourcecode 4. 8</i> Implementasi perhitungan proporsi	79
<i>Sourcecode 4. 9</i> Implementasi pengujian per <i>record</i>	81
<i>Sourcecode 4. 10</i> Implementasi pengujian data tabel	82
<i>Sourcecode 4. 11</i> Implementasi perhitungan <i>information gain</i> atribut semester	109
<i>Sourcecode 4. 12</i> Implementasi pembentukan <i>tree</i>	111
<i>Sourcecode 4. 13</i> Implementasi pembentukan aturan	113

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Prestasi mahasiswa biasanya diukur dari tingginya nilai indeks prestasi yang diraih. Pada tiap akhir ujian semester, evaluasi pada hasil indeks prestasi merupakan hal yang ramai diperbincangkan. Apalagi untuk mahasiswa baru, indeks prestasi yang diperoleh pada tahun pertama tentu dapat dijadikan sebagai dasar awal evaluasi studi dalam mencari pencapaian indeks prestasi selanjutnya.

Pada tahun pertama perkuliahan dan saat kelulusan, beragam nilai indeks prestasi yang mampu diraih oleh mahasiswa. Dari pengamatan pada data akademik mahasiswa Fakultas MIPA Universitas Brawijaya, ada kecenderungan nilai indeks prestasi mahasiswa pada tahun pertama yang tidak sejalan dengan indeks prestasi kelulusannya. Dari 1132 record data kelulusan akademik mahasiswa, 43 record mahasiswa yang mempunyai nilai indeks prestasi $> 3,50$ pada semester 1 dan semester 2, 16 % diantaranya memperoleh nilai kelulusan $< 3,50$ dan 243 record mahasiswa yang mempunyai nilai indeks prestasi $< 2,75$ pada semester 1 dan semester 2, 85 % diantaranya memperoleh nilai kelulusan $> 2,75$ (Hasil Komputasi pada Data Akademik Mahasiswa Fakultas MIPA Universitas Brawijaya Angkatan 2003-2006).

Dewasa ini, belum ada penelitian yang mencoba mencari aturan untuk mengetahui hubungan antara indeks prestasi mahasiswa pada tahun pertama dengan indeks prestasi kelulusannya. Untuk mencari aturan, dapat menggunakan metode *decision tree*. Metode *decision tree* sangat terkenal daripada metode yang lainnya, karena metode ini tidak membutuhkan pengetahuan yang lebih atau pengaturan parameter (Han dan Khamber, 2001). Algoritma *Iterative Dichotomiser 3* atau disingkat ID3, merupakan salah satu dari metode *decision tree* yang menggunakan teori informasi untuk menentukan atribut mana yang paling informatif. Seperti kebanyakan metode *decision tree* lainnya, ID3 juga menghasilkan keputusan dari data kategorial untuk prosesnya. Sedangkan data indeks prestasi mahasiswa merupakan bentuk data numerik. Penggunaan data

numerik pada *decision tree* memungkinkan dengan membentuk partisi pada data tersebut.

Salah satu cara yang dapat digunakan untuk menentukan batas partisi pada data numerik yaitu menggunakan logika *fuzzy*. Logika *fuzzy* mampu memodelkan fungsi-fungsi *input/output* yang sangat kompleks. Logika *fuzzy* merupakan logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) yang digunakan untuk melakukan penalaran (Kusumadewi, 2010). Penggabungan metode *decision tree* dengan *fuzzy* memungkinkan untuk menggunakan nilai-nilai numerik yang dihubungkan dengan atribut kuantitatif yang masing-masing memiliki nilai derajat keanggotaan. Menurut Liang, 2005, menggunakan teknik *fuzzy* dalam *decision tree* dapat memberikan suatu peningkatan atribut kuantitatif dalam melakukan penggolongan pada saat pelatihan. Selain itu, proses pengujian menggunakan metode *fuzzy* inferensi Mamdani untuk mendapatkan *output* yang baik berdasarkan nilai derajat keanggotaan masing-masing atribut.

Pada skripsi ini metode yang digunakan adalah *Fuzzy Iterative Dichotomiser 3* atau disingkat *Fuzzy ID3*. Metode *fuzzy decision tree* menggunakan algoritma ID3 ini telah dilakukan pada berbagai riset. Rumusan metode ini dituliskan pada riset yang dilakukan Romansyah tahun 2009 pada data diabetes dan menghasilkan tingkat akurasi 94,15%. Selain itu, metode ini juga telah diterapkan untuk diagnosis penyakit jantung koroner (Nurlaelasari, 2010). Riset mengenai perbandingan antara *decision tree* menggunakan algoritma ID3 dengan *decision tree* menggunakan algoritma C4.5 dalam spam e-mail menunjukkan bahwa *decision tree* dengan menggunakan algoritma ID3 memiliki tingkat akurasi yang lebih baik (Defiyanti, 2009). Penelitian *Fuzzy ID3* juga dilakukan Guoxiu Liang tahun 2005 yang menguji tentang *performance* dari *Fuzzy ID3* menunjukkan nilai tertinggi pada $\theta_{0,4}$ sebesar 91,4 % .

Berdasarkan latar belakang yang telah dikemukakan, maka judul yang diambil dalam skripsi ini adalah **“Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan Menggunakan *Fuzzy ID3*”**.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka dalam skripsi ini dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana menggunakan metode *Fuzzy ID3* dalam Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan.
2. Bagaimana aturan yang dihasilkan oleh metode *Fuzzy ID3* sebagai bentuk hubungan antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan.

1.3 Batasan Masalah

Untuk membatasi masalah yang dikaji, maka penulis membatasi dan menfokuskan permasalahan sebagai berikut:

1. Data yang dijadikan pelatihan dan pengujian pada skripsi ini adalah data mahasiswa yang telah lulus pada tahun akademik 2003 s/d 2006 Fakultas MIPA Universitas Brawijaya diambil dari Bagian Akademik fakultas MIPA Universitas Brawijaya dengan pembagian jurusan/program studi yaitu Ilmu Komputer, Statistika, Matematika, Fisika, Kimia dan Biologi.
2. Atribut yang digunakan yaitu indeks prestasi semester 1, indeks prestasi semester 2, jenis kelamin dan indeks prestasi kelulusan sebagai atribut target tanpa mempertimbangkan masa studi.
3. Tidak ada perbandingan metode dalam skripsi ini.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Untuk menerapkan metode *Fuzzy ID3* dalam Pembangkitan Aturan Untuk Mengetahui Hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan.
2. Untuk mengetahui aturan yang dihasilkan oleh metode *Fuzzy ID3* sebagai bentuk hubungan Antara Indeks Prestasi Mahasiswa Tahun Pertama Dengan Indeks Prestasi Kelulusan.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil pada penelitian ini adalah menggunakan aturan hasil penelitian ini sebagai dasar untuk memprediksi indeks prestasi kelulusan mahasiswa berdasarkan atribut yang telah diteliti, yaitu indeks prestasi semester 1, indeks prestasi semester 2 dan jenis kelamin.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori yang berhubungan dengan indeks prestasi tahun pertama, indeks prestasi kelulusan, decision tree, metode *Fuzzy Decision Tree*, dan algoritma Iterative Dichotomiser 3 (ID3), dan fuzzy inferensi Mamdani.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Bab ini berisi membahas tentang data penelitian, spesifikasi perangkat lunak, perancangan sistem, perhitungan manual, perancangan antarmuka dan perancangan uji coba.

4. BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisi implementasi sistem dan pengujian perangkat lunak yang dibangun.

5. BAB V PENUTUP

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

BAB II

TINJAUAN PUSTAKA

2.1 Data

Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian. Contohnya fakta mengenai biodata mahasiswa yang meliputi nama, alamat, jenis kelamin, agama, dan lain-lain. Data dapat dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol (Kusrini, 2007).

2.1.1 Data Akademik Mahasiswa

Data akademik mahasiswa adalah data tentang kegiatan pendidikan/pembelajaran, penelitian, dan pengabdian pada masyarakat. Data kegiatan pendidikan/pembelajaran adalah data-data mengenai proses belajar mengajar yang memuat dosen sebagai tenaga pengajar, mahasiswa sebagai peserta ajar, materi kuliah sebagai bahan ajar, transaksi kuliah sebagai proses belajar mengajar, manajemen registrasi, KRS, KHS, dan nilai sebagai bahan administrasi pembelajaran sejak mahasiswa masuk kuliah sampai dengan lulus (Suharto, 2010).

Fakultas MIPA Universitas Brawijaya memiliki 4 jurusan dan 8 program studi, yaitu sebagai berikut (Suharto, 2010) :

1. Jurusan Biologi
 - a. Program Studi Biologi
2. Jurusan Kimia
 - a. Program Studi Kimia
3. Jurusan Fisika
 - a. Program Studi Fisika
 - b. Program Studi Ilmu Instrumentasi
 - c. Program Studi Geofisika
4. Jurusan Matematika
 - a. Program Studi Matematika
 - b. Program Studi Statistika
 - c. Program Studi Ilmu Komputer

Data yang digunakan dalam skripsi ini adalah data akademik mahasiswa yang diambil dari Bagian Akademik Fakultas MIPA Universitas Brawijaya. Data-data ini berisi transaksi nilai dan indeks prestasi (IP) mahasiswa setiap semester genap maupun ganjil dari tahun akademik 2003 s/d 2006.

2.2 Evaluasi Keberhasilan Studi Program Sarjana (S1)

Keberhasilan studi mahasiswa dinyatakan dengan indek prestasi (IP), yang ditulis dengan angka. Evaluasi keberhasilan studi mahasiswa dilaksanakan sekurang-kurangnya tiap akhir semester.

2.2.1 Evaluasi Keberhasilan Studi Akhir Semester

Evaluasi keberhasilan studi akhir semester dilakukan pada setiap akhir semester, meliputi matakuliah yang diambil mahasiswa pada semester tersebut. Menurut Suharto 2010, rumusan yang digunakan untuk menghitung indeks prestasi yaitu :

$$IP = \frac{\sum_{i=1}^n K_i NA_i}{\sum_{i=1}^n K_i} \dots\dots\dots (2.1)$$

Dimana,

- IP : Indeks Prestasi, dapat berupa indek prestasi semester atau indeks prestasi kumulatif
 K : Jumlah sks masing-masing mata kuliah
 NA : Nilai akhir masing-masing mata kuliah
 n : Banyaknya matakuliah diambil

2.2.2 Evaluasi Keberhasilan Studi pada Akhir Studi Program Sarjana

Jumlah kredit yang harus dikumpulkan oleh seorang mahasiswa untuk menyelesaikan studi program sarjana adalah 144 – 160 sks termasuk skripsi/tugas lain yang ditentukan oleh fakultas. Jumlah sks minimum ditentukan oleh fakultas dalam batas sebaran tersebut. Mahasiswa yang telah mengumpulkan sekurang-kurangnya sejumlah sks minimum diatas dinyatakan telah menyelesaikan program studi sarjana apabila memenuhi syarat-syarat :

1. Indeks Prestasi Kumulatif (IPK) sekurang-kurangnya 2,00.
2. Nilai D/D+ tidak melebihi 10 % dari beban kredit total, kecuali untuk matakuliah tertentu yang tidak diperbolehkan memperoleh nilai D/D+ yang diatur tersendiri.
3. Tidak ada nilai E.
4. Lulus ujian sarjana.

Apabila indek prestasi yang dicapai kurang dari 2,00 maka mahasiswa yang bersangkutan harus memperbaiki nilai matakuliah selama batas masa studi belum dilampaui. Perbaikan harus dilakukan pada semester berikutnya saat matakuliah yang akan diperbaiki ditawarkan. Setiap matakuliah yang diperbaiki, nilai tertinggi yang digunakan untuk evaluasi.

Predikat kelulusan terdiri dari 3 tingkat, yaitu memuaskan, sangat memuaskan dan dengan pujian, yang dinyatakan pada transkrip akademik. Indeks Prestasi Kumulatif (IPK) sebagai dasar menentukan predikat kelulusan adalah :

Tabel 2. 1 Indeks Prestasi dalam Predikat Kelulusan

IPK	Predikat Kelulusan
2,00 – 2,75	Memuaskan
2,76 – 3,50	Sangat Memuaskan
3,51 – 4,00	<i>Cumlaude</i> (dengan pujian)

(Suharto,2010)

2.3 Logika Fuzzy

Kata *Fuzzy* merupakan kata sifat yang berarti kabur, tidak jelas. *Fuzziness* atau kekaburan atau ketidakjelasan atau ketidakpastian selalu meliputi keseharian manusia. Orang yang belum pernah mengenal logika *fuzzy* pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang rumit dan tidak menyenangkan. Namun, sekali seseorang mulai mengenalnya, pasti akan tertarik untuk ikut mempelajari logika *fuzzy*. Logika *fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *fuzzy* modern dan metode baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada sejak lama (Kusumadewi, 2002).

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan ruang *input* kedalam suatu ruang *output* (Kusumadewi, 2003). Konsep ini diperkenalkan dan dipublikasikan pertama kali oleh Lotfi A. Zadeh, seorang profesor dari University of California di Berkeley pada tahun 1965. Logika *fuzzy* menggunakan ungkapan bahasa untuk menggambarkan nilai variabel. Logika *fuzzy* bekerja dengan menggunakan derajat keanggotaan dari sebuah nilai yang kemudian digunakan untuk menentukan hasil yang ingin dihasilkan berdasarkan atas spesifikasi yang telah ditentukan. Telah disebutkan sebelumnya bahwa logika *fuzzy* memetakan ruang *input* ke ruang *output*. Antara *input* dan *output* ada suatu kotak hitam yang harus memetakan *input* ke *output* yang sesuai. Alasan mengapa orang menggunakan logika *fuzzy*, yaitu (Kusumadewi, 2010):

1. Konsep logika *fuzzy* mudah dimengerti karena menggunakan konsep matematis yang sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.4 Himpunan Fuzzy

Himpunan tegas (*crisp*) A didefinisikan oleh *item-item* yang ada pada himpunan itu. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu *item* x dalam suatu himpunan A ($\mu_A(x)$) memiliki dua kemungkinan (Kusumadewi, 2010), yaitu:

1. Satu (1), yang berarti bahwa suatu *item* menjadi anggota dalam suatu himpunan.

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

Himpunan *Fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu *item* dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran suatu *item* tidak hanya benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah. Himpunan *fuzzy* memiliki 2 atribut, yaitu (Kusumadewi, 2003):

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

1. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*.

2. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel.

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

4. Domain

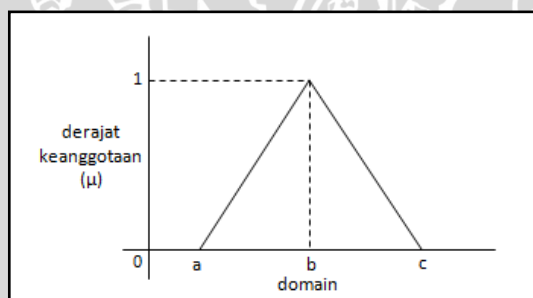
Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

2.5 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melakukan pendekatan fungsi. Terdapat beberapa fungsi yang dapat digunakan (Kusumadewi, 2010), yaitu :

1. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear).



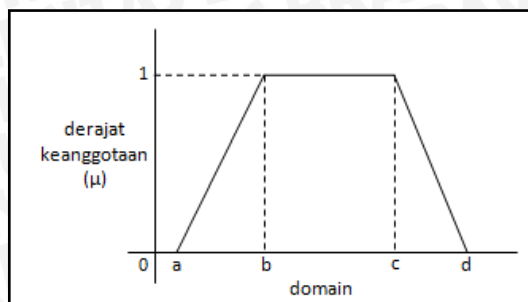
Gambar 2. 1 Kurva Segitiga

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; a \leq x \leq b \\ \frac{c-x}{c-b}; b \leq x \leq c \end{cases} \dots\dots\dots (2.2)$$

2. Representasi Kurva Trapezium

Kurva trapezium pada dasarnya seperti bentuk segitiga, hanya saja terdapat beberapa titik yang memiliki nilai keanggotaan 1.



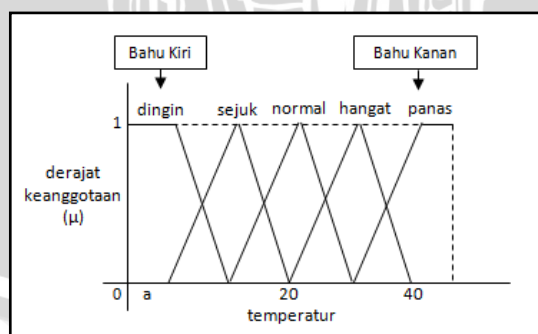
Gambar 2. 2 Kurva Trapesium

Fungsi keanggotaan :

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

3. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himpunan *fuzzy* ‘bahu’, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Sebagai contoh, himpunan *fuzzy* pada variabel TEMPERATUR dengan daerah bahunya.



Gambar 2. 3 Kurva Bentuk Bahu

2.6 Fungsi Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah sebagai berikut :

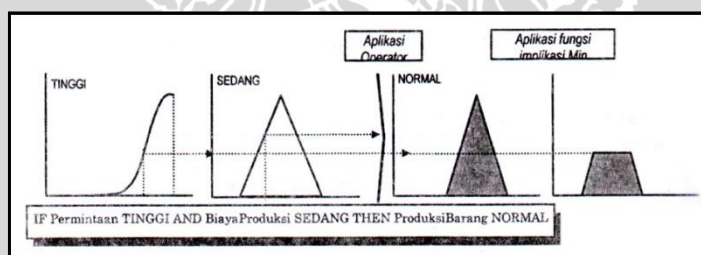
$$IF x \text{ is } A \text{ THEN } y \text{ is } B \quad \dots\dots\dots (2.4)$$

dengan x dan y adalah skalar, A dan B adalah himpunan *fuzzy*. Proposisi yang mengikuti IF disebut anteseden, sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy*, yaitu:

IF (x_1 is A_1) opt (x_2 is A_2) opt (x_3 is A_3) opt ... opt (x_N is A_N) THEN y is B
dengan opt OR atau AND.

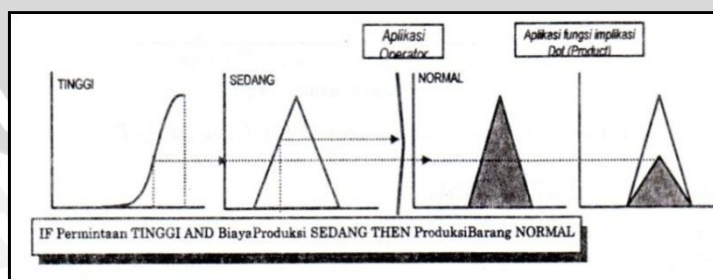
Secara umum, terdapat 2 fungsi implikasi yang dapat digunakan, yaitu:

1. *Min* (minimum). Fungsi ini akan memotong *output* himpunan *fuzzy*. Gambar 2.4 menunjukkan salah satu contoh penggunaan fungsi *min*.



Gambar 2. 4 Fungsi Implikasi *Min*

2. *Dot* (product). Fungsi ini akan melakukan skala *output* himpunan *fuzzy*. Salah satu penggunaan fungsi *dot* adalah sebagai berikut.



Gambar 2. 5 Fungsi Implikasi *DOT*

(Kusumadewi, 2004).

2.7 Pengertian *Decision Tree*

Decision tree merupakan salah satu fungsional dari data mining yang menggunakan representasi *tree* untuk menentukan aturan-aturan yang menghasilkan keputusan yang terbagi dalam kelas-kelas tertentu. *Decision tree* dapat juga dikatakan sebagai flowchart seperti struktur *tree*, dimana tiap *node* internal menunjukkan sebuah tes pada sebuah atribut, tiap cabang menunjukkan hasil dari tes, dan setiap *node leaf* menunjukkan kelas-kelas atau distribusi kelas. Metode *Decision tree* sangat terkenal daripada metode *decision tree* yang lainnya, karena metode ini tidak membutuhkan pengetahuan yang lebih atau pengaturan parameter (Han dan Khamber, 2001). Ada dua tipe *decision tree*, yaitu:

1. *Classification tree*

Memasukkan *record* ke kelas data yang telah dikategorikan untuk menghasilkan sebuah *node*.

2. *Regression tree*

Memprediksi suatu nilai (nilai numerik) dari sebuah variabel target.

Decision tree merupakan *supervised learning* dimana kelas dari sekumpulan data telah ditentukan sebelumnya. Maka dalam proses pembelajarannya membutuhkan kumpulan data yang banyak dengan kemungkinan variasi yang berbeda-beda pada tiap-tiap kelas yang telah didefinisikan. Hal ini disebabkan *decision tree* menganalisa dari data yang telah ada, apabila data tersebut secara sistematis kekurangan data pembelajarannya maka hasil akhir dari penentuan kelasnya menjadi kurang optimal.

Pada *decision tree* terdapat beberapa algoritma, seperti CLS, ID3, C4.5, CART, SLIQ, dan SPRINT. *Decision tree* sangat mudah untuk dipahami dan diinterpretasikan. Akan tetapi, *tree* bisa menjadi sangat kompleks, dimana suatu *tree* harus dimasukkan ke dalam database dengan ratusan atribut (Larose, 2005).

2.8 Algoritma *Iterative Dichotomiser 3* (ID3)

Iterative Dichotomiser 3 (ID3) adalah salah satu algoritma yang sering digunakan pada pembelajaran dan data mining karena mudah digunakan dan cukup efektif. Algoritma ini dikembangkan oleh J. Rose Quinlan pada tahun 1986. Algoritma ini membangun pohon keputusan dari beberapa data untuk proses

penentuan kelas dan menentukan kelas data yang baru. Data yang digunakan harus memiliki beberapa atribut dengan nilai yang berbeda pula (Liang, 2005).

Algoritma ID3 adalah sebagai berikut (Mitchell, 1997).

ID3 (Kumpulan Sampel, Atribut Target, Kumpulan Atribut)

- Buat simpul akar untuk *tree* yang akan dibuat.
- Jika semua atribut positif, maka berhenti dengan suatu pohon dengan satu simpul dan akar, beri label +.
- Jika semua atribut negatif, maka berhenti dengan suatu pohon dengan satu simpul dan akar, beri label -.
- Jika atribut kosong, maka berhenti dengan suatu pohon dengan satu simpul akar dengan label yang sesuai.

- Untuk yang lain :

Mulai

- $A \leftarrow$ atribut yang mengkan sampel dengan hasil terbaik (berdasarkan nilai *information gain*)
- Atribut keputusan untuk simpul akar (*root*) $\leftarrow A$
- Untuk setiap nilai (v_i), yang mungkin untuk A :
 - Tambahkan cabang dibawah akar yang berhubungan dengan $A = v_i$
 - Tentukan atribut S_{vi} sebagai *subset* dari atribut yang mempunyai nilai
 - Jika atribut S_{vi} kosong :
 - Dibawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label *training*
 - Jika tidak, tambahkan cabang baru dibawah cabang yang sekarang ID3 (sampel, label, atribut-[A])

Berhenti

2.8.1 Entropy dan Information Gain pada Algoritma ID3

ID3 adalah algoritma pembelajaran pohon keputusan yang terkenal. Misalkan, seseorang yang akan pergi bekerja memutuskan untuk mengendarai mobil atau menaiki angkutan umum berdasarkan cuaca dan kondisi lalu lintas, seperti data pada tabel berikut (Liang 2005):

Tabel 2. 2 Dataset Auto Driving

Day	Temperature	Wind	Traffic-Jam	Car Driving
D1	Hot	Weak	Long	No
D2	Hot	Strong	Long	No
D3	Hot	Weak	Long	Yes
D4	Mild	Weak	Long	Yes
D5	Cool	Weak	Short	Yes
D6	Cool	Strong	Short	No
D7	Cool	Strong	Short	Yes
D8	Mild	Weak	Long	No
D9	Cool	Weak	Short	Yes
D10	Mild	Weak	Short	Yes
D11	Mild	Strong	Short	Yes
D12	Mild	Strong	Long	Yes
D13	Hot	Weak	Short	Yes
D14	Mild	Strong	Long	No

Dasar dari metode ID3 adalah memilih masing-masing atribut dengan menggunakan metode statistik dimulai dari bagian atas pohon. Metode statistika tersebut dinamakan *information gain* yang didefinisikan sebagai ukuran nilai tiap atribut. Selain itu, terdapat *entropy* yang digunakan untuk mendefinisikan nilai *information gain*, untuk memilih atribut terbaik dari atribut-atribut yang tersedia (Liang, 2005).

$$H_s(s) = \sum_i^N -P_i * \log_2(P_i) \quad \dots\dots\dots (2.5)$$

Dimana P_i adalah rasio dari kelas C_i pada contoh $S=\{x_1, x_2, \dots, x_k\}$ seperti persamaan 2.6 dibawah ini.

$$P_i = \frac{\sum x_k \in C_i}{S} \quad \dots\dots\dots (2.6)$$

Berdasarkan tabel 2.2, terdapat 9 contoh positif dan 5 contoh negatif. Maka nilai *entropy* dari S adalah sebagai berikut.

$$H(9+,5-) = -(9/14)*\log_2(9/14) - (5/14) * \log_2(5/14)=0,940$$

Sedangkan untuk mendefinisikan *information gain* dari atribut A pada *sample set* S adalah seperti persamaan 2.7 sebagai berikut.

$$G(S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} * H(S_v) \quad \dots\dots\dots (2.7)$$

Dimana $W_i = \left| \frac{S_v}{S} \right|$ adalah rasio dari data dengan v atribut pada *sample set*.

Berdasarkan pada contoh di atas, S set (9+,5-) keseluruhan terdiri dari 14 contoh. Pada atribut *wind* dengan variabel *weak* terdapat 8 contoh yang terdiri dari 6 positif dan 2 negatif. Sedangkan sisanya pada variabel *strong*. Maka dapat dihitung *information gain* dari atribut *wind* sebagai berikut.

$$S = [9+,5-]$$

$$S(\text{weak}) = [6+,2-]$$

$$S(\text{strong}) = [3+,3-]$$

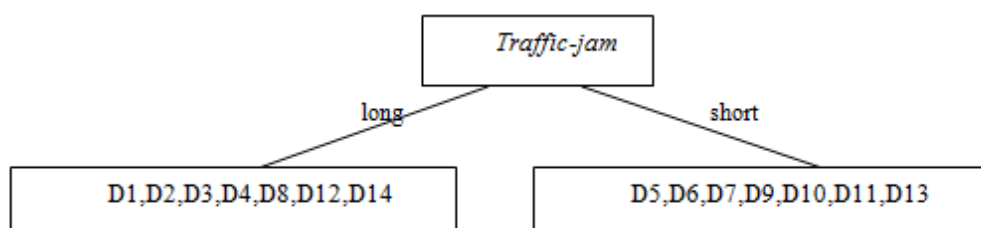
$$\begin{aligned} G(S, \text{wind}) &= H(S) - (8/14)H(\text{weak}) - (6/14)H(\text{strong}) \\ &= 0,940 - (8/14)*0,811 - (6/14)*1,0 \\ &= 0,048 \end{aligned}$$

Dengan menggunakan prinsip yang sama, dapat dilakukan perhitungan *information gain* dari atribut yang lain yaitu *temperature* dan *traffic-jam* adalah sebagai berikut.

$$\begin{aligned} G(S, \text{temp}) &= H(S) - (4/14)H(\text{hot}) - (6/14)H(\text{wild}) - (4/14)H(\text{cool}) \\ &= 0,940 - (4/14)*0,811 - (6/14)*0,918 - (4/14)*0,811 \\ &= 0,029 \end{aligned}$$

$$\begin{aligned} G(S, \text{traffic}) &= H(S) - (8/14)H(\text{weak}) - (6/14)H(\text{strong}) \\ &= 0,940 - (8/14)*0,811 - (6/14)*1,0 \\ &= 0,048 \end{aligned}$$

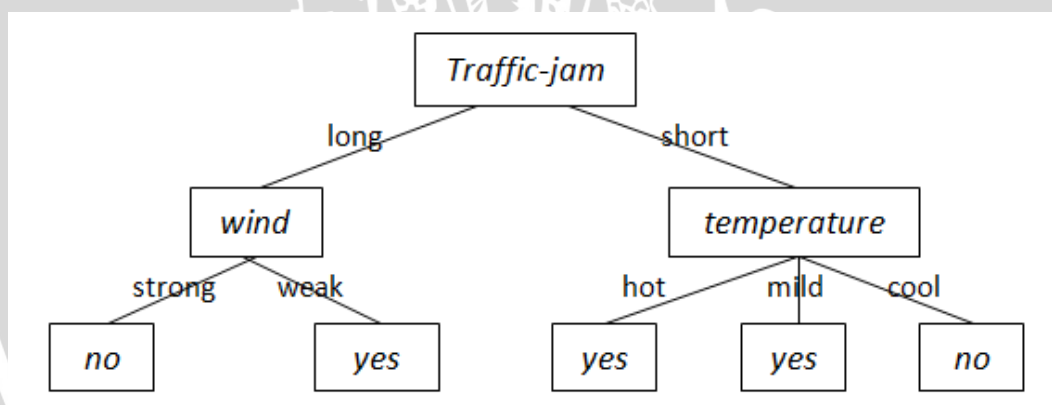
Information gain yang dipilih adalah *information gain* yang memiliki nilai terbesar, yaitu atribut *traffic-jam* yang akan digunakan sebagai *root* sebagai ekspansi pertama dari proses penentuan kelas.



Gambar 2. 6 Proses Pembentukan *Tree* Tahap Awal

2.8.2 Pembangunan *Decision Tree* dengan Algoritma ID3

Membangun *decision tree* menggunakan algoritma ID3 dimulai dengan menentukan *root* berdasarkan nilai *information gain* terbesar. Atribut yang telah digunakan untuk mengekspansi *tree* dihapuskan dan tidak digunakan lagi untuk proses ekspansi. Proses tersebut dilakukan berulang-ulang sampai tidak ada lagi *entropy* dari *node* (*equal null*) (Liang, 2005). Dari contoh proses ekspansi *tree* di atas, maka dihasilkan *tree* seperti gambar sebagai berikut.



Gambar 2. 7 *Decision Tree*

2.9 Fuzzy *Decision Tree* dengan Algoritma ID3

Fuzzy Decision tree merupakan suatu pendekatan yang sangat populer dan praktis dalam *machine learning* untuk menyelesaikan permasalahan penentuan kelas yang mengalami ketidakpastian. *Fuzzy decision tree* memungkinkan untuk menggunakan nilai-nilai *numeric-symbolic* selama konstruksi atau saat menentukan kelas pada kasus-kasus baru. Manfaat dari teori himpunan *fuzzy* dalam *decision tree* ialah meningkatkan kemampuan atribut-atribut kuantitatif, bahkan

dengan menggunakan teknik *fuzzy* dapat meningkatkan ketahanan saat melakukan penentuan kelas pada kasus-kasus baru (Romansyah,dkk, 2009).

Salah satu algoritma yang dapat digunakan untuk membangun *fuzzy decision tree* adalah algoritma *Iterative Dichotomiser 3* (ID3). *Iterative Dichotomiser 3* (ID3) adalah algoritma *decision tree learning* (algoritma pembelajaran pohon keputusan) yang paling dasar. Algoritma ini melakukan pencarian secara rakus /menyeluruh (*greedy*) pada semua kemungkinan pohon keputusan. Salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi *rekursif* (fungsi yang memanggil dirinya sendiri). Algoritma ID3 berusaha membangun *decision tree* (Wahyudin, 2009).

2.9.1 *Threshold* dalam *Fuzzy Decision Tree* dengan Algoritma ID3

Jika pada proses *learning* dari FDT dihentikan sampai semua data contoh pada masing-masing *leaf-node* menjadi anggota sebuah kelas, akan dihasilkan akurasi yang rendah. Oleh karena itu untuk meningkatkan akurasinya, proses *learning* harus dihentikan lebih awal atau melakukan pemotongan *tree* secara umum. Untuk itu diberikan 2 *threshold* yang harus terpenuhi jika *tree* akan diekspansi (Liang, 2005), yaitu:

1. *Fuzziness Control Threshold* (FCT) / θ_r

Jika proporsi dari himpunan data dari kelas C_k lebih besar atau sama dengan nilai *threshold* θ_r , maka hentikan ekspansi *tree*. Sebagai contoh: jika pada sebuah sub-dataset rasio dari kelas 1 adalah 90%, kelas 2 adalah 10% dan θ_r adalah 85% maka hentikan ekspansi *tree*

2. *Leaf Decision Threshold* (LDT) / θ_n

Jika banyaknya anggota himpunan data pada suatu *node* lebih kecil dari *threshold* θ_n , hentikan ekspansi *tree*. Sebagai contoh, sebuah himpunan data memiliki 600 contoh dengan θ_n adalah 2%. Jika jumlah data contoh pada sebuah *node* lebih kecil dari 12 (2% dari 600), maka hentikan ekspansi *tree*.

