

**KLASIFIKASI KELUARGA SEJAHTERA MENGGUNAKAN METODE
DECISION TREE DENGAN ALGORITMA ITERATIVE
DICHOTOMISER 3 (ID3) PADA KECAMATAN BATU**

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun oleh :

SELLY JOHANSYAH ISAMUDDIN

105090600111037

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

MALANG

2014

LEMBAR PERSETUJUAN

**KLASIFIKASI KELUARGA SEJAHTERA MENGGUNAKAN METODE
DECISION TREE DENGAN ALGORITMA ITERATIVE
DICHOTOMISER 3 (ID3) PADA KECAMATAN BATU**

**SKRIPSI
KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI**

**Untuk memenuhi sebagian persyaratan untuk
Mencapai gelar Sarjana Komputer**



Disusun Oleh:

SELLY JOHANSYAH ISAMUDDIN

105090600111037

**Telah diperiksa dan disetujui oleh
Dosen Pembimbing**

Dosen Pembimbing I

Dosen Pembimbing II

Candra Dewi, S.kom M.Sc

NIP. 19771114 200312 2 001

Rekyan Regasari Mardi Putri, ST, MT

NIP. 77041406120253

LEMBAR PENGESAHAN

**KLASIFIKASI KELUARGA SEJAHTERA MENGGUNAKAN METODE
DECISION TREE DENGAN ALGORITMA ITERATIVE
DICHOTOMISER 3 (ID3) PADA KECAMATAN BATU**

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

SELLY JOHANSYAH ISAMUDDIN

NIM. 105090600111037

Skripsi ini diuji dan dinyatakan lulus pada tanggal 15 Januari 2015

Penguji I,

Penguji II,

Dian Eka Ratnawati, S.Si, M.Kom

NIP. 19730619 200212 2 001

Lailil Muflikhah, S.Kom., M.Sc

NIP. 19741113 200501 2 001

Penguji III,

Drs. Mardji, M.T

NIP. 19670801 199203 1 001

Mengetahui,

Ketua Program Studi Informatika / Ilmu Komputer

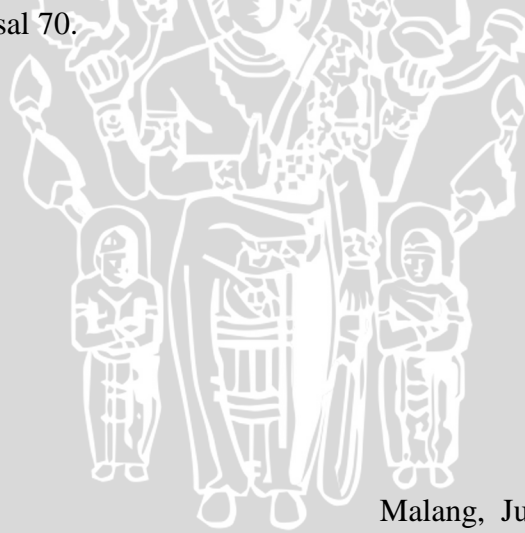
Drs. Marji, M.T.

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 di dalam 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, Juni 2014

Mahasiswa,

Selly Johansyah Isamuddin
NIM. 105090600111037

KATA PENGANTAR

Syukur dan alhamdulillah penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul: ” **KLASIFIKASI KELUARGA SEJAHTERA MENGGUNAKAN METODE DECISION TREE DENGAN ALGORITMA ITERATIVE DICHOTOMISER 3 (ID3) PADA KECAMATAN BATU** “.

Skripsi ini diajukan sebagai syarat ujian skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTIIK), Program Studi Informatika/Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, Penulis mengucapkan terima kasih kepada:

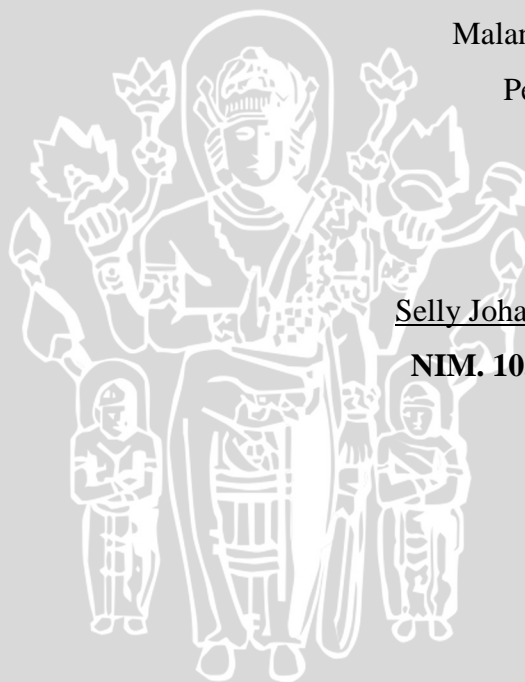
1. Ibu Candra Dewi, S.kom., M.Sc., selaku Dosen Pembimbing Skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Ibu Rekyan Regasari Mardi Putri, ST., MT., selaku Dosen Pembimbing Skripsi kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Drs. Marji, MT. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya
4. Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Ibu Sritiningsih, ayah Alm. Abdul Djalil, dan adek ferdi yang telah selalu mendukung dan mendoakan sehingga terselesaikannya skripsi ini.\
6. Teman – teman kontrakan kendalsari Aries, Alex, Bayu, Danny, Damar, Fadli, dan Alwi yang selalu memberi semangat dan membantu penulis menyelesaikan petualangan untuk meraih gelar sarjana di PTIIK.
7. Teman – teman kos Sigura Hill E-10 Dwy S.Kom, Ryan S.Kom, Apem S.Kom, Yhoga S.Kom, Restu, dan Dedi yang telah mengisi hari – hari dengan motivasi dan petualangan hingga terselesaikannya skripsi penulis.

8. Sahabat Penulis serta teman – teman ilmu komputer angkatan 2010 yang memotivasi dan saling menyemangati.
9. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
10. Segenap staff dan karyawan di Program Studi Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu dalam hal administrasi Penulis dalam pelaksanaan penyusunan skripsi ini.

Akhirnya penulis berharap agar skripsi ini dapat memberikan sumbangan dan manfaat bagi semua pihak yang berkepentingan.

Malang, Juni 2014

Penulis,



Selly Johansyah Isamuddin

NIM. 105090600111037

ABSTRAK

Selly Johansyah Isamuddin. 2014 : Klasifikasi Keluarga Sejahtera Menggunakan Metode Decision Tree dengan Algoritma Iterative Dichotomiser 3 (ID3) Pada Kecamatan Batu.

Dosen Pembimbing : Candra Dewi, S.kom M.Sc. dan Rekyan Regasari Mardi Putri, ST, MT.

Keluarga sejahtera adalah keluarga yang dibentuk berdasarkan atas perkawinan yang sah, mampu memenuhi kebutuhan hidup spiritual dan materiil yang layak, bertakwa kepada Tuhan Yang Maha Esa, memiliki hubungan yang serasi, selaras, dan seimbang antar anggota dan antar keluarga dengan masyarakat dan lingkungan. Pendataan Keluarga yang dilakukan bersama masyarakat ini merupakan data mikro keluarga di daerah yang sesungguhnya dapat menyediakan data dan informasi yang sangat penting untuk Program Pembangunan di Daerah, akan tetapi upaya untuk meningkatkan makna dan pemanfaatannya masih terkendala untuk dapat merekam data pendataan keluarga menjadi Database Keluarga di daerah, karena keterbatasan dari sisi kompetensi tenaga, penyediaan sarana serta program aplikasi yang mudah untuk dapat diimplementasikan di daerah. Banyak indikator untuk penentuan kategori keluarga juga menjadi masalah, meskipun proses pengelompokan telah ditentukan sebelumnya namun banyak kombinasi data terkadang membuat petugas kesulitan dalam menentukan kategori yang cocok dari data tersebut sehingga diperlukan adanya sistem yang dapat mengklasifikasikan kategori keluarga dengan tepat dan mudah diimplementasikan. Sistem ini dibangun menggunakan metode decision tree dengan algoritma iterative dichotomiser 3 (ID3). Hasil uji coba dari skenario uji coba didapatkan akurasi terbaik sebesar 88% dari 28 data latih seimbang.

Kata Kunci : Keluarga Sejahtera, Decision Tree, Iterative Dichotomiser 3, Klasifikasi

ABSTRACT

Selly Johansyah Isamuddin. 2014 : Family Prosper Classification Using Decision Tree with Iterative Dichotomiser 3 (ID3) Algorithm on Batu District
Advisor : Candra Dewi, S.kom M.Sc. and Rekyan Regasari Mardi Putri, ST, MT.

Prosperous family is a family that is formed based on a legal marriage, able to meet the spiritual and material needs of life that worth, fear of God Almighty, have relationships which are harmonious, and balanced between members and between families with the community and the environment. Family Data Collection made with the community is a family micro data in the area that actually can provide the data and information that very important for Regional Development Program, but efforts to improve the meaning and utilization is still constrained to be able to record data into a family of data collection in the Family Database , due to the limitations of the competence of personnel, provision of facilities and easy application program to be implemented in the area. Many indicators for the determination of the family category is also a problem, although the process of grouping a predetermined but many combinations of data sometimes makes attendant difficulty in determining the appropriate category of data so it needs a system that can classify the category of families with precise and easy to implement. The system is built using decision tree method using iterative dichotomiser 3 (ID3) algorithm. The test result of the test scenarios obtained the best accuracy by 88% from 28 balance training data.

Keywords : *Prosperous Family, Classification, Decision Tree, Iterative Dichotomiser 3.*

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS SKRIPSI.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
DAFTAR SOURCE CODE	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Kajian Pustaka.....	6
2.2 Keluarga Sejahtera	8
2.3 Indikator Tahapan Keluarga Sejahtera.....	9
2.4 Decision Tree	11
2.5 Algoritma <i>Iterative Dichotomiser 3 (ID3)</i>	12
2.6 <i>Entropy</i> dan <i>Information Gain</i>	13
2.7 <i>Precision, Recall, F-Measure</i> dan Akurasi	14
BAB III METODOLOGI PENELITIAN dan PERANCANGAN.....	16
3.1 Metodologi	16
3.2 Perancangan Sistem.....	16
3.2.1 Analisa Data.....	16
3.2.2 Analisa Proses.....	18

3.2.2.1 Pelatihan	19
3.2.2.2 Pembentukan Tree.....	19
3.2.2.3 Hitung <i>Entropy</i>	20
3.2.2.4 Perhitungan information gain.....	23
3.2.2.5 GetRoot	25
3.2.2.6 Gambar Tree	26
3.2.2.7 Rule	28
3.3 Perhitungan Manual	28
3.4 Perancangan Uji Coba	50
3.4.1 Skenario 1	50
3.4.2 Skenario 2	51
3.4.3 Skenario 3	51
3.4.4 Skenario 4	52
3.5 Rancangan Antarmuka	52
BAB IV IMPLEMENTASI	55
4.1 Lingkungan Implementasi	55
4.1.1 Lingkungan Perangkat Keras.....	55
4.1.2 Lingkungan Perangkat Lunak.....	55
4.2 Implementasi Program	55
4.2.1 Pembacaan Data (Read File).....	56
4.2.2 Perhitungan <i>Entropy</i>	56
4.2.3 Perhitungan Information Gain	58
4.2.4 GetRoot.....	58
4.2.5 Gambar Tree	59
4.3 Implementasi Antarmuka	61
BAB V PENGUJIAN DAN ANALISIS.....	67
5.1 Hasil Uji Coba Skenario 1	67
5.2 Hasil Uji Coba Skenario 2	71
5.3 Hasil Uji Coba Skenario 3	72
5.4 Hasil Uji Coba Skenario 4	73
BAB VI PENUTUP	74
6.1 Kesimpulan.....	74



6.2 Saran	74
DAFTAR PUSTAKA	75



DAFTAR GAMBAR

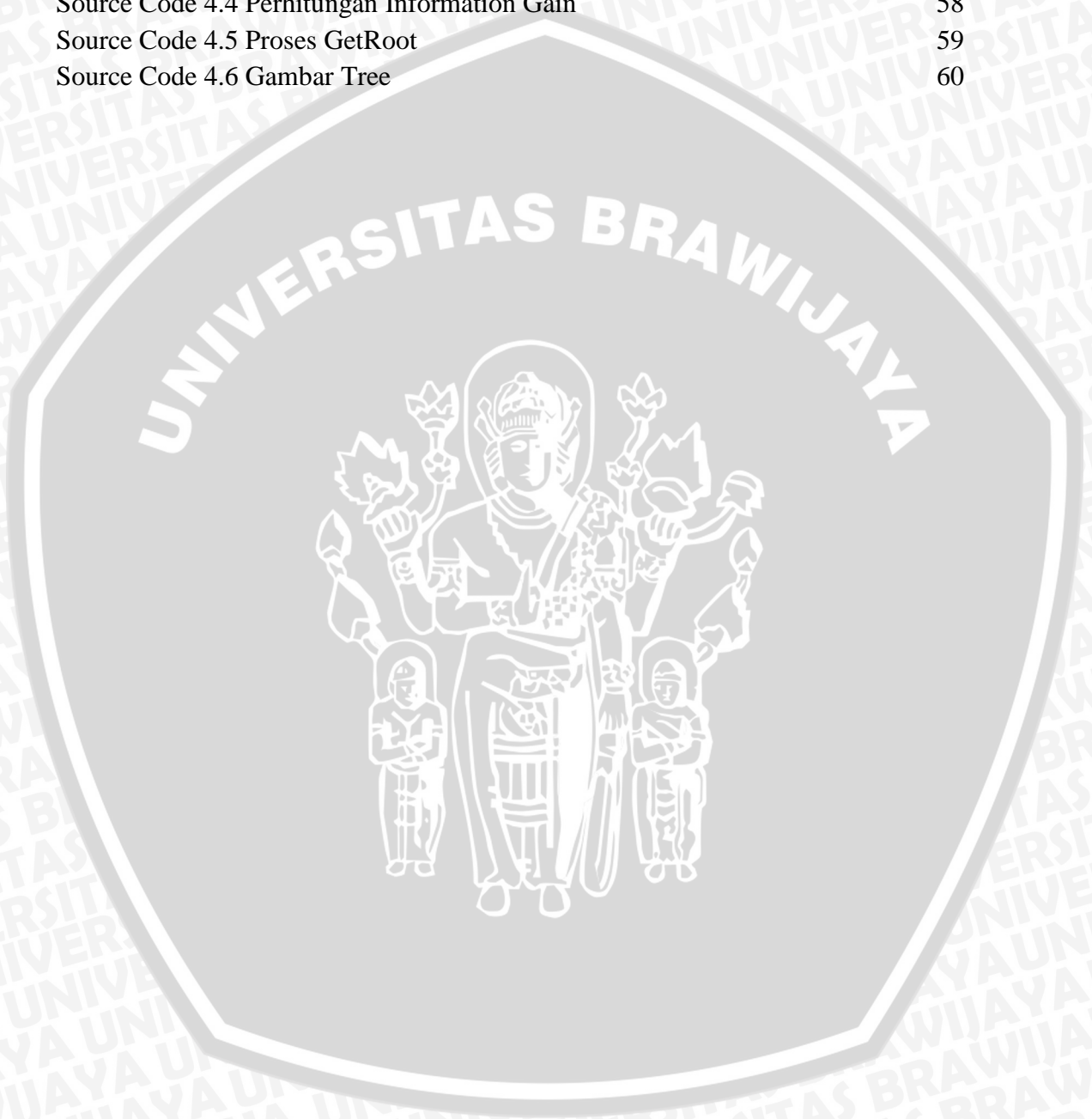
Gambar 3.1 Flowchart Sistem.....	18
Gambar 3.2 Flowchart Pelatihan.....	19
Gambar 3.3 Flowchart Pembentukan Tree.....	20
Gambar 3.4 Flowchart Perhitungan <i>Entropy</i> total.....	21
Gambar 3.5 Flowchart Perhitungan <i>Entropy</i> Nilai dalam Atribut.....	22
Gambar 3.6 Flowchart Perhitungan <i>Entropy</i> Atribut.....	23
Gambar 3.7 Flowchart Perhitungan Information Gain.....	24
Gambar 3.8 Flowchart GetRoot.....	25
Gambar 3.9 Flowchart Gambar Tree.....	28
Gambar 3.10 Tree hasil perhitungan manual tahap I.....	32
Gambar 3.11 Tree hasil perhitungan manual tahap II cabang ya.....	37
Gambar 3.12 Tree hasil perhitungan manual tahap II cabang tidak.....	41
Gambar 3.13 Tree hasil perhitungan manual tahap III.....	45
Gambar 3.14 Rancangan Antarmuka tab Tabel Data Latih.....	53
Gambar 3.15 Rancangan Antarmuka tab Rule.....	53
Gambar 3.16 Rancangan Antarmuka tab Pohon Keputusan.....	54
Gambar 3.17 Rancangan Antarmuka Pengujian.....	54
Gambar 4.1 Antar muka Tabel Data Latih.....	61
Gambar 4.2 Antar muka Tabel Data Latih setelah memilih data.....	62
Gambar 4.3 Antar muka Rule.....	63
Gambar 4.4 Antar muka Rule setelah data latih dimasukkan.....	63
Gambar 4.5 Antar muka Pohon Keputusan.....	64
Gambar 4.6 Antar muka Pohon Keputusan setelah data latih dimasukkan.....	65
Gambar 4.7 Antar muka Data Uji.....	65
Gambar 4.8 Antar muka Data Uji setelah memasukkan data uji.....	66
Gambar 5.1 Grafik Jumlah Aturan yang Terbentuk Skenario 1.....	70
Gambar 5.2 Grafik Akurasi Skenario 1.....	71
Gambar 5.3 Grafik Akurasi Skenario 2.....	72

DAFTAR TABEL

Tabel 3.1 Contoh data indikator dan status TKS	17
Tabel 3.2 Data indikator dan status tks perhitungan manual tahap I	29
Tabel 3.3 Nilai entropy atribut perhitungan manual tahap I	30
Tabel 3.4 Nilai <i>information gain</i> perhitungan manual tahap I	31
Tabel 3.5 Data indikator dan status tks perhitungan manual tahap II cabang ya..	34
Tabel 3.6 Nilai <i>entropy</i> perhitungan manual tahap II cabang ya	35
Tabel 3.7 Nilai <i>information gain</i> perhitungan manual tahap II cabang ya	36
Tabel 3.8 Data indikator dan status tks perhitungan manual tahap II cabang tidak	38
Tabel 3.9 Nilai <i>entropy</i> perhitungan manual tahap II cabang tidak	39
Tabel 3.10 Nilai <i>information gain</i> perhitungan manual tahap II cabang tidak	40
Tabel 3.11 Data indikator dan status tks perhitungan manual tahap III	42
Tabel 3.12 Nilai <i>entropy</i> perhitungan manual tahap III	43
Tabel 3.13 Nilai <i>information gain</i> perhitungan manual tahap III	44
Tabel 3.14 Data untuk pengujian rule	47
Tabel 3.15 Hasil Pengujian tabel 3.14	48
Tabel 3.16 Rancangan Uji Coba Sistem Skenario 1 Aturan	51
Tabel 3.17 Rancangan Uji Coba Sistem Skenario 2	51
Tabel 3.18 Rancangan Uji Coba Sistem Skenario 3	51
Tabel 3.19 Rancangan Uji Coba Sistem Skenario 4	52
Tabel 5.1 Hasil Uji Coba Sistem Skenario 1	67
Tabel 5.2 Hasil Uji Coba Sistem Skenario 2	71
Tabel 5.3 Hasil Uji Coba Sistem Skenario 3	72
Tabel 5.4 Hasil Uji Coba Sistem Skenario 4	73

DAFTAR SOURCE CODE

Source Code 4.1 Proses Perhitungan Entropy Total	57
Source Code 4.2 Proses Perhitungan Entropy Nilai dalam Atribut	57
Source Code 4.3 Proses Perhitungan Entropy Atribut	58
Source Code 4.4 Perhitungan Information Gain	58
Source Code 4.5 Proses GetRoot	59
Source Code 4.6 Gambar Tree	60



BAB I

PENDAHULUAN

1.1 Latar Belakang

Badan Kependudukan dan Keluarga Berencana Nasional (BKKBN) membagi jenis keluarga dalam masyarakat menjadi beberapa kategori keluarga yakni keluarga prasejahtera, keluarga sejahtera I, keluarga sejahtera II, keluarga sejahtera III, dan keluarga sejahtera III plus. Pembagian kategori keluarga tersebut dilakukan oleh petugas – petugas setempat dengan cara manual berdasarkan indikator – indikator yang didapat dari survey. Indikator untuk menentukan status terdiri dari 21 macam indikator dengan nilai indikator berupa data kategorikal yakni ya, tidak, dan ada indikator yang tidak diisi oleh masyarakat.

Pendataan Keluarga yang dilakukan bersama masyarakat ini merupakan data mikro keluarga di daerah yang sesungguhnya dapat menyediakan data dan informasi yang sangat penting untuk Program Pembangunan di Daerah, khususnya Program KKB Nasional, akan tetapi upaya untuk meningkatkan makna dan pemanfaatannya masih terkendala untuk dapat merekam data pendataan keluarga menjadi Database Keluarga di daerah, karena keterbatasan dari sisi kompetensi tenaga, penyediaan sarana serta program aplikasi yang mudah untuk dapat dimplementasikan di daerah. Pemanfaatan data yang dilakukan dewasa ini masih dilakukan dalam bentuk tabulasi pengolahan sederhana, terutama untuk manajemen operasional, pembuatan segmentasi potensi sasaran per wilayah dan pembuatan peta kerja operasional di lapangan [BKB-12]. Banyak indikator untuk penentuan kategori keluarga juga menjadi masalah, meskipun proses pengelompokan telah ditentukan sebelumnya namun banyak kombinasi data terkadang membuat petugas kesulitan dalam menentukan kategori yang cocok dari data tersebut.

Proses pengelompokan data/klasifikasi manual ini bisa diperbaiki dengan proses klasifikasi yang terkomputerisasi. Proses klasifikasi yang terkomputerisasi bisa membantu petugas menentukan kategori yang cocok dari suatu data yang kombinasinya rumit. Klasifikasi ini juga bisa mengcover opini – opini petugas

dalam menentukan kategori suatu data, sehingga tidak terjadi perbedaan kategori pada suatu data yang sama dikarenakan perbedaan opini antara petugas satu dengan lainnya. Kelebihan lain dari proses klasifikasi terkomputerisasi ini yakni mempermudah petugas dalam menjalankan tugasnya, serta apabila terdapat data keluarga yang amat banyak klasifikasi terkomputerisasi ini bisa saja lebih cepat daripada klasifikasi manual yang dilakukan oleh petugas. Langkah pertama proses klasifikasi terkomputerisasi ini dapat dilakukan dengan memilih metode yang tepat. Pemilihan metode mempengaruhi kinerja klasifikasi yang nantinya juga berpengaruh apakah dengan proses yang terkomputerisasi ini meningkatkan kualitas klasifikasi dan mempermudah petugas. Pemilihan metode untuk klasifikasi pada dasarnya di tentukan oleh jenis data yang akan diklasifikasikan. Data kategorikal misalnya, dapat menggunakan beberapa metode, salah satunya adalah decision tree dengan algoritma *Iterative Dichotomiser 3* (ID3). Decision tree ID3 telah banyak digunakan pada penelitian dengan data kategorikal dan memberikan hasil yang cukup baik.

Decision Tree adalah sebuah struktur pohon, dimana setiap node pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan node daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level node teratas dari sebuah *decision tree* adalah node akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Salah satu algoritma *decision tree* yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi *rekursif*. Algoritma ID3 berusaha membangun *decision tree* secara *top-down*, mulai dengan pertanyaan : “atribut mana yang pertama kali harus dicek dan diletakkan pada *root*?” pertanyaan ini dijawab dengan mengevaluasi semua atribut yang ada dengan menggunakan suatu ukuran statistik (*information gain*) untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan kumpulan sampel data [WYD-09].

Penelitian sebelumnya menggunakan algoritma *ID3* pernah dilakukan oleh Rashmi, G. Dona [RGD-14] untuk proses klasifikasi obesitas pada anak kecil. Penelitian tersebut menggunakan data kategorikal dan bekerja dengan baik. Hasil yang didapatkan dari algoritma *ID3* berupa rule yang kemudian digunakan untuk

mengklasifikasikan apakah anak tersebut termasuk dalam obesitas atau tidak. Penelitian yang lain yakni penelitian [RSS-13] ID3 digunakan untuk proses klasifikasi apakah seorang sarjana itu buruk, baik, atau luar biasa. Jenis data kontinyu dirubah dahulu ke dalam bentuk diskrit/kategorikal sebelum digunakan pada algoritma. Hasil dari penerapan algoritma ke dalam data berjalan dengan baik, algoritma dapat memberikan panduan pada para sarjana untuk mengevaluasi diri berdasarkan aturan yang didapat dari pohon yang terbentuk.

Hasil penelitian diatas dan juga permasalahan yang dihadapi dari proses klasifikasi manual yang dilakukan oleh petugas setempat membuat peneliti merasa perlu dilakukan penelitian lebih lanjut untuk proses pengklasifikasian ini secara terkomputerisasi agar dapat mempermudah petugas menentukan kategori suatu keluarga serta mengakomodir persepsi – persepsi petugas pada suatu data dengan kombinasi yang rumit. Berkaca dari penelitian Rashmi, G. Dona misalnya, dimana dia menggunakan *ID3* untuk melakukan klasifikasi dengan jenis data kategorikal dan berjalan dengan baik, maka dari itu dirasa bisa diterapkan pula dalam kasus klasifikasi keluarga sejahtera ini karena jenis data yang diklasifikasikan juga sama yakni kategorikal.

1.2 Rumusan Masalah

Rumusan masalah dalam skripsi ini adalah :

1. Bagaimana mengimplementasikan algoritma *ID3* untuk klasifikasi tahapan keluarga sejahtera?
2. Bagaimana tingkat akurasi algoritma *ID3* untuk klasifikasi tahapan keluarga sejahtera?

1.3 Batasan Masalah

Dari permasalahan yang dirumuskan diatas, maka batasan permasalahan yang digunakan untuk merancang dan membuat sistem ini adalah :

1. Indikator yang digunakan untuk klasifikasi tahapan keluarga sejahtera adalah indikator yang sudah ditetapkan oleh BKKBN.

2. Data yang digunakan adalah data hasil pemutakhiran keluarga tahun 2013 di Desa Tulungrejo, RT. 01, 02, 03 – RW. 04 dan RT. 01, 02, 03, 04, 05 – RW. 05, Kecamatan Bumi Aji, Kota Batu.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas maka tujuan dari skripsi ini adalah untuk mengimplementasikan algoritma *ID3* kedalam program untuk mengklasifikasi status tahapan keluarga sejahtera agar dapat membantu petugas.

1.5 Manfaat

Manfaat yang diharapkan dari hasil skripsi ini adalah mempermudah dan mempercepat petugas pendataan keluarga dalam proses klasifikasi tahapan keluarga sejahtera.

1.6 Sistematika Penulisan

Pembuatan tugas akhir ini dilakukan dengan sistematika penulisan sebagai berikut :

1. BAB I PENDAHULUAN

Berisi latar belakang, permasalahan, tujuan, batasan masalah, dan manfaat serta sistematika penulisan skripsi.

2. BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Berisi teori tentang kajian teori yang berhubungan dalam penelitian skripsi ini serta terori dasar tentang tahapan keluarga sejahtera, google maps, *Decision Tree*, algoritma *Iterative Dichotomiser 3* dan teori-teori yang berhubungan dalam skripsi ini.

3. BAB III METODOLOGI DAN PERANCANGAN

Berisi algoritma-algoritma yang digunakan dalam pembuatan sistem klasifikasi tahapan keluarga sejahtera dengan menggunakan algoritma *iterative dichotomiser 3*.

4. BAB IV IMPLEMENTASI

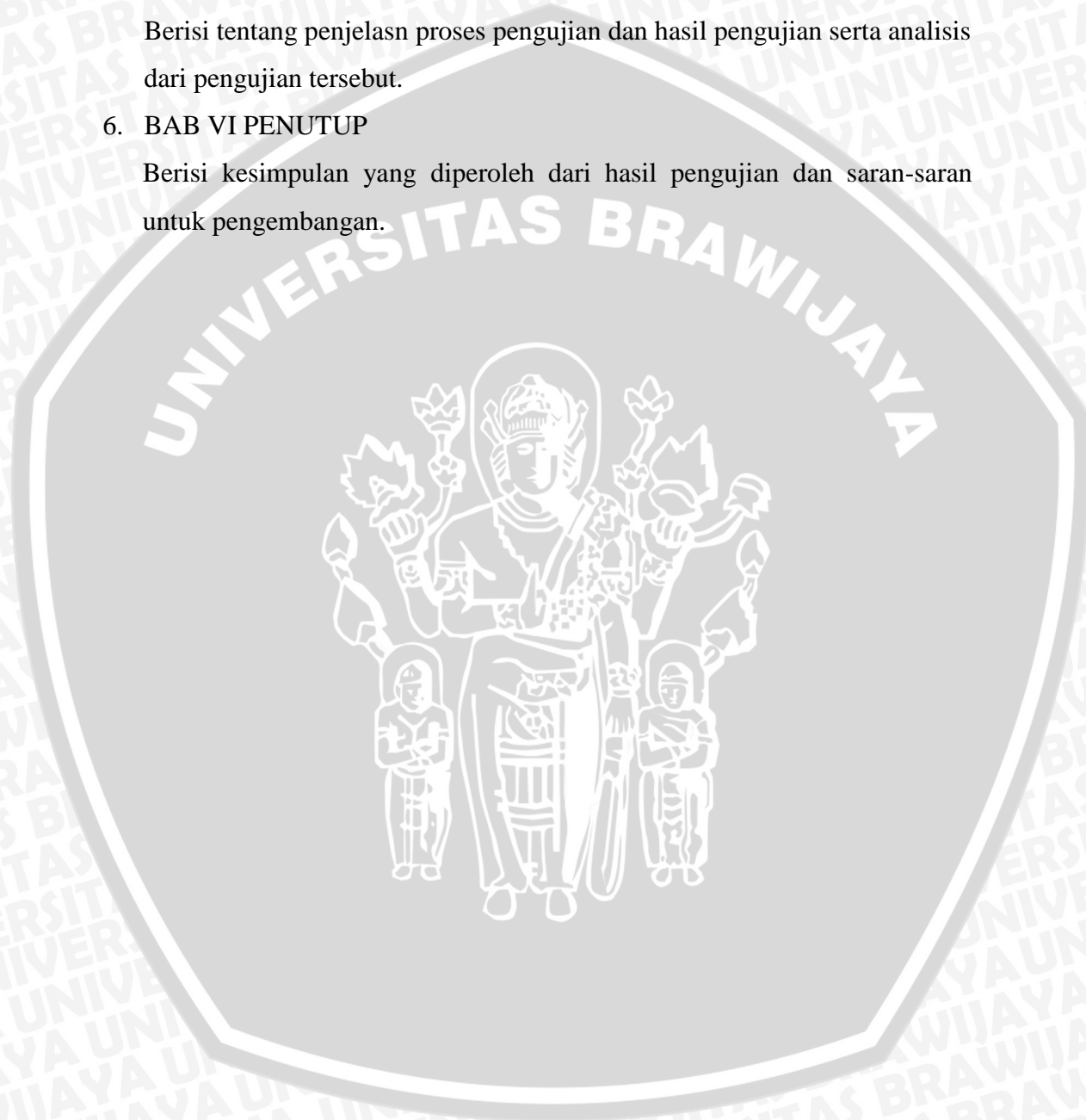
Berisi tentang penjelasan implementasi dari sistem, bagaimana user interface sistem dan *source code* untuk mengembangkan sistem.