2.9.2 Fuzzy Entropy dan Information Gain dalam Fuzzy Decision Tree dengan Algoritma ID3

Pada himpunan data *fuzzy* terdapat penyesuaian rumus untuk menghitung nilai *entropy* untuk atribut dan *information gain* karena adanya ekspresi data *fuzzy*. Persamaan 2.8 berikut adalah persamaan untuk mencari nilai *fuzzy entropy* dari keseluruhan data.

$$H_f(S) = H_s(s) = \sum_i^N -P_i * \log_2(P_i) \quad \dots\dots\dots (2.8)$$

Dimana P_i adalah rasio dari kelas C_i pada contoh $S = \{x_1, x_2, \dots, x_k\}$.

Untuk menentukan *fuzzy entropy* dan *information gain* dari suatu atribut digunakan persamaan 2.9 dan 2.10 sebagai berikut:

$$H_f(S, A) = - \sum_{i=1}^C \frac{\sum_j^N \mu_{ij}}{S} \log_2 \frac{\sum_j^N \mu_{ij}}{S} \quad \dots\dots\dots (2.9)$$

$$G_f(S, A) = H_f(S) - \sum_{v \in A} \frac{|S_v|}{|S|} * H_f(S_v, A) \quad \dots\dots\dots (2.10)$$

dengan μ_{ij} adalah nilai keanggotaan dari pola ke-j untuk kelas ke-i. $H_f(S)$ menunjukkan *entropy* dari himpunan S dari data pelatihan pada *node*. $|S_v|$ adalah ukuran dari *subset* $S_v \subseteq S$ dari data pelatihan x_j dengan atribut v . $|S|$ menunjukkan ukuran dari himpunan S .

2.9.3 Pembangunan Fuzzy Decision Tree dengan Algoritma ID3

Algoritma ini menggunakan teori informasi untuk menentukan atribut mana yang paling informatif, namun ID3 sangat tidak stabil dalam melakukan penggolongan berkenaan dengan gangguan kecil pada data pelatihan. logika *fuzzy* dapat memberikan suatu peningkatan untuk dalam melakukan penggolongan pada saat pelatihan (Romansyah,dkk, 2009).

Algoritma *fuzzy ID3* merupakan algoritma yang efisien untuk membuat suatu *fuzzy decision tree*. Algoritma *fuzzy ID3 decision tree* adalah sebagai berikut (Romansyah,dkk, 2009).

1. Membuat *root node* yang memiliki himpunan data *fuzzy* dengan nilai keanggotaan 1

2. Jika t node dengan himpunan data *fuzzy* D memenuhi kondisi berikut, maka itu adalah sebuah *node* daun dan ditugaskan oleh nama kelas.
 - proporsi C_k kelas lebih besar dari atau sama dengan Θ_r ,

$$\left| \frac{D_{ck}}{D} \right| \geq \Theta_r$$
 - jumlah *set* data kurang dari Θ_n tidak ada atribut untuk yang lebih
3. Jika *node* D tidak ada memenuhi kondisi di atas, maka bukan daun-*node*. dan sub-*node* baru dihasilkan sebagai berikut:
 - untuk A_i 's ($i=1, \dots, L$) menghitung mendapatkan informasi, lalu pilih atribut uji A_{max} yang memaksimalkan mereka.
 - membagi D ke $D_1, \dots, D_m$ *Fuzzy subset* menurut A_{max} , dimana nilai keanggotaan data dalam D_j adalah produk dari nilai keanggotaan dalam D dan nilai $F_{max, j}$ nilai A_{max} di D
 - $T_1 \dots T_m$ menghasilkan *node* baru, untuk *Fuzzy subset* $D_1, \dots, D_m$ dan label *Fuzzy set* $F_{max, j}$ ke tepi yang menghubungkan antara *node* t_j , dan t
 - mengganti D oleh D_j ($j = 1, 2, \dots, m$) dan ulangi dari 2 rekursif

2.10 Sistem Inferensi *Fuzzy* Metode Mamdani

Metode Mamdani diperkenalkan oleh Ibrahim Mamdani pada tahun 1975. Metode Mamdani sering juga disebut metode *Max-Min*. Untuk mendapatkan *output* diperlukan 4 tahapan yaitu :

1. Pembentukan Himpunan *Fuzzy*

Pada Metode Mamdani, baik *variable input* maupun *output* dibagi satu atau lebih himpunan *fuzzy*.

2. Aplikasi fungsi Implikasi (Aturan)

Pada metode Mamdani, fungsi implikasi yang digunakan adalah *minimum*.

3. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dari beberapa korelasi antar aturan. Aturan yang digunakan dalam melakukan inferensi sistem *fuzzy* yaitu menggunakan metode *MAX (Maximum)*.

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi

daerah *fuzzy* dan mengaplikasikannya ke *output* dengan menggunakan operator OR. Jika semua proposisi telah diaplikasikan maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] = \text{Max}(\mu_{sf}[x_i], \mu_{kf}[x_i]) \quad \dots\dots\dots (2.11)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen fuzzy aturan ke-i.

4. Penegasan (Defuzzifikasi)

Proses defuzzifikasi adalah proses pengubahan *crisp input* menjadi *fuzzy input*. Untuk mentransformasikan *crisp input* menjadi *fuzzy input*, diperlukan *membership function* untuk tiap *input*. Proses *fuzzy input* mengambil nilai *crisp input* dan membandingkannya dengan *membership function* yang telah ada untuk menghasilkan harga *fuzzy input*. Dalam proses defuzzifikasi dapat diolah lebih dari satu macam *crisp input*. Metode defuzzifikasi yang dipakai pada komposisi aturan Mamdani, yaitu metode *centroid (center of gravity)*

Pada metode ini setiap *membership function output* yang memiliki nilai di atas nilai *fuzzy output* akan dipotong. Pemotongan ini disebut lamda cut. Hasilnya (*membership function* yang terpotong) digabung dulu baru dihitung *center of gravity* keseluruhannya.

Defuzzifikasi pada komposisi aturan Mamdani dengan menggunakan metode *centroid*. Dimana pada metode ini solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Secara umum dirumuskan :

$$\mu(x) = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad \dots\dots\dots (2.12)$$

Ada dua keuntungan menggunakan *metode centroid*, yaitu:

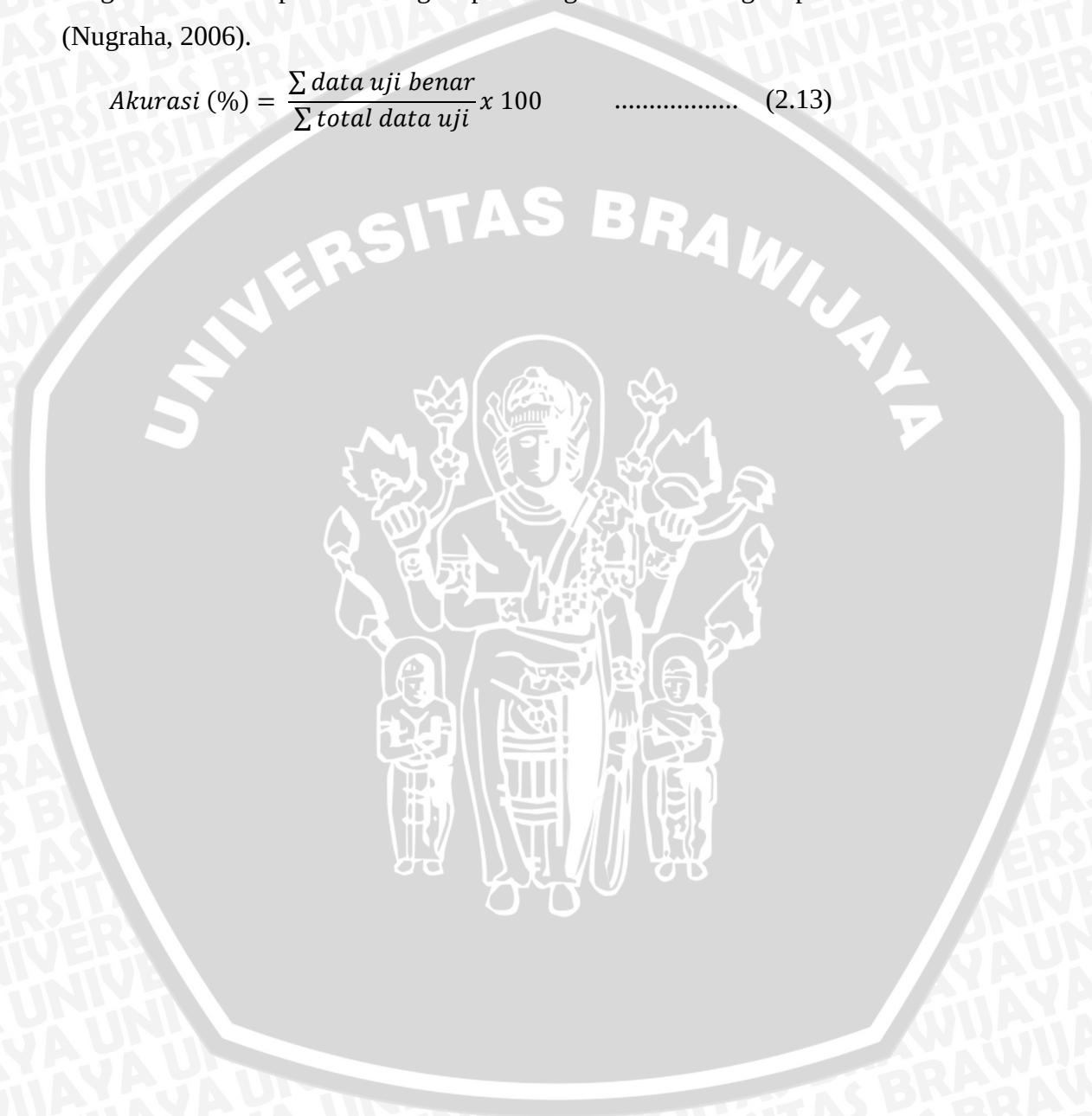
1. Nilai defuzzifikasi akan bergerak secara halus sehingga perubahan dari suatu himpunan *fuzzy* juga akan berjalan dengan halus.
2. Lebih mudah dalam perhitungan

(Kusumadewi, 2004).

2.11 Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (*true value* atau *reference value*). Dalam penelitian ini akurasi diagnosis dihitung dari jumlah diagnosis yang tepat dibagi dengan jumlah data. Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan 2.13 (Nugraha, 2006).

$$\text{Akurasi (\%)} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100 \quad \dots\dots\dots (2.13)$$



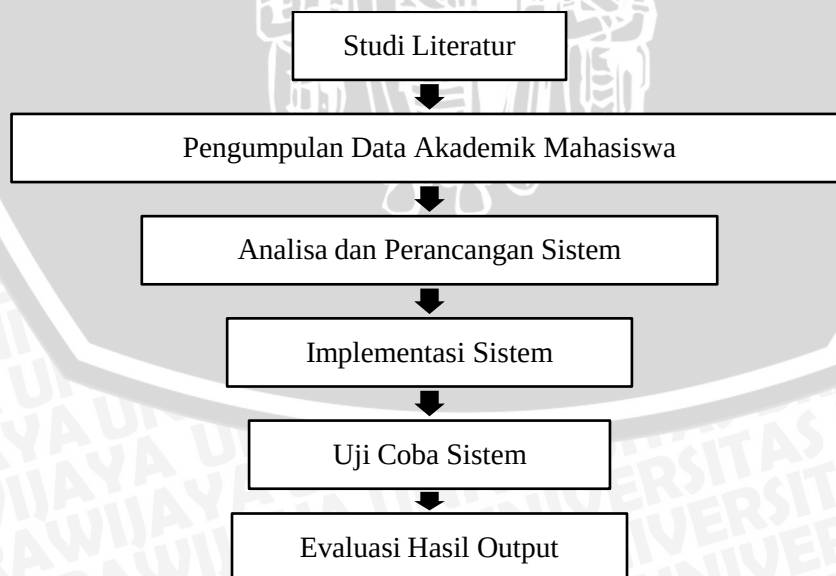
BAB III

METODOLOGI DAN PERANCANGAN

Pada bab metodologi dan perancangan ini akan dibahas metode, rancangan yang digunakan, dan langkah-langkah yang dilakukan dalam penelitian tentang *fuzzy decision tree* dengan algoritma ID3 dalam pembangkitan aturan untuk mengetahui hubungan antara indeks prestasi mahasiswa tahun pertama dengan indeks prestasi kelulusan. Langkah-langkah yang dijalankan dalam penelitian ini, yaitu:

1. Mempelajari literatur yang berhubungan dengan indeks prestasi mahasiswa tahun pertama, indeks prestasi kelulusan dan metode decision tree dengan algoritma ID3.
2. Melakukan pengumpulan data akademik mahasiswa.
3. Menganalisa dan melakukan perancangan sistem dengan metode *fuzzy decision tree* dengan algoritma ID3.
4. Membangun perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan (implementasi).
5. Melakukan uji coba terhadap perangkat lunak.
6. Mengevaluasi *output* yang dihasilkan oleh sistem.

Adapun langkah-langkah penelitian dapat digambarkan dalam bentuk diagram alir yang ditunjukkan pada gambar 3.1 sebagai berikut.



Gambar 3. 1 Diagram Alir Penelitian

3.1 Studi Literatur

Dalam penelitian ini dibutuhkan studi literatur untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori mengenai indeks prestasi mahasiswa, himpunan *fuzzy*, algoritma ID3, dan metode inferensi *fuzzy* Mamdani digunakan sebagai dasar penelitian yang diperoleh dari buku, jurnal dan *browsing* dari internet. Kemudian data yang diperoleh diubah sehingga dapat digunakan untuk analisis. Setelah dianalisis maka dapat diimplementasikan ke dalam program.

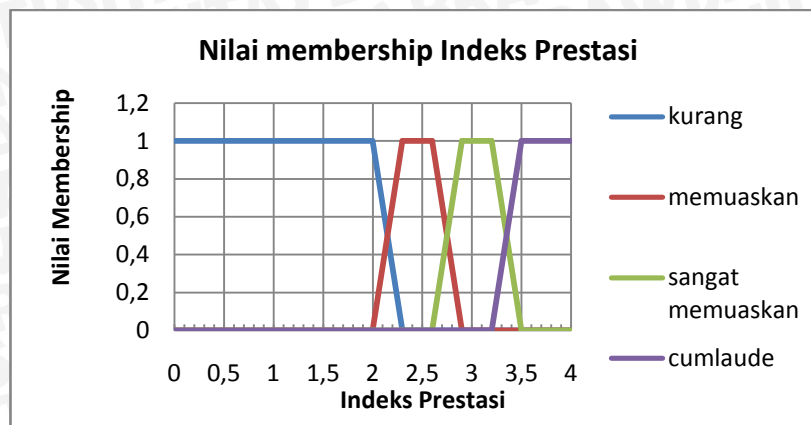
3.2 Data yang digunakan

Data yang digunakan dalam penelitian ini adalah data akademik mahasiswa angkatan 2003 s/d 2006 yang diambil di Bagian Akademik Fakultas MIPA Universitas Brawijaya. Pada *dataset* ini, atribut yang digunakan adalah jenis kelamin, indeks prestasi semester 1, indeks prestasi semester 2. Sedangkan kelas *output* yaitu indeks prestasi kelulusan tanpa mempertimbangkan masa studi dengan kategori:

- 1 (Cumlaude) = Indeks Prestasi $> 3,50$
- 2 (Sangat Memuaskan) = Indeks Prestasi $> 2,75$ dan $\leq 3,50$
- 3 (Memuaskan) = Indeks Prestasi $> 2,00$ dan $\leq 2,75$

Data tersebut akan dibagi menjadi 6 jurusan/program studi yaitu ilmu komputer, statistika, matematika, fisika, kimia dan biologi. Secara keseluruhan rata-rata indeks prestasi kelulusan dari data yang digunakan dalam penelitian ini berada pada kelas kategori sangat memuaskan.

Fungsi keanggotaan yang digunakan untuk atribut indeks prestasi terbagi dalam 4 *linguistik term* yaitu kurang (K), memuaskan (M), sangat memuaskan (SM), dan *cumlaude* (C).



Gambar 3. 2 Nilai Membership Indeks Prestasi

Fungsi keanggotaannya yaitu :

$$\mu_k[x] = \begin{cases} 0; & x > 2,3 \\ \frac{(2,3 - x)}{(0,3)}; & 2 \leq x \leq 2,3 \\ 1; & x < 2 \end{cases}$$

$$\mu_m[x] = \begin{cases} 0; & x < 2 \text{ atau } x > 2,9 \\ \frac{(x - 2)}{(0,3)}; & 2 \leq x < 2,3 \\ 1; & 2,3 \leq x < 2,6 \\ \frac{(2,9 - x)}{0,3}; & 2,6 \leq x < 2,9 \end{cases}$$

$$\mu_{sm}[x] = \begin{cases} 0; & x < 2,6 \text{ atau } x > 3,5 \\ \frac{(x - 2,6)}{(0,3)}; & 2,6 \leq x < 2,9 \\ 1; & 2,9 \leq x < 3,2 \\ \frac{(3,5 - x)}{0,3}; & 3,2 \leq x < 3,5 \end{cases}$$

$$\mu_c[x] = \begin{cases} 0; & x < 3,2 \\ \frac{(x - 3,5)}{(0,3)}; & 3,2 \leq x \leq 3,5 \\ 1; & x > 3,5 \end{cases}$$

3.3 Deskripsi Umum Sistem

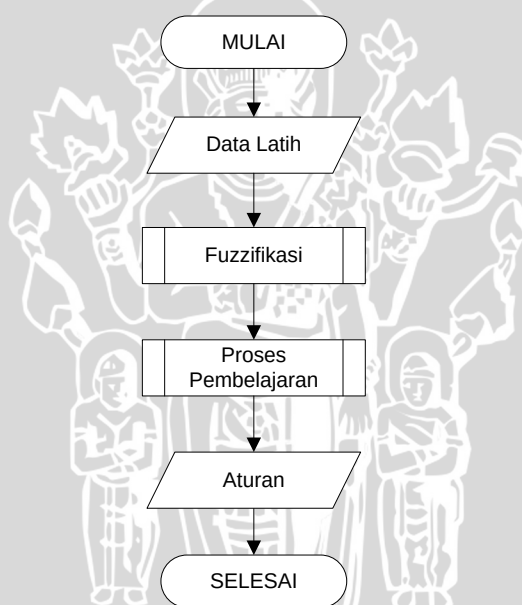
Secara umum sistem yang dibangun adalah suatu perangkat lunak untuk pembangkitan aturan pada data akademik mahasiswa yang mengimplementasikan metode *fuzzy decision tree* dengan algoritma *Iterative Dichotomiser 3* (ID3). Sistem ini bertujuan untuk mengetahui aturan yang berhubungan dengan indeks prestasi mahasiswa tahun pertama dengan indeks prestasi kelulusannya.

3.4 Perancangan Sistem

Pada perancangan sistem ini, diperlukan beberapa tahapan yang perlu dilalui agar dapat membentuk sistem pembangkitan aturan pada data akademik mahasiswa dengan metode *fuzzy Iterative Dichotomiser 3* (*fuzzy ID3*). Terdapat 2 tahap dalam sistem ini, yaitu pembangkitan aturan dan pengujian.

3.4.1 Pembangkitan Aturan

Dalam proses pembangkitan aturan dibutuhkan *input* berupa data latih dalam bentuk *crisp* kemudian akan dilakukan proses fuzzifikasi, selanjutnya membangun *decision tree* menggunakan Algoritma *Iterative Dichotomiser 3* (ID3) yang akan memberi *output* dalam bentuk aturan. Alur sistem ini digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



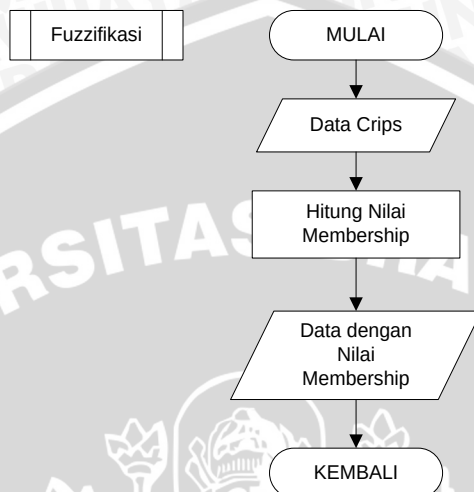
Gambar 3. 3 Pembangkitan Aturan

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Tahapan awal berupa *inputan* data latih yang mulanya dalam bentuk *crisp*
2. Kemudian data tersebut diubah ke dalam bentuk himpunan *fuzzy* dan memperoleh nilai *membership* untuk setiap *record* data
3. Data latih dengan nilai *membership* tersebut akan digunakan dalam membangun *decision tree*.
4. Hasil dari proses *decision tree* adalah aturan.

3.4.1.1 Proses Fuzzifikasi

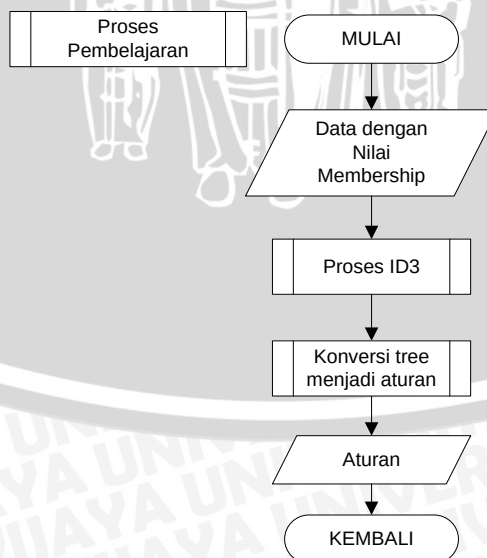
Dalam tahap ini diperlukan *input* berupa data dalam bentuk *crips*, kemudian akan dihitung nilai *membership*nya menggunakan fungsi keanggotaan dan diperoleh *output* berupa *record* data dengan nilai *membership*. Alur sistem ini digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



Gambar 3. 4 Proses Fuzzifikasi

3.4.1.2 Proses Pembelajaran

Setelah proses fuzzifikasi akan dilakukan proses pembelajaran. Dalam proses pembelajaran akan dibentuk *tree* menggunakan proses ID3 dan mengkonversi *tree* menjadi aturan yang nanti akan digunakan dalam proses pengujian.



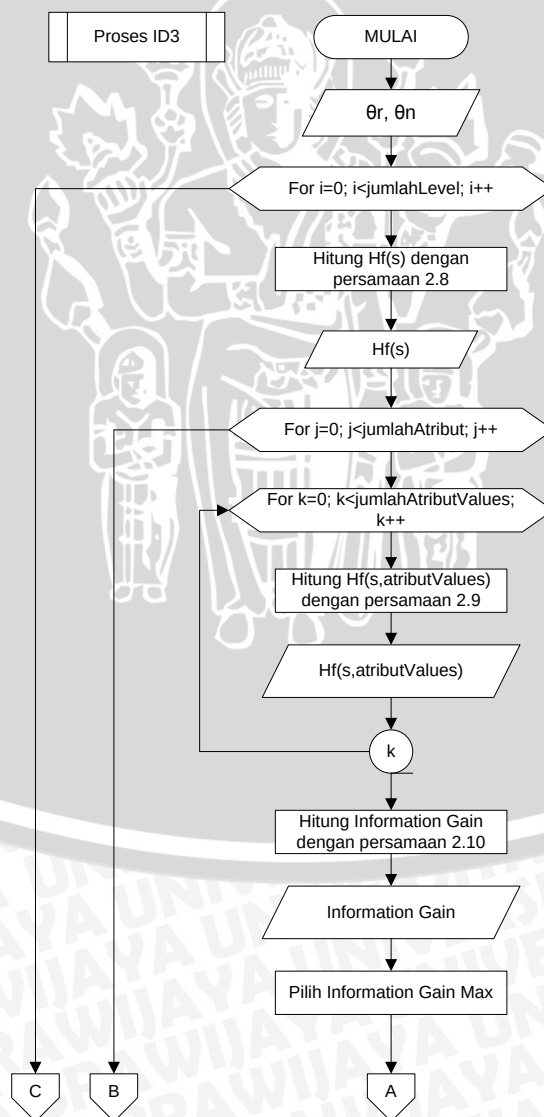
Gambar 3. 5 Proses Pembelajaran

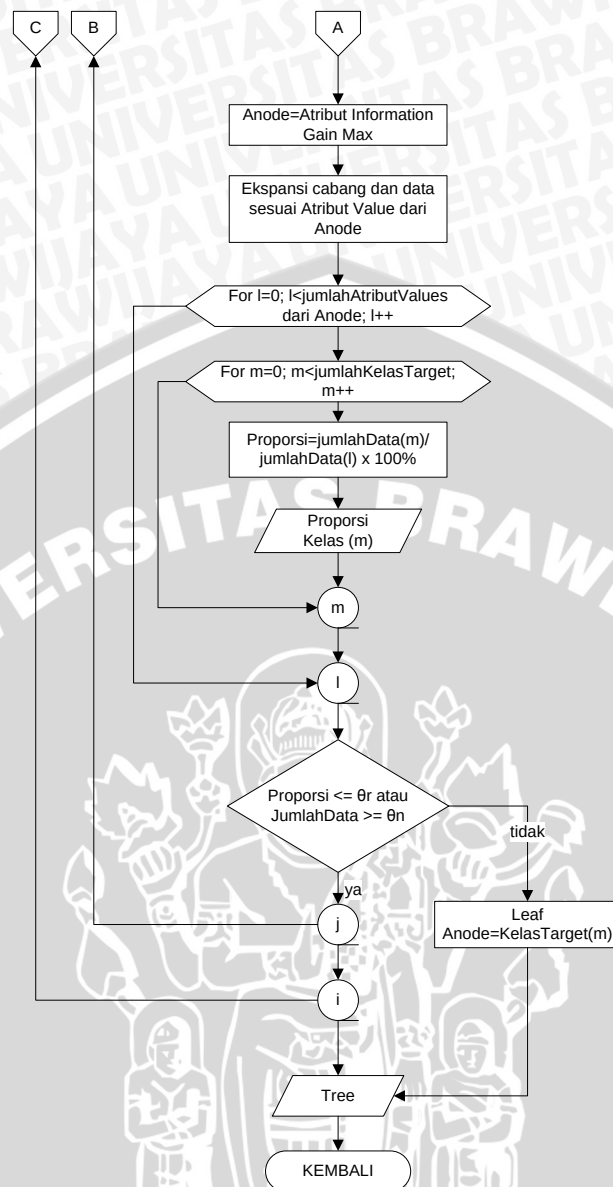
Berdasarkan gambar di atas, langkah-langkah dalam proses pembelajaran adalah sebagai berikut:

1. Sistem mendapatkan data latih berupa data dengan nilai *membership* pada masing-masing atribut value.
2. Sistem membangun *fuzzy decision tree* dari data latih dengan menggunakan algoritma ID3.
3. Hasil berupa *tree*, yang kemudian dikonversi menjadi aturan.
4. Sistem menampilkan aturan yang terbentuk

3.4.1.3 Proses ID3

Proses membangun *tree* dengan algoritma ID3 dapat dilihat pada gambar berikut :





Gambar 3. 6 Proses ID3

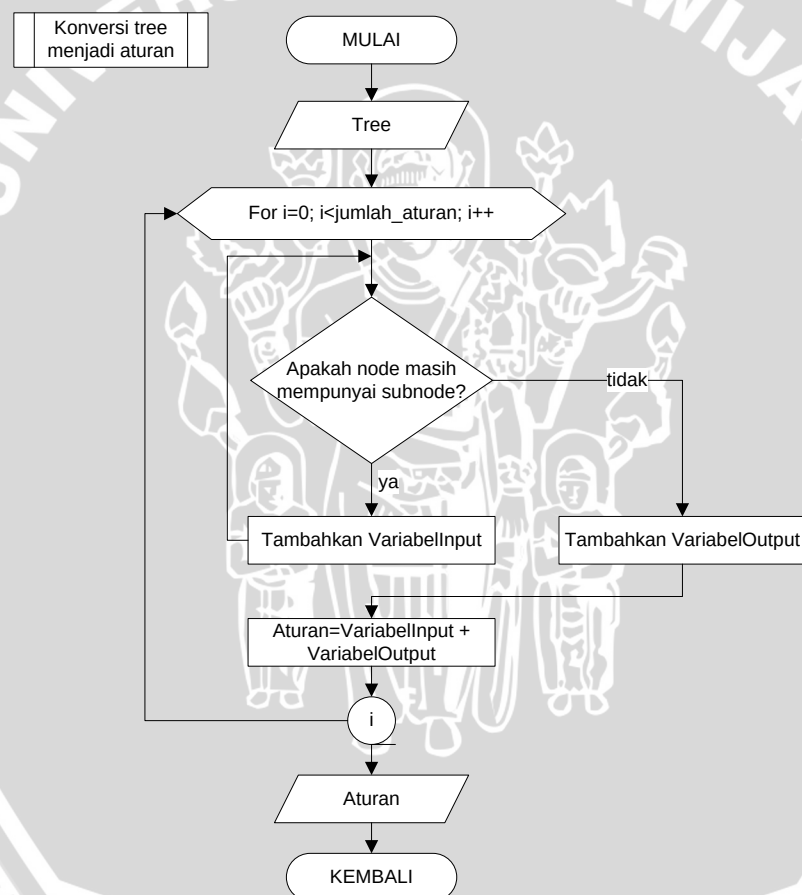
Berdasarkan gambar di atas, langkah-langkah dalam proses ID3 adalah sebagai berikut:

1. Input yang digunakan adalah nilai FCT θ_r dan LDT θ_n
2. Untuk setiap *node* dalam *tree* akan dihitung nilai $H_f(s)$
3. Untuk setiap atribut yang ada akan dihitung nilai $H_f(s, \text{atribut values})$ dan nilai *information gain* atau $G_f(s, \text{atribut})$
4. Akan dipilih nilai *information gain* terbesar untuk menjadi atribut dalam *node* tersebut
5. Tree akan diekspansi sesuai atribut values dari atribut yang terpilih

6. Akan dihitung nilai proporsi kelas target pada setiap cabang yang terbentuk
7. Jika proporsi kelas lebih kecil dari nilai FCT θ_r dan jumlah data masih lebih besar dari jumlah LDT θ_n , maka *tree* akan terus diekspansi
8. Jika proporsi kelas lebih besar atau sama dengan dari nilai FCT θ_r dan jumlah data masih lebih kecil atau sama dengan dari jumlah LDT θ_n , maka *leaf* ditemukan
9. Output yang dihasilkan adalah *tree*.

3.4.1.4 Konversi *Tree* menjadi Aturan

Proses konversi *tree* menjadi aturan dapat dilihat pada gambar berikut :



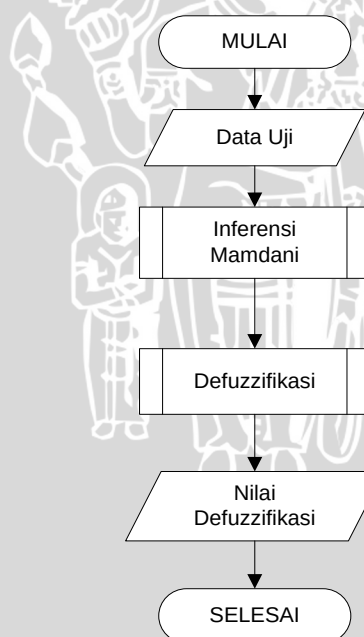
Gambar 3. 7 Konversi *Tree* menjadi Aturan

Berdasarkan gambar di atas, langkah-langkah dalam proses ID3 adalah sebagai berikut:

1. Input berupa *tree* hasil dari proses id3
2. Kemudian akan dilakukan iterasi sebanyak jumlah aturan
3. Untuk setiap cabang akan dicek setiap *node* apakah masih memiliki *subnode* di bawahnya
4. Jika ya, maka variabel *input* untuk aturan ditambahkan
5. Jika tidak, maka variabel *output* untuk aturan ditambahkan
6. Aturan merupakan penggabungan dari variabel *input* dan variabel *output*.

3.4.2 Pengujian Aturan

Aturan yang telah terbentuk pada tahap pembangkitan aturan sebelumnya diuji dengan menggunakan data uji. Pengujian aturan ini dibagi menjadi 2 proses yaitu proses *fuzzy inferensi* dengan metode Mamdani dan proses defuzzifikasi. Alur sistem ini digambarkan dalam bentuk *flowchart* sebagai berikut.



Gambar 3. 8 Pengujian Aturan

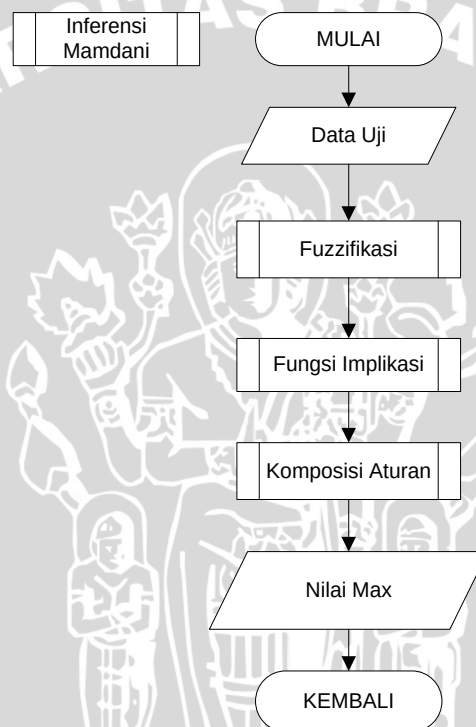
Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Tahapan awal dibutuhkan *input* berupa data uji kemudian data tersebut diproses menggunakan *fuzzy inferensi* dengan metode Mamdani.