5. BAB V PENGUJIAN DAN ANALISIS

Berisi tentang penjelasan proses pengujian dan hasil pengujian serta analisis dari pengujian tersebut.

6. BAB VI PENUTUP

Berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran untuk pengembangan.



BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Berdasarkan judul skripsi yang dibahas, penulis menemukan beberapa hasil penelitian yang relevan untuk mendukung penelitian dalam skripsi ini, antara lain: penelitian [RGD-14] yang menjelaskan algoritma *Iterative Dichotomiser 3* untuk mengklasifikasi data obesitas pada anak - anak. Data yang digunakan dalam jurnal sangat sedikit, dan data dikelompokkan menjadi dua class, yakni Healthy dan Obesity. Data yang diperoleh sebelum diproses diberi batasan terlebih dahulu. Perhitungan untuk penentuan atribut terbaik untuk memisahkan data ke dalam class pun dimulai setelah batasan ditentukan. Rashmi, G. Dona menyebutkan hasil dari algoritma *Iterative Dichotomiser 3* cukup baik, sayangnya dalam jurnal ini pembahasan sangat terbatas dan tidak dituliskan secara pasti angka akurasi serta performa dari algoritma tersebut.

Penelitian yang selanjutnya [NOC-12], menjelaskan penggunaan algoritma ID3 untuk klasifikasi kemampuan tanah yang ada di Maharashtra, India. ID3 dievaluasi menggunakan data dari 38 seri tanah (Sharma et. Al. 2008) dari Kabupaten Wardha, Maharashtra. Kedalaman tanah, kemiringan, drainase, tekstur, erosi, dan permeabilitas dipilih sebagai atribut untuk klasifikasi kemampuan lahan. Penerapan algoritma di mulai dari perhitungan nilai information gain tiap attribute dan kemudian memilih atribut dengan nilai information gain tertinggi untuk dijadikan node. Proses tersebut dilakukan secara rekursif hingga nilai information gain tidak ada yang lebih dari 0. Performa dari algoritma tersebut dijabarkan dari 38 sampel yang digunakan, 33 berhasil diklasifikasi dengan benar, dan 3 data diklasifikasi dengan tidak tepat, sedangkan 2 data tidak dapat diklasifikasikan. Keakuratan keseluruhan adalah 86,84%.

Dalam dunia nyata, dataset mengandung *noise*. Tidak semua nilai pada masing-masing atribut tersedia pada saat pembelajaran. Nilai atribut yang hilang ini dapat dianggap sebagai *noise*. Melihat cara Algoritma ID3 dirumuskan, sulit untuk berurusan dengan nilai-nilai yang hilang. Bahety dalam jurnalnya [BHA-09] mencoba untuk mengevaluasi beberapa dataset yang mempunyai nilai

yang hilang pada atribut yang ada menggunakan cara yang berbeda - beda. Pertama-tama dia hanya menjalankan algoritma tanpa modifikasi algoritma. Ini berarti bahwa setiap nilai yang hilang ("?") akan dianggap sebagai nilai atribut baru. Hasil penelitian menunjukkan bahwa ID3 tampaknya memberikan hasil yang dapat diterima untuk dataset bahkan dengan atribut yang mempunyai nilai yang hilang. Hasil tergantung pada jenis dataset. Kumpulan data Voting memiliki beberapa nilai yang hilang tapi semua atribut adalah boolean yang membuat proses klasifikasi mudah. Pohon keputusan mencoba untuk lebih menggeneralisasi data dengan mempertimbangkan "?" Sebagai nilai baru karena setiap nilai yang tidak terlihat/hilang untuk atribut akan dimasukkan dalam kategori ini [BHA-09].

Penelitian [KBR-13] menggunakan Metode pembangkitan aturan dengan Fuzzy Iterative Dichotomiser 3 diimplementasikan untuk klasifikasi pada data gangguan hati. Teknik yang dilakukan untuk mengawali adalah dengan pembentukan himpunan fuzzy pada data latih, kemudian pembentukan tree dengan algoritma Iterative Dicotomiser 3 (ID3) dan menghasilkan aturan-aturan. Aturan yang telah terbentuk mengalami proses pengujian dengan menggunakan metode inferensi Mamdani. Hasil dari proses defuzzifikasi pada inferensi Mamdani inilah yang digunakan untuk menentukan kelas output klasifikasi. Hasil dari penelitian tersebut menunjukkan algoritma ID3 dapat digunakan untuk klasifikasi pada data gangguan hati dengan hasil yang cukup baik.

Penggunaan decision tree dengan algoritma ID3 juga pernah dilakukan untuk pengklasifikasian kanker payudara. Penelitian [IUM-14] Membangun sebuah sistem untuk klasifikasi kanker payudara dengan data yang sudah ada. Decision Tree terbentuk dengan algoritma ID-3 dari sekumpulan data latih yang di masukkan, selanjutnya di uji dengan data yang lain untuk mendapatkan akurasi. Hasil klasifikasi dari user input terbentuk dari tree yang sudah dibentuk oleh sistem. Terbentuknya decision tree berubah sesuai data training yang dimasukkan. Pada keakuratan sistem juga dapat berubah sesuai data latih dan data uji yang dimasukkan. Dari hasil uji coba menunjukkan bahwa terjadi peningkatan akurasi dengan 500 data latih. Kesimpulan hasil dari uji coba 1 menghasilkan akurasi yang 96.7%, 97.6% dan 94.51%. Pada percobaan kedua

menggunakan menghasilkan akurasi 98.6%, 98% dan 97,33%. Pada uji coba 3 menghasilkan akurasi 99.3%, 100% dan 100%. Ketika data latih diuji dengan datanya sendiri maka menghasilkan akurasi 100%.

Jenis data dari kedua jurnal pertama kategorikal, dan juga dari paper [BHA-09] yang menyebutkan bahwa algoritma ID3 mampu mengatasi data yang hilang atau *noise* maka dari itu penulis mencoba menerapkan metode pada jenis data yang sama yakni kategorikal namun dengan variasi class yang lebih banyak daripada kedua jurnal tersebut yakni 5 class serta menyertakan *noise* atau *missing data*. Penelitian lain juga menyebutkan bahwa hasil penerapan decision tree dengan algoritma ID3 dapat berjalan baik dan menghasilkan akurasi yang cukup memuaskan.

2.2 Keluarga Sejahtera

Keluarga sejahtera adalah keluarga yang dibentuk berdasarkan atas perkawinan yang sah, mampu memenuhi kebutuhan hidup spiritual dan materiil yang layak, bertakwa kepada Tuhan Yang Maha Esa, memiliki hubungan yang serasi, selaras, dan seimbang antar anggota dan antar keluarga dengan masyarakat dan lingkungan (Undang – Undang Republik Indonesia nomor 52 tahun 2009).

Tingkat kesejahteraan keluarga dikelompokkan menjadi 5 (lima) tahapan, yaitu :

1. Tahapan Keluarga Pra Sejahtera (KPS)

Yaitu keluarga yang tidak memenuhi salah satu dari 6 (enam) indikator Keluarga Sejahtera I (KS I) atau indikator “kebutuhan dasar keluarga” (basic needs).

2. Tahapan Keluarga Sejahtera I (KS I)

Yaitu keluarga mampu memenuhi 6 (enam) indikator tahapan KS I, tetapi tidak memenuhi salah satu dari 8 (delapan) indikator Keluarga Sejahtera II atau indikator ”kebutuhan psikologis” (psychological needs) keluarga.

3. Tahapan Keluarga Sejahtera II

Yaitu keluarga yang mampu memenuhi 6 (enam) indikator tahapan KS I dan 8 (delapan) indikator KS II, tetapi tidak memenuhi salah satu dari 5 (lima) indikator Keluarga Sejahtera III (KS III), atau indikator "kebutuhan pengembangan" (developmental needs) dari keluarga.

4. Tahapan Keluarga Sejahtera III

Yaitu keluarga yang mampu memenuhi 6 (enam) indikator tahapan KS I, 8 (delapan) indikator KS II, dan 5 (lima) indikator KS III, tetapi tidak memenuhi salah satu dari 2 (dua) indikator Keluarga Sejahtera III Plus (KS III Plus) atau indikator "aktualisasi diri" (self esteem) keluarga.

5. Tahapan Keluarga Sejahtera III Plus

Yaitu keluarga yang mampu memenuhi keseluruhan dari 6 (enam) indikator tahapan KS I, 8 (delapan) indikator KS II, 5 (lima) indikator KS III, serta 2 (dua) indikator tahapan KS III Plus.

2.3 Indikator Tahapan Keluarga Sejahtera

a. Enam Indikator tahapan Keluarga Sejahtera I (KS I) atau indikator "kebutuhan dasar keluarga" (basic needs), dari 21 indikator keluarga sejahtera yaitu:

1. Pada umumnya anggota keluarga makan dua kali sehari atau lebih.
2. Anggota keluarga memiliki pakaian yang berbeda untuk di rumah, bekerja/sekolah dan bepergian.
3. Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik.
4. Bila ada anggota keluarga sakit dibawa ke sarana kesehatan.
5. Bila pasangan usia subur ingin ber KB pergi ke sarana pelayanan kontrasepsi.
6. Semua anak umur 7-15 tahun dalam keluarga bersekolah.

b. Delapan indikator Keluarga Sejahtera II (KS II) atau indikator "kebutuhan psikologis" (psychological needs) keluarga, dari 21 indikator keluarga sejahtera yaitu:

1. Pada umumnya anggota keluarga melaksanakan ibadah sesuai dengan agama dan kepercayaan masing-masing.
2. Paling kurang sekali seminggu seluruh anggota keluarga makan daging/ikan/telur.
3. Seluruh anggota keluarga memperoleh paling kurang satu stel pakaian baru dalam setahun.
4. Luas lantai rumah paling kurang 8 m² untuk setiap penghuni rumah.
5. Tiga bulan terakhir keluarga dalam keadaan sehat sehingga dapat melaksanakan tugas/fungsi masing-masing.
6. Ada seorang atau lebih anggota keluarga yang bekerja untuk memperoleh penghasilan.
7. Seluruh anggota keluarga umur 10 - 60 tahun bisa baca tulisan latin.
8. Pasangan usia subur dengan anak dua atau lebih menggunakan alat/obat kontrasepsi.

c. Lima indikator Keluarga Sejahtera III (KS III) atau indikator "kebutuhan pengembangan" (developmental needs), dari 21 indikator keluarga sejahtera yaitu:

1. Keluarga berupaya meningkatkan pengetahuan agama.
2. Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang.
3. Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi.
4. Keluarga ikut dalam kegiatan masyarakat di lingkungan tempat tinggal.
5. Keluarga memperoleh informasi dari surat kabar/majalah/radio/tv/internet.

d. Dua indikator Keluarga Sejahtera III Plus (KS III Plus) atau indikator "aktualisasi diri" (self esteem) dari 21 indikator keluarga, yaitu:

1. Keluarga secara teratur dengan suka rela memberikan sumbangan materiil untuk kegiatan sosial.
2. Ada anggota keluarga yang aktif sebagai pengurus perkumpulan sosial/yayasan/ institusi masyarakat.

2.4 Decision Tree

Decision tree merupakan salah satu fungsional dari data mining yang menggunakan representasi tree untuk menentukan aturan-aturan klasifikasi. *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 klasifikasi yang lainnya, karena metode ini tidak membutuhkan pengetahuan yang lebih atau pengaturan parameter [HAN-01].

Decision Tree adalah sebuah struktur pohon, dimana setiap node pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan node daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level node teratas dari sebuah *decision tree* adalah node akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Salah satu algoritma *decision tree* yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi *rekursif*. Algoritma ID3 berusaha membangun *decision tree* secara *top-down*, mulai dengan pertanyaan : “atribut mana yang pertama kali harus dicek dan diletakkan pada *root*?” pertanyaan ini dijawab dengan mengevaluasi semua atribut yang ada dengan menggunakan suatu ukuran statistik (*information gain*) untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan kumpulan sampel data [WYD-09].

Proses pembangunan tree ini dilakukan dengan cara membagi data secara rekursif atau terus menggunakan proses yang sama, hingga tiap bagian terdiri dari beberapa sampel yang berasal dari kelas yang sama. Ketika sampel tidak dalam kelas yang sama maka dilakukan perhitungan *information gain*. Suatu atribut yang telah dipilih menjadi tes atribut maka atribut tersebut tidak diikutkan lagi dalam perhitungan *information gain* [TAN-06].

Pembangunan tree dilakukan dengan sebuah algoritma induksi *decision tree*. Rangka algoritma induksi *decision tree* disebut TreeGrowth. Cara kerja algoritma ini yaitu dengan memilih atribut terbaik dengan cara menghitung

information gain dari atribut – atribut yang ada untuk memisahkan data secara rekursif dan mengembangkan *node leaf* pada tree sampai ditemui kriteria untuk berhenti [KOH-99].

Sebuah dataset S terdiri dari n kasus dengan m masukan atribut X dan satu output (tergantung) atribut Y (numerik atau kategori). Tujuan dari algoritma pohon keputusan (*Decision Tree*) adalah untuk menemukan model pohon keputusan itu, berdasarkan nilai-nilai X , mengklasifikasikan kasus ke dalam kelas Y (model klasifikasi), atau memperkirakan nilai Y (model regresi). Sebuah model pohon keputusan terdiri dari node, cabang, dan daun. Daun menentukan prediksi Y , dan aturan yang dapat diperoleh pada pohon keputusan (*Decision Tree*) secara menurun dari akar ke daun. Jumlah daun mendefinisikan jumlah aturan yang dapat diekstraksi dari pohon keputusan. Pohon keputusan (*Decision Tree*) cocok digunakan untuk mendapatkan aturan IF-THEN dalam struktur seperti pohon hirarkis [SUK-11].

2.5 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 klasifikasi dan menentukan klasifikasi data yang baru [LIG-05]. Algoritma ID3 membutuhkan data yang telah memiliki class untuk pembelajaran sebelum melakukan proses klasifikasi. Kurangnya data pembelajaran dapat berakibat kurang optimalnya klasifikasi yang dihasilkan dari algoritma ini.

Algoritma ID3 adalah sebagai berikut [MIT-97].

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 :
 - $A \leftarrow$ atribut yang mengklasifikasikan 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 kumpulan sampel (v_i) sebagai subset dari atribut yang mempunyai nilai
 - Jika kumpulan sampel (v_i) kosong :
 - Dibawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada kumpulan sampel
 - Jika tidak, tambahkan cabang baru dibawah cabang yang sekarang ID3 (kumpulan sampel (v_i), atribut target, atribut-[A])

2.6 Entropy dan Information Gain

Ukuran *information gain* digunakan untuk memilih tes atribut pada setiap simpul dalam *tree*. Atribut dengan informasi tertinggi (nilai pengurangan *entropy* yang terbesar) dipilih sebagai tes atribut untuk simpul tersebut. Atribut ini meminimalkan informasi yang dibutuhkan untuk mengklasifikasikan contoh pada proses pembagian dan mencerminkan ketidakmurnian (*impurity*) [KOH-99].

Persamaan perhitungan entropy adalah persamaan yang dijabarkan oleh Shannon pada tahun 1948 [SUK-11], yakni sebagai berikut :

$$E(S) = - \sum_{j=1}^n f_s(j) \log_2 f_s(j) \quad (2.1)$$

Dimana :

- E(S) adalah entropy dari himpunan S
 - n adalah jumlah nilai yang berbeda dari atribut di S
 - $f_s(j)$ adalah frekuensi dari nilai j dalam himpunan S
- entropy 0 menandakan satu set telah diklasifikasi dengan sempurna.