2. Kemudian aturan yang diperoleh dari *fuzzy* inferensi dengan metode Mamdani akan dihitung nilai defuzzifikasi menggunakan metode *centroid*.
3. *Output* yang diperoleh berupa nilai defuzzifikasi.

3.4.2.1 Proses Inferensi Mamdani

Aturan yang telah terbentuk disimpan dan akan dipakai untuk proses inferensi. Proses Inferensi yang dipakai yaitu inferensi Mamdani yang merupakan proses pengambilan keputusan dengan metode *Min-Max*. Alur sistem digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



Gambar 3. 9 Proses Inferensi Mamdani

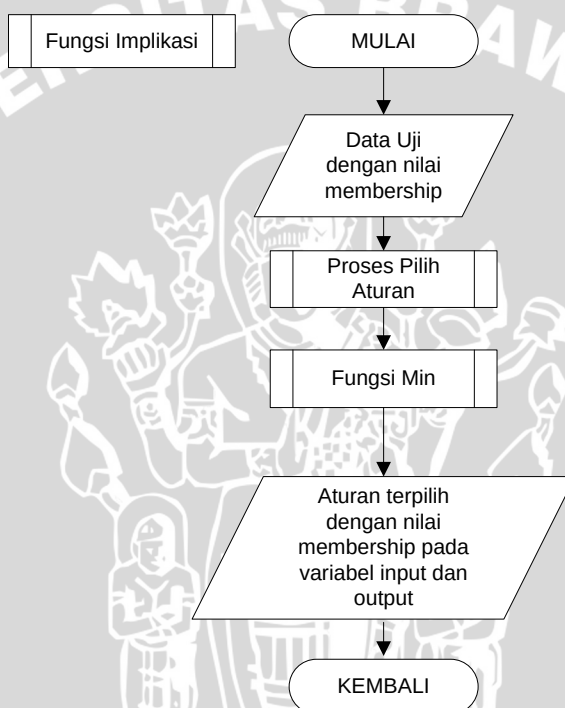
Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Tahapan awal berupa *input* data uji dalam himpunan *crisp*.
2. Kemudian pada data tersebut akan dilakukan proses fuzzifikasi dan diperoleh nilai *membership*nya.
3. Data dengan nilai *membership* akan diproses dalam fungsi implikasi untuk mencari aturan dengan aturan predikat dan memberi nilai *membership* variabel *output*.

4. Dalam komposisi aturan akan dipilih aturan dengan nilai *membership output* yang paling maksimal.
5. *Output* berupa aturan hasil dari proses komposisi aturan.

3.4.2.2 Fungsi Implikasi

Fungsi implikasi yang digunakan dalam inferensi Mamdani adalah fungsi *min*. Nilai *membership* atribut yang mengandung parameter data uji akan dicari nilai yang paling minimum. Alur sistem digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



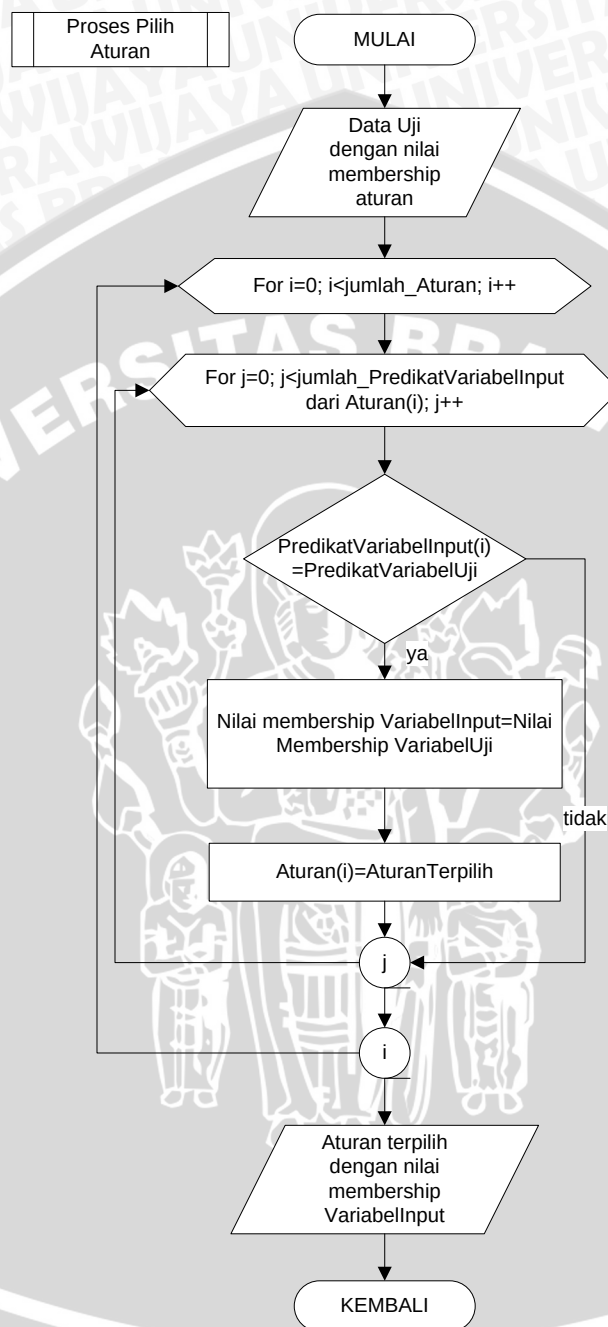
Gambar 3. 10 Fungsi Implikasi

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. *Input* berupa data uji dengan nilai *membership*nya.
2. Mencari dan memilih aturan yang mengandung parameter yang sesuai dengan data uji.
3. Memberikan nilai *membership* dari parameter data uji kepada setiap aturan yang terpilih untuk variabel *input* aturan tersebut.
4. Untuk setiap aturan yang terpilih akan dicari nilai *membership* variabel *input* aturan yang paling minimal dan menjadikannya sebagai nilai *membership* variabel *output* aturan.

3.4.2.3 Proses Pilih Aturan

Proses pilih aturan dapat dilihat pada gambar sebagai berikut :

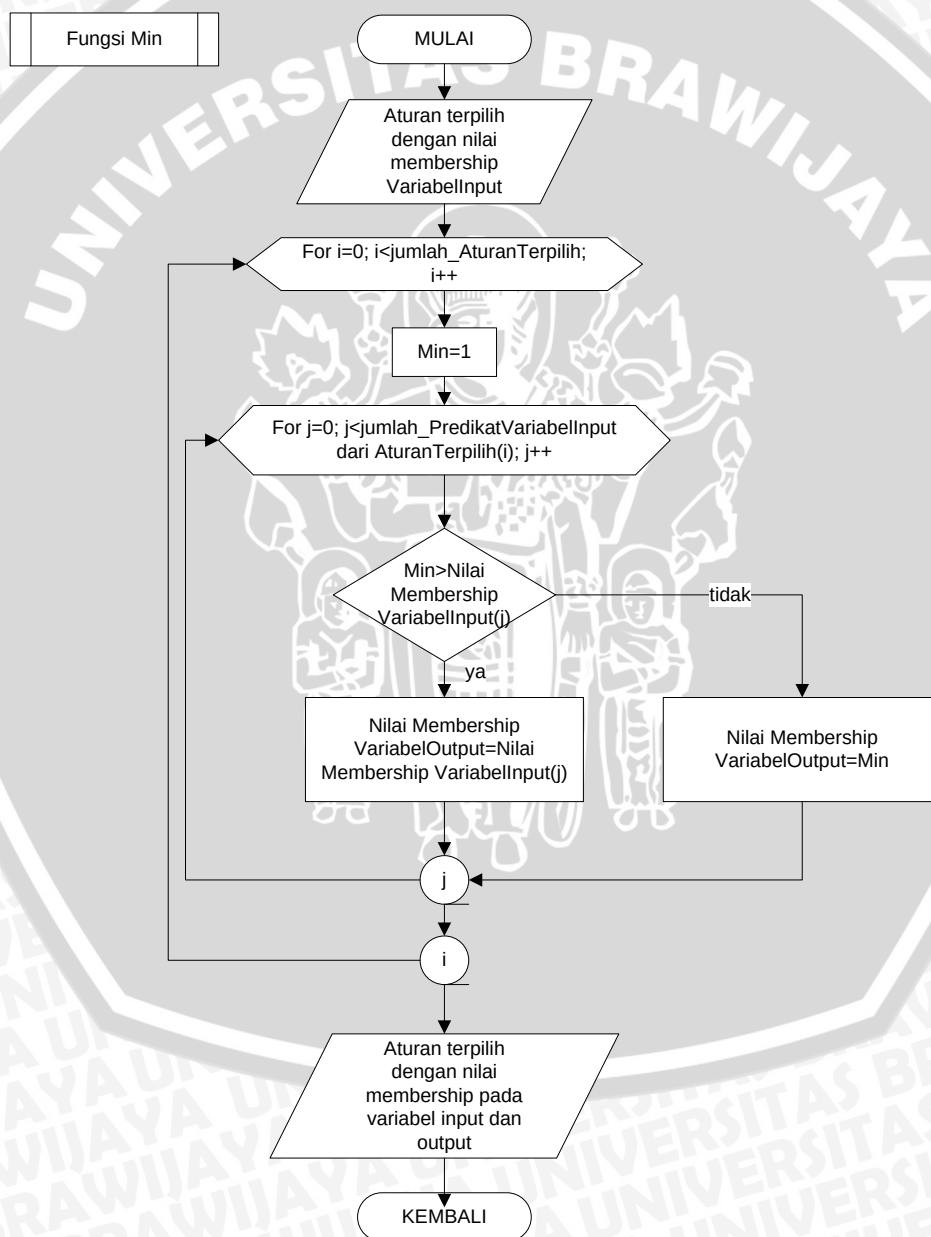


Gambar 3. 11 Proses Pilih Aturan

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Input berupa data uji dengan nilai *membership*nya dan aturan hasil learning.
2. Kemudian untuk setiap aturan learning akan dipilih aturan yang hanya memiliki predikat variabel *input* yang sesuai dengan predikat dari variabel *input* data uji.
3. Setelah itu aturan tersebut disimpan sebagai aturan terpilih.

3.4.2.4 Fungsi Min



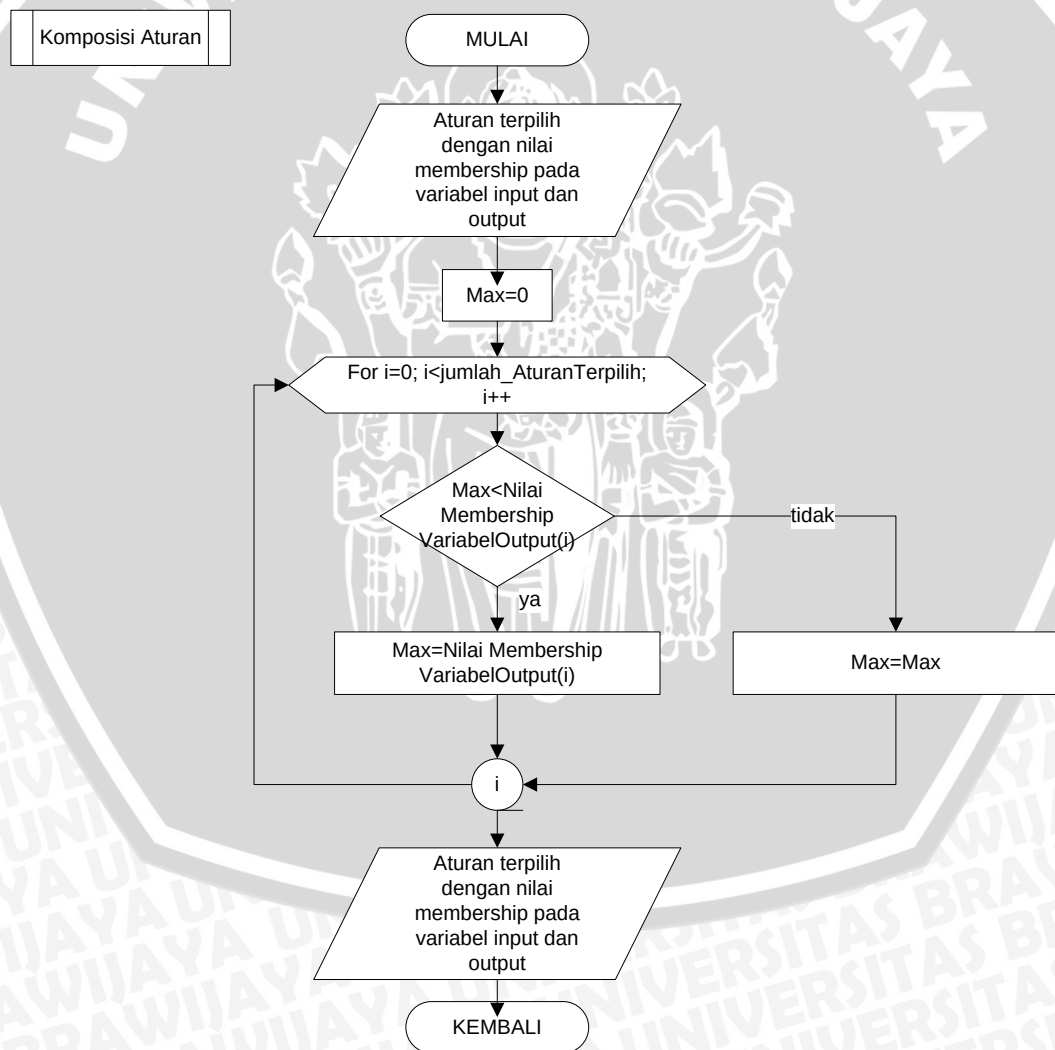
Gambar 3. 12 Fungsi Min

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Input berupa aturan terpilih dengan nilai *membership* pada variabel *input*.
2. Untuk setiap aturan yang terpilih akan dicari nilai paling kecil dari setiap nilai *membership* variabel *input*.
3. Kemudian nilai itu akan menjadi nilai variabel *output* dari aturan tersebut.

3.4.2.5 Komposisi Aturan

Fungsi komposisi adalah proses mencari nilai yang paling maksimum aturan dari nilai *min* yang didapatkan dari fungsi implikasi. Alur sistem digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



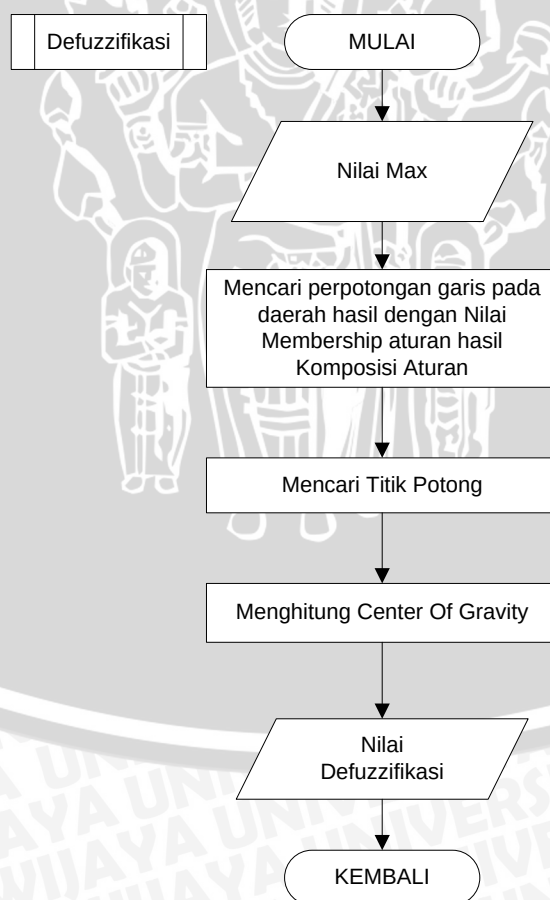
Gambar 3. 13 Komposisi Aturan

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Tahap awal berupa aturan dengan nilai *membership* hasil dari fungsi implikasi.
2. Memilih aturan dengan nilai *membership* variabel *output*nya yang paling maksimal.
3. *Output* berupa aturan dengan nilai *membership* variabel *output* yang paling maksimal.

3.4.2.6 Proses Defuzzifikasi

Proses defuzzifikasi adalah proses pengubahan *crisp input* menjadi *fuzzy input*. Metode defuzzifikasi yang dipakai pada komposisi aturan Mamdani yaitu metode *center of gravity (centroid)*. Alur sistem digambarkan dalam bentuk *flowchart* seperti pada gambar berikut.



Gambar 3. 14 Proses Defuzzifikasi

Berikut ini adalah penjelasan tahapan pada alur sistem:

1. Tahapan awal berupa *inputan* nilai *max* dari aturan hasil komposisi
2. Kemudian akan dicari perpotongan garis pada daerah hasil dengan nilai *membership* aturan tersebut.
3. Dari perpotongan garis tersebut akan dicari titik potongnya.
4. Perhitungan *center of gravity* berdasarkan persamaan 2.12 kemudian didapatkan nilai defuzzifikasi.

3.5 Contoh Perhitungan Manual

Berikut ini merupakan contoh perhitungan manual dalam pembangkitan aturan untuk mengetahui hubungan indeks prestasi mahasiswa tahun pertama dengan indeks prestasi kelulusannya menggunakan *fuzzy ID3* pada 35 *record* data pelatihan.

3.5.1 Pelatihan

Pada proses pelatihan ini akan dilakukan pembentukan *tree* dari 35 *record* data menggunakan metode *fuzzy ID3*. Data pelatihan yang akan digunakan akan ditunjukkan pada tabel berikut.

Tabel 3. 1 Data Pelatihan

Data Ke-	Jenis Kelamin	IP Semester 1	IP Semester 2	Kategori
1	L	2	2,07	2
2	L	2,22	2,78	2
3	L	2,44	1,97	3
4	L	2,47	2,25	3
5	L	2,5	2,72	2
6	P	2,53	1,47	2
7	L	2,61	3,31	2
8	P	2,69	2,88	2
9	P	2,69	3,07	2
10	L	2,75	2,11	2
11	L	2,78	2,93	2
12	L	2,78	2,74	2
13	P	2,86	3,24	2
14	L	2,89	2,36	2
15	P	2,92	3,6	2
16	L	2,94	2,81	2
17	L	3,03	2,5	2

18	P	3,03	3,38	2
19	L	3,1	2,46	3
20	P	3,14	3,21	2
21	L	3,17	2,24	3
22	P	3,19	3,41	1
23	L	3,25	3,48	2
24	L	3,27	2,79	2
25	P	3,28	3,5	2
26	L	3,28	2,71	2
27	L	3,31	3,52	1
28	L	3,39	3,26	1
29	L	3,47	3,57	2
30	P	3,47	3,52	1
31	L	3,53	3,24	1
32	L	3,55	2,68	2
33	P	3,56	3,33	2
34	L	3,56	3,17	2
35	P	3,67	3,41	1

Pada *dataset* ini, atribut yang digunakan adalah jenis kelamin, indeks prestasi semester 1, indeks prestasi semester 2. Sedangkan kelas *output* yaitu indeks prestasi kelulusan yang disimbolkan dengan kategori yaitu :

Kategori 1 (Cumlaude) = Indeks Prestasi $> 3,50$

Kategori 2 (Sangat Memuaskan) = Indeks Prestasi $> 2,75$ dan $\leq 3,50$

Kategori 3 (Memuaskan) = Indeks Prestasi $> 2,00$ dan $\leq 2,75$

Kategori Cumlaude yang dimaksudkan dalam penelitian hanya terkait indeks prestasi yang diraih tanpa mempertimbangkan masa studi mahasiswa.

Berikut ini adalah tahapan dalam perhitungan manual.

1. **Fuzzifikasi**

Dalam tahap fuzzifikasi ini kita akan merubah variabel *input* yang pada awalnya berada dalam himpunan *crisp* akan dirubah ke dalam bentuk himpunan *fuzzy*. Langkah awal yang akan dilakukan yaitu membentuk fungsi keanggotaan. Fungsi keanggotaan untuk variabel *input* dibuat dalam bentuk kurva setengah trapesium dan segitiga.

a. Atribut Jenis Kelamin

Dalam atribut jenis kelamin ada 2 *linguistik term* yaitu laki-laki dan perempuan. Karena pada kenyataannya atribut jenis kelamin tidak dapat

dibentuk dalam himpunan *fuzzy*, maka atribut ini tetap dalam himpunan *crisp*.

b. Atribut IP Semester 1

Sesuai dengan fungsi keanggotaan untuk himpunan *fuzzy*, maka langkah selanjutnya yaitu menghitung nilai *membership* untuk tiap *record* data pada setiap atribut.

Tabel 3. 2 Nilai *Membership* IP Semester 1

Data Ke-	IP Semester 1	Mf Kurang	Mf Memuaskan	Mf Sangat Memuaskan	Mf Cumlaude	Kategori
1	2	1,000	0,000	0,000	0,000	2
2	2,22	0,267	0,733	0,000	0,000	2
3	2,44	0,000	1,000	0,000	0,000	3
4	2,47	0,000	1,000	0,000	0,000	3
5	2,5	0,000	1,000	0,000	0,000	2
6	2,53	0,000	1,000	0,000	0,000	2
7	2,61	0,000	0,967	0,033	0,000	2
8	2,69	0,000	0,700	0,300	0,000	2
9	2,69	0,000	0,700	0,300	0,000	2
10	2,75	0,000	0,500	0,500	0,000	2
11	2,78	0,000	0,400	0,600	0,000	2
12	2,78	0,000	0,400	0,600	0,000	2
13	2,86	0,000	0,133	0,867	0,000	2
14	2,89	0,000	0,033	0,967	0,000	2
15	2,92	0,000	0,000	1,000	0,000	2
16	2,94	0,000	0,000	1,000	0,000	2
17	3,03	0,000	0,000	1,000	0,000	2
18	3,03	0,000	0,000	1,000	0,000	2
19	3,1	0,000	0,000	1,000	0,000	3
20	3,14	0,000	0,000	1,000	0,000	2
21	3,17	0,000	0,000	1,000	0,000	3
22	3,19	0,000	0,000	1,000	0,000	1
23	3,25	0,000	0,000	0,833	0,167	2
24	3,27	0,000	0,000	0,767	0,233	2
25	3,28	0,000	0,000	0,733	0,267	2
26	3,28	0,000	0,000	0,733	0,267	2
27	3,31	0,000	0,000	0,633	0,367	1
28	3,39	0,000	0,000	0,367	0,633	1
29	3,47	0,000	0,000	0,100	0,900	2
30	3,47	0,000	0,000	0,100	0,900	1
31	3,53	0,000	0,000	0,000	1,000	1
32	3,55	0,000	0,000	0,000	1,000	2
33	3,56	0,000	0,000	0,000	1,000	2
34	3,56	0,000	0,000	0,000	1,000	2
35	3,67	0,000	0,000	0,000	1,000	1
SUM		1,267	8,567	16,433	8,733	

Keterangan tabel:

Mf : *Membership Fuzzy*

Kategori 1 = Indeks Prestasi > 3,50

Kategori 2 = Indeks Prestasi > 2,75 dan $\leq 3,50$

Kategori 3 = Indeks Prestasi > 2,00 dan $\leq 2,75$

c. Atribut IP Semester 2

Sesuai dengan fungsi keanggotaan untuk himpunan *fuzzy*, maka langkah selanjutnya yaitu menghitung nilai *membership* untuk tiap *record* data pada setiap atribut.

Tabel 3. 3 Nilai *Membership* IP Semester 2

Data Ke-	IP Semester 2	Mf Kurang	Mf Memuaskan	Mf Sangat Memuaskan	Mf Cumlaude	Kategori
1	2,07	0,767	0,233	0,000	0,000	2
2	2,78	0,000	0,400	0,600	0,000	2
3	1,97	1,000	0,000	0,000	0,000	3
4	2,25	0,167	0,833	0,000	0,000	3
5	2,72	0,000	0,600	0,400	0,000	2
6	1,47	1,000	0,000	0,000	0,000	2
7	3,31	0,000	0,000	0,633	0,367	2
8	2,88	0,000	0,067	0,933	0,000	2
9	3,07	0,000	0,000	1,000	0,000	2
10	2,11	0,633	0,367	0,000	0,000	2
11	2,93	0,000	0,000	1,000	0,000	2
12	2,74	0,000	0,533	0,467	0,000	2
13	3,24	0,000	0,000	0,867	0,133	2
14	2,36	0,000	1,000	0,000	0,000	2
15	3,6	0,000	0,000	0,000	1,000	2
16	2,81	0,000	0,300	0,700	0,000	2
17	2,5	0,000	1,000	0,000	0,000	2
18	3,38	0,000	0,000	0,400	0,600	2
19	2,46	0,000	1,000	0,000	0,000	3
20	3,21	0,000	0,000	0,967	0,033	2
21	2,24	0,200	0,800	0,000	0,000	3
22	3,41	0,000	0,000	0,300	0,700	1
23	3,48	0,000	0,000	0,067	0,933	2
24	2,79	0,000	0,367	0,633	0,000	2
25	3,5	0,000	0,000	0,000	1,000	2
26	2,71	0,000	0,633	0,367	0,000	2
27	3,52	0,000	0,000	0,000	1,000	1
28	3,26	0,000	0,000	0,800	0,200	1
29	3,57	0,000	0,000	0,000	1,000	2
30	3,52	0,000	0,000	0,000	1,000	1
31	3,24	0,000	0,000	0,867	0,133	1
32	2,68	0,000	0,733	0,267	0,000	2
33	3,33	0,000	0,000	0,567	0,433	2
34	3,17	0,000	0,000	1,000	0,000	2
35	3,41	0,000	0,000	0,300	0,700	1
SUM		3,767	8,867	13,133	9,233	

Keterangan tabel:

Mf : *Membership Fuzzy*

Kategori 1 = Indeks Prestasi > 3,50

Kategori 2 = Indeks Prestasi > 2,75 dan ≤ 3,50

Kategori 3 = Indeks Prestasi > 2,00 dan ≤ 2,75

2. Level 0 Iterasi ke-1

Setelah memperoleh nilai *membership* dari setiap atribut *input* yang digunakan, langkah selanjutnya yaitu menghitung nilai *entropy* dan *information gain* dari seluruh data.

a. Perhitungan nilai *entropy* seluruh data

$$H_f(s) = -\left(\frac{6}{35} \log_2 \frac{6}{35}\right) - \left(\frac{25}{35} \log_2 \frac{25}{35}\right) - \left(\frac{4}{35} \log_2 \frac{4}{35}\right) = 1,14054$$

b. Perhitungan nilai *entropy* dan *information gain* tiap atribut

Selanjutnya akan dihitung nilai *entropy* dari setiap atribut values dan nilai *information gain* dari tiap atribut.

Atribut Semester 1

$$H_f(smt_1, k) = -(0) - \left(\frac{1,267}{1,267} \log_2 \frac{1,267}{1,267}\right) - (0) = 0$$

$$H_f(smt_1, m) = -(0) - \left(\frac{6,567}{8,567} \log_2 \frac{6,567}{8,567}\right) - \left(\frac{2}{8,567} \log_2 \frac{2}{8,567}\right) = 0,78398$$

$$H_f(smt_1, sm) = -\left(\frac{2,1}{16,433} \log_2 \frac{2,1}{16,433}\right) - \left(\frac{12,333}{16,433} \log_2 \frac{12,333}{16,433}\right) - \left(\frac{2}{16,433} \log_2 \frac{2}{16,433}\right) = 1,05987$$

$$H_f(smt_1, c) = -\left(\frac{3,9}{8,733} \log_2 \frac{3,9}{8,733}\right) - \left(\frac{4,833}{8,733} \log_2 \frac{4,833}{8,733}\right) - (0) = 0,99175$$

$$G_f(s, smt_1) = 1,14054 - \left(\frac{1,267}{35} \times 0\right) - \left(\frac{8,567}{35} \times 0,78398\right) - \left(\frac{16,433}{35} \times 1,05987\right) - \left(\frac{8,733}{35} \times 0,99175\right) = 0,20356$$

Atribut Semester 2

$$H_f(smt_2, k) = -(0) - \left(\frac{2,4}{3,767} \log_2 \frac{2,4}{3,767}\right) - \left(\frac{1,367}{3,767} \log_2 \frac{1,367}{3,767}\right) = 0,94505$$

$$H_f(smt_2, m) = -(0) - \left(\frac{6,234}{8,867} \log_2 \frac{6,234}{8,867}\right) - \left(\frac{2,633}{8,867} \log_2 \frac{2,633}{8,867}\right) = 0,87752$$

$$H_f(smt_2, sm) = -\left(\frac{2,267}{13,133} \log_2 \frac{2,267}{13,133}\right) - \left(\frac{10,866}{13,133} \log_2 \frac{10,866}{13,133}\right) - (0) = 0,66366$$

$$H_f(smt_2, c) = -\left(\frac{3,733}{9,233} \log_2 \frac{3,733}{9,233}\right) - \left(\frac{5,5}{9,233} \log_2 \frac{5,5}{9,233}\right) - (0) = 0,97342$$

$$G_f(s, smt_2) = 1,14054 - \left(\frac{3,767}{35} \times 0,94505\right) - \left(\frac{8,867}{35} \times 0,87752\right) - \left(\frac{13,133}{35} \times 0,66366\right) - \left(\frac{9,233}{35} \times 0,97342\right) = 0,31069$$

Atribut Jenis Kelamin

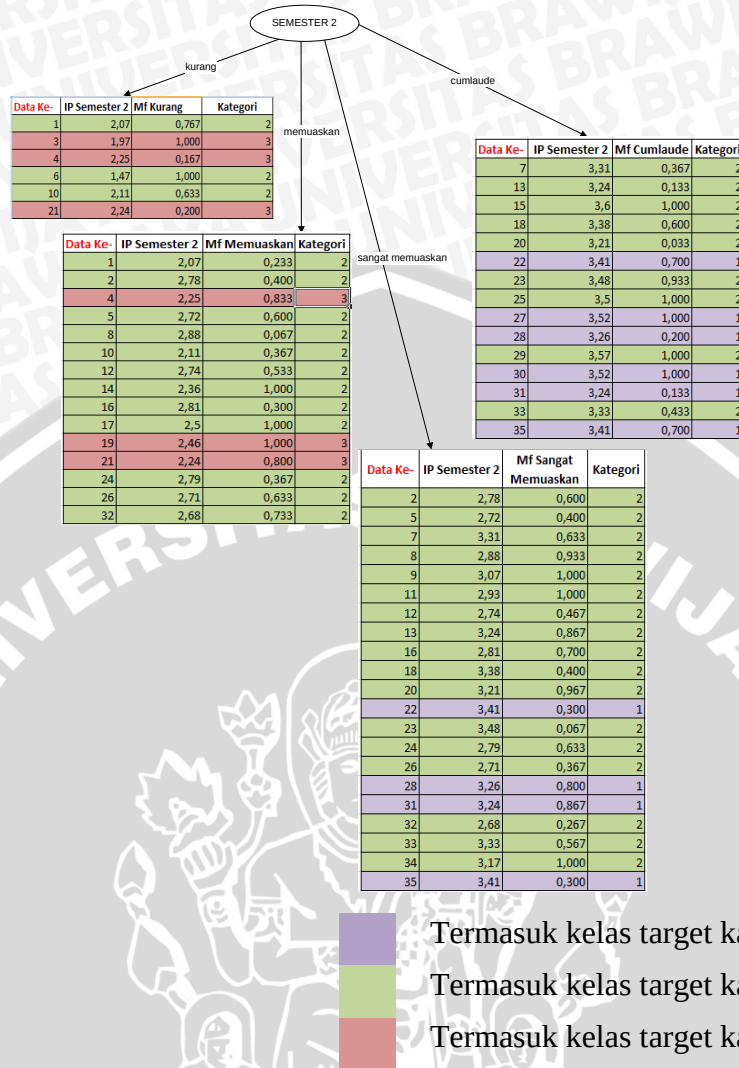
$$H_f(jk, l) = -\left(\frac{3}{23} \log_2 \frac{3}{23}\right) - \left(\frac{16}{23} \log_2 \frac{16}{23}\right) - \left(\frac{4}{23} \log_2 \frac{4}{23}\right) = 1,18639$$

$$H_f(jk, p) = -\left(\frac{3}{12} \log_2 \frac{3}{12}\right) - \left(\frac{9}{12} \log_2 \frac{9}{12}\right) - (0) = 0,81128$$

$$G_f(s, jk) = 1,14054 - \left(\frac{23}{35} \times 1,18639\right) - \left(\frac{12}{35} \times 0,81128\right) = 0,08275$$

c. Pembentukan *tree*

Dari 3 atribut, *information gain* terbesar dimiliki oleh atribut IP Semester 2, sehingga atribut ini yang akan menjadi *node root*.



Gambar 3. 15 Tree Tahap 1

d. **Perhitungan proporsi kelas**

Kemudian akan dihitung proporsi kelas dari tiap cabang yang terbentuk.

Cabang Semester 2 → Kurang

$$c_1 = 0\%$$

$$c_2 = \frac{0,767 + 0,633 + 1}{3,767} \times 100\% = 63,21\%$$

$$c_3 = \frac{1 + 0,167 + 0,2}{3,767} \times 100\% = 36,29\%$$

Cabang Semester 2 → Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{6,234}{8,867} \times 100\% = 70,3\%$$

$$c_3 = \frac{2,633}{8,867} \times 100\% = 29,7\%$$

Cabang Semester 2 → Sangat Memuaskan

$$c_1 = \frac{2,267}{13,133} \times 100\% = 17,26\%$$

$$c_2 = \frac{10,866}{13,133} \times 100\% = 82,74\%$$

$$c_3 = 0\%$$

Cabang Semester 2 → Cumlaude

$$c_1 = \frac{3,733}{9,233} \times 100\% = 40,43\%$$

$$c_2 = \frac{5,5}{9,233} \times 100\% = 59,57\%$$

$$c_3 = 0\%$$

Fuzziness Control Threshold (FCT) / ϕ_r yang digunakan adalah 80% sehingga cabang “semester 2 → sangat memuaskan” akan berhenti diekspansi karena proporsi kelasnya lebih besar dari ϕ_r yang berarti 80% data pada cabang tersebut sudah berada pada kelas yang sama. Sedangkan cabang yang lain akan terus diekspansi karena proporsi kelasnya lebih kecil dari ϕ_r .

3. Level 1 Iterasi ke-1

Tabel 3. 4 Level 1 Iterasi ke-1

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Kurang	Mf Memuaskan	Mf Sangat Memuaskan	Mf Cumlaude	IP Semester 2	Mf Kurang	Kategori
1	L	2	1,000	0,000	0,000	0,000	2,07	0,767	2
3	L	2,44	0,000	1,000	0,000	0,000	1,97	1,000	3
4	L	2,47	0,000	1,000	0,000	0,000	2,25	0,167	3
6	P	2,53	0,000	1,000	0,000	0,000	1,47	1,000	2
10	L	2,75	0,000	0,500	0,500	0,000	2,11	0,633	2
21	L	3,17	0,000	0,000	1,000	0,000	2,24	0,200	3
SUM			1,000	3,500	1,500	0,000		3,767	



Termasuk kelas target kategori 1

Termasuk kelas target kategori 2

Termasuk kelas target kategori 3

a. Perhitungan nilai *entropy* seluruh data

$$H_f(s) = -(0) - \left(\frac{3}{6} \log_2 \frac{3}{6}\right) - \left(\frac{3}{6} \log_2 \frac{3}{6}\right) = 1$$

b. Perhitungan nilai *entropy* dan *information gain* tiap atribut

Selanjutnya akan dihitung nilai *entropy* dari setiap atribut values dan nilai *information gain* dari tiap atribut

Atribut Semester 1

$$H_f(smt_1, k) = -(0) - \left(\frac{1}{1} \log_2 \frac{1}{1}\right) - (0) = 0$$

$$H_f(smt_1, m) = -(0) - \left(\frac{1,5}{3,5} \log_2 \frac{1,5}{3,5}\right) - \left(\frac{2}{3,5} \log_2 \frac{2}{3,5}\right) = 0,98523$$

$$H_f(smt_1, sm) = -(0) - \left(\frac{0,5}{1,5} \log_2 \frac{0,5}{1,5}\right) - \left(\frac{1}{1,5} \log_2 \frac{1}{1,5}\right) = 0,91830$$

$$H_f(smt_1, c) = 0$$

$$G_f(s, smt_1) = 1 - \left(\frac{3,5}{6} \times 0,98523\right) - \left(\frac{1,5}{6} \times 0,91830\right) = 0,19570$$

Atribut Jenis Kelamin

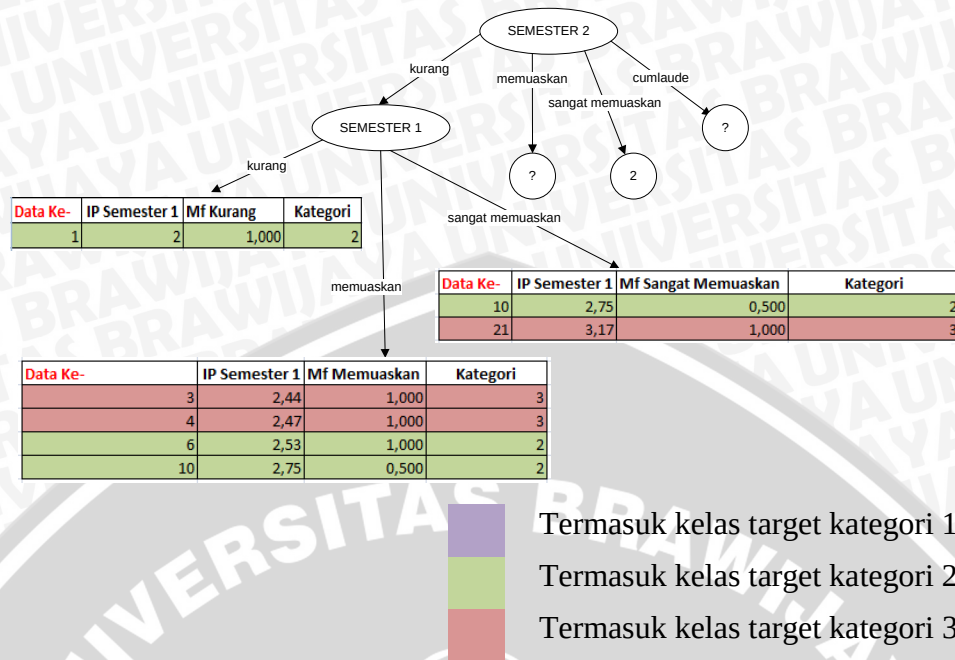
$$H_f(jk, l) = -(0) - \left(\frac{2}{5} \log_2 \frac{2}{5}\right) - \left(\frac{3}{5} \log_2 \frac{3}{5}\right) = 0,97095$$

$$H_f(jk, p) = -(0) - \left(\frac{1}{1} \log_2 \frac{1}{1}\right) - (0) = 0$$

$$G_f(s, jk) = 1 - \left(\frac{5}{6} \times 0,98523\right) = 0,19087$$

c. Pembentukan *tree*

Dari 2 atribut, *information gain* terbesar dimiliki oleh atribut IP Semester 1, sehingga atribut ini yang akan menjadi *node root* selanjutnya.



Gambar 3.16 Tree Tahap 2

d. Perhitungan proporsi kelas

Kemudian akan dihitung proporsi kelas dari tiap cabang yang terbentuk.

Cabang Semester 1 → Kurang

$$c_1 = 0\%$$

$$c_2 = \frac{1}{1} \times 100\% = 100\%$$

$$c_3 = 0\%$$

Cabang Semester 1 → Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{1,5}{3,5} \times 100\% = 42,85\%$$

$$c_3 = \frac{2}{3,5} \times 100\% = 57,15\%$$

Cabang Semester 1 → Sangat Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{0,5}{1,5} \times 100\% = 33,33\%$$

$$c_3 = \frac{1}{1,5} \times 100\% = 66,67\%$$

Fuzziness Control Threshold (FCT) / ϕ_r yang digunakan adalah 80% sehingga cabang “semester 1 →kurang” akan berhenti diekspansi karena proporsi kelasnya lebih besar dari ϕ_r . Sedangkan cabang yang lain akan terus diekspansi karena proporsi kelasnya lebih kecil dari ϕ_r .

4. Level 2 Iterasi ke-1

Setiap cabang yang proporsi kelasnya lebih kecil dari ϕ_r akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 5 Level 2 Iterasi ke-1

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Memuaskan	Kategori
3	L	2,44	1,000	3
4	L	2,47	1,000	3
6	P	2,53	1,000	2
10	L	2,75	0,500	2



Termasuk kelas target kategori 1

Termasuk kelas target kategori 2

Termasuk kelas target kategori 3

a. Perhitungan proporsi kelas

Cabang Jenis Kelamin → Laki-Laki

$$c_1 = 0\%$$

$$c_2 = \frac{1}{3} \times 100\% = 33,33\%$$

$$c_3 = \frac{2}{3} \times 100\% = 66,67\%$$

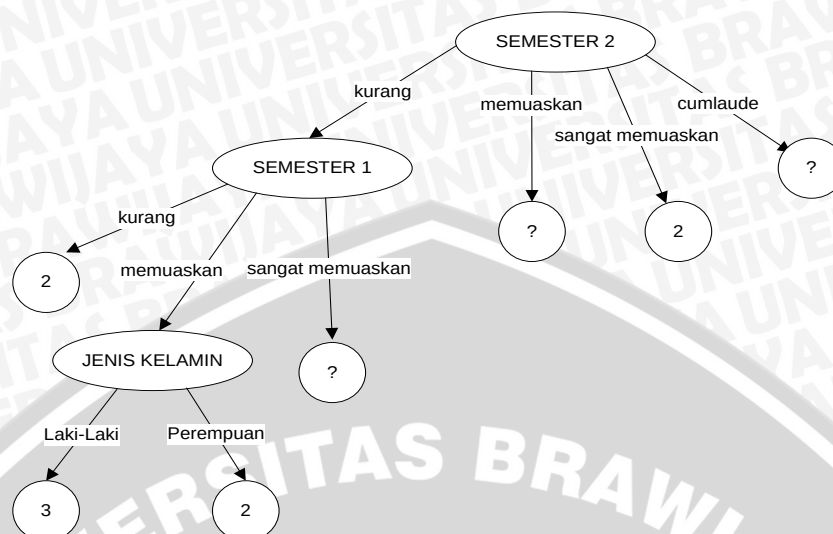
Cabang Jenis Kelamin → Perempuan

$$c_1 = 0\%$$

$$c_2 = \frac{1}{1} \times 100\% = 100\%$$

$$c_3 = \frac{2}{3} \times 100\% = 66,67\%$$

b. Pembentukan tree



Gambar 3. 17 Tree Tahap 3

5. Level 2 Iterasi ke-2

Setiap cabang yang proporsi kelasnya lebih kecil dari $\emptyset_r$ akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 6 Level 2 Iterasi ke-2

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Sangat Memuaskan	Kategori
10	L	2,75	0,500	2
21	L	3,17	1,000	3



Termasuk kelas target kategori 1

Termasuk kelas target kategori 2

Termasuk kelas target kategori 3

a. Perhitungan proporsi kelas

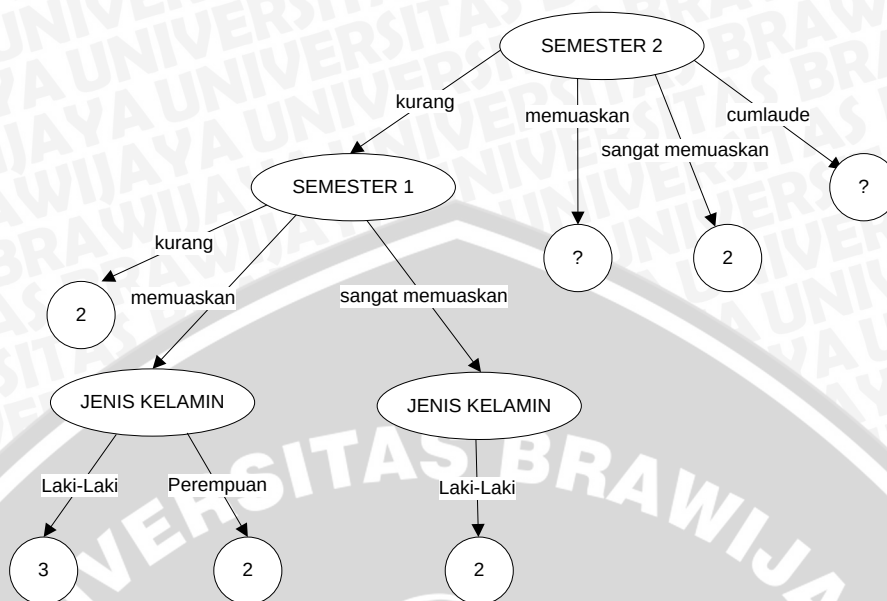
Cabang Jenis Kelamin → Laki-Laki

$$c_1 = 0\%$$

$$c_2 = \frac{1}{2} \times 100\% = 50\%$$

$$c_3 = \frac{1}{2} \times 100\% = 50\%$$

b. Pembentukan tree



Gambar 3. 18 Tree Tahap 4

6. Level 1 Iterasi ke-2

Tabel 3. 7 Level 1 Iterasi ke-2

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Kurang	Mf Memuaskan	Mf Sangat Memuaskan	Mf Cumlaude	IP Semester 2	Mf Memuaskan	Kategori
1	L	2	1,000	0,000	0,000	0,000	2,07	0,233	2
2	L	2,22	0,267	0,733	0,000	0,000	2,78	0,400	2
4	L	2,47	0,000	1,000	0,000	0,000	2,25	0,833	3
5	L	2,5	0,000	1,000	0,000	0,000	2,72	0,600	2
8	P	2,69	0,000	0,700	0,300	0,000	2,88	0,067	2
10	L	2,75	0,000	0,500	0,500	0,000	2,11	0,367	2
12	L	2,78	0,000	0,400	0,600	0,000	2,74	0,533	2
14	L	2,89	0,000	0,033	0,967	0,000	2,36	1,000	2
16	L	2,94	0,000	0,000	1,000	0,000	2,81	0,300	2
17	L	3,03	0,000	0,000	1,000	0,000	2,5	1,000	2
19	L	3,1	0,000	0,000	1,000	0,000	2,46	1,000	3
21	L	3,17	0,000	0,000	1,000	0,000	2,24	0,800	3
24	L	3,27	0,000	0,000	0,767	0,233	2,79	0,367	2
26	L	3,28	0,000	0,000	0,733	0,267	2,71	0,633	2
32	L	3,55	0,000	0,000	0,000	1,000	2,68	0,733	2



Termasuk kelas target kategori 1

Termasuk kelas target kategori 2

Termasuk kelas target kategori 3

a. **Perhitungan nilai *entropy* seluruh data**

$$H_f(s) = -(0) - \left(\frac{12}{15} \log_2 \frac{12}{15}\right) - \left(\frac{3}{15} \log_2 \frac{3}{15}\right) = 0,72193$$

b. **Perhitungan nilai *entropy* dan *information gain* tiap atribut**

Selanjutnya akan dihitung nilai *entropy* dari setiap atribut values dan nilai *information gain* dari tiap atribut

Atribut Semester 1

$$H_f(smt_1, k) = -(0) - \left(\frac{1,267}{1,267} \log_2 \frac{1,267}{1,267}\right) - (0) = 0$$

$$H_f(smt_1, m) = -(0) - \left(\frac{3,366}{4,366} \log_2 \frac{3,366}{4,366}\right) - \left(\frac{1}{4,366} \log_2 \frac{1}{4,366}\right) = 0,77634$$

$$H_f(smt_1, sm) = -(0) - \left(\frac{5,867}{7,867} \log_2 \frac{5,867}{7,867}\right) - \left(\frac{2}{7,867} \log_2 \frac{2}{7,867}\right) = 0,81791$$

$$H_f(smt_1, c) = -(0) - \left(\frac{1,5}{1,5} \log_2 \frac{1,5}{1,5}\right) - (0) = 0$$

$$G_f(s, smt_1) = 0,72193 - \left(\frac{4,366}{15} \times 0,77634\right) - \left(\frac{7,867}{15} \times 0,81791\right) = 0,06699$$

Atribut Jenis Kelamin

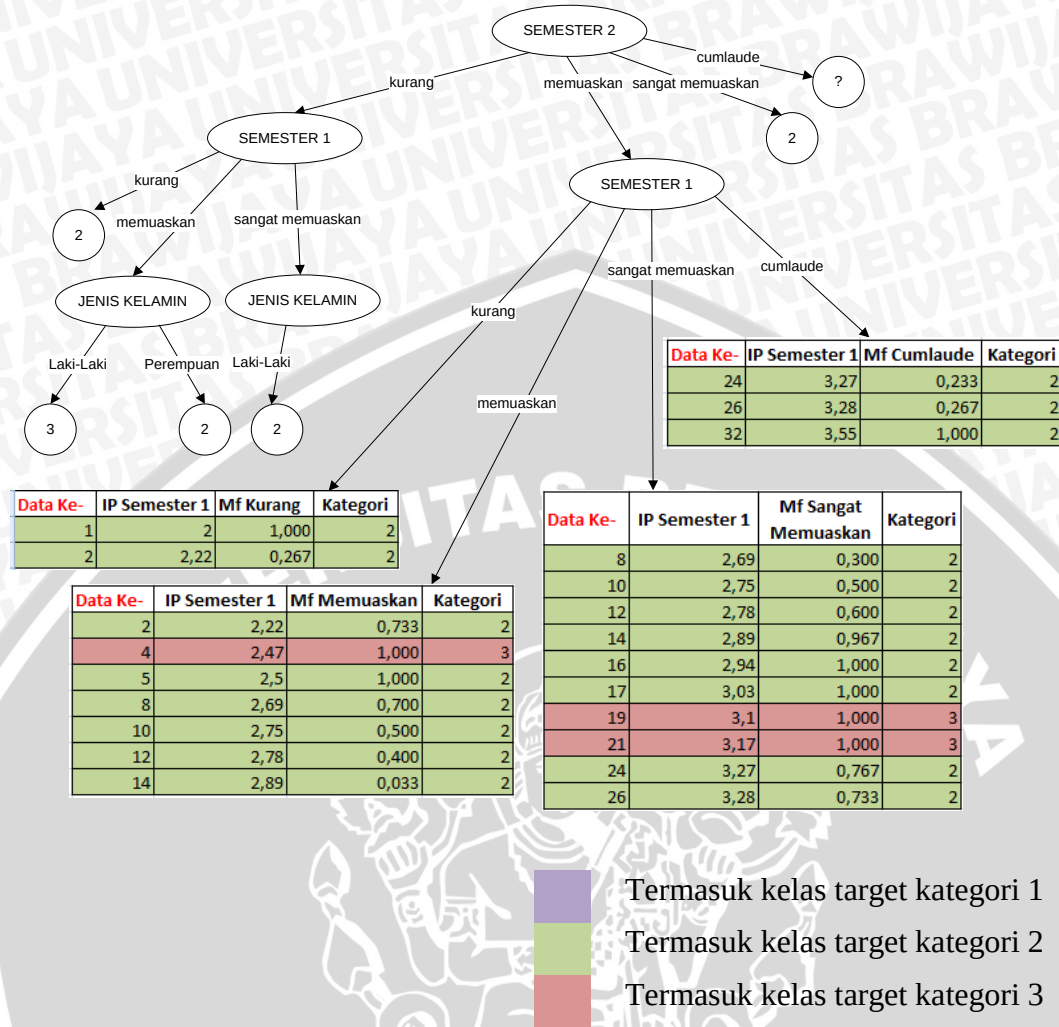
$$H_f(jk, l) = -(0) - \left(\frac{11}{14} \log_2 \frac{11}{14}\right) - \left(\frac{3}{14} \log_2 \frac{3}{14}\right) = 0,74960$$

$$H_f(jk, p) = -(0) - \left(\frac{1}{1} \log_2 \frac{1}{1}\right) - (0) = 0$$

$$G_f(s, jk) = 0,72193 - \left(\frac{14}{15} \times 0,74960\right) = 0,02230$$

c. **Pembentukan *tree***

Dari 2 atribut, *information gain* terbesar dimiliki oleh atribut IP Semester 1, sehingga atribut ini yang akan menjadi *node* root selanjutnya.



Gambar 3. 19 Tree Tahap 5

d. Perhitungan proporsi kelas

Kemudian akan dihitung proporsi kelas dari tiap cabang yang terbentuk.

Cabang Semester 1 → Kurang

$$c_1 = 0\%$$

$$c_2 = \frac{2}{2} \times 100\% = 100\%$$

$$c_3 = 0\%$$

Cabang Semester 1 → Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{3,366}{4,366} \times 100\% = 77,1\%$$

$$c_3 = \frac{1}{4,366} \times 100\% = 22,9\%$$

Cabang Semester 1 → Sangat Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{5,867}{7,867} \times 100\% = 74,5\%$$

$$c_3 = \frac{2}{7,867} \times 100\% = 25,4\%$$

Cabang Semester 1 → Cumlaude

$$c_1 = 0\%$$

$$c_2 = \frac{3}{3} \times 100\% = 100\%$$

$$c_3 = 0\%$$

Fuzziness Control Threshold (FCT) / ϕ_r yang digunakan adalah 80% sehingga cabang “semester 1 → kurang” dan “semester 1 → cumlaude” akan berhenti diekspansi karena proporsi kelasnya lebih besar dari ϕ_r . Sedangkan cabang yang lain akan terus diekspansi karena proporsi kelasnya lebih kecil dari ϕ_r .

7. Level 2 Iterasi ke-3

Setiap cabang yang proporsi kelasnya lebih kecil dari ϕ_r akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 8 Level 2 Iterasi ke-3

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Memuaskan	Kategori
2	L	2,22	0,733	2
4	L	2,47	1,000	3
5	L	2,5	1,000	2
8	P	2,69	0,700	2
10	L	2,75	0,500	2
12	L	2,78	0,400	2
14	L	2,89	0,033	2

- Termasuk kelas target kategori 1
- Termasuk kelas target kategori 2
- Termasuk kelas target kategori 3

a. Perhitungan proporsi kelas

Cabang Jenis Kelamin → Laki-Laki

$$c_1 = 0\%$$

$$c_2 = \frac{5}{6} \times 100\% = 83,33\%$$

$$c_3 = \frac{1}{6} \times 100\% = 16,67\%$$

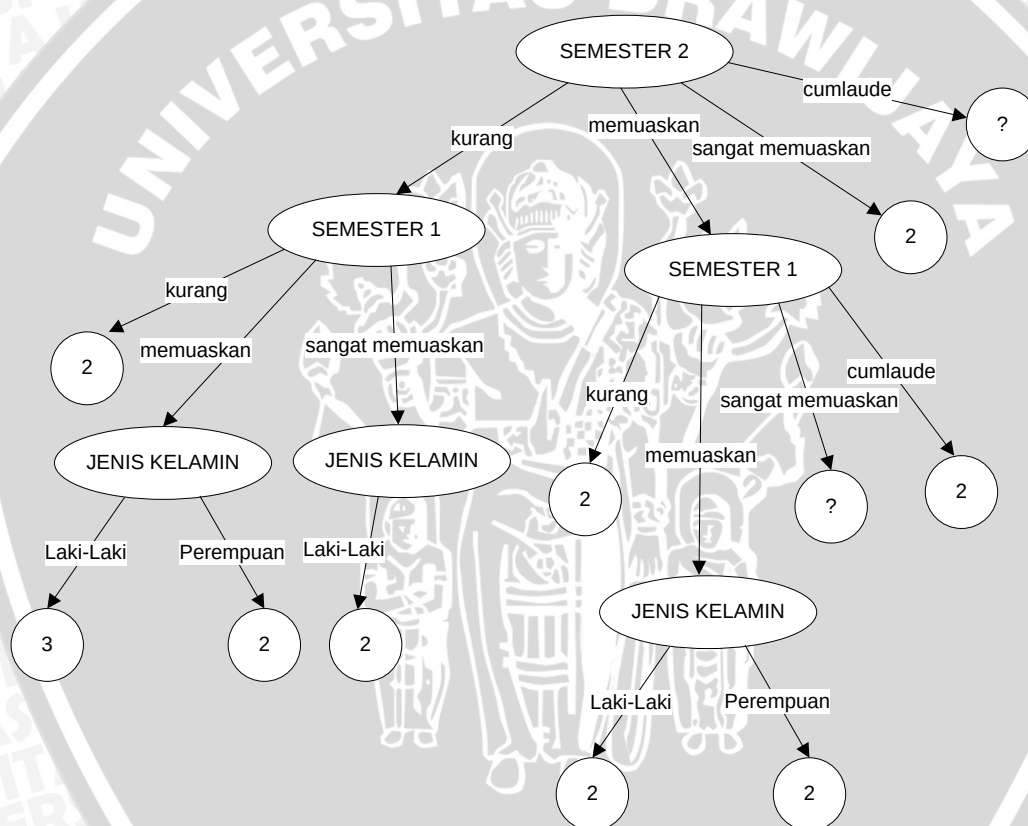
Cabang Jenis Kelamin → Perempuan

$$c_1 = 0\%$$

$$c_2 = \frac{1}{1} \times 100\% = 100\%$$

$$c_3 = 0\%$$

b. Pembentukan tree



Gambar 3. 20 Tree Tahap 6

8. Level 2 Iterasi ke-4

Setiap cabang yang proporsi kelasnya lebih kecil dari $\emptyset_r$ akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 9 Level 2 Iterasi ke-4

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Sangat Memuaskan	Kategori
8	P	2,69	0,300	2
10	L	2,75	0,500	2
12	L	2,78	0,600	2
14	L	2,89	0,967	2
16	L	2,94	1,000	2
17	L	3,03	1,000	2
19	L	3,1	1,000	3
21	L	3,17	1,000	3
24	L	3,27	0,767	2
26	L	3,28	0,733	2



Termasuk kelas target kategori 1

Termasuk kelas target kategori 2

Termasuk kelas target kategori 3

a. Perhitungan proporsi kelas

Cabang Jenis Kelamin → Laki-Laki

$$c_1 = 0\%$$

$$c_2 = \frac{7}{9} \times 100\% = 77,77\%$$

$$c_3 = \frac{2}{9} \times 100\% = 22,23\%$$

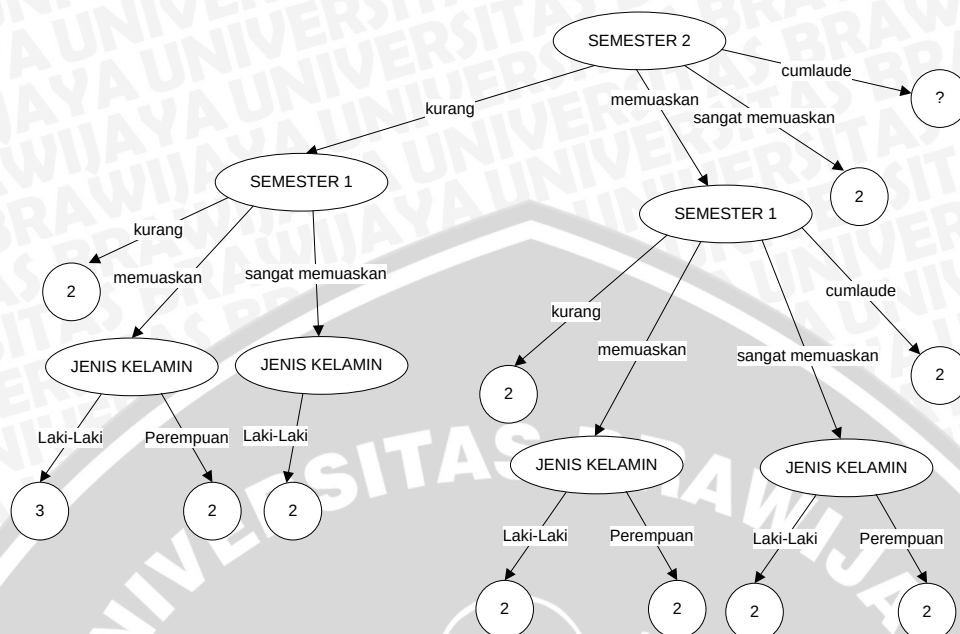
Cabang Jenis Kelamin → Perempuan

$$c_1 = 0\%$$

$$c_2 = \frac{1}{1} \times 100\% = 100\%$$

$$c_3 = 0\%$$

b. Pembentukan tree



Gambar 3. 21 Tree Tahap 7

9. Level 1 Iterasi ke-3

Tabel 3. 10 Level 1 Iterasi ke-3

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Kurang	Mf Memuaskan	Mf Sangat Memuaskan	Mf Cumaude	IP Semester 2	Mf Cumaude	Kategori
7	L	2,61	0,000	0,967	0,033	0,000	3,31	0,367	2
13	P	2,86	0,000	0,133	0,867	0,000	3,24	0,133	2
15	P	2,92	0,000	0,000	1,000	0,000	3,6	1,000	2
18	P	3,03	0,000	0,000	1,000	0,000	3,38	0,600	2
20	P	3,14	0,000	0,000	1,000	0,000	3,21	0,033	2
22	P	3,19	0,000	0,000	1,000	0,000	3,41	0,700	1
23	L	3,25	0,000	0,000	0,833	0,167	3,48	0,933	2
25	P	3,28	0,000	0,000	0,733	0,267	3,5	1,000	2
27	L	3,31	0,000	0,000	0,633	0,367	3,52	1,000	1
28	L	3,39	0,000	0,000	0,367	0,633	3,26	0,200	1
29	L	3,47	0,000	0,000	0,100	0,900	3,57	1,000	2
30	P	3,47	0,000	0,000	0,100	0,900	3,52	1,000	1
31	L	3,53	0,000	0,000	0,000	1,000	3,24	0,133	1
33	P	3,56	0,000	0,000	0,000	1,000	3,33	0,433	2
35	P	3,67	0,000	0,000	0,000	1,000	3,41	0,700	1

Termasuk kelas target kategori 1
 Termasuk kelas target kategori 2
 Termasuk kelas target kategori 3

a. **Perhitungan nilai *entropy* seluruh data**

$$H_f(s) = -\left(\frac{6}{15} \log_2 \frac{6}{15}\right) - \left(\frac{9}{15} \log_2 \frac{9}{15}\right) - (0) = 0,97095$$

b. **Perhitungan nilai *entropy* dan *information gain* tiap atribut**

Selanjutnya akan dihitung nilai *entropy* dari setiap atribut values dan nilai *information gain* dari tiap atribut

Atribut Semester 1

$$H_f(smt_1, k) = -(0) - (0) - (0) \\ = 0$$

$$H_f(smt_1, m) = -(0) - \left(\frac{1,1}{1,1} \log_2 \frac{1,1}{1,1}\right) - (0) \\ = 0$$

$$H_f(smt_1, sm) = -\left(\frac{2,1}{7,666} \log_2 \frac{2,1}{7,666}\right) - \left(\frac{5,566}{7,666} \log_2 \frac{5,566}{7,666}\right) - (0) \\ = 0,84706$$

$$H_f(smt_1, c) = -\left(\frac{3,9}{6,234} \log_2 \frac{3,9}{6,234}\right) - \left(\frac{2,334}{6,234} \log_2 \frac{2,334}{6,234}\right) - (0) \\ = 0,95399$$

$$G_f(s, smt_1) = 0,97095 - \left(\frac{7,666}{15} \times 0,84706\right) - \left(\frac{6,234}{15} \times 0,95399\right) \\ = 0,14156$$

Atribut Jenis Kelamin

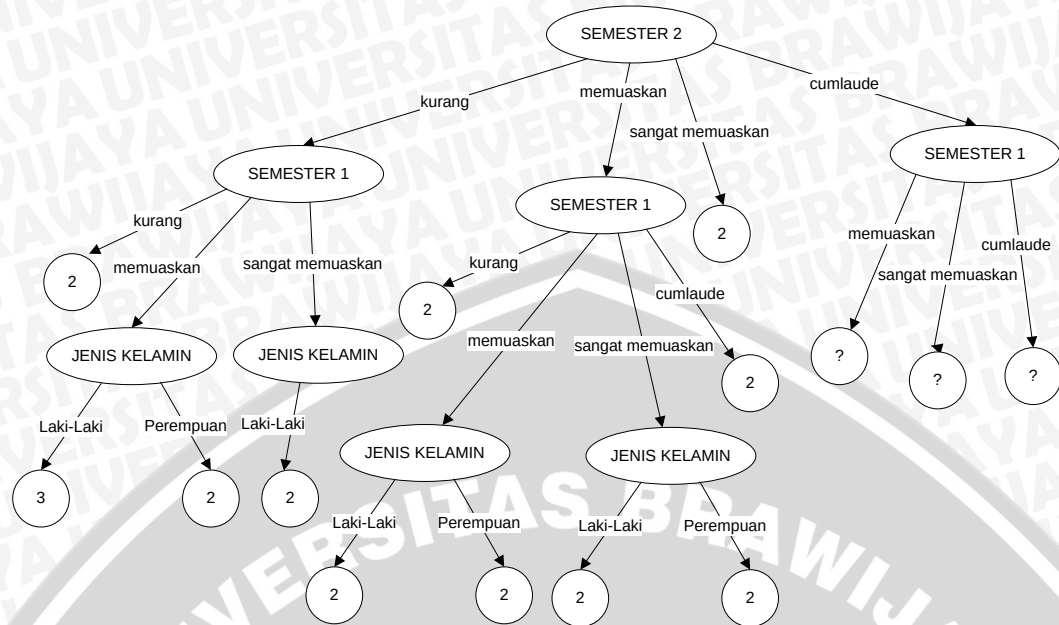
$$H_f(jk, l) = -\left(\frac{3}{6} \log_2 \frac{3}{6}\right) - \left(\frac{3}{6} \log_2 \frac{3}{6}\right) - (0) = 1$$

$$H_f(jk, p) = -\left(\frac{3}{9} \log_2 \frac{3}{9}\right) - \left(\frac{6}{9} \log_2 \frac{6}{9}\right) - (0) = 0,91830$$

$$G_f(s, jk) = 0,97095 - \left(\frac{6}{15} \times 1\right) - \left(\frac{9}{15} \times 0,91830\right) \\ = 0,01997$$

c. **Pembentukan *tree***

Dari 2 atribut, *information gain* terbesar dimiliki oleh atribut IP Semester 1, sehingga atribut ini yang akan menjadi *node root* selanjutnya.