Perhitungan *information gain* kemudian dilakukan setelah nilai *entropy* didapat dengan menggunakan persamaan :

$$G(S, A) = E(S) - \sum_{i=1}^m f_s(A_i) E(S_{A_i}) \quad (2.2)$$

Dimana :

- G(S, A) adalah nilai gain dari atribut A pada himpunan S
- m adalah jumlah nilai – nilai yang berbeda dari atribut A di S
- $f_s(A_i)$ frekuensi dari item yang memiliki A_i sebagai nilai dari A di S
- A_i adalah i^{th} nilai yang mungkin dari A
- S_{A_i} adalah bagian dari S yang berisi semua item dimana nilai A adalah A_i

Algoritma *ID3* dalam proses pembentukan treenya dimulai dengan mencari *best attribute*, yakni atribut yang memiliki nilai terbaik untuk memisahkan data ke dalam classnya. Penghitungan *best attribute* itu dimulai dengan menghitung *entropy* dan dilanjutkan dengan perhitungan *information gain*. Atribut dengan *information gain* yang terbesar lah yang merupakan *best attribute* yang kemudian dijadikan simpul seperti kutipan diatas.

2.7 Precision, Recall, F-Measure dan Akurasi

Precision adalah jumlah dari sampel data yang benar – benar merupakan class x dari semua sampel data yang diklasifikasikan sebagai class x. Perhitungan *Precision* dapat dilakukan menggunakan persamaan 2.3 [NOC-12].

$$Precision = \frac{TP}{(TP+FP)} \quad (2.3)$$

Recall adalah ukuran kemampuan model prediksi untuk memilih contoh dari kelas tertentu dari satu set data. Hal ini biasa juga disebut sensitivitas, dan sesuai dengan tingkat positif benar. Hal ini didefinisikan oleh persamaan 2.4 [NOC-12].

$$Recall = \frac{TP}{(TP+FN)} \quad (2.4)$$

F-Measure adalah ukuran gabungan untuk presisi dan recall dan dihitung dengan persamaan 2.5 [NOC-12].

$$F - Measure = \frac{2*Precision*Recall}{(Precision+Recall)} \quad (2.5)$$

Akurasi adalah nilai derajat kedekatan dari pengukuran kuantitas untuk nilai sebenarnya (*true*). Nilai akurasi didapatkan dari hasil *rule* yang dihasilkan dari perhitungan *decision tree* kemudian di uji coba kan pada data testing dan menghasilkan derajat keakuratan dari *rule* tersebut setelah di uji coba kan pada data testing. Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan 2.6 [KBR-13].

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Prediksi} \times 100\% \quad (2.6)$$

BAB III

METODOLOGI PENELITIAN dan PERANCANGAN

3.1 Metodologi

Pada bab ini akan dibahas algoritma dan langkah-langkah yang digunakan untuk melakukan penerapan algoritma *iterative dichotomiser 3 (ID3)* untuk klasifikasi tahapan keluarga sejahtera.

Penelitian dilakukan dengan langkah-langkah sebagai berikut:

1. Melakukan studi literatur tentang keluarga sejahtera, decision tree, dan algoritma *ID3*.
2. Pengumpulan data untuk proses klasifikasi status tahapan keluarga sejahtera dari hasil pendataan keluarga tahun 2013 yang didapatkan dari Posyandu Kecamatan Bumi Aji, Kota Batu.
3. Perancangan sistem decision tree dengan algoritma *ID3* pada data status tahapan keluarga sejahtera.
4. Implementasi sistem berdasarkan perancangan dan analisi yang telah dilakukan tentang decision tree menggunakan algoritma *ID3*.
5. Pengujian terhadap sistem yang telah diimplementasikan.
6. Melakukan evaluasi (analisis hasil) yang diperoleh dari uji coba sistem dan menarik kesimpulan dari hasil uji coba yang dilakukan.

3.2 Perancangan Sistem

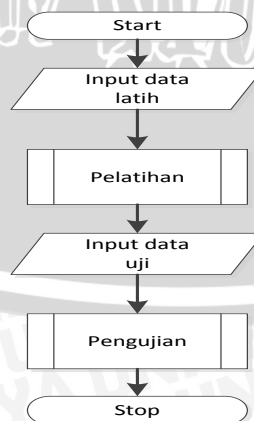
3.2.1 Analisa Data

Data yang digunakan dalam skripsi ini didapatkan dari Posyandu Kecamatan Bumi Aji, Kota Batu. Data yang diambil adalah data indikator dan status TKS pada data hasil pemutakhiran keluarga tahun 2013 di Desa/Kelurahan Tulungrejo, RT. 01, 02, 03 - RW. 04 dan RT. 01, 02, 03, 04, 05 - RW. 05, Kecamatan Bumi Aji, Kota Batu. Contoh data ditampilkan pada tabel 3.1.

Data tersebut adalah data indikator yang menentukan tahapan keluarga sejahtera (TKS) tiap keluarga. Indikator yang digunakan ada 21 dengan 5 macam status TKS. Nilai yang terdapat pada tiap indikator berbeda – beda, terdapat 3 macam nilai dari 21 indikator yang ada yakni nilai ya yang dilambangkan dengan tanda (v), nilai tidak yang dilambangkan dengan tanda (x), dan nilai kosong/*missing values* yang dilambangkan dengan tanda ().

3.2.2 Analisa Proses

Pada bagian ini dijelaskan rancangan proses serta gambaran umum bagaimana proses kerja sistem membentuk suatu decision tree dan pengklasifikasian menggunakan algoritma *ID3*. Inputan sistem dapat menggunakan 2 macam jenis data, yakni text dan excel, sedangkan untuk hasil dari sistem ini menggunakan database untuk penyimpanan datanya. Proses pertama nantinya ialah pembacaan data training yang bisa diinputkan menggunakan text atau excel, setelah data training masuk maka dilanjutkan dengan memasukkan data uji. Hasil dari pemrosesan data uji akan disimpan didatabase. Decision tree yang terbentuk dari proses pengolahan data kemudian akan diekstrak rulenya. Rule yang didapatkan dari decision tree yang terbentuk kemudian diterapkan ke data untuk mengklasifikasikan status TKS secara otomatis. Penjelasan gambaran umum proses yang dimulai dari pembacaan data hingga proses akhir akan digambarkan pada gambar 3.1.

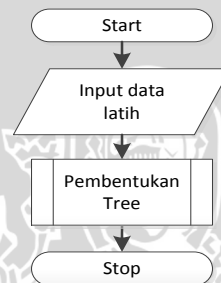


Gambar 3.1 Flowchart Sistem

Gambaran umum proses-proses utama yang akan dilakukan oleh sistem dalam penelitian ini dijelaskan pada Gambar 3.1. Detail dari masing-masing proses pada Gambar 3.1 akan dijelaskan lebih lanjut pada sub bab ini mulai dari proses perhitungan *entropy* hingga proses mendapatkan rule.

3.2.2.1 Pelatihan

Proses pelatihan ini adalah proses dimana tree nantinya dibentuk dari inputan data latih yang dimasukkan user. Tree nantinya digunakan untuk proses klasifikasi pada tahap pengujian. Flowchart proses pelatihan ditunjukkan pada gambar 3.2.



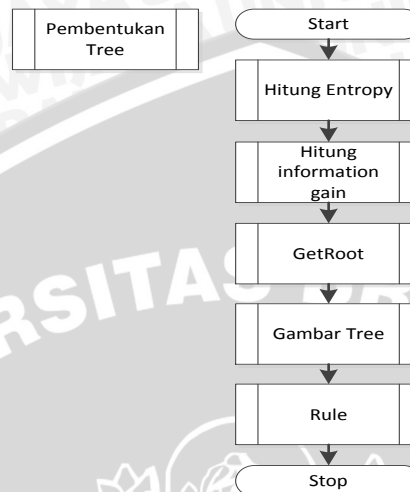
Gambar 3.2 Flowchart Pelatihan

3.2.2.2 Pembentukan Tree

Tahap pembentukan tree dibagi menjadi lima proses utama, yakni menghitung entropy total, kemudian menghitung entropy dari nilai yang ada dalam atribut dan menghitung entropy atribut tersebut. Entropy itu nantinya digunakan untuk menghitung nilai information gain, lalu GetRoot yakni proses pemilihan suatu atribut menjadi node, dan yang terakhir Gambar Tree dimana node yang terpilih dari GetRoot dibentuk menjadi tree. Proses terakhir dari tahap pelatihan adalah proses mendapatkan Rule dari tree yang telah terbentuk.

Penanganan missing value didasarkan dari penelitian yang telah dilakukan [BHA-09] yakni dengan menjadikan missing value suatu nilai baru yang berarti suatu cabang baru pada pohonnya nanti. Penerapan yang dilakukan pada penelitian ini atribut ya (v) diganti dengan nilai 1, atribut tidak (x) diganti dengan nilai 0, dan atribut missing value diganti dengan nilai -1. Penginisialisasian nilai

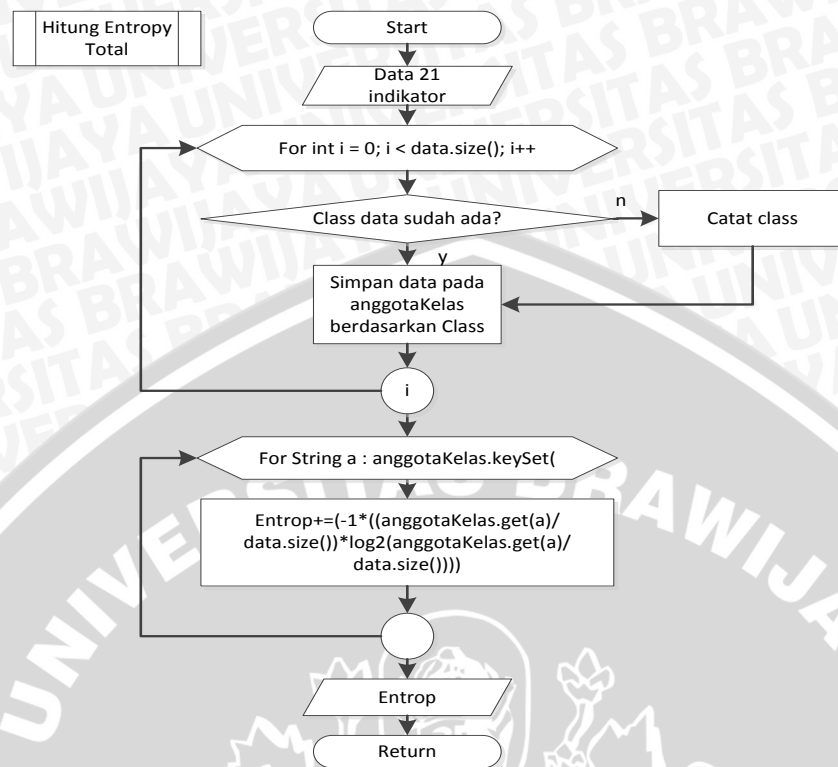
dengan angka 1, 0, dan -1 pada tiap atribut ini dilakukan hanya agar memudahkan dalam penerapan proses perulangan pada aplikasi. Flowchart proses pembentukan tree ditunjukkan pada gambar 3.3.



Gambar 3.3 Flowchart Pembentukan Tree

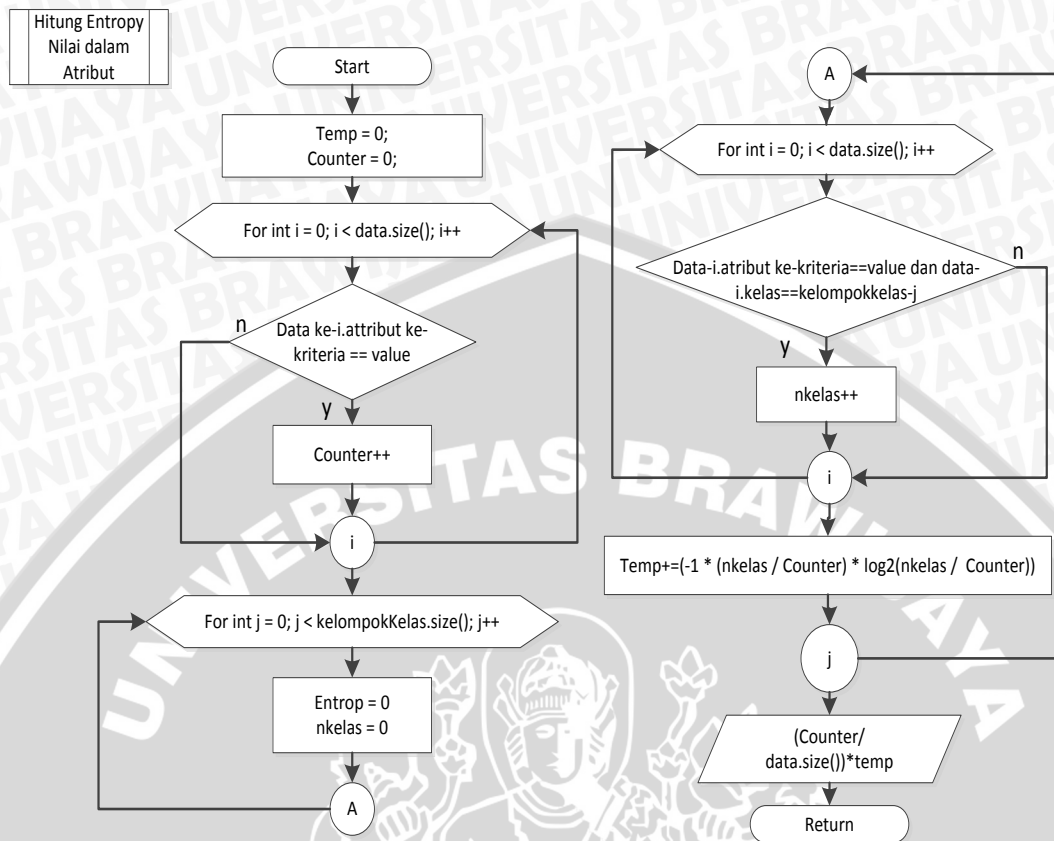
3.2.2.3 Hitung *Entropy*

Proses Hitung *Entropy* terdiri dari 3 macam perhitungan, yakni Hitung *Entropy* Total, Hitung *Entropy* Nilai dalam Atribut, dan yang terakhir Hitung *Entropy* Atribut. Flowchart perhitungan *Entropy* Total, *Entropy* Nilai dalam Atribut, dan *Entropy* Atribut ditunjukkan pada gambar 3.4, 3.5, dan 3.6



Gambar 3.4 Flowchart Perhitungan *Entropy* total

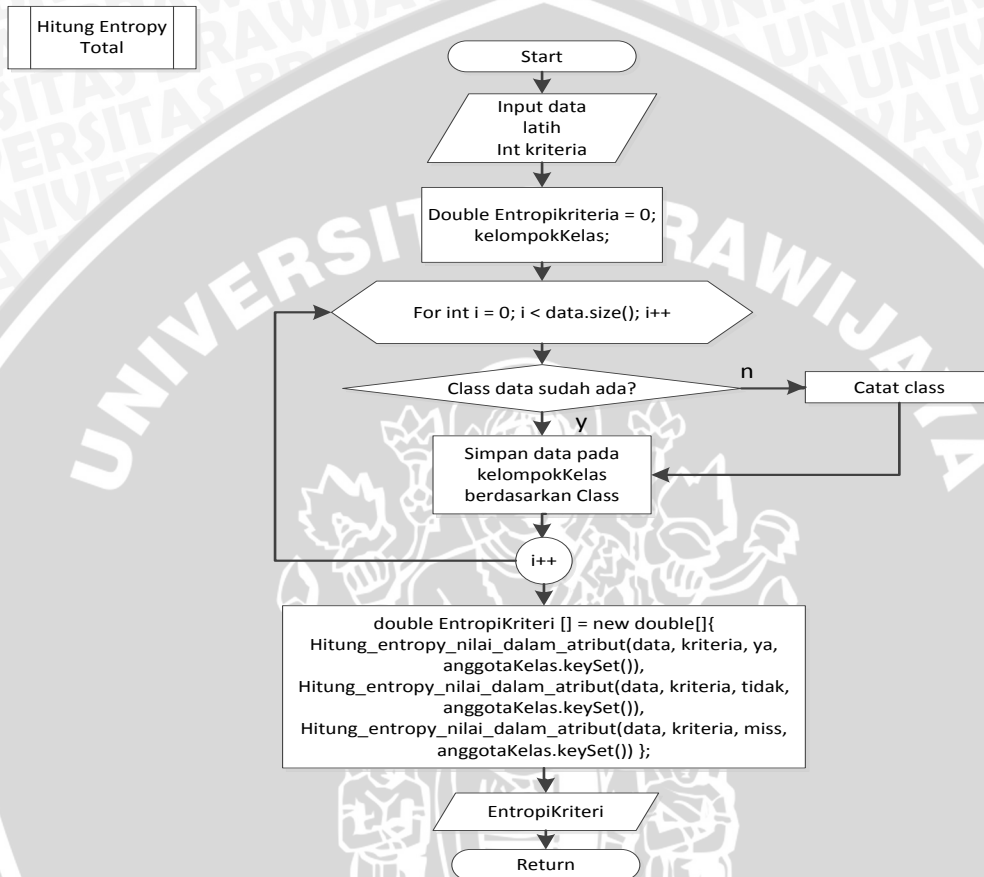
Proses Hitung *Entropy* Total diawali dengan mengecek class dari data yang dimasukkan, jika class telah tercatat maka simpan data tersebut ke dalam classnya, sedangkan jika class belum tercatat maka buat class baru kemudian masukkan data tersebut ke classnya, proses tersebut diulang sebanyak jumlah data. Setelah semua data tercatat ke dalam classnya masing – masing kemudian data di masukkan ke dalam persamaan 2.1 dan proses tersebut dilakukan sebanyak jumlah class yang telah tercatat. Hasil akhir dari proses ini adalah hasil penjumlahan dari proses perhitungan persamaan 2.1 yang diulang sebanyak jumlah class.