Gambar 3. 22 Tree Tahap 8

d. Perhitungan proporsi kelas

Kemudian akan dihitung proporsi kelas dari tiap cabang yang terbentuk.

Cabang Semester 1 → Memuaskan

$$c_1 = 0\%$$

$$c_2 = \frac{1,1}{1,1} \times 100\% = 100\%$$

$$c_3 = 0$$

Cabang Semester 1 → Sangat Memuaskan

$$c_1 = \frac{2,1}{7,666} \times 100\% = 27,39\%$$

$$c_2 = \frac{5,566}{7,666} \times 100\% = 72,61\%$$

$$c_3 = 0\%$$

Cabang Semester 1 → Cumlaude

$$c_1 = \frac{3,9}{6,234} \times 100\% = 62,56\%$$

$$c_2 = \frac{2,334}{6,234} \times 100\% = 37,44\%$$

$$c_3 = 0\%$$

Fuzziness Control Threshold (FCT) / ϕ_r yang digunakan adalah 80% sehingga cabang “semester 1 →memuaskan” akan berhenti diekspansi karena proporsi kelasnya lebih besar dari ϕ_r . Sedangkan cabang yang lain akan terus diekspansi karena proporsi kelasnya lebih kecil dari ϕ_r .

10. Level 2 Iterasi ke-5

Setiap cabang yang proporsi kelasnya lebih kecil dari ϕ_r akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 11 Level 2 Iterasi ke-5

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Sangat Memuaskan	Kategori
7	L	2,61	0,033	2
13	P	2,86	0,867	2
15	P	2,92	1,000	2
18	P	3,03	1,000	2
20	P	3,14	1,000	2
22	P	3,19	1,000	1
23	L	3,25	0,833	2
25	P	3,28	0,733	2
27	L	3,31	0,633	1
28	L	3,39	0,367	1
29	L	3,47	0,100	2
30	P	3,47	0,100	1


 Termasuk kelas target kategori 1
 Termasuk kelas target kategori 2
 Termasuk kelas target kategori 3

a. **Perhitungan proporsi kelas**

Cabang Jenis Kelamin → Laki-Laki

$$c_1 = \frac{2}{5} \times 100\% = 40\%$$

$$c_2 = \frac{3}{5} \times 100\% = 60\%$$

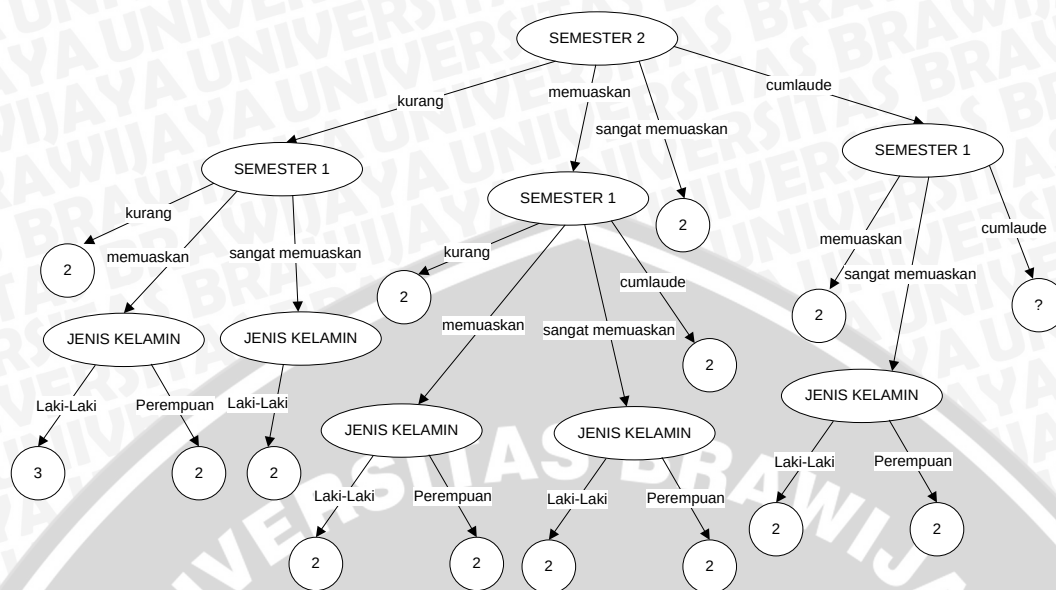
$$c_3 = 0\%$$

Cabang Jenis Kelamin → Perempuan

$$c_1 = \frac{2}{7} \times 100\% = 28,57\%$$

$$c_2 = \frac{5}{7} \times 100\% = 71,43\%$$

$$c_3 = 0\%$$

b. Pembentukan *tree*Gambar 3. 23 *Tree* Tahap 911. Level 2 Iterasi ke-6

Setiap cabang yang proporsi kelasnya lebih kecil dari ϕ_r akan diekspansi dengan atribut yang tersisa yaitu atribut Jenis Kelamin.

Tabel 3. 12 Level 2 Iterasi ke-6

Data Ke-	Jenis Kelamin	IP Semester 1	Mf Cumlaude	Kategori
23	L	3,25	0,167	2
25	P	3,28	0,267	2
27	L	3,31	0,367	1
28	L	3,39	0,633	1
29	L	3,47	0,900	2
30	P	3,47	0,900	1
31	L	3,53	1,000	1
33	P	3,56	1,000	2
35	P	3,67	1,000	1

- Termasuk kelas target kategori 1
- Termasuk kelas target kategori 2
- Termasuk kelas target kategori 3

a. **Perhitungan proporsi kelas**

Cabang Jenis Kelamin → Laki-Laki

$$c_1 = \frac{3}{5} \times 100\% = 60\%$$

$$c_2 = \frac{2}{5} \times 100\% = 40\%$$

$$c_3 = 0\%$$

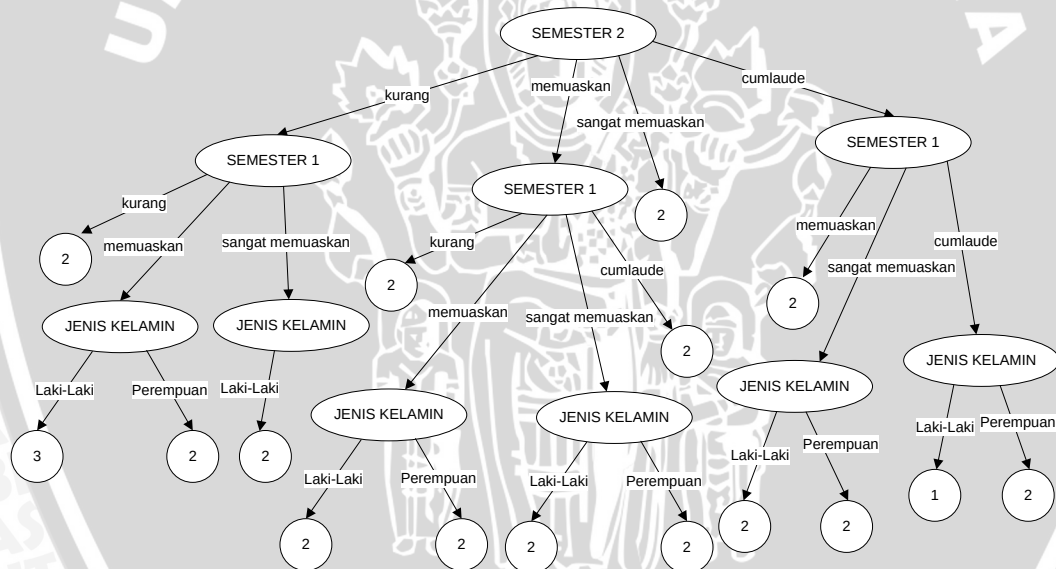
Cabang Jenis Kelamin → Perempuan

$$c_1 = \frac{2}{4} \times 100\% = 50\%$$

$$c_2 = \frac{2}{4} \times 100\% = 50\%$$

$$c_3 = 0\%$$

b. **Pembentukan tree**



Gambar 3. 24 Tree Tahap 10

Berdasarkan perhitungan pembentukan *fuzzy decision tree* pada data sampel dengan algoritma ID3 diperoleh hubungan antara indeks prestasi tahun dengan indeks prestasi kelulusan yaitu berupa 16 aturan. Aturan yang diperoleh sebagai berikut:

1. IF IP semester 2 kurang AND IP semester 1 kurang THEN IP kelulusan antara 2,76 - 3,50
2. IF IP semester 2 kurang AND IP semester 1 memuaskan AND jenis kelamin laki-laki THEN IP kelulusan antara 2,00 – 2,75
3. IF IP semester 2 kurang AND IP semester 1 memuaskan AND jenis kelamin perempuan THEN IP kelulusan antara 2,76 - 3,50
4. IF IP semester 2 kurang AND IP semester 1 sangat memuaskan AND jenis kelamin laki-laki THEN IP kelulusan antara 2,76 – 3,50
5. IF IP semester 2 memuaskan AND IP semester 1 kurang THEN IP kelulusan antara 2,76 – 3,50
6. IF IP semester 2 memuaskan AND IP semester 1 memuaskan AND jenis kelamin laki-laki THEN IP kelulusan antara 2,76 – 3,50
7. IF IP semester 2 memuaskan AND IP semester 1 memuaskan AND jenis kelamin perempuan THEN IP kelulusan antara 2,76 – 3,50
8. IF IP semester 2 memuaskan AND IP semester 1 sangat memuaskan AND jenis kelamin laki-laki THEN IP kelulusan antara 2,76 – 3,50
9. IF IP semester 2 memuaskan AND IP semester 1 sangat memuaskan AND jenis kelamin perempuan THEN IP kelulusan antara 2,76 – 3,50
10. IF IP semester 2 memuaskan AND IP semester 1 cumlaude THEN IP kelulusan antara 2,76 – 3,50
11. IF IP semester 2 sangat memuaskan THEN IP kelulusan antara 2,76 – 3,50
12. IF IP semester 2 cumlaude AND IP semester 1 memuaskan THEN IP kelulusan antara 2,76 – 3,50
13. IF IP semester 2 cumlaude AND IP semester 1 sangat memuaskan AND jenis kelamin laki-laki THEN IP kelulusan antara 2,76 – 3,50
14. IF IP semester 2 cumlaude AND IP semester 1 sangat memuaskan AND jenis kelamin perempuan THEN IP kelulusan antara 2,76 – 3,50

15. IF IP semester 2 cumlaude AND IP semester 1 cumlaude AND jenis kelamin laki-laki THEN IP kelulusan antara 3,51 – 4,00

16. IF IP semester 2 cumlaude AND IP semester 1 cumlaude AND jenis kelamin perempuan THEN IP kelulusan antara 2,76 – 3,50

3.5.2 Inferensi Mamdani

Data yang digunakan dalam pengujian adalah data uji. Berikut contoh perhitungan dalam proses pengujian.

Input data uji:

Jenis Kelamin = Laki-Laki

IP Semester 1 = 2,22

IP Semester 2 = 2,78

1. Fuzzifikasi

- Jenis kelamin laki-laki termasuk dalam himpunan *crisp* laki-laki dengan nilai 1.

- IP semester 1 = 2,22 termasuk dalam himpunan *fuzzy* kurang dan memuaskan.

$$\mu_{kurang}(2,22) = 0,267$$

$$\mu_{memuaskan}(2,22) = 0,733$$

- IP semester 2 = 2,78 termasuk dalam himpunan *fuzzy* memuaskan dan sangat memuaskan.

$$\mu_{memuaskan}(2,78) = 0,4$$

$$\mu_{sangatmemuaskan}(2,78) = 0,6$$

2. Rule Fuzzy

Proses fuzzifikasi tersebut memakai 8 aturan yang memenuhi, yaitu:

- IF IP semester 2 = memuaskan(0,4) AND IP semester 1 = kurang(0,267) THEN kelas = 2 (0,267)

- IF IP semester 2 = memuaskan(0,4) AND IP semester 1 = memuaskan(0,733) AND jenis kelamin = laki-laki(1) THEN kelas = 2 (0,4)

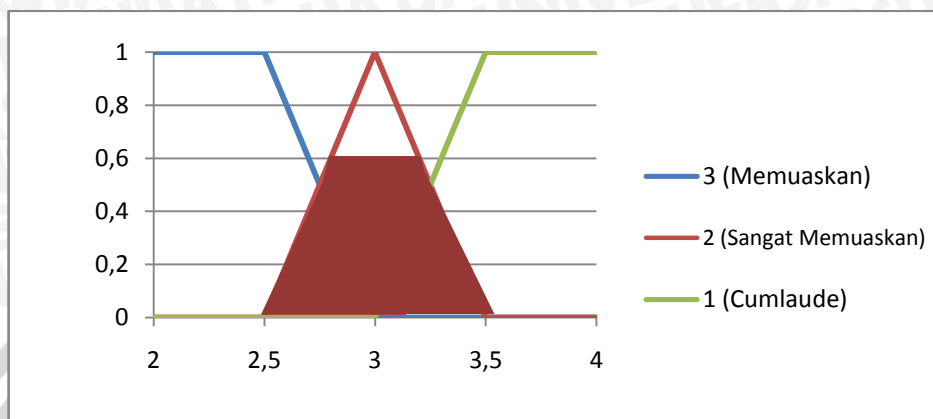
- IF IP semester 2 = sangat memuaskan(0,6) THEN kelas = 2 (0,6)

Menghasilkan:

2(0,267), 2(0,4), 2(0,6)

Keanggotaan diperoleh dengan mengambil nilai terbesar setiap elemen sehingga didapatkan : kelas = 2 (0,6)

Proses komposisinya sebagai berikut.



Gambar 3. 25 Daerah Hasil Komposisi

Berdasarkan daerah hasil komposisi pada Gambar 3.26 kemudian akan dicari titik potongnya yaitu sebagai berikut.

a. Titik A

$$A = (3 - 2,5) * 0,6 + 2,5 = 2,8$$

b. Titik B

$$B = (3,5 - 3) * 0,6 + 3 = 3,3$$

3. Defuzzifikasi

Setelah diketahui titik-titik potongnya maka dihitung hasil defuzzifikasinya dengan menggunakan metode *centroid* sesuai dengan persamaan 2.16 yaitu sebagai berikut.

$$\mu = \frac{(A + B) * 0,6}{2 * 0,6}$$

$$\mu = \frac{(2,8 + 3,3) * 0,6}{2 * 0,6}$$

$$\mu = 3,05$$

Jadi mahasiswa tersebut mendapat IP kelulusan pada kategori 2 dengan nilai defuzzifikasi sebesar 3,05.

3.5.3 Pengujian Tingkat Akurasi

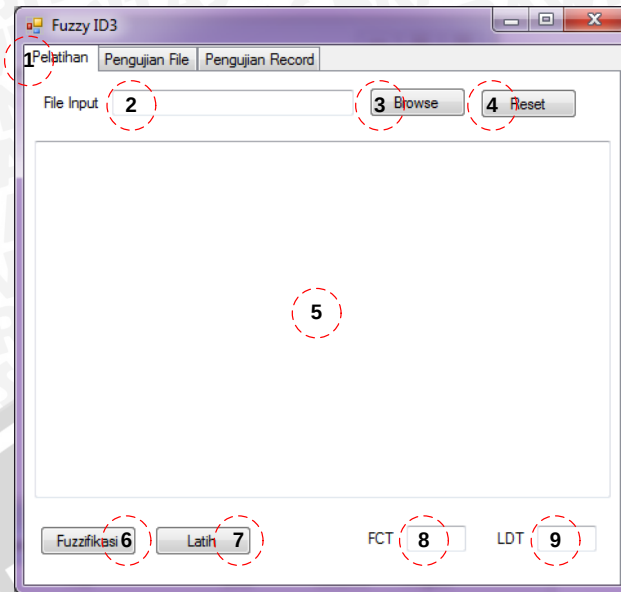
Pengujian tingkat akurasi dilakukan dengan membandingkan kelas kategori output yang dihasilkan sistem dengan kelas kategori data asli. Misalkan menggunakan data jurusan/program studi Ilmu Komputer yang memiliki jumlah data 202 record, dengan proporsi data latih 50%, data uji 50%, nilai FCT 75% dan nilai LDT 10% akan menghasilkan tabel perhitungan akurasi seperti pada lampiran, sehingga jika dihitung menggunakan persamaan 2.13 yaitu:

$$Akurasi(\%) = \frac{93}{101} \times 100 = 92,079$$

3.6 Perancangan Antar Muka

Pada bagian ini akan dijelaskan antarmuka sistem yang akan dibuat. Gambar 3.27 menunjukkan tampilan rancangan antarmuka proses pelatihan atau proses pembentukan *decision tree*. Berikut adalah penjelasan Gambar 3.27 berdasarkan nomor:

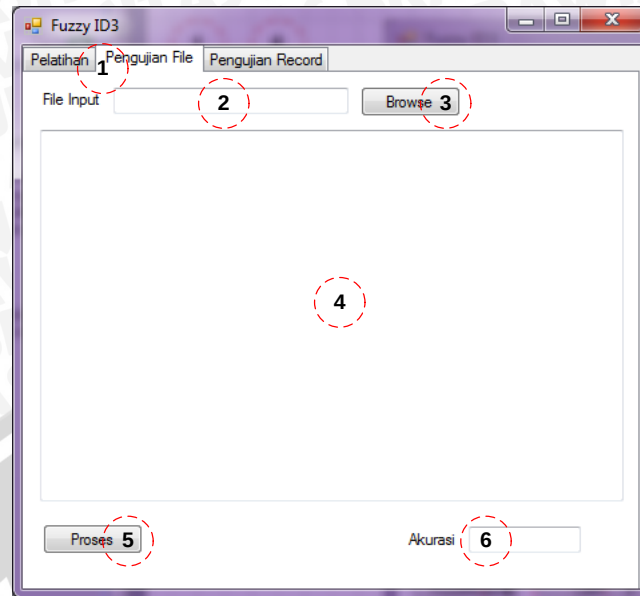
1. Tab pelatihan untuk mengarahkan pada halaman pelatihan *fuzzy ID3*.
2. Bagian ini menampilkan nama file yang akan digunakan sebagai data latih
3. Tombol *browse* untuk menampilkan kotak dialog *open file* yang digunakan untuk memilih file *input* data yang akan diproses
4. Tombol *reset* untuk melakukan reset pada aplikasi
5. Text area untuk menampilkan data latih dan hasil pelatihan
6. Tombol fuzzifikasi untuk melakukan proses fuzzifikasi pada data latih.
7. Tombol latih untuk melakukan proses pelatihan.
8. Input parameter FCT untuk proses pelatihan.
9. Input parameter LDT untuk proses pelatihan.



Gambar 3. 26 Tampilan Rancangan Antarmuka Pelatihan

Antarmuka pengujian ditunjukkan pada Gambar 3.28, penjelasan berdasarkan nomor pada Gambar 3.28 adalah sebagai berikut:

1. Tab pengujian untuk mengarahkan pada halaman pengujian *fuzzy* ID3.
2. Bagian ini menampilkan nama file yang akan digunakan sebagai data uji
3. Tombol browse untuk menampilkan kotak dialog open file yang digunakan untuk memilih file *input* data yang akan diproses
4. Bagian ini menampilkan data uji dalam jumlah banyak.
5. Tombol proses untuk memulai proses pengujian berdasar aturan-aturan yang telah terbentuk pada proses pelatihan.
6. *Textarea* untuk menampilkan tingkat akurasi data uji.



Gambar 3. 27 Tampilan Rancangan Antarmuka Penguian

Antarmuka proses penguian *record* ditunjukkan pada Gambar 3.29, penjelasan berdasarkan nomor pada Gambar adalah sebagai berikut:

1. Tab penguian untuk mengarahkan pada halaman penguian *fuzzy ID3*.
2. *Textarea* untuk memasukkan data nilai IP semester 1 yang akan diproses dalam penguian
3. *Textarea* untuk memasukkan data nilai IP semester 2 yang akan diproses dalam penguian
4. *Radiobutton* laki-laki untuk memasukkan data jenis kelamin
5. *Radiobutton* perempuan untuk memasukkan data jenis kelamin
6. Tombol untuk melakukan proses penguian
7. *Textarea* untuk menampilkan hasil aturan yang terpilih dalam proses penguian
8. *Textarea* untuk menampilkan kelas *output* dari hasil proses penguian
9. *Textarea* untuk menampilkan nilai defuzzifikasi dari hasil metode inferensi
10. Tombol untuk mereset aplikasi

Gambar 3. 28 Tampilan Rancangan Antarmuka Penguian Record

3.7 Perancangan Uji Coba

Pada bagian perancangan uji coba akan dijelaskan mengenai pengujian data. Pengujian dilakukan dengan mengubah-ubah nilai *Fuzziness Control Threshold* (FCT) dan *Leaf Decision Threshold* (LDT) yang dilakukan pada 6 jurusan/program studi yaitu ilmu komputer, statistika, matematika, fisika, kimia dan biologi.

3.7.1 Pengujian Jumlah Aturan yang Terbentuk

Pengujian ini untuk mengetahui jumlah aturan yang terbentuk dari kombinasi nilai *Fuzziness Control Threshold* (FCT) dan *Leaf Decision Threshold* (LDT) yang berbeda-beda pada proses pelatihan atau pembentukan *decision tree*. Hasil jumlah aturan yang terbentuk dalam setiap proses pelatihan dicatat dalam Tabel 3.13 sebagai berikut.

Tabel 3. 13 Rancangan Pengujian Jumlah Aturan

Jumlah Aturan					
FCT	LDT				

Keterangan :

- Kolom jumlah aturan akan menampilkan jumlah aturan yang terbentuk dari proses pelatihan dengan mengubah nilai FCT dan LDT.
- Pada kolom FCT dan LDT, terlihat pengujian akan dilakukan pada perubahan nilai FCT dan LDT yang berbeda-beda.

3.7.2 Pengujian Tingkat Akurasi

Pengujian tingkat akurasi dilakukan untuk membandingkan kelas *output* menggunakan *fuzzy ID3* dengan kelas data aktual. Pengujian ini menggunakan aturan yang telah terbentuk setiap perubahan nilai *Fuzziness Control Threshold* (FCT) dan *Leaf Decision Threshold* (LDT) seperti pada proses pengujian sebelumnya. Hasil perbandingan tersebut kemudian dihitung dengan rumus seperti persamaan 2.13. Hasil dari perhitungan akurasi kemudian disimpan dalam Tabel 3.14 sebagai berikut.

Tabel 3. 14 Rancangan Pengujian Akurasi

FCT	Tingkat Akurasi				
	LDT				

Keterangan :

- Kolom tingkat akurasi akan menampilkan hasil perhitungan akurasi kelas *output* dengan kelas data aktual berdasarkan aturan yang telah terbentuk dari proses pelatihan dengan mengubah nilai FCT dan LDT.
- Pada kolom FCT dan LDT, terlihat pengujian akan dilakukan perubahan nilai FCT dan LDT yang berbeda-beda.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Implementasi perangkat lunak ini berupa aplikasi pemrograman yang menerapkan metode *Fuzzy ID3* dalam pembangkitan aturan untuk mengetahui hubungan antara indeks prestasi tahun pertama mahasiswa dengan indeks prestasi kelulusannya. Atribut yang dipakai dalam data tersebut yaitu indeks prestasi semester 1, indeks prestasi semester 2, jenis kelamin sebagai atribut input dan indeks prestasi kelulusan sebagai atribut target. Adapun lingkungan implementasi akan dijelaskan ke dalam sub bab lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan *Fuzzy ID3* pada skripsi ini adalah :

1. Prosesor Intel (R) Core (TM) i3-2310M CPU @ 2.10 GHz
2. Memori 2 Gb
3. Harddisk 300 Gb
4. Monitor 14'
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan *Fuzzy ID3* pada skripsi ini adalah :

1. Sistem operasi Windows 7 Home Premium
2. NetBeans IDE 7.1.1
3. My SQL

4.2 Implementasi Program

Berdasarkan analisa dan perancangan proses yang telah dipaparkan pada Bab III, maka pada bab ini akan dijelaskan proses-proses implementasinya.

4.2.1 Struktur Data

Aplikasi *fuzzy id3* ini membutuhkan koneksi basis data untuk proses penyimpanan data yang akan digunakan, struktur data koneksi basis data akan diimplementasikan sebagai berikut.

```

1 // Class Connection
2 private Connection connection = null;
3 // Qualified Database Host
4 private String url = "jdbc:mysql://localhost:3306/fuzzy";
5 // Nama Driver database untuk mysql
6 private String driver = "com.mysql.jdbc.Driver";
7 // Database Username
8 private String userName = "root";
9 // Database Password
10 private String password = "";

```

Sourcecode 4. 1 Struktur data koneksi basis data

Struktur data untuk *tree* data, *fct*, *ldt*, data latih, *range* dan atribut direpresentasikan pada *sourcecode 4.2* sebagai berikut.

```

1 public final class ID3FuzzyTree
2 {
3     private ID3TreeModel treeModel;
4     private TreeDataTableModel tableModel;
5     private ArrayList data;
6
7     private Connection connection;
8     private String programStudi;
9     private String limit;
10
11     private double fct = 0.00;
12     private double ldt = 0.00;
13
14     String[] semesterCategories = {"Kurang", "Memuaskan",
15 "Sangat Memuaskan", "Cumlaude"};
16     String[] jenisKelaminCategories = {"Laki-laki",
17 "Perempuan"};

```

Sourcecode 4. 2 Struktur data pelatihan

Sedangkan untuk menyimpan data uji dan nilai akurasi akan diimplementasikan pada struktur data berikut.


```

1 public class TestingTableModel extends
2 ApplicationTableModel{
3     private Connection connection;
4     private Tester tester;
5     private String programStudi;
6     private double percentage;
7     private String limit;

```

Sourcecode 4. 3 Struktur data pengujian

4.2.2 Implementasi Pelatihan

Berdasarkan analisa dan perancangan proses yang telah dipaparkan pada Bab III, ada dua proses utama yg dikerjakan yaitu pelatihan dan pengujian. Dalam implementasi pelatihan ini ada beberapa proses yang berkaitan dengan pembangunan *tree* yang nantinya akan menghasilkan hubungan antara IP tahun pertama dengan IP kelulusan yaitu berupa aturan-aturan.

4.2.2.1 Implementasi Mengambil Data

Proses untuk mengambil data pada aplikasi ini menggunakan *query* kemudian hasilnya di masukkan pada sebuah *resultset*, diimplementasikan pada *sourcecode* sebagai berikut.

```

1 public void build()
2 {
3     ResultSet resultSet;
4
5     try {
6         PreparedStatement preparedStatement =
7         DBHelper.getPreparedStatement(connection, programStudi,
8         limit);
9
10        resultSet = preparedStatement.executeQuery();
11
12        int i =1;
13
14        while (resultSet.next()){
15
16            List row = new ArrayList();
17
18            row.add(i);
19            row.add(resultSet.getString("jenis_kelamin"));
20            row.add(resultSet.getDouble("IP_semester_1"));
21            row.add(resultSet.getDouble("IP_semester_2"));
22            row.add(resultSet.getInt("kategori"));
23

```

```

24      addRow(row.toArray());
25
26      i++;
27  }
28  resultSet.close();
29  preparedStatement.close();
30  } catch (SQLException ex) {
31      Logger.getLogger(ID3Frame.class.getName()).log(Level.SEVERE,
32      null, ex);
33  }
34  }
35
36
37

```

Sourcecode 4. 4 Implementasi mengambil data

4.2.2.2 Implementasi Perhitungan Derajat Keanggotaan

Proses fuzzifikasi terhadap data latih atribut semester terbagi menjadi 4 fungsi yaitu kurang, memuaskan, sangat memuaskan dan cumlaude, akan diimplementasikan pada *sourcecode* sebagai berikut.

```

1      public double kurang()
2      {
3          if(getNilaiIPSemester1() < 2){
4              return 1.00;
5          }
6          else if((getNilaiIPSemester1() >= 2) &&
7          (getNilaiIPSemester1() <= 2.3)){
8              return FormatHelper.decimalFormat((2.3 -
9              getNilaiIPSemester1()) / 0.3);
10         }
11         else{
12             return 0.00;
13         }
14     }
15     public double memuaskan()
16     {
17         if((getNilaiIPSemester1() < 2) && (getNilaiIPSemester1() >
18         2.9)){
19             return 0.00;
20         }
21         else if((getNilaiIPSemester1() >= 2) &&
22         (getNilaiIPSemester1() < 2.3)){
23             return FormatHelper.decimalFormat((getNilaiIPSemester1() -
24             2) / 0.3);
25         }
26         else if((getNilaiIPSemester1() >= 2.3) &&
27         (getNilaiIPSemester1() < 2.6)){
28             return 1.00;

```

```
29     }
30     else if((getNilaiIPSemester1() >= 2.6) &&
31 (getNilaiIPSemester1() < 2.9)){
32         return FormatHelper.decimalFormat((2.9 -
33 getNilaiIPSemester1()) / 0.3);
34     }
35     else{
36         return 0.00;
37     }}
38     public double sangatMemuaskan()
39     {
40         if((getNilaiIPSemester1() >= 2.6) && (getNilaiIPSemester1()
41 < 2.9)){
42             return FormatHelper.decimalFormat((getNilaiIPSemester1() -
43 2.6) / 0.3);
44         }
45         else if((getNilaiIPSemester1() >= 2.9) &&
46 (getNilaiIPSemester1() < 3.2)){
47             return 1.00;
48         }
49         else if((getNilaiIPSemester1() >= 3.2) &&
50 (getNilaiIPSemester1() < 3.5)){
51             return FormatHelper.decimalFormat((3.5 -
52 getNilaiIPSemester1()) / 0.3);
53         }
54         else if((getNilaiIPSemester1() < 2.6) &&
55 (getNilaiIPSemester1() > 3.5)){
56             return 0.00;
57         }
58         else{
59             return 0.00;
60         }
61     }
62     public double cumlaude()
63     {
64         if(getNilaiIPSemester1() > 3.5){
65             return 1.00;
66         }
67         else if((getNilaiIPSemester1() >= 3.2) &&
68 (getNilaiIPSemester1() <= 3.5)){
69             return FormatHelper.decimalFormat((getNilaiIPSemester1() -
70 3.2) / 0.3);
71         }
72         else if(getNilaiIPSemester1() < 3.2){
73             return 0.00;
74         }
75         else{
76             return 0.00;
77         }}

```

Sourcecode 4. 5 Implementasi perhitungan derajat keanggotaan

4.2.2.3 Implementasi Perhitungan *Fuzzy Entropy*

Dalam proses pembentukan *tree* ada perhitungan *fuzzy entropy* terhadap data latih, hal ini akan diimplementasikan pada *sourcecode* sebagai berikut.

```

1  private void calculateEntropy()
2  {
3      double total = 0;
4      double kategori1 = 0;
5      double kategori2 = 0;
6      double kategori3 = 0;
7      for(Iterator iter = getData().iterator(); iter.hasNext();)
8      {
9          Vector dataRow = (Vector) iter.next();
10
11         int kategori = (int) dataRow.get(13);
12         if(kategori == 1){
13             kategori1++;
14         }else if(kategori == 2){
15             kategori2++;
16         }else if(kategori == 3){
17             kategori3++;
18         }
19
20         total++;
21     }
22     setEntropy(-(MathHelper.subtract(kategori1, total) *
23 MathHelper.log2(kategori1/total)) -
24 (MathHelper.subtract(kategori2, total) *
25 MathHelper.log2(kategori2/total)) -
26 (MathHelper.subtract(kategori3, total) *
27 MathHelper.log2(kategori3/total)));
28 }

```

Sourcecode 4. 6 Implementasi perhitungan *fuzzy entropy*

4.2.2.4 Implementasi Perhitungan *Information Gain*

Setelah menghitung *entropy* kemudian menghitung *information gain* untuk setiap atribut. Proses perhitungan *information gain* terhadap data latih untuk atribut semester akan diimplementasikan *sourcecode* 4.7 pada lampiran.

Sedangkan perhitungan *information gain* untuk atribut jenis kelamin akan diimplementasikan pada *sourcecode* sebagai berikut.

```

1  public double getJenisKelaminGain()
2  {
3      int total = 0;
4
5      double lakiLakiKategori1 = 0.00;
6      double lakiLakiKategori2 = 0.00;

```

```
7      double lakiLakiKategori3 = 0.00;
8
9      double perempuanKategori1 = 0.00;
10     double perempuanKategori2 = 0.00;
11     double perempuanKategori3 = 0.00;
12
13     for(Iterator iter = getData().iterator(); iter.hasNext();)
14     {
15
16         Vector dataRow = (Vector) iter.next();
17
18         int kategori = (int) dataRow.get(13);
19         String type = (String) dataRow.get(12);
20
21         if(kategori == 1){
22             if(type.equalsIgnoreCase("L")){
23                 lakiLakiKategori1++;
24             }else{
25                 perempuanKategori1++;
26             }
27         }else if(kategori == 2){
28             if(type.equalsIgnoreCase("L")){
29                 lakiLakiKategori2++;
30             }else{
31                 perempuanKategori2++;
32             }
33         }else if(kategori == 3){
34             if(type.equalsIgnoreCase("L")){
35                 lakiLakiKategori3++;
36             }else{
37                 perempuanKategori3++;
38             }
39         }
40         total++;
41     }
42
43     double lakiLakiTotal = lakiLakiKategori1 +
44     lakiLakiKategori2 + lakiLakiKategori3;
45     double perempuanTotal = perempuanKategori1 +
46     perempuanKategori2 + perempuanKategori3;
47
48     double lakiLaki = -(MathHelper.substract(lakiLakiKategori1,
49     lakiLakiTotal) *
50     MathHelper.log2(lakiLakiKategori1/lakiLakiTotal)) -
51     (MathHelper.substract(lakiLakiKategori2, lakiLakiTotal) *
52     MathHelper.log2(lakiLakiKategori2/lakiLakiTotal)) -
53     (MathHelper.substract(lakiLakiKategori3, lakiLakiTotal) *
54     MathHelper.log2(lakiLakiKategori3/lakiLakiTotal));
55     double perempuan = -
56     (MathHelper.substract(perempuanKategori1, perempuanTotal) *
57     MathHelper.log2(perempuanKategori1/perempuanTotal)) -
```

```

58 (MathHelper.subtract(perempuanKategori2, perempuanTotal) *
59 MathHelper.log2(perempuanKategori2/perempuanTotal)) -
60 (MathHelper.subtract(perempuanKategori3, perempuanTotal) *
61 MathHelper.log2(perempuanKategori3/perempuanTotal));
62
63 double finalResult = getEntropy() -
64 (MathHelper.subtract(lakiLakiTotal, total) * lakiLaki) -
65 (MathHelper.subtract(perempuanTotal, total) * perempuan);
66
67 return finalResult;
68 }

```

Sourcecode 4. 7 Implementasi perhitungan *information gain* atribut jenis kelamin

4.2.2.5 Implementasi Pembentukan *Tree*

Setelah menghitung *information gain* selanjutnya pembentukan *tree*. *Tree* ini maksimal memiliki 3 level sesuai banyak atribut. Dalam pembentukan *tree* juga ada perhitungan nilai FCT dan LDT sebagai *threshold* . Proses untuk pembentukan *tree* akan diimplementasikan *sourcecode* 4.9 pada lampiran.

4.2.2.6 Implementasi Perhitungan Proporsi

Proses perhitungan proporsi setiap kelas kategori *output* akan diimplementasikan pada *sourcecode* sebagai berikut.

```

1 private HashMap calculateNodePercentage(double total, double
2 category1, double category2, double category3)
3 {
4
5 HashMap result = new HashMap();
6
7 result.put("category1",
8 FormatHelper.decimalFormat(MathHelper.calculatePercent(catego
9 ry1, total)));
10 result.put("category2",
11 FormatHelper.decimalFormat(MathHelper.calculatePercent(catego
12 ry2, total)));
13 result.put("category3",
14 FormatHelper.decimalFormat(MathHelper.calculatePercent(catego
15 ry3, total)));
16
17 ArrayList percentage = new ArrayList();
18
19 percentage.add(MathHelper.calculatePercent(category1,
20 total));
21 percentage.add(MathHelper.calculatePercent(category2,

```



```

22 total));
23 percentage.add(MathHelper.calculatePercent(category3,
24 total));
25
26 double minValue = (double) Collections.min(percentage);
27 double maxValue = (double) Collections.max(percentage);
28
29 result.put("min", minValue);
30 result.put("max", maxValue);
31 result.put("class", findClass(percentage));
32
33 return result;
34 }

```

Sourcecode 4. 8 Implementasi perhitungan proporsi

4.2.2.7 Implementasi Pembentukan Aturan

Proses pembentukan aturan yang dihasilkan oleh *tree* akan dilakukan pengecekan pada setiap level untuk mencari *leaf*, jika bukan maka akan dilanjutkan ke level selanjutnya kemudian dirubah dalam bentuk *string*. Hal ini diimplementasikan *sourcecode* 4.11 pada lampiran.

4.2.3 Implementasi Pengujian

Setelah mendapatkan aturan dari proses pelatihan sebelumnya, dalam implementasi pengujian ini akan mengimplementasikan proses pengujian dengan inferensi Mamdani dalam dua bentuk yaitu berupa individual data dan data tabel.

4.2.3.1 Implementasi Pengujian Record

Proses pengujian per *record* atau individual data dalam aplikasi fuzzy ID3 ini akan diimplementasikan pada *sourcecode* sebagai berikut.

```

1 public HashMap<String, Object>
2 doPersonTest(ArrayList<Object> person)
3 {
4     HashMap<String, Object> result = new HashMap<>();
5
6     IPSemester1 iPSemester1 = new
7     IPSemester1(Double.parseDouble((String)person.get(0)));
8     IPSemester2 iPSemester2 = new
9     IPSemester2(Double.parseDouble((String)person.get(1)));
10     JenisKelamin jenisKelamin = new
11     JenisKelamin((String)person.get(2));
12

```

```
13 InferensiMamdani inferensiMamdani = new
14 InferensiMamdani(iPSemester1, iPSemester2, jenisKelamin);
15
16 inferensiMamdani.setId3TreeModel(getId3TreeModel());
17
18 ArrayList mamdaniResult = inferensiMamdani.process();
19
20 String rule = "";
21
22 ArrayList classOutput = new ArrayList();
23 ArrayList classCategory = new ArrayList();
24
25 for(Iterator iter = mamdaniResult.iterator();
26 iter.hasNext();)
27 {
28     ArrayList row = (ArrayList) iter.next();
29
30     rule += row.get(0) + "\n\n";
31
32     classOutput.add(row.get(1));
33     classCategory.add(row.get(2));
34 }
35
36 int classCategoryInt = 3;
37 double classValue = 3;
38 if(classOutput.size() > 0){
39     classValue = (double) Collections.max(classOutput);
40
41     classCategoryInt = (int)
42     classCategory.get(classOutput.indexOf(classValue));
43 }
44
45 double defuzzifikasi=0;
46 System.out.print(+classValue);
47 if(classCategoryInt==3)
48 {
49     defuzzifikasi = (((getPerpotongan1(classValue) +
50 getPerpotongan2(classValue)) * classValue) / (2 *
51 classValue));
52 }
53
54 else if(classCategoryInt==2)
55 {
56     defuzzifikasi = (((getPerpotongan2(classValue) +
57 getPerpotongan3(classValue)) * classValue) / (2*
58 classValue));
59 }
60
61 else
62 {
63     defuzzifikasi = (((getPerpotongan3(classValue) +
64 getPerpotongan4(classValue)) * classValue) / (2 *
```

```

64 classValue));
65 }
66
67 result.put("ClassOutput", classCategoryInt);
68 result.put("Rule", rule);
69 result.put("ClassValue", classValue);
70 result.put("Defuzzifikasi", defuzzifikasi);
71
72 return result;
73 }
74
75 private double getPerpotongan1(double value)
76 {
77     return (((2.5-2) * value) + 2);
78 }
79 private double getPerpotongan2(double value)
80 {
81     return (((3-2.5) * value) + 2.5);
82 }
83 private double getPerpotongan3(double value)
84 {
85     return (((3.5-3) * value) + 3);
86 }
87 private double getPerpotongan4(double value)
88 {
89     return (4-3.5)* value + 4;
90 }

```

Sourcecode 4. 9 Implementasi pengujian per record

4.2.3.2 Implementasi Pengujian Data Tabel

Proses pengujian selanjutnya adalah pengujian dengan jumlah data yang lebih dari satu kemudian nanti akan dihitung nilai akurasi menggunakan rumus pada persamaan 2.13.

```

1 public void build()
2 {
3     ResultSet resultSet;
4     int all = 0;
5     int same = 0;
6
7     try {
8         PreparedStatement preparedStatement =
9 DBHelper.getPreparedStatement(connection, programStudi,
10 getLimit());
11
12     resultSet = preparedStatement.executeQuery();
13     int i =1;
14
15     while (resultSet.next())

```



```

16 {
17
18     ArrayList data = new ArrayList();
19
20     data.add(String.valueOf(resultSet.getDouble("IP_semester_1"))
21 );
22
23     data.add(String.valueOf(resultSet.getDouble("IP_semester_2"))
24 );
25     data.add(resultSet.getString("jenis_kelamin"));
26
27     HashMap<String, Object> result =
28     getTester().doPersonTest(data);
29
30     List row = new ArrayList();
31
32     row.add(i);
33     row.add(resultSet.getString("jenis_kelamin"));
34     row.add(resultSet.getDouble("IP_semester_1"));
35     row.add(resultSet.getDouble("IP_semester_2"));
36     row.add(resultSet.getInt("kategori"));
37     row.add(String.valueOf(result.get("ClassOutput")));
38
39     row.add(String.valueOf(FormatHelper.decimalFormat((double)res
40 ult.get("Defuzzifikasi"))));
41
42     if(result.get("ClassOutput") ==
43 resultSet.getInt("kategori")){
44
45         row.add("Sama");
46         same++;
47     }else{
48         row.add("Tidak Sama");
49     }
50     addRow(row.toArray());
51     i++;
52     all++;
53 }
54 setPercentage((double)same*100/all);
55
56 resultSet.close();
57 preparedStatement.close();
58 }
59 catch (SQLException ex) {
60     Logger.getLogger(ID3Frame.class.getName()).log(Level.SEVERE,
61 null, ex);
62 }
63 }

```

Sourcecode 4. 10 Implementasi pengujian data tabel

4.3 Implementasi Antarmuka

Implementasi antarmuka sistem ini terdiri atas 3 bagian utama, yaitu:

1. Tab Pelatihan

Pada bagian ini digunakan sebagai antarmuka untuk melakukan pelatihan data sehingga menghasilkan *decision tree* yang akan dikonversi dalam bentuk aturan-aturan.

2. Tab Pengujian

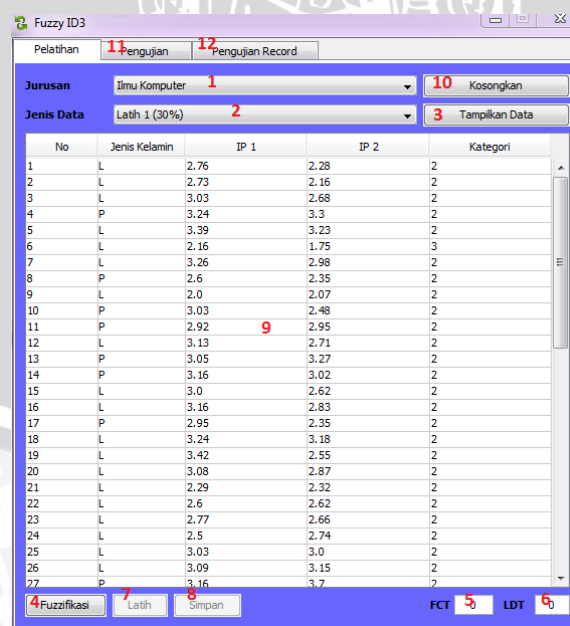
Pada tab pengujian digunakan untuk mencari akurasi dari pengujian aturan menggunakan inferensi mamdani dengan data uji yang disediakan.

3. Tab Pengujian Record

Tab pengujian record digunakan untuk menguji data per record menggunakan inferensi mamdani dengan menginputkan *value* yang dibutuhkan.

4.3.1 Tab Pelatihan

Pada tab pelatihan ini akan dilakukan pelatihan data yang akan membentuk *decision tree* yang nantinya akan dikonversi dalam bentuk aturan-aturan yang akan digunakan pada tab pengujian dan tab pengujian record.

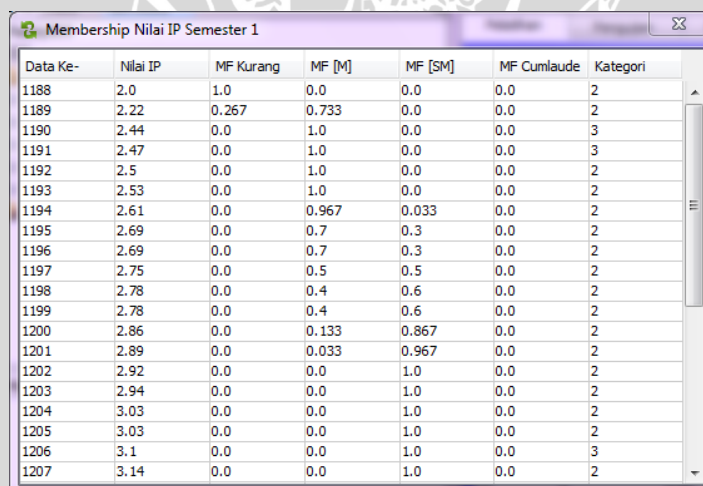


Gambar 4. 1 Tab Pelatihan

Keterangan gambar :

1. *ComboBox* Jurusan untuk memilih jurusan data yang diinginkan
2. *ComboBox* Jenis Data untuk memilih data latih yang ingin ditampilkan
3. Tombol Tampilkan Data untuk menampilkan data yang sesuai dengan pilihan pada *combobox* sebelumnya
4. Tombol Fuzzifikasi untuk melakukan fuzzifikasi pada data
5. *TextField* FCT untuk menginputkan nilai FCT yang diinginkan
6. *TextField* LDT untuk menginputkan nilai FDT yang diinginkan
7. Tombol Latih untuk melakukan pelatihan pada data
8. Tombol Simpan untuk menyimpan *decision tree* yang dihasilkan
9. Tabel Latih untuk menampilkan data latih
10. Tombol Kosongkan untuk mengembalikan tab pelatihan ke dalam posisi *default* dan mengosongkan data pada tabel latih
11. Tab Pengujian untuk menuju ke halaman pengujian
12. Tab Pengujian *Record* untuk menuju ke halaman pengujian *record*.

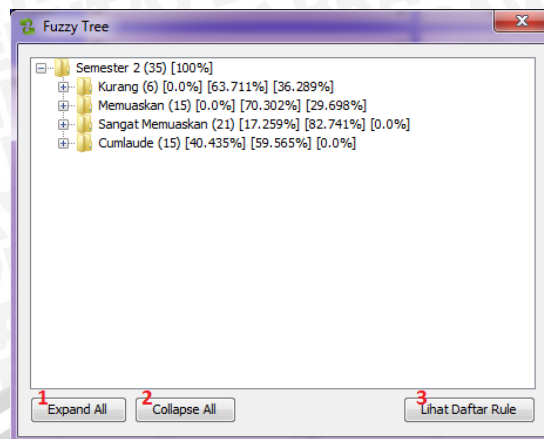
Setelah menekan tombol fuzzifikasi, maka akan muncul halaman fuzzifikasi untuk setiap atribut data yaitu IP Semester 1 dan IP Semester 2.



Data Ke-	Nilai IP	MF Kurang	MF [M]	MF [SM]	MF Cumlaude	Kategori
1188	2.0	1.0	0.0	0.0	0.0	2
1189	2.22	0.267	0.733	0.0	0.0	2
1190	2.44	0.0	1.0	0.0	0.0	3
1191	2.47	0.0	1.0	0.0	0.0	3
1192	2.5	0.0	1.0	0.0	0.0	2
1193	2.53	0.0	1.0	0.0	0.0	2
1194	2.61	0.0	0.967	0.033	0.0	2
1195	2.69	0.0	0.7	0.3	0.0	2
1196	2.69	0.0	0.7	0.3	0.0	2
1197	2.75	0.0	0.5	0.5	0.0	2
1198	2.78	0.0	0.4	0.6	0.0	2
1199	2.78	0.0	0.4	0.6	0.0	2
1200	2.86	0.0	0.133	0.867	0.0	2
1201	2.89	0.0	0.033	0.967	0.0	2
1202	2.92	0.0	0.0	1.0	0.0	2
1203	2.94	0.0	0.0	1.0	0.0	2
1204	3.03	0.0	0.0	1.0	0.0	2
1205	3.03	0.0	0.0	1.0	0.0	2
1206	3.1	0.0	0.0	1.0	0.0	3
1207	3.14	0.0	0.0	1.0	0.0	2

Gambar 4. 2 Halaman *Membership*

Kemudian saat menekan tombol latih, maka akan dilakukan pelatihan data yang akan menghasilkan *decision tree* yang ditampilkan pada halaman *tree*.

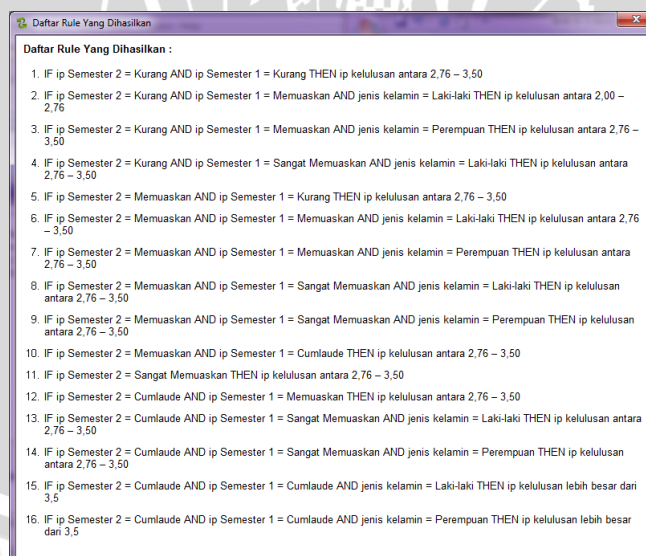


Gambar 4. 3 Halaman *Tree*

Keterangan Gambar:

1. Tombol *Expand All* untuk meng-*expand* semua cabang *tree* yang dihasilkan
2. Tombol *Collapse All* untuk men-*collapse* semua cabang *tree* yang dihasilkan
3. Tombol *Lihat Daftar Rule* untuk menampilkan *rule* yang dihasilkan oleh *tree*.

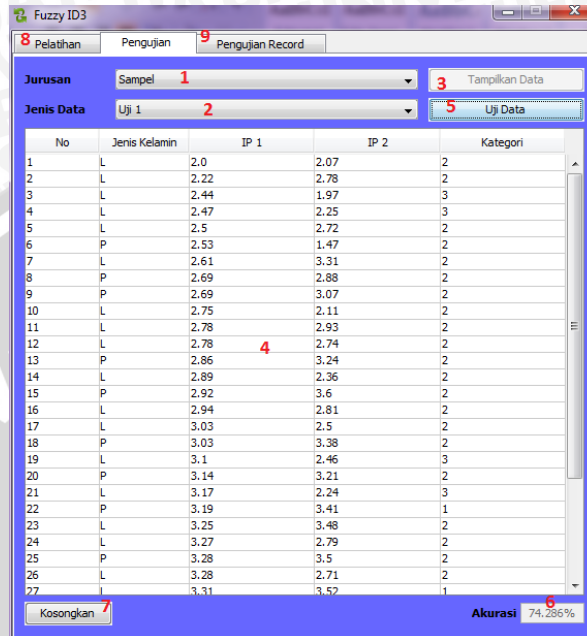
Setelah menekan tombol *Lihat Daftar Rule* maka akan menampilkan halaman yang berisi rule hasil konversi *tree* yang dihasilkan.



Gambar 4. 4 Halaman Aturan yang Terbentuk

4.3.2 Tab Pengujian

Pada tab pengujian ini digunakan untuk melakukan pengujian data dalam jumlah banyak menggunakan inferensi Mamdani yang nantinya akan dihitung nilai akurasi.



Gambar 4. 5 Tab Pengujian

Keterangan gambar :

1. *ComboBox* Jurusan untuk memilih jurusan data yang diinginkan
2. *ComboBox* Jenis Data untuk memilih data yang ingin ditampilkan
3. Tombol Tampilkan Data untuk menampilkan data yang sesuai dengan pilihan pada *comboBox* sebelumnya
4. Tabel Uji untuk menampilkan data yang akan diuji
5. Tombol Uji Data untuk melakukan pengujian pada data
6. *TextField* Akurasi untuk menampilkan nilai akurasi
7. Tombol Kosongkan untuk mengembalikan tab pengujian pada tampilan *default* dan mengosongkan data pada tabel uji
8. Tab Pelatihan untuk menuju ke halaman pelatihan
9. Tab Pengujian Record untuk menuju ke halaman pengujian record.

Setelah memilih data pada *comboBox* dan menekan tombol tampilkan, maka data akan ditampilkan pada tabel uji. Selanjutnya menekan tombol Uji Data dan nilai akurasi akan ditampilkan pada *textField* akurasi.

4.3.3 Tab Pengujian Record

Pada tab pengujian *record* digunakan untuk menguji data per *record* dengan memasukkan nilai IP Semester 1, IP Semester 2 dan jenis kelamin kemudian akan dilakukan inferensi Mamdani.

Gambar 4. 6 Tab Pengujian Record

Keterangan gambar :

1. *TextField* IP Semester 1 untuk menginputkan nilai IP Semester 1
2. *TextField* IP Semester 2 untuk menginputkan nilai IP Semester 2
3. *RadioButton* Laki-laki untuk *input* jenis kelamin laki-laki
4. *RadioButton* Perempuan untuk *input* jenis kelamin perempuan
5. Tombol *Uji Record* untuk melakukan pengujian dengan inferensi Mamdani
6. *TextField* Rule Terpilih untuk menampilkan aturan yang memenuhi hasil inferensi Mamdani
7. *TextField* Kelas Output untuk menampilkan kelas yang terpilih dari hasil inferensi Mamdani
8. *TextField* Nilai Defuzzifikasi untuk menampilkan hasil nilai defuzzifikasi dari inferensi Mamdani
9. Tombol *Reset* untuk mereset tampilan tab pengujian *record* ke tampilan *default*
10. Tab Pelatihan untuk menuju ke halaman pelatihan

11. Tab Pengujian untuk menuju ke halaman pengujian.

Setelah mengisi data inputan pada *textField* IP Semester 1, IP Semester 2, dan jenis kelamin kemudian menekan tombol uji *record*. Akan dilakukan inferensi Mamdani yang hasilnya akan ditampilkan pada *textField* rule terpilih, kelas *output* dan nilai defuzzifikasi.

4.4 Sistematika Pengujian

Sesuai dengan penjelasan pada bab 3 tentang rancangan pengujian, maka pada sistematika pengujian ini akan dilakukan sebanyak 2 macam. Sistematika pengujian tersebut adalah sebagai berikut :

1. Uji jumlah aturan yang terbentuk dari proses pelatihan
2. Uji tingkat akurasi kelas kategori *output*

. Berikut adalah tabel komposisi data latih dan data uji yang digunakan untuk proses pengujian pada setiap jurusan/program studi.

Tabel 4. 1 Tabel Proporsi Data Latih dan Data Uji

ILMU KOMPUTER				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
202	30	60	70	142
	40	81	60	121
	50	101	50	101
	60	121	40	81
STATISTIKA				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
139	30	41	70	98
	40	56	60	83
	50	69	50	70
	60	83	40	56

MATEMATIKA				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
155	30	46	70	109
	40	62	60	93
	50	77	50	78
	60	93	40	62
FISIKA				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
156	30	47	70	109
	40	62	60	94
	50	78	50	78
	60	94	40	62
KIMIA				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
314	30	94	70	126
	40	126	60	188
	50	157	50	157
	60	188	40	126
BIOLOGI				
Total Data	Data Latih		Data Uji	
	Persen	Jumlah Data	Persen	Jumlah Data
148	30	44	70	104
	40	59	60	89
	50	74	50	74
	60	89	40	59

4.4.1 Sistematika Uji Jumlah Aturan yang Terbentuk

Pengujian yang pertama adalah pengujian jumlah aturan yang terbentuk. Pengujian ini dilakukan pada setiap jurusan/program studi. Adapun jurusan/program studi yang digunakan yaitu Matematika, Statistika, Ilmu

Komputer, Fisika, Kimia dan Biologi. Setiap jurusan/program studi tersebut nanti akan terbagi menjadi 4 data latih dengan jumlah proporsi yang berbeda yaitu 30 %, 40%, 50% dan 60%. Sisanya akan digunakan sebagai data uji. Masing-masing data latih dilakukan pengujian sebanyak 50 kali dengan memasukkan nilai kombinasi FDT dan LDT yang berbeda-beda pada proses pelatihan. Nilai FDT yang diujikan antara 50 % sampai 98 %, sedangkan nilai LDT yang diujikan adalah antara 5% sampai 15 %. Dengan memasukkan kombinasi nilai FDT dan LDT ini akan diperoleh hubungan antara IP tahun pertama dengan IP kelulusan berupa jumlah aturan hasil proses pelatihan

4.4.2 Sistematika Uji Tingkat Akurasi Kelas Kategori Output

Pengujian yang kedua adalah pengujian tingkat akurasi kelas kategori *output*. Pada pengujian ini, kelas *output* yang dihasilkan dibandingkan dengan kelas *output* pada data asli. Pengujian ini dilakukan sebanyak 50 kali dengan kombinasi nilai FCT dan LDT yang berbeda-beda, pada 5 jenis data latih yang berbeda sebagaimana pada proses pengujian jumlah aturan sebelumnya dengan masing-masing menggunakan semua data yang termasuk dalam jurusan/program studi terkait selain data latih.

4.5 Uji Coba Jurusan/Program Studi Ilmu Komputer

Uji coba yang pertama menggunakan data jurusan/program studi Ilmu Komputer. Jumlah keseluruhan data ini yaitu 203 baris. Data ini diperoleh dari bagian akademik Fakultas MIPA yang merupakan data kelulusan dari tahun 2003 sampai tahun 2006.

4.5.1 Pengujian Jumlah Aturan Data Latih 30%

Data latih pertama yang digunakan adalah data dengan jumlah data latih sebanyak 30%. Sesuai dengan sistematika pengujian, pengujian dilakukan sebanyak 50 kali dengan mengubah-ubah kombinasi nilai FCT dan LDT untuk mendapatkan jumlah aturan yang dibangkitkan. Hasil dari pengujian jumlah aturan pada data latih 30% ditunjukkan pada tabel 4.2 sebagai berikut.

Tabel 4. 2 Pengujian Jumlah Aturan dengan Data Latih 30%

FCT (%)	JUMLAH ATURAN				
	LDT (%)				
	5	8	10	13	15
50	4	4	4	4	4
55	4	4	4	4	4
60	6	6	6	6	4
65	6	6	6	6	4
70	6	6	6	6	4
75	6	6	6	6	4
80	6	6	6	6	4
85	6	6	6	6	4
90	9	9	9	9	5
95	13	13	13	13	9
98	13	13	13	13	9

Hasil pengujian jumlah aturan pada tabel 4.2 diperoleh jumlah aturan terkecil yaitu 4, terletak pada semua kombinasi nilai LDT dengan FCT 50% dan 55% serta LDT 15% dengan FCT 60% sampai 80% . Sedangkan jumlah aturan terbanyak yaitu 13, terletak di nilai FCT 95% dan 98% dengan semua kombinasi LDT kecuali nilai LDT 15%.

4.5.2 Pengujian Jumlah Aturan Data Latih 40%

Data latih kedua yang digunakan adalah data dengan jumlah data latih sebanyak 40%. Sesuai dengan sistematika pengujian, pengujian dilakukan sebanyak 50 kali dengan mengubah-ubah kombinasi nilai FCT dan LDT untuk mendapatkan jumlah aturan yang dibangkitkan. Hasil dari pengujian jumlah aturan pada data latih 40% ditunjukkan pada tabel 4.3 sebagai berikut.

Tabel 4. 3 Pengujian Jumlah Aturan dengan Data Latih 40%

FCT (%)	JUMLAH ATURAN				
	LDT (%)				
	5	8	10	13	15
50	4	4	4	4	4
55	6	6	6	6	4
60	6	6	6	6	4
65	6	6	6	6	4
70	6	6	6	6	4
75	9	9	8	8	6
80	9	9	8	8	6
85	14	14	13	13	11
90	14	14	13	13	11
95	20	20	19	18	16
98	20	20	19	18	16

Hasil pengujian jumlah aturan pada tabel 4.3 diperoleh jumlah aturan terkecil yaitu 4, terletak pada nilai FCT 50% dengan semua kombinasi nilai LDT dan FCT 55% sampai 70% dengan LDT 15%. Sedangkan jumlah aturan terbanyak yaitu 20, terletak di kombinasi FCT 95% dan 98% dengan LDT 5% dan 8%.

4.5.3 Pengujian Jumlah Aturan Data Latih 50%

Data latih ketiga yang digunakan adalah data dengan jumlah data latih sebanyak 50%. Sesuai dengan sistematika pengujian, pengujian dilakukan sebanyak 50 kali dengan mengubah-ubah kombinasi nilai FCT dan LDT untuk mendapatkan jumlah aturan yang dibangkitkan. Hasil dari pengujian jumlah aturan pada data latih 50% ditunjukkan pada tabel 4.4 sebagai berikut.

Tabel 4. 4 Pengujian Jumlah Aturan dengan Data Latih 50%

FCT (%)	JUMLAH ATURAN				
	LDT (%)				
	5	8	10	13	15
50	4	4	4	4	4
55	6	6	6	4	4
60	6	6	6	4	4
65	6	6	6	4	4
70	6	6	6	4	4
75	9	9	9	6	6
80	9	9	9	6	6
85	10	10	10	7	7
90	14	14	14	11	11
95	14	14	14	11	11
98	20	20	20	16	16

Hasil pengujian jumlah aturan pada tabel 4.4 diperoleh jumlah aturan terkecil yaitu 4, terletak pada FCT 50% dengan semua kombinasi nilai LDT dan FCT 55% sampai 70% dengan LDT 13% dan 15%. Sedangkan jumlah aturan terbanyak yaitu 20, terletak pada nilai FCT 98% dengan LDT 5%, 8% dan 10%

4.5.4 Pengujian Jumlah Aturan Data Latih 60%

Data latih keempat yang digunakan adalah data dengan jumlah data latih sebanyak 60%. Sesuai dengan sistematika pengujian, pengujian dilakukan sebanyak 50 kali dengan mengubah-ubah kombinasi nilai FCT dan LDT untuk mendapatkan jumlah aturan yang dibangkitkan. Hasil dari pengujian jumlah aturan pada data latih 60% ditunjukkan pada tabel 4.5 sebagai berikut.

Tabel 4. 5 Pengujian Jumlah Aturan dengan Data Latih 60%

FCT (%)	JUMLAH ATURAN				
	LDT (%)				
	5	8	10	13	15
50	4	4	4	4	4
55	4	4	4	4	4
60	6	6	6	4	4
65	6	6	6	4	4
70	6	6	6	4	4
75	9	9	9	7	6
80	9	9	9	7	6
85	10	10	10	8	7
90	14	14	14	12	11
95	14	14	14	12	11
98	20	20	20	17	16

Hasil pengujian jumlah aturan pada tabel 4.5 diperoleh jumlah aturan terkecil yaitu 4, terletak pada nilai FCT 50% dan 55% dengan semua kombinasi LDT dan FCT 60% sampai 70% dengan LDT 13% dan 15%. Sedangkan jumlah aturan terbanyak yaitu 20, terletak pada nilai FCT 98% dengan LDT 5%, 8% dan 10%

4.5.5 Pengujian Tingkat Akurasi Data Latih 30%

Data latih pertama yang digunakan adalah data latih dengan jumlah data latih sebanyak 30%. Data latih yang telah mengalami proses pelatihan akan membangkitkan aturan seperti pada pengujian sebelumnya. Aturan tersebut akan disesuaikan dengan data uji sebagaimana sistematisa pengujian tingkat akurasi yang sebelumnya telah dipaparkan. Hasil pengujian akurasi dengan data latih 30% ditunjukkan pada tabel 4.6.

Tabel 4. 6 Pengujian Tingkat Akurasi dengan Data Latih 30%

FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)				
	5	8	10	13	15
50	85,211	85,211	85,211	85,211	85,211
55	85,211	85,211	85,211	85,211	85,211
60	85,915	85,915	86,62	86,62	85,211
65	85,915	85,915	86,62	86,62	85,211
70	85,915	85,915	86,62	86,62	85,211
75	85,915	85,915	86,62	86,62	85,211
80	85,915	85,915	86,62	86,62	85,211
85	85,915	85,915	86,62	86,62	85,211
90	85,915	85,915	86,62	86,62	85,211
95	85,915	85,915	86,62	86,62	85,211
98	85,915	85,915	86,62	86,62	85,211

Hasil pengujian akurasi pada tabel 4.6 diperoleh akurasi terkecil yaitu 85,211% , terletak pada FCT 50% dan 55% dengan semua kombinasi nilai LDT dan untuk masing-masing nilai FCT dengan LDT 15% . Sedangkan akurasi terbesar yaitu 86,62 % , terletak pada FCT 60% sampai 98% dengan LDT 10% dan 13%.