Gambar 3.5 Flowchart Perhitungan *Entropy* Nilai dalam Atribut

Proses selanjutnya setelah didapatkan nilai *entropy* total adalah menghitung *entropy* nilai dalam atribut. Nilai dalam atribut yang dimaksud disini adalah nilai ya (v), tidak (x), dan missing value. Suatu atribut bisa memiliki 1, 2 atau 3 macam nilai, dan untuk mendapatkan *entropy* atribut perlu dilakukan perhitungan *entropy* dari nilai yang ada dalam atribut tersebut. Pertama dibuat 2 variable dengan nilai awal nol (0) yakni counter dan temp, kemudian dilakukan pengecekan apakah nilai dari data ke-i atribut ke-kriteria (terdapat 21 atribut) ialah value (ya, tidak, miss), jika ya maka jumlah counter bertambah satu, pengecekan tersebut dilakukan sejumlah banyak data. Proses selanjutnya didefinisikan variable nkelas bernilai nol, lalu dilakukan proses pengecekan apa nilai dari data ke-i atribut ke-kriteria dan apa class dari data tersebut adalah class ke-j, jika ya maka nkelas bertambah satu pengecekan tersebut dilakukan sebanyak jumlah data. Nilai nkelas dan counter yang telah didapat kemudian dimasukkan ke dalam persamaan 2.1 dan didefinisikan sebagai variable temp. Proses diatas

dilakukan sebanyak jumlah class, sehingga nilai temp bertambah tiap kali proses perulangan, setelah perulangan selesai hasil akhir adalah nilai counter dibagi jumlah data dikalikan temp. Nilai ini nantinya digunakan di proses menghitung *entropy* atribut.

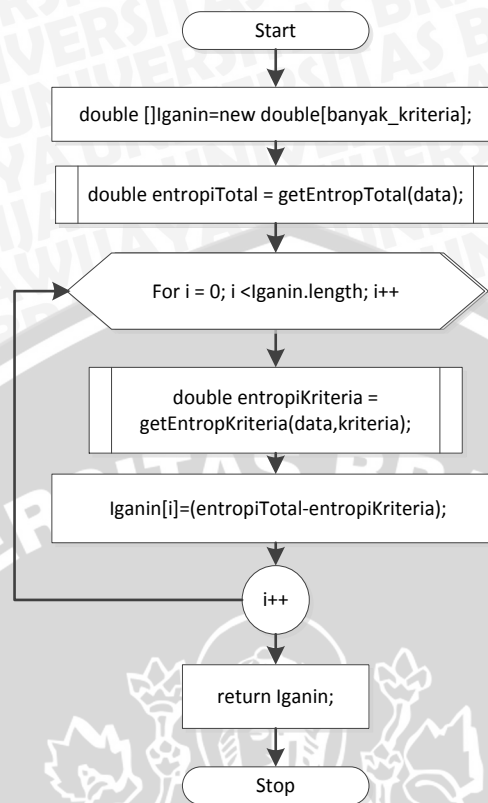


Gambar 3.6 Flowchart Perhitungan *Entropy* Atribut

Hitung entropy atribut memanggil method hitung *entropy* nilai dalam atribut berdasarkan nilai yang ada.

3.2.2.4 Perhitungan information gain

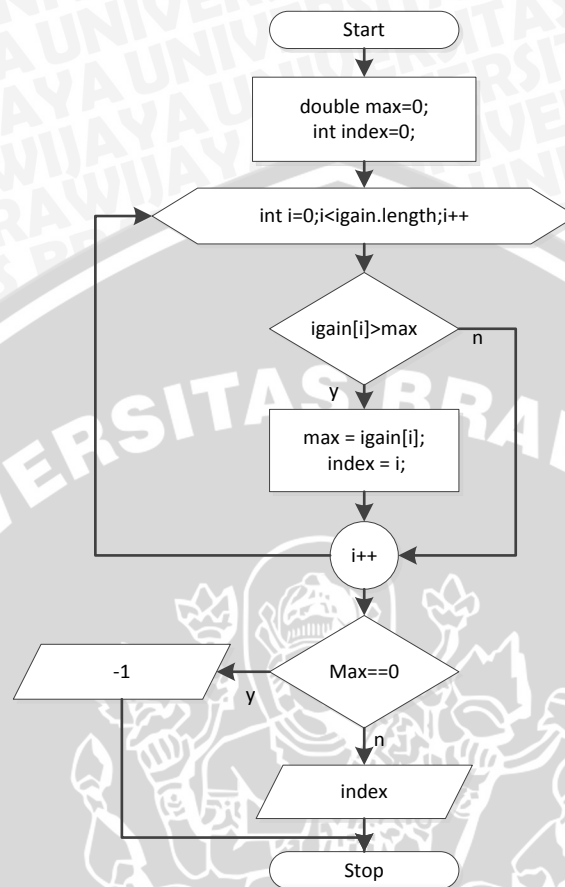
Penentuan node dalam pembentukan tree pada algoritma *ID3* menggunakan perhitungan yang dinamakan information gain. Hasil perhitungan *entropy* pada proses sebelumnya dimasukkan ke dalam persamaan 2.2.



Gambar 3.7 Flowchart Perhitungan Information Gain

Perhitungan information gain dilakukan sebanyak jumlah atribut, memakai nilai *entropy* total yang dikurangkan *entropy* atribut, hasil ini nantinya digunakan untuk proses GetRoot.

3.2.2.5 GetRoot



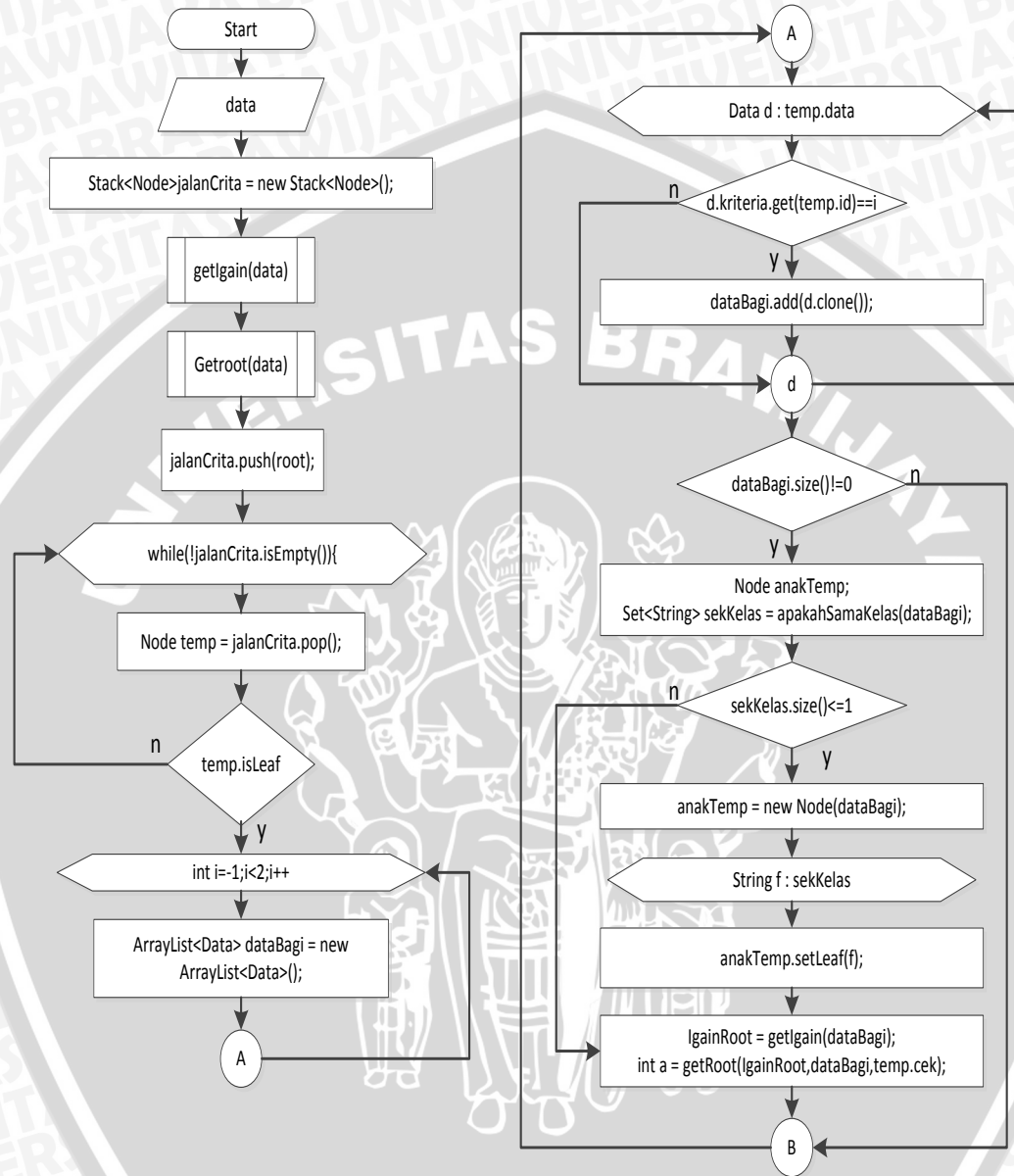
Gambar 3.8 Flowchart GetRoot

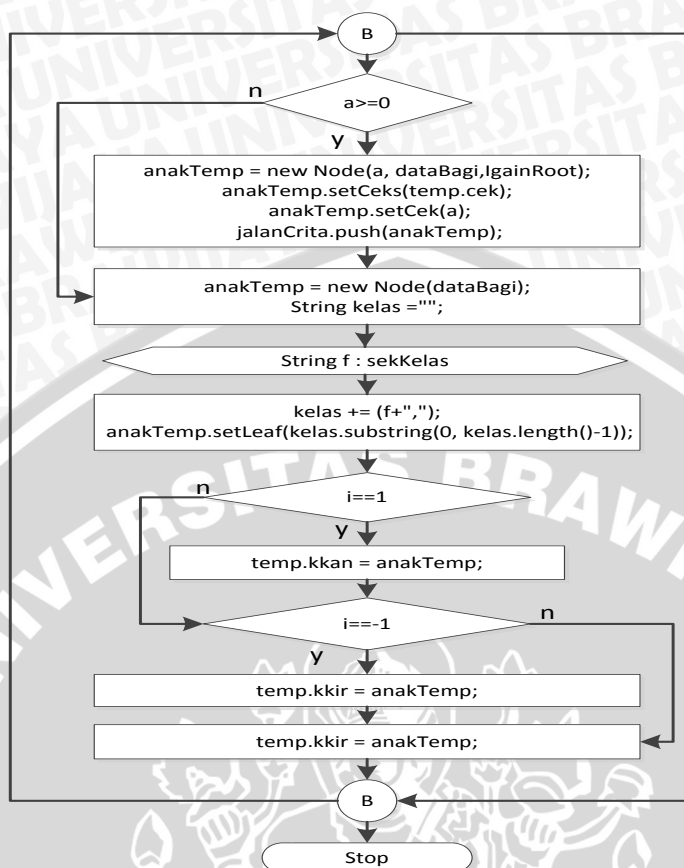
Method GetRoot ini digunakan untuk mengambil nilai information gain tertinggi. Di inialisasikan variabel max dan index dengan nilai awal nol, kemudian dilakukan pengecekan apakah igain pada index i lebih dari max, jika ya maka nilai max diganti dengan nilai igain dan index di isi dengan nilai index dari igain tersebut, pengecekan dilakukan sebanyak jumlah igain. Jika nilai max adalah nol maka mengembalikan nilai -1, dan jika tidak lebih dari 0 maka nilai yang dikembalikan dari method ini adalah index yang telah terisi tadi.

3.2.2.6 Gambar Tree

Gambar Tree adalah proses dimana tree mulai dibentuk. Mengambil nilai information gain yang telah diproses sebelumnya dan menggunakan method GetRoot didapat atribut yang terpilih sebagai node. Node tersebut disimpan didalam Stack. Node yang ada dalam stack dikeluarkan, dan dicek apakah node tersebut adalah leaf, jika tidak maka berlanjut ke proses selanjutnya. Membuat variable databagi, databagi ini adalah data yang akan diproses, untuk membentuk tree. Databagi di cek apakah jumlahnya lebih dari nol, jika ya maka databagi di cek lagi menggunakan method apakahSamaKelas, jika nilai dari apakahSamaKelas kurang dari sama dengan 1 maka Node tersebut diset sebagai leaf. Jika nilai dari apakahSamaKelas lebih dari 1 maka menghitung nilai information gain tertinggi lagi dari databagi dan mencari atribut yang terpilih dengan information gain tertinggi menggunakan GetRoot dan di simpan dalam variabel a. Jika a lebih dari 0 maka node tersebut dimasukkan ke dalam stack jalanCrita. Jika a sama dengan 0 berarti dalam perhitungan tersebut tidak ada nilai information gain yang lebih dari 0 namun masih terdapat lebih dari satu class. Jika nilai i adalah 1 maka node tersebut ditaruh pada anak kanan, atau anak yang bernilai ya. Jika i adalah -1 maka node tersebut ditaruh pada anak kiri, atau anak yang bernilai tidak, dan jika i tidak 1 ataupun -1 maka node tersebut ditaruh pada anak tengah, atau anak yang bernilai missing value. Flowchart proses Gambar Tree ditunjukkan pada gambar 3.9.

Gambar Tree





Gambar 3.9 Flowchart Gambar Tree

3.2.2.7 Rule

Rule didapat dari node tree yang berstatus sebagai leaf, dari leaf – leaf tersebut diambil aturan dengan menelusuri dari node paling atas (root) hingga ke leaf. Aturan yang didapat tersebut kemudian diterapkan ke data uji untuk mendapatkan class prediksi berdasarkan pohon yang terbentuk.

3.3 Perhitungan Manual

Dalam menjelaskan proses pembentukan decision tree dengan menggunakan algoritma *ID3* diberikan contoh perhitungan manual untuk gambaran awal kerja sistem nanti dalam melakukan pembentukan tree serta dapat digunakan untuk pengecekan hasil akhir memastikan apakah sistem telah bekerja sesuai dengan algoritma yang benar. Berikut contoh perhitungan manual dengan menggunakan sebagian sampel data indikator dan status TKS yang didapat dari Posyandu Kecamatan Bumiaji, Kota Batu.