4.5.6 Pengujian Tingkat Akurasi Data Latih 40%

Data latih kedua yang digunakan adalah data latih dengan jumlah data latih sebanyak 40%. Data latih yang telah mengalami proses pelatihan akan membangkitkan aturan seperti pada pengujian sebelumnya. Aturan tersebut akan disesuaikan dengan data uji sebagaimana sistematisa pengujian tingkat akurasi yang sebelumnya telah dipaparkan. Hasil pengujian akurasi dengan data latih 40% ditunjukkan pada tabel 4.7.

Tabel 4. 7 Pengujian Tingkat Akurasi dengan Data Latih 40%

FCT (%)	TINGKAT AKURASI				
	LDT (%)				
	5	8	10	13	15
50	89,256	89,256	89,256	89,256	89,256
55	90,083	90,083	90,083	90,083	89,256
60	89,256	89,256	89,256	90,083	89,256
65	89,256	89,256	89,256	90,083	89,256
70	89,256	89,256	89,256	90,083	89,256
75	90,909	90,909	88,43	89,256	88,43
80	90,909	90,909	88,43	89,256	88,43
85	90,909	90,909	88,43	89,256	88,43
90	90,909	90,909	88,43	89,256	88,43
95	90,909	90,909	88,43	89,256	88,43
98	90,909	90,909	88,43	89,256	88,43

Hasil pengujian akurasi pada tabel 4.7 diperoleh akurasi terkecil yaitu 88,43% , terletak pada FCT 75% sampai 98% dengan LDT 10% dan 15%. Sedangkan akurasi terbesar yaitu 90,909%, terletak pada FCT 75% sampai 98% dengan LDT 5% dan 8%.

4.5.7 Pengujian Tingkat Akurasi Data Latih 50%

Data latih ketiga yang digunakan adalah data latih dengan jumlah data latih sebanyak 50%. Data latih yang telah mengalami proses pelatihan akan membangkitkan aturan seperti pada pengujian sebelumnya. Aturan tersebut akan disesuaikan dengan data uji sebagaimana sistematisa pengujian tingkat akurasi yang sebelumnya telah dipaparkan. Hasil pengujian akurasi dengan data latih 50% ditunjukkan pada tabel 4.8.

Tabel 4. 8 Pengujian Tingkat Akurasi dengan Data Latih 50%

FCT (%)	TINGKAT AKURASI				
	LDT (%)				
	5	8	10	13	15
50	89,109	89,109	89,109	89,109	89,109
55	90,099	90,099	90,099	89,109	89,109
60	89,109	89,109	90,099	89,109	89,109
65	89,109	89,109	90,099	89,109	89,109
70	89,109	89,109	90,099	89,109	89,109
75	91,089	91,089	92,079	89,109	89,109
80	91,089	91,089	92,079	89,109	89,109
85	91,089	91,089	92,079	89,109	89,109
90	91,089	91,089	92,079	89,109	89,109
95	91,089	91,089	92,079	89,109	89,109
98	91,089	91,089	92,079	89,109	89,109

Hasil pengujian akurasi pada tabel 4.8 diperoleh akurasi terkecil yaitu 89,109% , terletak pada sebagian besar titik FCT 50%, 60%, 65% dan 70% dengan LDT 5%, 8% dan 10%. Untuk LDT 13% dan 15% dengan semua kombinasi FCT juga memperoleh akurasi 89,109%. Sedangkan akurasi terbesar yaitu 92,079%, terletak pada LDT 10% dengan kombinasi FCT 75% sampai 98%.

4.5.8 Pengujian Tingkat Akurasi Data Latih 60%

Data latih keempat yang digunakan adalah data latih dengan jumlah data latih sebanyak 60%. Data latih yang telah mengalami proses pelatihan akan membangkitkan aturan seperti pada pengujian sebelumnya. Aturan tersebut akan disesuaikan dengan data uji sebagaimana sistematisa pengujian tingkat akurasi yang sebelumnya telah dipaparkan. Hasil pengujian akurasi dengan data latih 60% ditunjukkan pada tabel 4.9.

Tabel 4. 9 Pengujian Tingkat Akurasi dengan Data Latih 60%

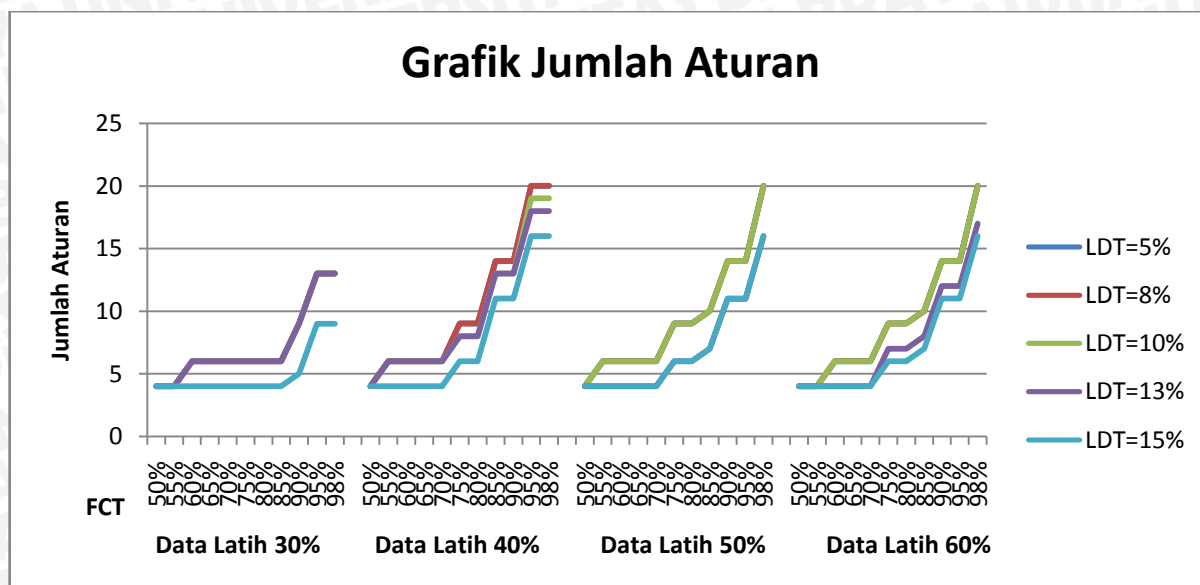
FCT (%)	TINGKAT AKURASI				
	LDT (%)				
	5	8	10	13	15
50	90,244	90,244	90,244	90,244	90,244
55	90,244	90,244	90,244	90,244	90,244
60	90,244	90,244	90,244	90,244	90,244
65	90,244	90,244	90,244	90,244	90,244
70	89,024	89,024	90,244	90,244	90,244
75	89,024	89,024	90,244	90,244	90,244
80	89,024	89,024	90,244	90,244	90,244
85	89,024	89,024	90,244	90,244	90,244
90	89,024	89,024	90,244	90,244	90,244
95	89,024	89,024	90,244	90,244	90,244
98	89,024	89,024	90,244	90,244	90,244

Hasil pengujian akurasi pada tabel 4.9 diperoleh akurasi terkecil yaitu 89,024% , terletak pada nilai LDT 5% dan 8% dengan kombinasi nilai FCT 70% sampai 98%. Sedangkan akurasi terbesar yaitu 90,244%, terletak pada semua kombinasi nilai FCT dan LDT kecuali di beberapa nilai LDT 5% dan 8%.

4.6 Analisa Hasil Jurusan/Program Studi Ilmu Komputer

Setelah dilakukan pengujian jumlah aturan dan tingkat akurasi, maka akan dilakukan 2 bentuk analisa yaitu analisa jumlah aturan yang terbentuk dan analisa tingkat akurasi kelas kategori *output*.

4.6.1 Analisa Hasil Jumlah Aturan yang Terbentuk



Gambar 4. 7 Grafik Jumlah Aturan yang Terbentuk

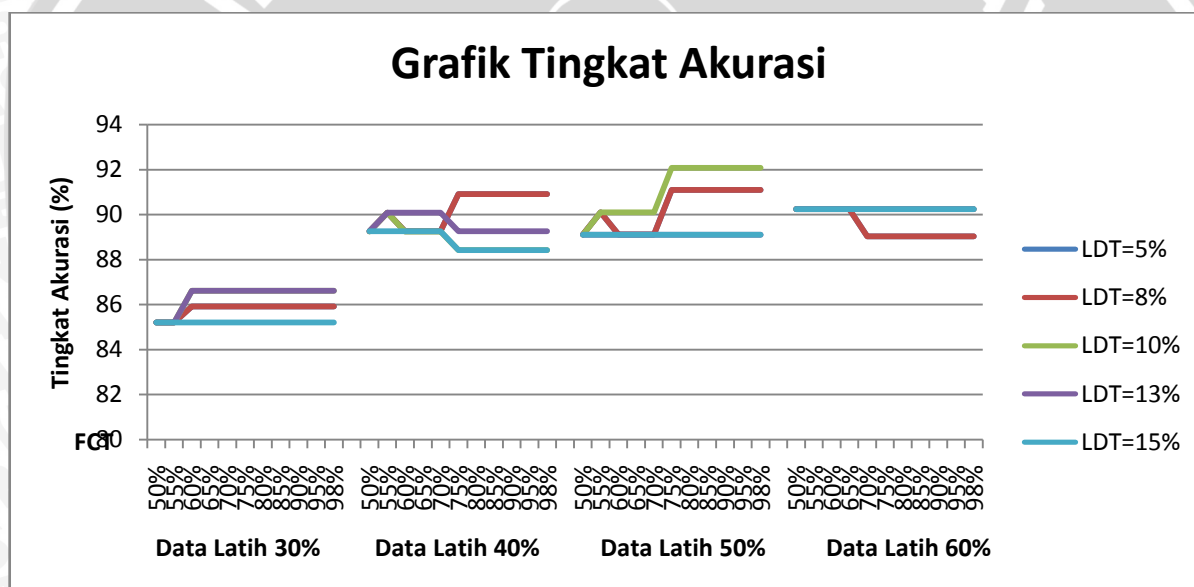
Grafik menunjukkan bahwa semakin tinggi nilai FCT akan menyebabkan jumlah aturan yang dihasilkan juga meningkat. Hal ini disebabkan karena apabila proporsi dari masing-masing kelas dibawah nilai FCT, maka *tree* akan terus diekspansi. Pada grafik, kenaikan jumlah aturan yang cukup signifikan terjadi dari nilai FCT 85% ke 90% dan 95% ke 98%. Hal ini berarti bahwa pada nilai FCT 85%, ada banyak *sub-node* yang memiliki prosentase proporsi kelas kategori yang sama berada di atas nilai 85%, sehingga *tree* berhenti diekspansi. Namun ketika nilai FCT dinaikkan menjadi 95% dan 98%, banyak *sub-node* yang prosentase proporsi kelas kategori yang sama berada di bawah nilai 90% dan 98%, sehingga *tree* terus diekspansi. Untuk kenaikan nilai FCT dari 95% ke 98% pada data latih 30% dan 40%, jumlah aturan yang dihasilkan cenderung sama, hal ini berarti pada nilai FCT 95% banyak *sub-node* yang memiliki nilai prosentasi proporsi kelas kategori yang sama di bawah nilai 95% menyebabkan *tree* harus diekspansi sampai seluruh atribut telah digunakan.

Berdasarkan grafik dapat dilihat bahwa kenaikan nilai LDT menyebabkan penurunan jumlah aturan yang terbentuk. Hal ini disebabkan karena semakin tinggi nilai LDT maka semakin tinggi batasan jumlah data tiap *sub-node*. Ekspansi *tree* akan dihentikan apabila jumlah data pada *sub-node* kurang dari prosentase LDT dikalikan dengan jumlah data pada root node. Pada grafik,

penurunan jumlah aturan signifikan terjadi untuk nilai LDT 15%. Hal ini berarti ketika nilai LDT dinaikkan menjadi 15% banyak *sub-node* yang mengalami *prunning* atau pemotongan sehingga jumlah aturan yang dihasilkan menurun.

Nilai FCT yang terlalu rendah dan atau nilai LDT yang terlalu tinggi akan menghasilkan *tree* dengan ukuran yang kecil sehingga jumlah aturan yang dihasilkan juga sedikit. Hal ini terjadi karena *tree* yang sedang dibangun mengalami *pruning* atau pemotongan. Sebaliknya jika nilai FCT terlalu tinggi atau nilai LDT terlalu rendah dapat menghasilkan *tree* dengan ukuran yang besar dan jumlah aturan yang banyak karena *tree* akan diekspansi sampai *leaf-node* terdalam atau sampai tidak ada atribut lagi.

4.6.2 Analisa Hasil Tingkat Akurasi Kelas Kategori Output



Gambar 4. 8 Grafik Tingkat Akurasi Kelas Kategori Output

Grafik tersebut menunjukkan bahwa tingkat akurasi terbaik yaitu sebesar 92,079% terdapat data latih 50% pada nilai FCT 75% sampai 98% dengan LDT 10% yang memiliki rentang jumlah aturan 9 sampai 20. Pada grafik juga terlihat bahwa rata-rata akurasi mulai stabil pada rentang FCT 75% sampai 98% . Untuk LDT 15%, rata-rata akurasinya mengalami penurunan. Penyebaran data pada kelas kategori target yang digunakan dalam data latih dan data uji juga dapat mempengaruhi tingkat akurasi.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan penelitian maka dapat disimpulkan bahwa :

1. Metode *fuzzy* ID3 dapat diterapkan dalam pembangkitan aturan dengan *Fuzzy Iterative Dichotomiser 3* untuk mengetahui hubungan antara IP tahun pertama dengan IP Kelulusan pada data akademik FMIPA tahun 2003 sampai tahun 2006. Langkah awal dilakukan dengan pembentukan himpunan *fuzzy* pada data latih. Selanjutnya dilakukan pembentukan *tree* dengan algoritma *Iterative Dicotomiser 3* (ID3) untuk menghasilkan aturan-aturan. Aturan yang terbentuk mengalami proses pengujian dengan menggunakan inferensi Mamdani. Hasil dari proses inferensi Mandani inilah yang digunakan untuk menentukan kelas kategori *output*.
2. Berdasarkan pengujian yang dilakukan, jumlah aturan yang dihasilkan dipengaruhi oleh nilai *threshold* proporsi kelas (FCT) dan *threshold* jumlah data dalam suatu node (LDT). Nilai FCT yang terlalu rendah dan atau nilai LDT yang terlalu tinggi akan menghasilkan *tree* dengan ukuran yang kecil sehingga jumlah aturan yang dihasilkan juga sedikit. Sebaliknya jika nilai FCT terlalu tinggi atau nilai LDT terlalu rendah dapat menghasilkan *tree* dengan ukuran yang besar dan jumlah aturan yang banyak. Jumlah aturan terbanyak yang dihasilkan yaitu 21 aturan dan yang terendah yaitu 4 aturan.
3. Dari hasil pengujian menggunakan data akademik FMIPA tahun 2003 sampai tahun 2006 ternyata perubahan nilai FCT dan LDT tidak memberi pengaruh yang signifikan terhadap akurasi dari aturan yang dihasilkan. Penyebaran data pada kelas kategori target yang digunakan dalam data latih dan data uji juga dapat mempengaruhi tingkat akurasi. Tingkat akurasi tertinggi yang diraih yaitu pada jurusan/program studi Fisika sebesar 93,548%.

5.2 Saran

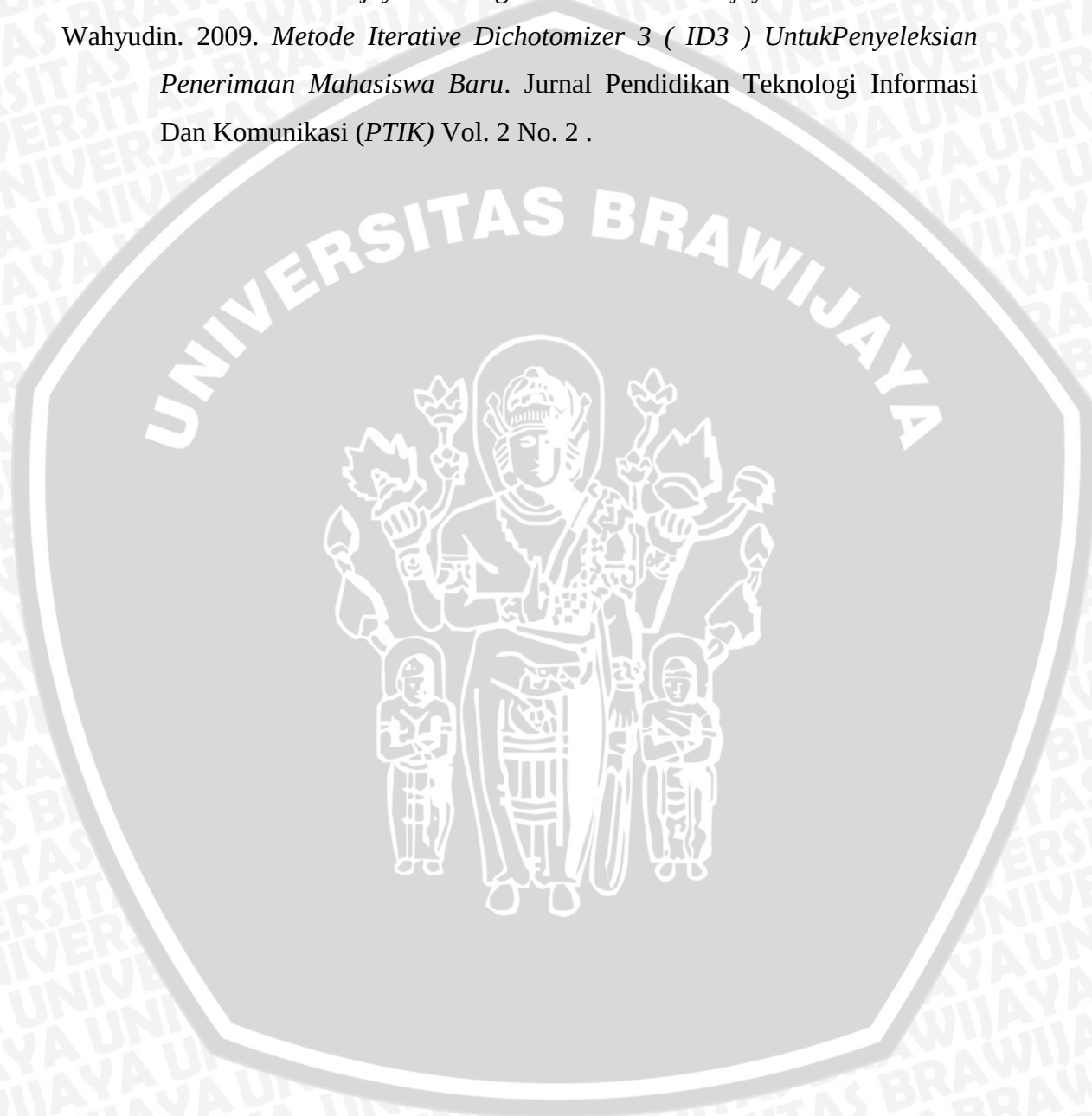
Pada penelitian ini metode pembangkitan aturan dengan *Fuzzy Iterative Dichotomiser 3* menggunakan data akademik FMIPA tahun 2003 sampai 2006 yang memiliki jumlah atribut sebanyak 3 atribut dan atribut indeks prestasi kelulusan sebagai atribut target. Oleh karena itu disarankan untuk penelitian selanjutnya dapat melakukan penambahan atribut terkait masa studi dan menggunakan penyebaran data yang lebih merata sehingga dapat dilihat perbedaan aturan dan tingkat akurasi yang lebih signifikan.



DAFTAR PUSTAKA

- Defiyanti, Sofi. 2010. *Perbandingan Kinerja Algoritma Id3 Dan C4.5 Dalam Spam-Mail*. <http://openstorage.gunadarma.ac.id/~mwiryan/KOMMIT/per-artikel/03-02-004-Perbandingan%5BSofi%5D.pdf>, diakses tanggal 21 Maret 2012.
- Han, Jiawei dan Micheline Khamber. 2001. *Data Mining : Concepts and Technique*. San Francisco, USA : Morgan Kaufmann Publisher.
- Kohavi, R., Quinlan. 1999. *Decision Tree Discovery*. AAAI and The MIT, Pres, 1-16
- Kusrini, M.Kom. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta : CV Andi Offset.
- Kusumadewi, Sri . 2002. *Analisis & Desain Sistem Fuzzy Menggunakan Toolbox MATLAB*. Yogyakarta: Graha Ilmu.
- _____. 2003. *Artificial Intelligence (Teknik & Aplikasinya)*. Jogjakarta: Graha Ilmu.
- Kusumadewi, Sri dan Purnomo, H. 2004. *Aplikasi Logika Fuzzy untuk Mendukung Keputusan*. Yogyakarta : Graha Ilmu.
- _____. 2010. *Aplikasi Logika Fuzzy untuk Mendukung Keputusan: Jilid 2*. Yogyakarta.: Graha Ilmu.
- Larose, Daniel T. 2005. *Discovering Knowledge in Data. An Introduction to Data mining*. John willey & Sons. New Jersey.
- Liang, G. 2005. *A Comparative Study of Three Decision Tree Algorithms: I D3, Fuzzy ID3 and Probalistic Fuzzy ID3*. Informatics & Economics Erasmus University Rotterdam, The Netherlands.
- Mitchell, Tom M. 1997. *Machine Learning*. Singapore : McGraw Hill.
- Nurlaelasari, Fitri. 2010. *Implementasi Algoritma Fuzzy Iterative Dichotomiser 3 (FID3) Pada Fuzzy Decision Tree Untuk Diagnosa Penyakit Jantung*. Universitas Pendidikan Indonesia, Bandung.
- Nugraha, Dany, dkk. 2006. *Diagnosis Gangguan Sistem Urinari pada Anjing dan Kucing Menggunakan VFI 5*. Institut Pertanian Bogor.

- Romansyah, I, Sitanggang S, Nurdiati S. 2009. *Fuzzy Decision Tree dengan Algoritme ID3 pada Data Diabetes*. Internetworking Indonesia Journal. Vol. 1, No. 2: Special Issue on Data Mining.
- Suharto,Bambang, dkk. 2010. *Pedoman Pendidikan Tahun Akademik 2010/2011 Universitas Brawijaya*. Malang : Universitas Brawijaya.
- Wahyudin. 2009. *Metode Iterative Dichotomizer 3 (ID3) UntukPenyeleksian Penerimaan Mahasiswa Baru*. Jurnal Pendidikan Teknologi Informasi Dan Komunikasi (PTIK) Vol. 2 No. 2 .



LAMPIRAN

Tabel Perhitungan Akurasi

Data Ke-	Jenis Kelamin	IP Semester 1	IP Semester 2	Kategori Asli	Kategori Sistem	Akurasi Sistem
1	P	3.0	3.21	2	2	Benar
2	P	3.0	3.67	2	2	Benar
3	L	3.28	3.24	2	2	Benar
4	P	3.03	3.33	2	2	Benar
5	P	3.28	3.5	2	2	Benar
6	P	2.78	2.88	2	2	Benar
7	L	2.92	2.62	2	2	Benar
8	L	2.58	2.22	2	2	Benar
9	L	2.61	2.91	2	2	Benar
10	L	2.86	2.79	2	2	Benar
11	P	2.83	2.91	2	2	Benar
12	L	3.13	2.79	2	2	Benar
13	P	2.56	2.86	2	2	Benar
14	P	3.36	3.48	1	1	Benar
15	P	2.86	2.93	2	2	Benar
16	P	2.69	2.64	2	2	Benar
17	P	2.69	2.94	2	2	Benar
18	P	3.36	3.52	1	1	Benar
19	L	2.44	2.03	2	2	Benar
20	L	2.47	2.25	3	2	Salah
21	P	2.53	2.98	2	2	Benar
22	L	2.69	2.5	2	2	Benar
23	P	2.57	2.81	2	2	Benar
24	L	2.89	2.36	2	2	Benar
25	P	2.61	3.21	2	2	Benar
26	P	3.56	3.33	2	2	Benar
27	L	2.89	3.07	2	2	Benar
28	P	2.81	3.07	2	2	Benar
29	L	1.78	2.3	3	2	Salah
30	P	2.81	3.02	2	2	Benar
31	L	2.67	2.88	2	2	Benar
32	L	2.81	3.17	2	2	Benar
33	L	2.53	2.69	3	2	Salah
34	L	2.64	2.91	2	2	Benar
35	P	2.81	3.14	2	2	Benar
36	P	2.86	3.0	2	2	Benar
37	L	2.97	2.76	2	2	Benar
38	L	2.86	2.83	2	2	Benar
39	L	2.5	2.72	2	2	Benar
40	L	3.22	3.0	2	2	Benar
41	P	3.36	2.81	2	2	Benar
42	L	3.0	2.97	2	2	Benar
43	P	3.44	3.24	2	2	Benar
44	L	2.78	2.74	2	2	Benar
45	P	3.5	3.41	2	1	Salah

46	L	2.81	2.79	2	2	Benar
47	L	3.11	2.95	2	2	Benar
48	P	3.07	3.55	2	2	Benar
49	L	3.31	2.95	2	2	Benar
50	P	2.69	3.02	2	2	Benar
51	L	3.42	3.45	2	2	Benar
52	P	2.37	2.53	2	2	Benar
53	P	3.17	2.88	2	2	Benar
54	L	3.25	3.48	2	2	Benar
55	L	3.28	2.93	2	2	Benar
56	P	3.28	3.0	2	2	Benar
57	P	3.14	3.31	2	2	Benar
58	L	2.78	3.25	2	2	Benar
59	P	2.72	2.6	2	2	Benar
60	L	3.28	2.71	2	2	Benar
61	L	2.56	2.81	2	2	Benar
62	P	2.67	2.79	2	2	Benar
63	L	2.36	2.56	3	2	Salah
64	P	2.33	2.28	2	2	Benar
65	L	2.64	2.33	2	2	Benar
66	L	2.19	2.25	2	2	Benar
67	P	2.33	2.67	2	2	Benar
68	P	2.69	2.95	2	2	Benar
69	P	2.44	2.47	2	2	Benar
70	L	2.44	1.97	3	2	Salah
71	P	2.53	1.47	2	2	Benar
72	P	2.36	2.36	2	2	Benar
73	L	3.27	2.79	2	2	Benar
74	P	3.14	2.91	2	2	Benar
75	P	2.83	2.6	2	2	Benar
76	L	3.0	3.21	2	2	Benar
77	P	2.69	2.5	2	2	Benar
78	L	3.39	2.52	2	2	Benar
79	L	2.22	2.78	2	2	Benar
80	L	2.61	3.31	2	2	Benar
81	L	3.47	3.12	2	2	Benar
82	P	3.08	3.38	2	2	Benar
83	P	3.47	3.52	1	1	Benar
84	L	3.56	3.17	2	2	Benar
85	P	3.06	3.45	2	2	Benar
86	L	2.69	3.24	2	2	Benar
87	L	2.69	3.38	2	2	Benar
88	L	3.53	3.31	2	2	Benar
89	L	2.33	2.86	2	2	Benar
90	P	2.69	2.88	2	2	Benar
91	P	3.31	3.45	1	2	Salah
92	L	2.81	2.95	2	2	Benar
93	L	3.39	3.26	1	2	Salah
94	P	2.69	3.07	2	2	Benar
95	P	3.17	2.69	2	2	Benar
96	L	3.0	3.29	2	2	Benar

97	P	3.25	3.12	2	2	Benar
98	P	3.19	3.56	2	2	Benar
99	L	2.58	2.88	2	2	Benar
100	P	2.92	3.1	2	2	Benar
101	P	2.42	3.03	2	2	Benar

Sourcecode perhitungan information gain untuk atribut IP semester 1 dan 2

```

1      public double getSemesterGain(int semester)
2      {
3          int total = 0;
4
5          double kurangKategori1 = 0.00;
6          double kurangKategori2 = 0.00;
7          double kurangKategori3 = 0.00;
8
9          double memuaskanKategori1 = 0.00;
10         double memuaskanKategori2 = 0.00;
11         double memuaskanKategori3 = 0.00;
12
13         double sangatMemuaskanKategori1 = 0.00;
14         double sangatMemuaskanKategori2 = 0.00;
15         double sangatMemuaskanKategori3 = 0.00;
16
17         double cumlaudeKategori1 = 0.00;
18         double cumlaudeKategori2 = 0.00;
19         double cumlaudeKategori3 = 0.00;
20
21         ArrayList<Integer> semesterDataIndex = new ArrayList<>();
22
23         if(semester == 1)
24         {
25             semesterDataIndex.add(4);
26             semesterDataIndex.add(5);
27             semesterDataIndex.add(6);
28             semesterDataIndex.add(7);
29         }
30         else if(semester == 2)
31         {
32             semesterDataIndex.add(8);
33             semesterDataIndex.add(9);
34             semesterDataIndex.add(10);
35             semesterDataIndex.add(11);
36         }
37
38         for(Iterator iter = getData().iterator(); iter.hasNext();)
39         {
40
41             Vector dataRow = (Vector) iter.next();
42
43             int kategori = (int) dataRow.get(13);
44
45             if(kategori == 1)
46             {
47                 kurangKategori1 += (double)
48                 dataRow.get(semesterDataIndex.get(0));
49                 memuaskanKategori1 += (double)

```

```

50  dataRow.get(semesterDataIndex.get(1));
51      sangatMemuaskanKategori1 += (double)
52  dataRow.get(semesterDataIndex.get(2));
53      cumlaudeKategori1 += (double)
54  dataRow.get(semesterDataIndex.get(3));
55  }
56  else if(kategori == 2)
57  {
58      kurangKategori2 += (double)
59  dataRow.get(semesterDataIndex.get(0));
60      memuaskanKategori2 += (double)
61  dataRow.get(semesterDataIndex.get(1));
62      sangatMemuaskanKategori2 += (double)
63  dataRow.get(semesterDataIndex.get(2));
64      cumlaudeKategori2 += (double)
65  dataRow.get(semesterDataIndex.get(3));
66  }
67  else if(kategori == 3)
68  {
69      kurangKategori3 += (double)
70  dataRow.get(semesterDataIndex.get(0));
71      memuaskanKategori3 += (double)
72  dataRow.get(semesterDataIndex.get(1));
73      sangatMemuaskanKategori3 += (double)
74  dataRow.get(semesterDataIndex.get(2));
75      cumlaudeKategori3 += (double)
76  dataRow.get(semesterDataIndex.get(3));
77  }
78
79      total++;
80  }
81
82      double kurangTotal = kurangKategori1 + kurangKategori2 +
83  kurangKategori3;
84      double memuaskanTotal = memuaskanKategori1 +
85  memuaskanKategori2 + memuaskanKategori3;
86      double sangatMemuaskanTotal = sangatMemuaskanKategori1 +
87  sangatMemuaskanKategori2 + sangatMemuaskanKategori3;
88      double cumlaudeTotal = cumlaudeKategori1 +
89  cumlaudeKategori2 + cumlaudeKategori3;
90
91      double kurang = -(MathHelper.substract(kurangKategori1,
92  kurangTotal) * MathHelper.log2(kurangKategori1/kurangTotal)) -
93  (MathHelper.substract(kurangKategori2, kurangTotal) *
94  MathHelper.log2(kurangKategori2/kurangTotal)) -
95  (MathHelper.substract(kurangKategori3, kurangTotal) *
96  MathHelper.log2(kurangKategori3/kurangTotal));
97      double memuaskan = -
98  (MathHelper.substract(memuaskanKategori1, memuaskanTotal) *
99  MathHelper.log2(memuaskanKategori1/memuaskanTotal)) -
100  (MathHelper.substract(memuaskanKategori2, memuaskanTotal) *
101  MathHelper.log2(memuaskanKategori2/memuaskanTotal)) -
102  (MathHelper.substract(memuaskanKategori3, memuaskanTotal) *
103  MathHelper.log2(memuaskanKategori3/memuaskanTotal));
104      double sangatMemuaskan = -
105  (MathHelper.substract(sangatMemuaskanKategori1,
106  sangatMemuaskanTotal) *
107  MathHelper.log2(sangatMemuaskanKategori1/sangatMemuaskanTotal))
108  - (MathHelper.substract(sangatMemuaskanKategori2,

```



```

109 sangatMemuaskanTotal) *
110 MathHelper.log2(sangatMemuaskanKategori2/sangatMemuaskanTotal))
111 - (MathHelper.subtract(sangatMemuaskanKategori3,
112 sangatMemuaskanTotal) *
113 MathHelper.log2(sangatMemuaskanKategori3/sangatMemuaskanTotal));
114 double cumlaude = -(MathHelper.subtract(cumlaudeKategori1,
115 cumlaudeTotal) *
116 MathHelper.log2(cumlaudeKategori1/cumlaudeTotal)) -
117 (MathHelper.subtract(cumlaudeKategori2, cumlaudeTotal) *
118 MathHelper.log2(cumlaudeKategori2/cumlaudeTotal)) -
119 (MathHelper.subtract(cumlaudeKategori3, cumlaudeTotal) *
120 MathHelper.log2(cumlaudeKategori3/cumlaudeTotal));
121
122 double finalResult = getEntropy() -
123 (MathHelper.subtract(kurangTotal, total) * kurang) -
124 (MathHelper.subtract(memuaskanTotal, total) * memuaskan) -
125 (MathHelper.subtract(sangatMemuaskanTotal, total) *
126 sangatMemuaskan) - (MathHelper.subtract(cumlaudeTotal, total) *
127 cumlaude);
128
129 return finalResult;
130 }