Dari 10 data tersebut terdapat 21 indikator yakni IND-1, IND-2, hingga IND-21 yang nantinya menjadi atribut atau node dalam tree serta 5 macam status TKS yang dijadikan sebagai class pada penelitian ini, yakni class KPS, KS I, KS II, KS III, KS III+. Nilai pada tiap atribut seperti yang telah dijelaskan pada analisis data nantinya akan menjadi cabang dari atribut yang menjadi node.

Tahap pertama dalam proses pengklasifikasian menggunakan ID3 adalah menghitung nilai *entropy* dari tiap atribut.

Perhitungan *Entropy* sampel data (S):

$$E(S) = \left(-\frac{1}{10} \log_2 \left(\frac{1}{10}\right)\right) + \left(-\frac{1}{10} \log_2 \left(\frac{1}{10}\right)\right) + \left(-\frac{2}{10} \log_2 \left(\frac{2}{10}\right)\right) + \left(-\frac{4}{10} \log_2 \left(\frac{4}{10}\right)\right) + \left(-\frac{2}{10} \log_2 \left(\frac{2}{10}\right)\right)$$

$$E(S) = 0,332 + 0,332 + 0,464 + 0,528 + 0,464 \\ = 2,12$$

Perhitungan *Entropy* tiap atribut :

$$E(S, \text{IND-1ya}) = \left(-\frac{1}{10} \log_2 \left(\frac{1}{10}\right)\right) + \left(-\frac{1}{10} \log_2 \left(\frac{1}{10}\right)\right) + \left(-\frac{2}{10} \log_2 \left(\frac{2}{10}\right)\right) + \left(-\frac{4}{10} \log_2 \left(\frac{4}{10}\right)\right) + \left(-\frac{2}{10} \log_2 \left(\frac{2}{10}\right)\right)$$

$$E(S, \text{IND-1ya}) = 0,332 + 0,332 + 0,464 + 0,528 + 0,464 \\ = 2,12$$

Nilai *Entropy* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.3 berikut.

Tabel 3.3 Nilai *entropy* atribut perhitungan manual tahap I

Atribut	<i>Entropy</i>		
	Ya	Tidak	Kosong
IND-2	2,12	-	-
IND-3	1,837	0	-
IND-4	2,12	-	-
IND-5	2,059	-	0
IND-6	2,25	-	0,811
IND-7	2,12	-	-
IND-8	2,12	-	-

IND-9	2,12	-	-
IND-10	1,837	0	-
IND-11	2,12	-	-
IND-12	2,12	-	-
IND-13	2,12	-	-
IND-14	2,196	-	0
IND-15	2,12	-	-
IND-16	1,5	1	-
IND-17	1,75	0	-
IND-18	2,12	-	-
IND-19	2,12	-	-
IND-20	0	1,75	-
IND-21	0	1,75	-

Tahap selanjutnya setelah perhitungan *entropy* adalah perhitungan *information gain* yang nantinya akan digunakan untuk memilih atribut yang dijadikan node.

Perhitungan *Information gain*:

$$IG(S, \text{IND-1}) = 2,12 - (10/10)2,12 = 0$$

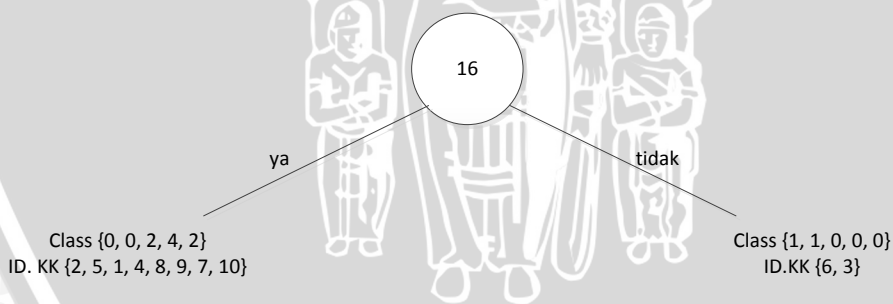
Nilai *Information gain* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.4 berikut.

Tabel 3.4 Nilai *information gain* perhitungan manual tahap I

Attribute	<i>Information gain</i>
IND-2	0
IND-3	0,467
IND-4	0
IND-5	0,267
IND-6	0,446
IND-7	0
IND-8	0

IND-9	0
IND-10	0,467
IND-11	0
IND-12	0
IND-13	0
IND-14	0,144
IND-15	0
IND-16	0,72
IND-17	0,72
IND-18	0
IND-19	0
IND-20	0,72
IND-21	0,72

Berdasarkan perhitungan nilai *information gain* diatas terdapat 4 nilai yang sama tingginya yakni 0,72 pada atribut IND-16, IND-17, IND-20, serta IND-21, maka yang akan diambil menjadi root adalah atribut IND-16.



Gambar 3.10 Tree hasil perhitungan manual tahap I.

Keterangan Class :

KPS = Angka pertama anggota Class KS III = Angka keempat anggota Class

KS I = Angka kedua anggota Class KS III+ = Angka kelima anggota Class

KS II = Angka ketiga anggota Class

Tree diatas menunjukkan pada cabang ya dari 5 class yang ada, cabang tersebut mempunyai 3 class yang tersebar pada data ID. KK 2, 5, 1, 4, 8, 9, 7 10, pada dan pada cabang tidak dari 5 class terdapat 2 class pada cabang tersebut yang tersebar pada data ID. KK 6, 3. Tree selanjutnya akan dikembangkan dari cabang ya dan tidak, karena pada cabang tersebut masih terdapat lebih dari 1 class (nilai information gainnya masih lebih besar dari 0). Pembentukan node selanjutnya dilakukan dengan 2 tahap, pertama pembentukan node selanjutnya dari cabang ya kemudian pembentukan node selanjutnya dari cabang tidak. Perhitungan dilakukan seperti diatas namun hanya menggunakan data yang ada pada cabang yang bersangkutan, serta tidak menyertakan atribut IND-16 karena atribut tersebut telah terpilih sebagai node dalam hal ini menjadi root (node awal) dalam tree. Attribute yang telah terpilih sebagai node tidak dapat dipilih sebagai node lagi dalam satu cabang yang sama.



Perhitungan *Entropy* :

$$E(S) = (-2/8) \log_2 (2/8) + (-4/8) \log_2 (4/8) + (-2/8) \log_2 (2/8) \\ = 0,5 + 0,5 + 0,5 = 1,5$$

$$E(S, \text{IND-1ya}) = (-2/8) \log_2 (2/8) + (-4/8) \log_2 (4/8) + (-2/8) \log_2 (2/8)$$

$$E(S, \text{IND-1ya}) = 0,5 + 0,5 + 0,5 \\ = 1,5$$

Nilai *Entropy* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.6 berikut.

Tabel 3.6 Nilai *entropy* perhitungan manual tahap II cabang ya

Atribut	<i>Entropy</i>		
	Ya	Tidak	Kosong
IND-2	1,5	-	-
IND-3	1,5	-	-
IND-4	1,5	-	-
IND-5	1,378	-	0
IND-6	1,5	-	0,811
IND-7	1,5	-	-
IND-8	1,5	-	-
IND-9	1,5	-	-
IND-10	1,5	-	-
IND-11	1,5	-	-
IND-12	1,5	-	-
IND-13	1,5	-	-
IND-14	1,556	-	0
IND-15	1,5	-	-
IND-17	0,919	0	-
IND-18	1,5	-	-
IND-19	1,5	-	-
IND-20	0	0,919	-
IND-21	0	0,919	-

Perhitungan *Information gain*:

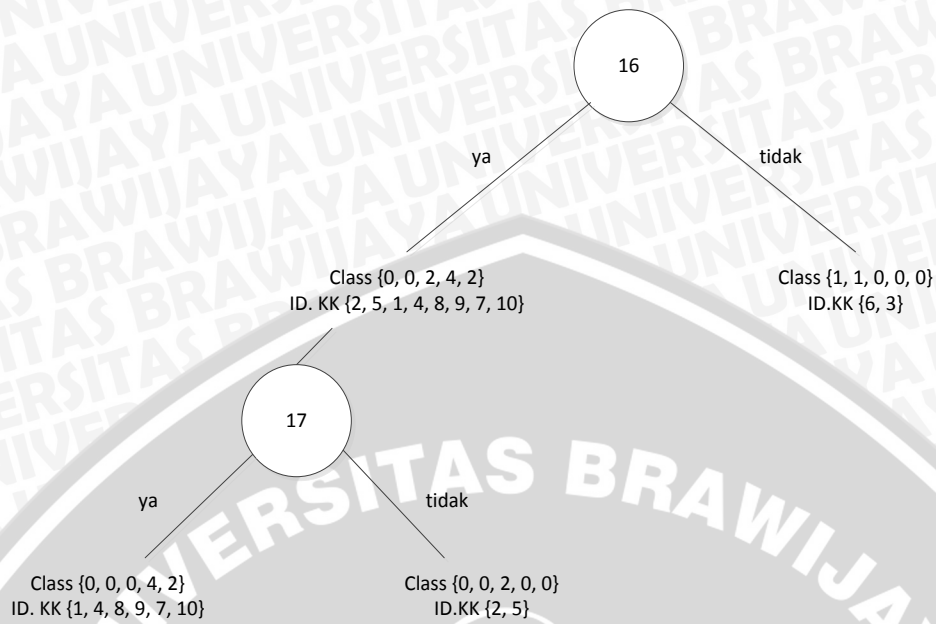
$$IG(S, IND-1) = 1,5 - (8/8)1,5 = 1,5 - 1,5 = 0$$

Nilai *Information gain* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.7 berikut.

Tabel 3.7 Nilai *information gain* perhitungan manual tahap II cabang ya

Atribute	<i>Information gain</i>
IND-2	0
IND-3	0
IND-4	0
IND-5	0,294
IND-6	0,344
IND-7	0
IND-8	0
IND-9	0
IND-10	0
IND-11	0
IND-12	0
IND-13	0
IND-14	0,138
IND-15	0
IND-17	0,811
IND-18	0
IND-19	0
IND-20	0,811
IND-21	0,811

Berdasarkan perhitungan nilai *information gain* diatas terdapat 3 nilai yang sama tingginya yakni 0,811 pada atribut IND-17, IND-20, serta IND-21, maka yang akan diambil menjadi node selanjutnya adalah atribut IND-17.



Gambar 3.11 Tree hasil perhitungan manual tahap II cabang ya.

Tree diatas menunjukkan pada cabang ya di node 17 dari 5 class yang ada, cabang tersebut mempunyai 2 class yang tersebar pada data ID. KK 1, 4, 8, 9, 7, 10, dan pada cabang tidak dari 5 class cabang tersebut hanya memiliki 1 class. Tree selanjutnya akan dikembangkan dari cabang ya karena pada cabang tersebut masih terdapat lebih dari 1 class (nilai information gainnya masih lebih besar dari 0), sedangkan cabang tidak sudah selesai, karena data yang terdapat pada cabang tersebut hanya mempunyai satu class saja (nilai information gainnya tidak lebih besar dari 0). Pembentukan node selanjutnya dilakukan dengan perhitungan seperti diatas namun hanya menggunakan data yang ada pada cabang ya, serta tidak menyertakan atribut IND-16 dan IND-17 karena atribut tersebut telah terpilih sebagai node dalam satu cabang yang sama.

Tahap II (cabang tidak)

Data yang akan digunakan pada perhitungan pembentukan node cabang ya ini adalah data awal pada tabel 3.2 yang telah melalui pemrosesan pada tahap I :

Tabel 3.8 Data indikator dan status tks perhitungan manual tahap II cabang tidak

ID. KK	ID. ISTKS	INDIKATOR																				STATUS TKS
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17	18	19	20	21	
6	6	v	v	x	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x	x	KPS
3	3	v	v	v	v	v	v	v	v	v	x	v	v	v	v	v	v	v	v	x	x	KSI

Perhitungan *Entropy* :

$$\begin{aligned} E(S) &= (-(1/2) \log_2 (1/2)) + (-(1/2) \log_2 (1/2)) \\ &= 0,5 + 0,5 = 1 \end{aligned}$$

$$\begin{aligned} E(S, \text{IND-1ya}) &= (-(1/2) \log_2 (1/2)) + (-(1/2) \log_2 (1/2)) \\ &= 0,5 + 0,5 = 1 \end{aligned}$$

Nilai *Entropy* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.9 berikut.

Tabel 3.9 Nilai *entropy* perhitungan manual tahap II cabang tidak

Atribut	<i>Entropy</i>		
	Ya	Tidak	Kosong
IND-2	1	-	-
IND-3	0	0	-
IND-4	1	-	-
IND-5	1	-	-
IND-6	1	-	-
IND-7	1	-	-
IND-8	1	-	-
IND-9	1	-	-
IND-10	0	0	-
IND-11	1	-	-
IND-12	1	-	-
IND-13	1	-	-
IND-14	1	-	-
IND-15	1	-	-
IND-17	1	-	-
IND-18	1	-	-
IND-19	1	-	-
IND-20	-	1	-
IND-21	-	1	-

Perhitungan *Information gain*:

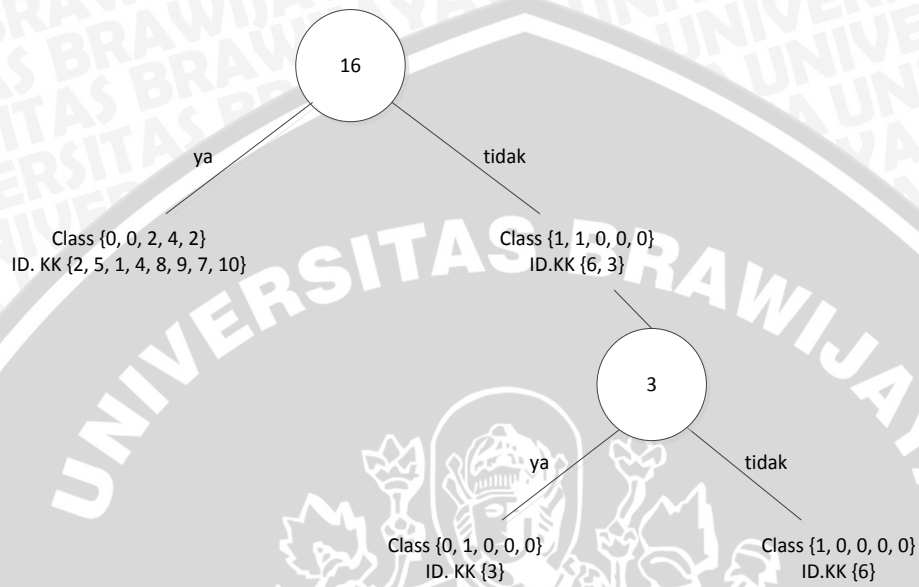
$$IG(S, IND-1) = 1 - (2/2)1 = 1 - 1 = 0$$

Nilai *Information gain* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.10 berikut.

Tabel 3.10 Nilai *information gain* perhitungan manual tahap II cabang tidak

Atribute	<i>Information gain</i>
IND-2	0
IND-3	1
IND-4	0
IND-5	0
IND-6	0
IND-7	0
IND-8	0
IND-9	0
IND-10	1
IND-11	0
IND-12	0
IND-13	0
IND-14	0
IND-15	0
IND-17	0
IND-18	0
IND-19	0
IND-20	0
IND-21	0

Berdasarkan perhitungan nilai *information gain* diatas terdapat 2 nilai yang sama tingginya yakni 1 pada atribut IND-3, serta IND-10, maka yang akan diambil menjadi node selanjutnya adalah atribut IND-3.



Gambar 3.12 Tree hasil perhitungan manual tahap II cabang tidak

Tree diatas menunjukkan pada cabang ya di node 3 dari 5 class yang ada, cabang tersebut mempunyai 1 class yang tersebar pada data ID. KK 3, dan pada cabang tidak dari 5 class cabang tersebut juga hanya memiliki 1 class. Tahap II pada cabang tidak ini dinyatakan telah selesai karena data yang ada pada cabang sudah tersebar ke masing – masing 1 class saja (nilai *information gain*nya tidak lebih besar dari 0). Tree selanjutnya akan dikembangkan dari cabang ya hasil pembentukan tree pada tahap II (cabang ya) yang telah dibahas sebelumnya.

Perhitungan *Entropy* :

$$\begin{aligned} E(S) &= -(4/6) \log_2 (4/6) + -(2/6) \log_2 (2/6) \\ &= 0,391 + 0,528 = 0,919 \end{aligned}$$

$$\begin{aligned} E(S, \text{IND-1yes}) &= -(4/6) \log_2 (4/6) + -(2/6) \log_2 (2/6) \\ &= 0,391 + 0,528 \\ &= 0,919 \end{aligned}$$

Nilai *Entropy* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.12 berikut.

Tabel 3.12 Nilai *entropy* perhitungan manual tahap III

Atribut	<i>Entropy</i>		
	Ya	Tidak	Kosong
IND-2	0,919	-	-
IND-3	0,919	-	-
IND-4	0,919	-	-
IND-5	0,919	-	-
IND-6	0,919	-	0
IND-7	0,919	-	-
IND-8	0,919	-	-
IND-9	0,919	-	-
IND-10	0,919	-	-
IND-11	0,919	-	-
IND-12	0,919	-	-
IND-13	0,919	-	-
IND-14	0,971	-	0
IND-15	0,919	-	-
IND-18	0,919	-	-
IND-19	0,919	-	-
IND-20	0	0	-
IND-21	0	0	-

Perhitungan *Information gain*:

$$IG(S, IND-1) = 0,919 - (6/6)0,919 - (0/6)0 - (0/6)0 = 0,919 - 0,919 - 0 - 0 = 0$$

Nilai *Information gain* dari tiap atribut selanjutnya dapat dilihat pada tabel 3.13 berikut.

Tabel 3.13 Nilai *information gain* perhitungan manual tahap III

Atribute	<i>Information gain</i>
IND-2	0
IND-3	0
IND-4	0
IND-5	0
IND-6	0,459
IND-7	0
IND-8	0
IND-9	0
IND-10	0
IND-11	0
IND-12	0
IND-13	0
IND-14	0,11
IND-15	0
IND-18	0
IND-19	0
IND-20	0,919
IND-21	0,919

Berdasarkan perhitungan nilai *information gain* diatas terdapat 2 nilai yang sama tingginya yakni 0,919 pada atribut IND-20 serta IND-21, maka yang akan diambil menjadi node selanjutnya adalah atribut IND-20.



Gambar 3.13 Tree hasil perhitungan manual tahap III

Tree diatas menunjukkan pada cabang yes dan no sama – sama hanya memiliki 1 class, sehingga tree sudah menunjukkan bentuk akhirnya karena sudah dapat memisahkan semua data ke masing – masing classnya (nilai information gain pada tiap node sudah tidak ada yang lebih besar dari 0). Proses pembentukan decision tree dengan algoritma *ID3* berhenti dan menjalankan proses terakhir yakni peng-ekstrak-an rule untuk mengklasifikasi data.

Proses selanjutnya setelah tree berhasil memisahkan tiap data ke classnya masing – masing ialah peng-ekstrak-an rule/aturan untuk proses klasifikasi data. Rule yang dihasilkan dari tree diatas yakni :

1. If 16 = “tidak” and 3 = “tidak” then class = KPS
2. If 16 = “tidak” and 3 = “ya” then class = KS I
3. If 16 = “ya” and 17 = “tidak” then class = KS II
4. If 16 = “ya” and 17 = “ya” and 20 = “tidak” then class = KS III
5. If 16 = “ya” and 17 = “ya” and 20 = “ya” then class = KS III+

Kelima rule tersebut selanjutnya akan digunakan untuk proses klasifikasi data. Rule tersebut akan coba diterapkan pada data tabel 3.14 berikut ini.



Tabel 3.14 Data untuk pengujian rule

ID. KK	ID. ISTKS	INDIKATOR																				STATUS TKS	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		21
11	11	v	v	v	v	v		v	v	v	v	v	v	x		v	x	v	v	v	x	x	KS I
12	12	v	v	v	v	v		v	v	v	v	v	v	v	v	v	x	v	v	v	x	x	KS II
13	13	v	v	v	v			v	v	v	v	v	v	x		v	v	v	v	v	x	x	KS I
14	14	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x	x	KS III
15	15	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x	x	KS III
16	16	v	v	v	v	v		v	v	v	v	v	v	v	v	v	v	x	v	v	x	x	KS II
17	17	v	v	v	v	v	v	v	v	v	v	v	v	x	v	v	v	v	v	v	x	x	KS I
18	18	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x	x	KS III
19	19	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	x	v	v	x	x	KS II
20	20	v	v	x	v	v	v	v	v	v	v	v	v	v	v	v	x	v	v	v	x	x	KPS

Menggunakan rule yang telah di dapatkan dari proses diatas maka hasil pengujian yang didapatkan dari tabel 3.14 ditunjukkan pada tabel 3.15.

Tabel 3.15 Hasil Pengujian tabel 3.14

Data	Status TKS Prediksi	Status TKS Aktual
11	KS I	KS I
12	KS I	KS II
13	KS III	KS I
14	KS III	KS III
15	KS III	KS III
16	KS II	KS II
17	KS III	KS I
18	KS III	KS III
19	KS II	KS II
20	KPS	KPS

Tabel 3.15 diatas adalah tabel dengan status TKS yang asli dan status TKS prediksi hasil penerapan rule yang di dapatkan dari tree tang telah terbentuk. Penghitungan akurasi dari decision tree yakni dengan membagi data yang diklasifikasikan dengan benar dengan banyaknya data.

Hasil yang didapatkan dari pengecekan 2 tabel tersebut yakni dari 10 data, 7 data diklasifikasikan dengan benar, dan 3 data diklasifikasikan dengan salah, maka akurasi yang didapatkan yakni :

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}} \times 100\%$$

$$\text{Akurasi} = \frac{7}{10} \times 100\%$$

$$= 70\%$$

Precision, Recall, dan F-Measure :

$$Precision = \frac{TP}{(TP + FP)}$$

$$\text{Class KPS : Precision} = \frac{1}{(1+0)} = 1$$

$$\text{Class KS II : Precision} = \frac{2}{(2+0)} = 1$$

$$\text{Class KS I : Precision} = \frac{1}{(1+1)} = 0,5$$

$$\text{Class KS III : Precision} = \frac{3}{(3+2)} = 0,6$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$\text{Class KPS : Recall} = \frac{1}{(1+0)} = 1$$

$$\text{Class KS I : Recall} = \frac{1}{(1+2)} = 0,33$$

$$\text{Class KS II : Recall} = \frac{2}{(2+1)} = 0,67$$

$$\text{Class KS III : Recall} = \frac{3}{(3+0)} = 1$$

$$F - Measure = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

$$\text{Class KPS : F - Measure} = \frac{2*1*1}{(1+1)} = 1$$

$$\text{Class KS II : F - Measure} = \frac{2*1*0,67}{(1+0,67)} = 0,8$$

$$\text{Class KS I : F - Measure} = \frac{2*0,5*0,33}{(0,5+0,33)} = 0,4$$

$$\text{Class KS III : F - Measure} = \frac{2*0,6*1}{(0,6+1)} = 0,75$$

F - Measure rata - rata = 0,74

Akurasi yang didapatkan dari penerapan rule yakni 70% dan F - Measure rata - ratanya ialah 0,74. Akurasi dari metode ID3 ini dipengaruhi oleh jumlah data training yang ada, dalam perhitungan manual ini data training yang digunakan untuk mendapatkan rule berjumlah 10 data. Jumlah data training yang lebih banyak mungkin bisa meningkatkan akurasi dari algoritma ini karena

dengan jumlah data training yang lebih banyak, tree yang dibentuk akan lebih baik untuk mengklasifikasikan data.

3.4 Perancangan Uji Coba

Perancangan uji coba menggambarkan skenario proses pengujian yang akan dilakukan pada sistem. Proses uji coba yang akan dilakukan terdiri dari 3 skenario yakni skenario 1 yang ditujukan untuk mengetahui jumlah aturan/rule yang terbentuk serta akurasi dari 2 jenis data yakni data seimbang dan data tidak seimbang. Data seimbang dalam hal ini maksudnya adalah jumlah kelas yang terdapat pada data latih antara 1 dan lainnya sama, semisal class KS I terdapat 2 data maka class lain juga masing – masing hanya 2 data. Skenario 2 ditujukan untuk mengetahui apakah aturan yang terbaik yang diperoleh dari skenario 2 merupakan aturan yang terbaik saat diterapkan ke beberapa data uji yang berbeda dan skenario 3 ditujukan untuk mengetahui akurasi dari kombinasi data latih dan data uji.

3.4.1 Skenario 1

Skenario 1 menguji 2 jenis data yakni data seimbang dan tidak seimbang. Pengujian ini dilakukan untuk membandingkan jumlah aturan yang terbentuk serta akurasi dari tree. Maksud dari data seimbang dan data tidak seimbang adalah jumlah class yang ada pada dataset dimana untuk data seimbang jumlah data pada tiap class disamakan, sedangkan pada data tidak seimbang jumlah data pada tiap class berbeda – beda. Pada skenario ini terdapat proses pengolahan data terlebih dahulu, dikarenakan jumlah data class KS III+ sangat sedikit, maka class KS III+ tidak diikuti sertakan (dihapus dari dataset). Pengujian dilakukan dengan 25 data uji untuk tiap kali pengujian, yang berbeda hanya jumlah data latih, hal ini dilakukan agar mempermudah pengambilan kesimpulan pada proses analisa.

Tabel 3.16 Rancangan Uji Coba Sistem Skenario 1 Aturan

Data Latih	Aturan yang Terbentuk		Akurasi	
	Data Seimbang	Data Tidak Seimbang	Data Seimbang	Data Tidak Seimbang
28 data				
52 data				
76 data				
100 data				

3.4.2 Skenario 2

Skenario 2 ditujukan untuk mengetahui apakah aturan yang terbaik yang diperoleh dari skenario 1 merupakan aturan yang terbaik saat diterapkan ke beberapa data uji yang berbeda. Pengujian dilakukan empat kali pengujian yakni dengan 25 data uji, 50 data uji, 75 data uji, serta 100 data uji.

Tabel 3.17 Rancangan Uji Coba Sistem Skenario 2

Data Uji	Akurasi
25 data	
50 data	
75 data	
100 data	

3.4.3 Skenario 3

Skenario 3 menguji validitas yakni melakukan pelatihan dan pengujian menggunakan data yang sama.

Tabel 3.18 Rancangan Uji Coba Sistem Skenario 3

Data Uji	Akurasi
8 data	
16 data	
24 data	
32 data	

3.4.4 Skenario 4

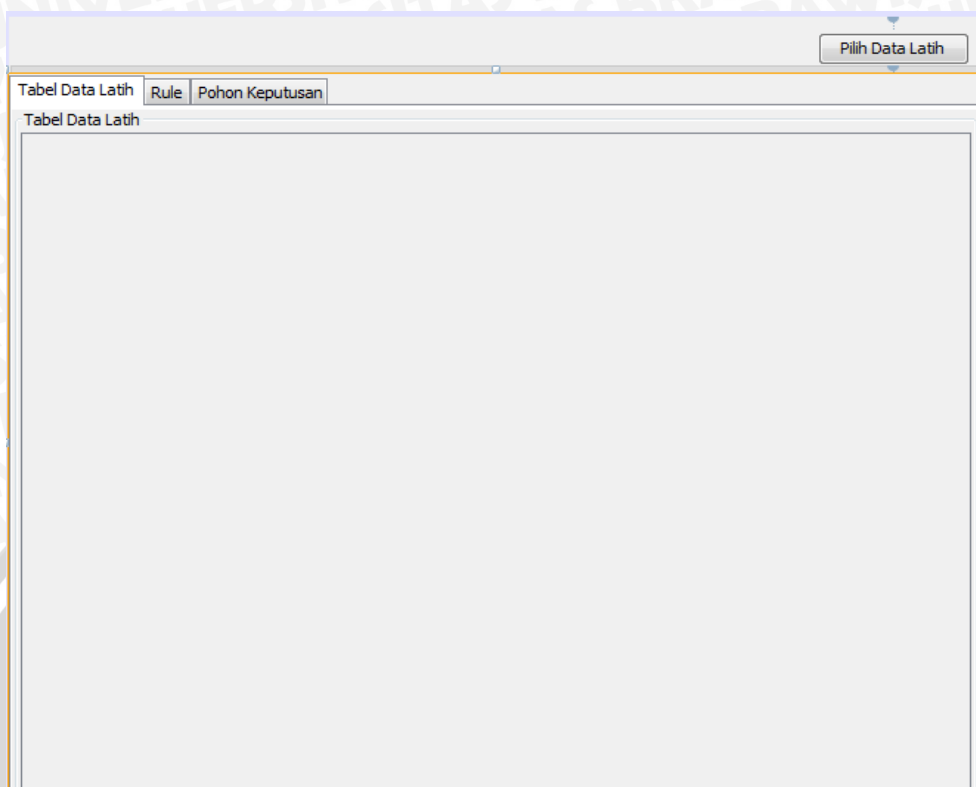
Skenario 4 menguji data tahapan keluarga sejahtera namun dengan mengganti semua data missing value dengan data ya.

Tabel 3.199 Rancangan Uji Coba Sistem Skenario 4

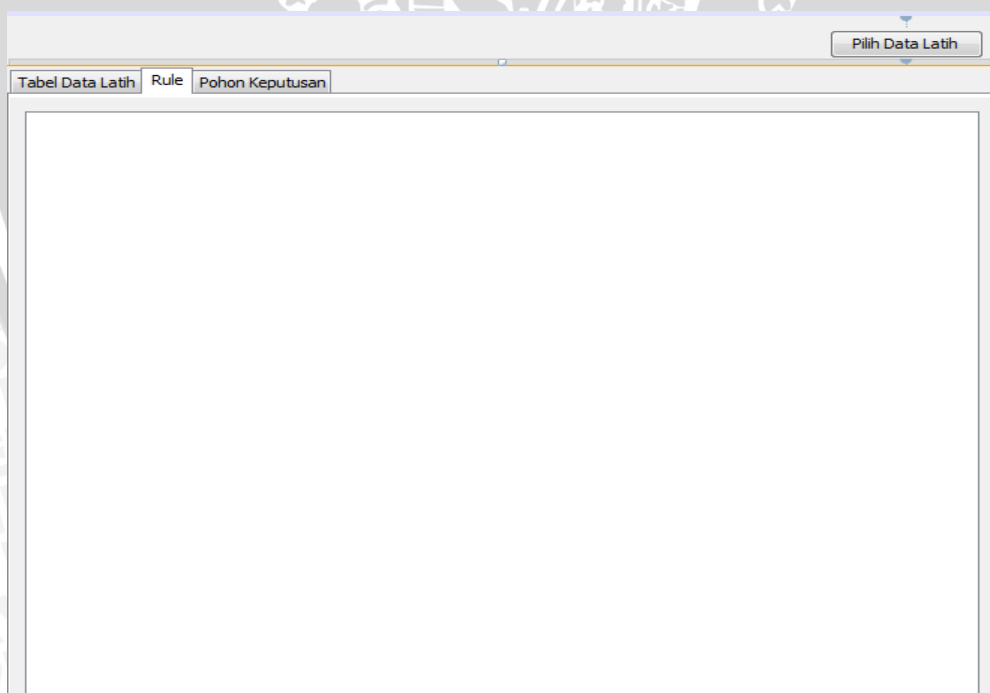
Data Latih	Aturan yang Terbentuk		Akurasi	
	Data Seimbang	Data Tidak Seimbang	Data Seimbang	Data Tidak Seimbang
28 data				
52 data				
76 data				
100 data				