```

Sourcecode 4. 11 Implementasi perhitungan *information gain* atribut semester

Sourcecode pembentukan tree

```

1 private void buildTreeModel()
2 {
3     int totalData = getData().size();
4     ArrayList rootGainsCalculation = calculateLevelGain(0,
5     getData(), null, null);
6     HashMap rootMetadata = (HashMap)
7     rootGainsCalculation.get(0);
8     DefaultMutableTreeNode root = new
9     DefaultMutableTreeNode(rootMetadata.get(0) + " (" +
10     String.valueOf(totalData) + ") [100%]");
11     int rootCounter = 4;
12     int ldtTotal = (int) Math.round((getLdt()/100) *
13     totalData);
14     String[] rootCategories =
15     getAttributeCategories((String)rootMetadata.get(0));
16     for(int rootCategory = 0; rootCategory <
17     rootCategories.length; rootCategory++)
18     {
19         ArrayList rootData = calculateData(getData(),
20         rootCategories[rootCategory], (String)rootMetadata.get(0));
21         HashMap rootCalculation =
22         calculateNodePercentage((double) rootData.get(3),
23         (double)rootData.get(0), (double)rootData.get(1),
24         (double)rootData.get(2));
25
26         int rootCategoryTotal = (int) rootData.get(rootCounter);
27         DefaultMutableTreeNode rootCategoryNode =
28         createAndAttachNode(root, rootCategories[rootCategory] + " (" +
29         + String.valueOf(rootCategoryTotal) + ") " + "[" +
30         rootCalculation.get("category1") + "%] " + "[" +
31         rootCalculation.get("category2") + "%] " + "[" +
32         rootCalculation.get("category3") + "%]");
33         if(rootCategoryTotal >= ldtTotal &&

```

```

34 (double)rootCalculation.get("max") <= getFct())
35 {
36     System.out.println("(1)"+ldtTotal);
37     ArrayList branchGainsCalculation = calculateLevelGain(1,
38 (ArrayList)rootData.get(8), (String)rootMetadata.get(0),
39 null);
40     HashMap branchMetadata = (HashMap)
41 branchGainsCalculation.get(0);
42
43     DefaultMutableTreeNode branchNode =
44 createAndAttachNode(rootCategoryNode, branchMetadata.get(0) +
45 "" + " (" + String.valueOf(rootCategoryTotal) + ")");
46     int branchCounter = 4;
47     String[] branchCategories =
48 getAttributeCategories((String)branchMetadata.get(0));
49     for(int branchCategory = 0; branchCategory <
50 branchCategories.length; branchCategory++)
51     {
52         ArrayList branchData =
53 calculateData((ArrayList)rootData.get(8),
54 branchCategories[branchCategory],
55 (String)branchMetadata.get(0));
56
57         int branchCategoryTotal =
58 (int)branchData.get(branchCounter);
59         HashMap branchCalculation =
60 calculateNodePercentage((double) branchData.get(3), (double)
61 branchData.get(0), (double) branchData.get(1), (double)
62 branchData.get(2));
63         if(branchCategoryTotal > 0)
64         {
65             DefaultMutableTreeNode branchCategoryNode =
66 createAndAttachNode(branchNode,
67 branchCategories[branchCategory] + " (" +
68 String.valueOf(branchCategoryTotal) + ") " + "[" +
69 branchCalculation.get("category1") + "%] " + "[" +
70 branchCalculation.get("category2") + "%] " + "[" +
71 branchCalculation.get("category3") + "%]");
72             if(branchCategoryTotal >= ldtTotal &&
73 (double)branchCalculation.get("max") <= getFct())
74             {
75                 System.out.println("(2)"+ldtTotal);
76                 ArrayList lastGainsCalculation =
77 calculateLevelGain(2, (ArrayList)branchData.get(8),
78 (String)rootMetadata.get(0), (String)branchMetadata.get(0));
79                 HashMap lastMetadata = (HashMap)
80 lastGainsCalculation.get(0);
81                 DefaultMutableTreeNode lastNode =
82 createAndAttachNode(branchCategoryNode, lastMetadata.get(0) +
83 "" + " (" + String.valueOf(branchCategoryTotal) + ")");
84
85                 String[] lastCategories =
86 getAttributeCategories((String)lastMetadata.get(0));
87                 for(int lastCategory = 0; lastCategory <
88 lastCategories.length; lastCategory++)
89                 {
90                     ArrayList lastBranchData =
91 calculateData((ArrayList)branchData.get(8),
92 lastCategories[lastCategory], (String)lastMetadata.get(0));

```



```

93         HashMap lastCalculation =
94         calculateNodePercentage((double)lastBranchData.get(3),
95         (double) lastBranchData.get(0), (double)
96         lastBranchData.get(1), (double) lastBranchData.get(2));
97
98         if((double)lastBranchData.get(3) > 0)
99         {
100             String lastBranchPercent = "[" +
101             lastCalculation.get("category1") + "%" + "[" +
102             lastCalculation.get("category2") + "%" + "[" +
103             lastCalculation.get("category3") + "%]";
104             DefaultMutableTreeNode lastCategoryNode =
105             createAndAttachNode(lastNode, lastCategories[lastCategory] +
106             "(" +
107             String.valueOf(Math.round((double)lastBranchData.get(3))) +
108             ")" + lastBranchPercent);
109             createAndAttachNode(lastCategoryNode,
110             (int)lastCalculation.get("class"));
111         }
112         lastBranchData.clear();
113     }
114 }
115 else
116 {
117     createAndAttachNode(branchCategoryNode,
118     (int)branchCalculation.get("class"));
119 }
120 }
121 branchData.clear();
122 branchCounter++;
123 }
124 }
125 else
126 {
127     createAndAttachNode(rootCategoryNode,
128     (int)rootCalculation.get("class"));
129 }
130 rootData.clear();
131 rootCounter++;
132 }
133 treeModel = new ID3TreeModel(root);
134 }

```

Sourcecode 4. 12 Implementasi pembentukan tree

Sourcecode pembentukan aturan

```

1      public ArrayList getRules()
2      {ArrayList result = new ArrayList<>();
3        DefaultMutableTreeNode rootNode = (DefaultMutableTreeNode)
4        getId3TreeModel().getRoot();
5        String rootType = (String)rootNode.getUserObject();
6        Enumeration rootCategories = rootNode.children();
7        while (rootCategories.hasMoreElements())
8        {
9          DefaultMutableTreeNode rootCategory =
10         (DefaultMutableTreeNode) rootCategories.nextElement();
11          DefaultMutableTreeNode branch = (DefaultMutableTreeNode)
12         rootCategory.getFirstChild();
13          String rootChildType =
14         (String)rootCategory.getUserObject();
15          if(!branch.isLeaf())
16          {
17            Enumeration branchCategories = branch.children();
18
19            String rootChildLeafChildrenType =
20            (String)branch.getUserObject();
21            while (branchCategories.hasMoreElements())
22            {
23              DefaultMutableTreeNode branchCategory =
24              (DefaultMutableTreeNode) branchCategories.nextElement();
25              DefaultMutableTreeNode last = (DefaultMutableTreeNode)
26              branchCategory.getFirstChild();
27
28              String branchCategoryType =
29              (String)branchCategory.getUserObject();
30
31              if (!last.isLeaf())
32              {
33                Enumeration lastCategories = last.children();
34                while (lastCategories.hasMoreElements())
35                {
36                  DefaultMutableTreeNode lastCategory =
37                  (DefaultMutableTreeNode) lastCategories.nextElement();
38                  if (!lastCategory.isLeaf())
39                  {
40                    DefaultMutableTreeNode classNode =
41                    (DefaultMutableTreeNode) lastCategory.getFirstChild();
42                    String lastCategoryType =
43                    (String)lastCategory.getUserObject();
44                    int classType =
45                    getClassType(classNode.getUserObject());
46                    result.add("IF IP " + rootType.substring(0, 10) + "
47                    = " + this.getTypeString(rootChildType) + " AND IP " +
48                    rootChildLeafChildrenType.substring(0, 10) + " = " +
49                    this.getTypeString(branchCategoryType) + " AND jenis kelamin
50                    = " + this.getTypeString(lastCategoryType) + " THEN " +
51                    this.getRangeIPKelulusan(classType));
52                  }
53                  else
54                  {
55                    int classType = getClassType(last.getUserObject());
56
57                    result.add("IF IP " + rootType.substring(0, 10) + "

```

```

58 = " + this.getTypeString(rootChildType) + " AND IP " +
59 rootChildLeafChildrenType.substring(0, 10) + " = " +
60 this.getTypeString(branchCategoryType) + " THEN " +
61 this.getRangeIPKelulusan(classType));
62     }
63     }
64     }
65     else{
66         int classType = getClassType(last.getUserObject());
67
68         result.add("IF IP " + rootType.substring(0, 10) + " = "
69 + this.getTypeString(rootChildType) + " AND IP " +
70 rootChildLeafChildrenType.substring(0, 10) + " = " +
71 this.getTypeString(branchCategoryType) + " THEN " +
72 this.getRangeIPKelulusan(classType));
73     }
74     }
75     }
76     else
77     {
78         int classType = getClassType(branch.getUserObject());
79
80         result.add("IF IP " + rootType.substring(0, 10) + " = "
81 + this.getTypeString(rootChildType) + " THEN " +
82 this.getRangeIPKelulusan(classType));
83     }
84     }
85     return result;
86     }

```

Sourcecode 4. 13 Implementasi pembentukan aturan

Tabel hasil pengujian jurusan/program studi Matematika

Data Latih 30%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	88,889	88,889	88,889	88,889	88,889
55	4	4	4	4	4	55	88,889	88,889	88,889	88,889	88,889
60	5	5	4	4	4	60	89,815	89,815	88,889	88,889	88,889
65	5	5	4	4	4	65	89,815	89,815	88,889	88,889	88,889
70	6	5	4	4	4	70	89,815	89,815	88,889	88,889	88,889
75	9	8	7	7	4	75	89,815	89,815	88,889	88,889	88,889
80	9	8	7	7	4	80	89,815	89,815	88,889	88,889	88,889
85	9	8	7	7	4	85	85,185	89,815	88,889	88,889	88,889
90	9	8	7	7	4	90	85,185	89,815	88,889	88,889	88,889
95	11	10	9	9	6	95	84,259	88,889	87,963	87,963	87,963
98	11	10	9	9	6	98	84,259	88,889	87,963	87,963	87,963

Tabel hasil pengujian jurusan/program studi Matematika

Data Latih 40%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	86,957	86,957	86,957	86,957	86,957
55	5	5	4	4	4	55	86,957	86,957	86,957	86,957	86,957
60	5	5	4	4	4	60	86,957	86,957	86,957	86,957	86,957
65	5	5	4	4	4	65	86,957	86,957	86,957	86,957	86,957
70	6	5	4	4	4	70	86,957	86,957	86,957	86,957	86,957
75	6	5	4	4	4	75	86,957	86,957	86,957	86,957	86,957
80	11	9	8	7	7	80	88,043	86,957	86,957	86,957	86,957
85	11	9	8	7	7	85	88,043	86,957	86,957	86,957	86,957
90	11	9	8	7	7	90	88,043	86,957	86,957	86,957	86,957
95	14	12	11	10	10	95	88,043	86,957	86,957	86,957	86,957
98	14	12	11	10	10	98	88,043	86,957	86,957	86,957	86,957

Tabel hasil pengujian jurusan/program studi Matematika

Data Latih 50%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	85,526	85,526	85,526	85,526	85,526
55	4	4	4	4	4	55	85,526	85,526	85,526	85,526	85,526
60	4	4	4	4	4	60	85,526	85,526	85,526	85,526	85,526
65	5	5	4	4	4	65	86,842	86,842	85,526	85,526	85,526
70	9	8	7	7	4	70	86,842	80,263	78,947	78,947	85,526
75	11	8	7	7	4	75	86,842	80,263	78,947	78,947	85,526
80	12	9	7	7	4	80	86,842	80,263	78,947	78,947	85,526
85	12	9	7	7	4	85	86,842	80,263	78,947	78,947	85,526
90	12	9	7	7	4	90	86,842	80,263	78,947	78,947	85,526
95	12	9	7	7	4	95	86,842	80,263	78,947	78,947	85,526
98	15	12	10	10	7	98	86,842	80,263	78,947	78,947	85,526

Tabel hasil pengujian jurusan/program studi Matematika

Data Latih 60%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	81,967	81,967	81,967	81,967	81,967
55	4	4	4	4	4	55	81,967	81,967	81,967	81,967	81,967
60	4	4	4	4	4	60	81,967	81,967	81,967	81,967	81,967
65	4	4	4	4	4	65	81,967	81,967	81,967	81,967	81,967
70	4	4	4	4	4	70	81,967	81,967	81,967	81,967	81,967
75	9	8	7	7	4	75	83,607	77,049	75,41	75,41	81,967
80	10	9	7	7	4	80	83,607	77,049	75,41	75,41	81,967
85	10	9	7	7	4	85	83,607	77,049	75,41	75,41	81,967
90	10	9	7	7	4	90	83,607	77,049	75,41	75,41	81,967
95	10	9	7	7	4	95	83,607	77,049	75,41	75,41	81,967
98	13	12	10	10	7	98	83,607	77,049	75,41	75,41	81,967

Tabel hasil pengujian jurusan/program studi Kimia

Data Latih 30%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	85,845	85,845	85,845	85,845	85,845
55	4	4	4	4	4	55	85,845	85,845	85,845	85,845	85,845
60	4	4	4	4	4	60	85,845	85,845	85,845	85,845	85,845
65	4	4	4	4	4	65	85,845	85,845	85,845	85,845	85,845
70	4	4	4	4	4	70	85,845	85,845	85,845	85,845	85,845
75	7	7	7	7	6	75	86,301	86,301	86,301	86,301	85,845
80	7	7	7	7	6	80	86,301	86,301	86,301	86,301	85,845
85	14	14	13	12	10	85	84,932	84,932	84,932	84,932	84,475
90	14	14	13	12	10	90	84,932	84,932	84,932	84,932	84,475
95	14	14	13	12	10	95	84,932	84,932	84,932	84,932	84,475
98	14	14	13	12	10	98	84,932	84,932	84,932	84,932	84,475

Tabel hasil pengujian jurusan/program studi Kimia

Data Latih 40%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	84,574	84,574	84,574	84,574	84,574
55	4	4	4	4	4	55	84,574	84,574	84,574	84,574	84,574
60	4	4	4	4	4	60	84,574	84,574	84,574	84,574	84,574
65	7	7	7	7	6	65	85,106	85,106	85,106	85,106	84,574
70	7	7	7	7	6	70	85,106	85,106	85,106	85,106	84,574
75	7	7	7	7	6	75	85,106	85,106	85,106	85,106	84,574
80	7	7	7	7	6	80	85,106	85,106	85,106	85,106	84,574
85	8	8	8	8	6	85	83,511	83,511	83,511	83,511	84,574
90	14	14	13	13	10	90	83,511	83,511	83,511	83,511	84,574
95	14	14	13	13	10	95	83,511	83,511	83,511	83,511	84,574
98	14	14	13	13	10	98	83,511	83,511	83,511	83,511	84,574

Tabel hasil pengujian jurusan/program studi Kimia

Data Latih 50%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	84,615	84,615	84,615	84,615	84,615
55	6	6	6	6	6	55	84,615	87,821	87,821	87,821	87,821
60	7	7	7	6	6	60	87,821	87,821	87,821	87,821	87,821
65	7	7	7	6	6	65	87,821	87,821	87,821	87,821	87,821
70	7	7	7	6	6	70	87,821	87,821	87,821	87,821	87,821
75	7	7	7	6	6	75	87,821	87,821	87,821	87,821	87,821
80	7	7	7	6	6	80	87,821	87,821	87,821	87,821	87,821
85	8	8	8	7	6	85	87,821	87,821	87,821	87,821	87,821
90	13	13	12	11	9	90	87,821	87,821	87,821	87,821	87,821
95	14	14	13	12	10	95	87,821	87,821	87,821	87,821	87,821
98	19	19	18	17	15	98	87,821	87,821	87,821	87,821	87,821

Tabel hasil pengujian jurusan/program studi Kimia

Data Latih 60%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	86,400	86,400	86,400	86,400	86,400
55	7	7	7	7	7	55	87,200	87,200	87,200	87,200	87,200
60	8	8	8	8	8	60	86,400	86,400	86,400	86,400	86,400
65	8	8	8	8	8	65	86,400	86,400	86,400	86,400	86,400
70	8	8	8	8	8	70	86,400	86,400	86,400	86,400	86,400
75	8	8	8	8	8	75	86,400	86,400	86,400	86,400	86,400
80	9	9	9	9	9	80	86,400	86,400	86,400	86,400	86,400
85	9	9	9	9	9	85	86,400	86,400	86,400	86,400	86,400
90	14	14	13	12	12	90	86,400	86,400	86,400	86,400	86,400
95	15	15	14	13	13	95	86,400	86,400	86,400	86,400	86,400
98	15	15	14	13	13	98	86,400	86,400	86,400	86,400	86,400

Tabel hasil pengujian jurusan/program studi Fisika

Data Latih 30%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	80,734	80,734	80,734	80,734	80,734
55	7	7	7	7	7	55	88,073	88,073	88,073	88,073	88,073
60	8	8	8	8	8	60	88,073	88,073	88,073	88,073	88,073
65	8	8	8	8	8	65	88,073	88,073	88,073	88,073	88,073
70	8	8	8	8	8	70	88,073	88,073	88,073	88,073	88,073
75	8	8	8	8	8	75	88,073	88,073	88,073	88,073	88,073
80	8	8	8	8	8	80	88,073	88,073	88,073	88,073	88,073
85	13	13	13	13	12	85	88,073	88,073	88,073	88,073	88,073
90	14	14	14	14	13	90	88,073	88,073	88,073	88,073	88,073
95	14	14	14	14	13	95	88,073	88,073	88,073	88,073	88,073
98	14	14	14	14	13	98	88,073	88,073	88,073	88,073	88,073

Tabel hasil pengujian jurusan/program studi Fisika

Data Latih 40%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	93,548	93,548	93,548	93,548	93,548
55	4	4	4	4	4	55	93,548	93,548	93,548	93,548	93,548
60	7	7	7	7	6	60	93,548	93,548	93,548	93,548	93,548
65	7	7	7	7	6	65	93,548	93,548	93,548	93,548	93,548
70	8	8	8	8	7	70	93,548	93,548	93,548	93,548	93,548
75	8	8	8	8	7	75	93,548	93,548	93,548	93,548	93,548
80	8	8	8	8	7	80	93,548	93,548	93,548	93,548	93,548
85	12	12	12	12	11	85	93,548	93,548	93,548	93,548	93,548
90	14	13	13	13	12	90	91,398	93,548	93,548	93,548	93,548
95	17	16	16	16	15	95	91,398	93,548	93,548	93,548	93,548
98	21	20	20	20	19	98	91,398	93,548	93,548	93,548	93,548

Tabel hasil pengujian jurusan/program studi Fisika

Data Latih 50%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	92,208	92,208	92,208	92,208	92,208
55	4	4	4	4	4	55	92,208	92,208	92,208	92,208	92,208
60	4	4	4	4	4	60	92,208	92,208	92,208	92,208	92,208
65	7	7	7	6	6	65	92,208	92,208	92,208	92,208	92,208
70	7	7	7	6	6	70	92,208	92,208	92,208	92,208	92,208
75	8	8	8	7	7	75	92,208	92,208	92,208	92,208	92,208
80	8	8	8	7	7	80	92,208	92,208	92,208	92,208	92,208
85	8	8	8	7	7	85	92,208	92,208	92,208	92,208	92,208
90	12	12	12	11	11	90	92,208	92,208	92,208	92,208	92,208
95	17	16	16	15	14	95	89,61	92,208	92,208	92,208	92,208
98	21	20	20	19	17	98	89,62	92,208	92,208	92,208	92,208

Tabel hasil pengujian jurusan/program studi Fisika

Data Latih 60%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	91,935	91,935	91,935	91,935	91,935
55	4	4	4	4	4	55	91,935	91,935	91,935	91,935	91,935
60	4	4	4	4	4	60	91,935	91,935	91,935	91,935	91,935
65	4	4	4	4	4	65	91,935	91,935	91,935	91,935	91,935
70	7	7	6	6	6	70	91,935	91,935	91,935	91,935	91,935
75	9	8	7	7	6	75	88,71	91,935	91,935	91,935	91,935
80	10	9	8	8	7	80	88,72	91,935	91,935	91,935	91,935
85	10	9	8	8	7	85	88,73	91,935	91,935	91,935	91,935
90	14	13	12	12	11	90	88,74	91,935	91,935	91,935	91,935
95	17	16	15	15	14	95	88,75	91,935	91,935	91,935	91,935
98	21	20	19	19	17	98	88,76	91,935	91,935	91,935	91,935

Tabel hasil pengujian jurusan/program studi Statistika

Data Latih 30%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	80,412	80,412	80,412	80,412	80,412
55	4	4	4	4	4	55	80,412	80,412	80,412	80,412	80,412
60	4	4	4	4	4	60	80,412	80,412	80,412	80,412	80,412
65	6	6	6	6	6	65	84,536	84,536	84,536	84,536	84,536
70	6	6	6	6	6	70	84,536	84,536	84,536	84,536	84,536
75	6	6	6	6	6	75	84,536	84,536	84,536	84,536	84,536
80	6	6	6	6	6	80	84,536	84,536	84,536	84,536	84,536
85	9	9	9	9	9	85	82,474	82,474	82,474	82,474	82,474
90	11	11	11	11	11	90	82,474	82,474	82,474	82,474	82,474
95	17	17	17	17	17	95	82,474	82,474	82,474	82,474	82,474
98	17	17	17	17	17	98	82,474	82,474	82,474	82,474	82,474

Tabel hasil pengujian jurusan/program studi Statistika

Data Latih 40%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	79,518	79,518	79,518	79,518	79,518
55	4	4	4	4	4	55	79,518	79,518	79,518	79,518	79,518
60	4	4	4	4	4	60	79,518	79,518	79,518	79,518	79,518
65	4	4	4	4	4	65	79,518	79,518	79,518	79,518	79,518
70	4	4	4	4	4	70	79,518	79,518	79,518	79,518	79,518
75	6	6	6	6	6	75	84,337	84,337	84,337	84,337	84,337
80	9	9	9	9	9	80	81,928	81,928	81,928	81,928	81,928
85	9	9	9	9	9	85	81,928	81,928	81,928	81,928	81,928
90	10	10	10	10	10	90	81,928	81,928	81,928	81,928	81,928
95	16	16	16	16	16	95	81,928	81,928	81,928	81,928	81,928
98	16	16	16	16	16	98	81,928	81,928	81,928	81,928	81,928

Tabel hasil pengujian jurusan/program studi Statistika

Data Latih 50%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	79,71	79,71	79,71	79,71	79,71
55	4	4	4	4	4	55	79,71	79,71	79,71	79,71	79,71
60	4	4	4	4	4	60	79,71	79,71	79,71	79,71	79,71
65	4	4	4	4	4	65	79,71	79,71	79,71	79,71	79,71
70	6	6	6	6	6	70	84,058	84,058	84,058	84,058	84,058
75	6	6	6	6	6	75	84,058	84,058	84,058	84,058	84,058
80	9	9	9	9	9	80	84,058	84,058	84,058	84,058	84,058
85	9	9	9	9	9	85	84,058	84,058	84,058	84,058	84,058
90	9	9	9	9	9	90	84,058	84,058	84,058	84,058	84,058
95	14	14	14	14	14	95	84,058	84,058	84,058	84,058	84,058
98	16	16	16	16	16	98	84,058	84,058	84,058	84,058	84,058

Tabel hasil pengujian jurusan/program studi Statistika

Data Latih 60%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	78,182	78,182	78,182	78,182	78,182
55	4	4	4	4	4	55	78,182	78,182	78,182	78,182	78,182
60	6	6	6	6	6	60	83,636	83,636	83,636	83,636	83,636
65	6	6	6	6	6	65	83,636	83,636	83,636	83,636	83,636
70	6	6	6	6	6	70	83,636	83,636	83,636	83,636	83,636
75	7	7	7	7	7	75	83,636	83,636	83,636	83,636	83,636
80	8	8	8	8	8	80	80,000	80,000	80,000	80,000	80,000
85	11	11	11	11	11	85	80,000	80,000	80,000	80,000	80,000
90	11	11	11	11	11	90	80,000	80,000	80,000	80,000	80,000
95	16	16	16	16	16	95	80,000	80,000	80,000	80,000	80,000
98	18	18	18	18	17	98	80,000	80,000	80,000	80,000	80,000

Tabel hasil pengujian jurusan/program studi Biologi

Data Latih 30%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	78,641	78,641	78,641	78,641	78,641
55	4	4	4	4	4	55	78,641	78,641	78,641	78,641	78,641
60	4	4	4	4	4	60	78,641	78,641	78,641	78,641	78,641
65	4	4	4	4	4	65	78,641	78,641	78,641	78,641	78,641
70	7	7	7	7	7	70	78,641	78,641	78,641	78,641	78,641
75	7	7	7	7	7	75	78,641	78,641	78,641	78,641	78,641
80	7	7	7	7	7	80	78,641	78,641	78,641	78,641	78,641
85	11	11	11	11	11	85	77,67	77,67	77,67	77,67	77,68
90	18	18	18	18	18	90	75,728	75,728	75,728	75,728	75,728
95	18	18	18	18	18	95	75,728	75,728	75,728	75,728	75,728
98	18	18	18	18	18	98	75,728	75,728	75,728	75,728	75,728

Tabel hasil pengujian jurusan/program studi Biologi

Data Latih 40%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	79,545	79,545	79,545	79,545	79,545
55	4	4	4	4	4	55	79,545	79,545	79,545	79,545	79,545
60	4	4	4	4	4	60	79,545	79,545	79,545	79,545	79,545
65	7	7	7	7	7	65	78,409	78,409	78,409	78,409	78,409
70	7	7	7	7	7	70	78,409	78,409	78,409	78,409	78,409
75	11	11	11	11	11	75	78,409	78,409	78,409	78,409	78,409
80	12	12	12	12	12	80	78,409	78,409	78,409	78,409	78,409
85	13	13	13	13	13	85	78,409	78,409	78,409	78,409	78,409
90	13	13	13	13	13	90	78,409	78,409	78,409	78,409	78,409
95	17	17	17	17	17	95	78,409	78,409	78,409	78,409	78,409
98	17	17	17	17	17	98	78,409	78,409	78,409	78,409	78,409

Tabel hasil pengujian jurusan/program studi Biologi

Data Latih 50%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	79,452	79,452	79,452	79,452	79,452
55	4	4	4	4	4	55	79,452	79,452	79,452	79,452	79,452
60	7	7	7	7	7	60	78,082	78,082	78,082	78,082	78,082
65	7	7	7	7	7	65	78,082	78,082	78,082	78,082	78,082
70	7	7	7	7	7	70	78,082	78,082	78,082	78,082	78,082
75	7	7	7	7	7	75	78,082	78,082	78,082	78,082	78,082
80	12	12	12	12	11	80	78,082	78,082	78,082	78,082	78,082
85	13	13	13	13	12	85	78,082	78,082	78,082	78,082	78,082
90	18	18	18	18	17	90	78,082	78,082	78,082	78,082	78,082
95	18	18	18	18	17	95	78,082	78,082	78,082	78,082	78,082
98	18	18	18	18	17	98	78,082	78,082	78,082	78,082	78,082

Tabel hasil pengujian jurusan/program studi Biologi

Data Latih 60%

FCT (%)	JUMLAH ATURAN					FCT (%)	TINGKAT AKURASI (%)				
	LDT (%)						LDT (%)				
	5	8	10	13	15		5	8	10	13	15
50	4	4	4	4	4	50	79,661	79,661	79,661	79,661	79,661
55	4	4	4	4	4	55	79,661	79,661	79,661	79,661	79,661
60	7	7	7	7	7	60	76,271	76,271	76,271	76,271	76,271
65	7	7	7	7	7	65	76,271	76,271	76,271	76,271	76,271
70	7	7	7	7	7	70	76,271	76,271	76,271	76,271	76,271
75	7	8	8	8	8	75	76,271	76,271	76,271	76,271	76,271
80	8	8	8	8	8	80	76,271	76,271	76,271	76,271	76,271
85	11	11	11	11	11	85	76,271	76,271	76,271	76,271	76,271
90	15	15	15	15	15	90	76,271	76,271	76,271	76,271	76,271
95	18	18	18	18	18	95	76,271	76,271	76,271	76,271	76,271
98	18	18	18	18	18	98	76,271	76,271	76,271	76,271	76,271

Daftar akurasi terbaik yang dihasilkan setiap jurusan.

a. **Ilmu Komputer**

Jumlah data latih	: 50%
FCT	: 75%
LDT	: 10%
Jumlah Aturan	: 9
Akurasi	: 92,079 %

Aturan yang dihasilkan :

1. IF IP Semester 2 = Kurang AND IP Semester 1 = Kurang THEN IP kelulusan antara 2,76 – 3,50
2. IF IP Semester 2 = Kurang AND IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
3. IF IP Semester 2 = Kurang AND IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
4. IF IP Semester 2 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
5. IF IP Semester 2 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
6. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
7. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
8. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Cumlaude AND jenis kelamin = Laki-laki THEN IP kelulusan antara 2,76 – 3,50
9. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Cumlaude AND jenis kelamin = Perempuan THEN IP kelulusan lebih besar dari 3,5

b. Matematika

Jumlah data latih	: 30%
FCT	: 60%
LDT	: 8%
Jumlah Aturan	: 5
Akurasi	: 89,815 %
Aturan yang dihasilkan	:

1. IF IP Semester 2 = Kurang AND IP Jenis Kela = Laki-laki THEN IP kelulusan antara 2,00 – 2,76
2. IF IP Semester 2 = Kurang AND IP Jenis Kela = Perempuan THEN IP kelulusan antara 2,76 – 3,50
3. IF IP Semester 2 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
4. IF IP Semester 2 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
5. IF IP Semester 2 = Cumlaude THEN IP kelulusan lebih besar dari 3,5

c. Kimia

Jumlah data latih	: 50%
FCT	: 55%
LDT	: 8%
Jumlah Aturan	: 6
Akurasi	: 87,821 %
Aturan yang dihasilkan	:

1. IF IP Semester 2 = Kurang THEN IP kelulusan antara 2,76 – 3,50
2. IF IP Semester 2 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
3. IF IP Semester 2 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
4. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50

5. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50

6. IF IP Semester 2 = Cumlaude AND IP Semester 1 = Cumlaude THEN IP kelulusan lebih besar dari 3,5

d. **Fisika**

Jumlah data latih : 40%
 FCT : 50%
 LDT : 5%
 Jumlah Aturan : 4
 Akurasi : 93,548%
 Aturan yang dihasilkan :

1. IF IP Semester 1 = Kurang THEN IP kelulusan antara 2,76 – 3,50
2. IF IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
3. IF IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50
4. IF IP Semester 1 = Cumlaude THEN IP kelulusan antara 2,76 – 3,50

e. **Statistika**

Jumlah data latih : 30%
 FCT : 65%
 LDT : 5%
 Jumlah Aturan : 6
 Akurasi : 84,536%
 Aturan yang dihasilkan :

1. IF IP Semester 1 = Kurang THEN IP kelulusan antara 2,76 – 3,50
2. IF IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 – 3,50
3. IF IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara 2,76 – 3,50

4. IF IP Semester 1 = Cumlaude AND Jenis Kelamin = Laki-laki
THEN IP kelulusan lebih besar dari 3,5
5. IF IP Semester 1 = Cumlaude AND Jenis Kelamin = Perempuan
AND IP Semester 2 = Sangat Memuaskan THEN IP kelulusan
antara 2,76 – 3,50
6. IF IP Semester 1 = Cumlaude AND Jenis Kelamin = Perempuan
AND IP Semester 2 = Cumlaude THEN IP kelulusan lebih besar
dari 3,5

f. **Biologi**

Jumlah data latih	: 50%
FCT	: 60%
LDT	: 5%
Jumlah Aturan	: 4
Akurasi	: 78,082%
Aturan yang dihasilkan	:

1. IF IP Semester 1 = Kurang THEN IP kelulusan antara 2,76 – 3,50
2. IF IP Semester 1 = Memuaskan THEN IP kelulusan antara 2,76 –
3,50
3. IF IP Semester 1 = Sangat Memuaskan THEN IP kelulusan antara
2,76 – 3,50
4. IF IP Semester 1 = Cumlaude AND IP Semester 2 = Memuaskan
THEN IP kelulusan lebih besar dari 3,5
5. IF IP Semester 1 = Cumlaude AND IP Semester 2 = Sangat
Memuaskan THEN IP kelulusan antara 2,76 – 3,50
6. IF IP Semester 1 = Cumlaude AND IP Semester 2 = Cumlaude
AND jenis kelamin = Laki-laki THEN IP kelulusan antara 2,76 –
3,50
7. IF IP Semester 1 = Cumlaude AND IP Semester 2 = Cumlaude
AND jenis kelamin = Perempuan THEN IP kelulusan antara 2,76 –
3,50