3.5 Rancangan Antarmuka

Antarmuka yang nantinya akan dibuat diharapkan bersifat *user friendly* agar pengguna mudah dalam menjalankan aplikasi. Antarmuka sistem terdiri dari 2 layer yakni layer pertama untuk pelatihan dan layer kedua untuk pengujian. Pada layer pelatihan terdapat 3 tab yakni tab Tabel Data Latih, tab Rule, dan tab Pohon Keputusan, sedangkan pada layer pengujian hanya terdiri dari 1 tab untuk menampilkan data dan hasil akurasi dan f-measure sistem yang terdapat pada bagian bawah. Gambar rancangan antarmuka ditunjukkan pada gambar 3.14, 3.15, 3.16, dan 3.17

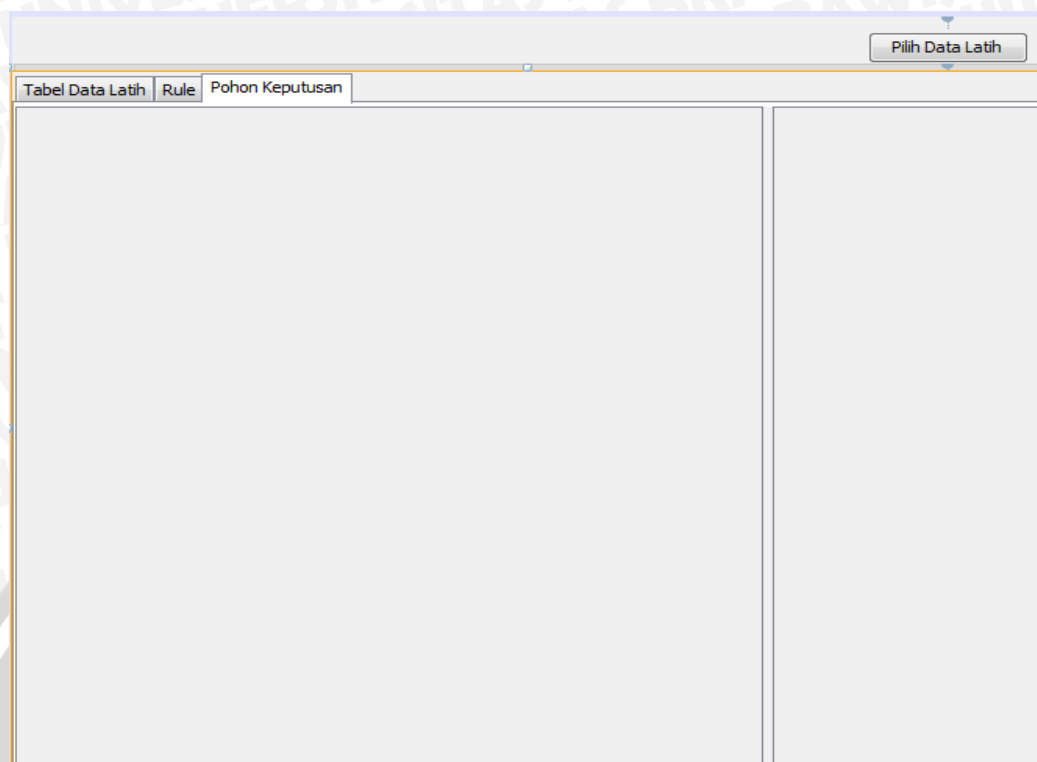


Gambar 3.14 Rancangan Antarmuka tab Tabel Data Latih

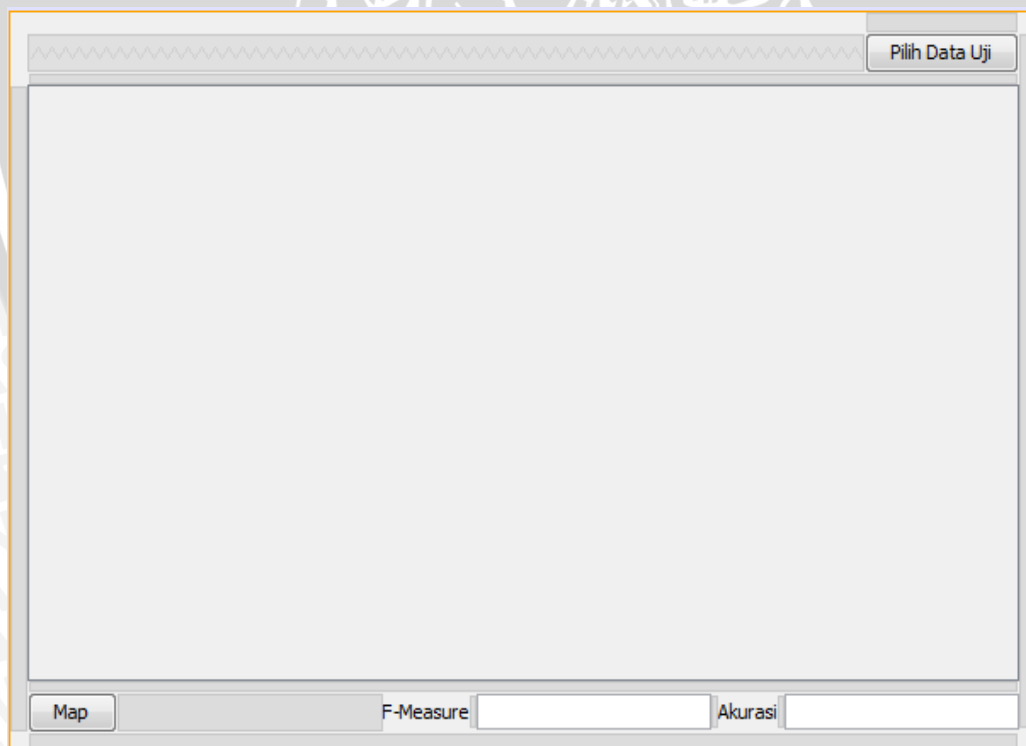


Gambar 3.15 Rancangan Antarmuka tab Rule





Gambar 3.16 Rancangan Antarmuka tab Pohon Keputusan



Gambar 3.17 Rancangan Antarmuka Pengujian

BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam sub bab ini adalah lingkungan implementasi perangkat lunak dan perangkat keras yang digunakan dalam mengimplementasikan sistem yang telah dibuat dalam penelitian ini.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam mengembangkan perangkat lunak dalam penelitian ini memiliki spesifikasi sebagai berikut :

1. Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz (4 CPUs)
2. Memori 6 GB
3. Harddisk 500 GB
4. Monitor 14"

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam mengembangkan sistem dan penelitian ini terdiri dari :

1. Sistem Operasi Windows 7 Home Basic 64bit
2. NetBeans IDE 7.1

4.2 Implementasi Program

Berdasarkan analisa dan perancangan sistem yang terdapat pada bab 3, maka pada subbab ini akan dijelaskan implementasi program yang membahas proses-proses yang terjadi pada sistem yang dirancang menggunakan bahasa pemrograman Java. Jalan awal program dimulai dengan pembacaan inputan data yang berupa file .xls, kemudian perhitungan nilai entropy, perhitungan nilai information gain, pembentukan tree, dan perhitungan akurasi. Penjelasan lebih lanjut akan dijelaskan dalam subbab dibawah ini.

4.2.1 Pembacaan Data (Read File)

Dalam penelitian ini sistem dapat menerima inputan berupa data yang berformat .xls. Pembacaan data .xls dimulai dengan menghitung kolom 0 terdapat berapa baris, kemudian data pada tiap baris dan kolom tersebut disimpan ke dalam array. Penerapan dalam bahasa pemrograman ditunjukkan pada source code 4.1.

```
public class FileProcessing {
    public static ArrayList<Data> bacaFiles(String namafile){
        ArrayList<Data> keluarga = new ArrayList<Data>();
        File file = new File(namafile);
        if (!file.isFile() || !file.getName().endsWith(".xls")) {
            return keluarga;
        }
        try {
            Workbook workbook = Workbook.getWorkbook(file);
            Sheet sheet = workbook.getSheet(0);
            for (int i = 0; i < sheet.getColumn(0).length; i++) {
                Cell[] temp = sheet.getRow(i);
                ArrayList<Integer> kriteria = new
                ArrayList<Integer>();
                for(int j=3;j<temp.length-1;j++)
                kriteria.add((temp[j].getContents().equals("v"))?
                1:((temp[j].getContents().equals(""))?-1:0));
                Data excel = new Data(temp[0].getContents(),
                temp[1].getContents(), temp[2].getContents(),
                temp[temp.length-1].getContents(), kriteria);
                keluarga.add(excel);
            }
            workbook.close();
            return keluarga;
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex,
            "Error!!", JOptionPane.ERROR_MESSAGE);
        }
        return keluarga;
    }
}
```

Source Code 4.1 Fungsi Pembacaan File (Read File)

4.2.2 Perhitungan Entropy

Proses awal dari pembentukan tree ini dimulai dengan perhitungan entropy, mulai dari perhitungan entropy total, entropy nilai dalam atribut hingga entropy dari tiap atribut. Perhitungan entropy total dimulai dengan menghitung berapa banyak data dan data tersebut termasuk class apa. Data dimasukkan ke dalam arraylist, dan kemudian dihitung menggunakan persamaan 2.1. Proses yang sama digunakan untuk mendapatkan entropy dari tiap atribut. Penjelasan

bagaimana proses perhitungan ditunjukkan pada source code 4.2 untuk entropy total, source code 4.3 untuk entropy nilai dalam atribut, dan source code 4.4 untuk entropy atribut.

```
public double getEntropTotal(ArrayList<Data> data){
    double Entrop = 0;
    HashMap<String,Integer> anggotaKelas = new
    HashMap<String,Integer>();
    for (int i = 0; i < data.size(); i++) {
        if(!anggotaKelas.containsKey(data.get(i).kelas)){
            anggotaKelas.put(data.get(i).kelas, 1);
        }else{
            int a = anggotaKelas.get(data.get(i).kelas);
            a++;
            anggotaKelas.put(data.get(i).kelas, a);
        }
    }
    for (String a : anggotaKelas.keySet()) {
        if(anggotaKelas.get(a)!=0){
            Entrop+=(-1*((double)anggotaKelas.get(a)/
            (double)data.size())*log2(((double)anggotaKelas.
            get(a)/(double)data.size())));
        }
    }
    return Entrop;
}
```

Source Code 4.1 Proses Perhitungan Entropy Total

```
public double getE(ArrayList<Data> data,int kriteria,int
value,Set<String> Kelas ){
    double temp = 0;
    double Counter = 0;
    for (int i = 0; i < data.size(); i++)
        if (data.get(i).kriteria.get(kriteria) == value)
            Counter++;
    for(String kelas : Kelas){
        double Entrop = 0;
        double nkelas = 0;
        for (int i = 0; i < data.size(); i++)
            if (data.get(i).kriteria.get(kriteria)==value &&
            data.get(i).kelas.equals(kelas))
                nkelas++;
        if (nkelas != 0) {
            if (Counter != 0) {
                Entrop = (-1 * (((double) nkelas / (double)
                Counter) * log2(((double) nkelas / (double)
                Counter))));
                temp+=Entrop;
            }
        }
    }
    return (Counter/data.size())*temp;
}
```

Source Code 4.2 Proses Perhitungan Entropy Nilai dalam Atribut

```

public double[] getEntropKriteria(ArrayList<Data> data,int
kriteria){
    HashMap<String,Integer> anggotaKelas = new
HashMap<String,Integer>();
    for (int i = 0; i < data.size(); i++) {
        if(!anggotaKelas.containsKey(data.get(i).kelas)){
            anggotaKelas.put(data.get(i).kelas, 1);
        }else{
            int a = anggotaKelas.get(data.get(i).kelas);
            a++;
            anggotaKelas.put(data.get(i).kelas, a);
        }
    }
    double EntropiKriteri [] = new double[]{
        getE(data, kriteria, 1, anggotaKelas.keySet()),
        getE(data, kriteria, 0, anggotaKelas.keySet()),
        getE(data, kriteria, -1, anggotaKelas.keySet())
    };
    return EntropiKriteri;
}

```

Source Code 4.3 Proses Perhitungan Entropy Atribut

4.2.3 Perhitungan Information Gain

Nilai Entropy yang didapat pada proses sebelumnya kemudian dimasukkan ke rumus 2.2 untuk mendapatkan nilai information gain yang nantinya digunakan untuk pemilihan atribut sebagai node. Proses tersebut dilakukan dalam perulangan sebanyak atribut yang ada pada data. Sourcode 4.3 akan menjelaskan detail method tersebut seperti berikut.

```

public double [] getIgain(ArrayList<Data> data){
    double []Iganin=new double[banyak_kriteria];
    double entropiTotal = getEntropTotal(data);
    for (int i = 0; i <Iganin.length; i++) {
        double [] entropiKriteria = getEntropKriteria(data,i);
        Iganin[i]=(entropiTotal-(entropiKriteria[0])-(
entropiKriteria[1])-(entropiKriteria[2]));
    }
    return Iganin;
}

```

Source Code 4.4 Perhitungan Information Gain

4.2.4 GetRoot

Proses pembentukan tree untuk memilih atribut yang dijadikan node dengan memilih nilai information gain yang tertinggi, untuk itu setelah proses perhitungan nilai information gain maka terdapat proses GetRoot untuk memilih information yang terbesar. Implementasi proses GetRoot dalam sistem ditunjukkan pada source code 4.6.


```

public int getRoot(double [] igain,ArrayList<Data> data,int []
cek){
    double max=0;
    int index=0;
    boolean al = false;
    for(int i=0;i<igain.length;i++){
        if(cek[i]!=0)
            continue;
        if(igain[i]>max){
            max = igain[i];
            index = i;
            al = true;
        }
    }
    if(!al){
        return -1;
    }
    return index;
}

```

Source Code 4.5 Proses GetRoot

4.2.5 Gambar Tree

Pada proses ini method perhitungan information gain dan GetRoot digunakan untuk membentuk sebuah decision tree dari data yang di inputkan. Setelah tree terbentuk kemudian tree ditelusuri dari root hingga ke leaf untuk mendapatkan aturan untuk mengklasifikasikan data. Source code 4.7 menjelaskan proses pembentukan tree.

```

public void prosesID3(ArrayList<Data> data){
    Stack<Node>jalanCrita = new Stack<Node>();
    int [] newCek = new int [banyak_kriteria];
    double [] IgainRoot = getIgain(data);
    int idKriteria = getRoot(IgainRoot,data,newCek);
    newCek[idKriteria]=1;
    Node root = new Node(idKriteria, data,IgainRoot);
    root.setCeks(newCek);
    this.root = root;
    jalanCrita.push(root);
    while(!jalanCrita.isEmpty()){
        Node temp = jalanCrita.pop();
        if(temp.isLeaf) continue;
        for(int i=-1;i<2;i++){
            ArrayList<Data> dataBagi = new ArrayList<Data>();
            for(Data d : temp.data){
                if(d.kriteria.get(temp.id)==i){
                    dataBagi.add(d.clone());
                }
            }
            if(dataBagi.size()!=0){

```

```

Node anakTemp;
Set<String> sekKelas =
    apakahSamaKelas(dataBagi);
if(sekKelas.size()<=1){
    anakTemp = new Node(dataBagi);
    for(String f : sekKelas)
        anakTemp.setLeaf(f);
}else{
    IgainRoot = getIgain(dataBagi);
    int a =
        getRoot(IgainRoot,dataBagi,temp.cek);
    if(a>=0){
        anakTemp = new Node(a,
            dataBagi,IgainRoot);
        anakTemp.setCeks(temp.cek);
        anakTemp.setCek(a);
        jalanCrita.push(anakTemp);
    }
    else{
        anakTemp = new Node(dataBagi);
        String kelas ="";
        for(String f : sekKelas)
            kelas += (f+",");
        anakTemp.setLeaf(kelas.substring(0,
            kelas.length()-1));
    }
}
if(i==1){
    temp.kkan = anakTemp;
}else if(i==-1){
    temp.kkir = anakTemp;
}else{
    temp.kten = anakTemp;
}
}
}
}
}

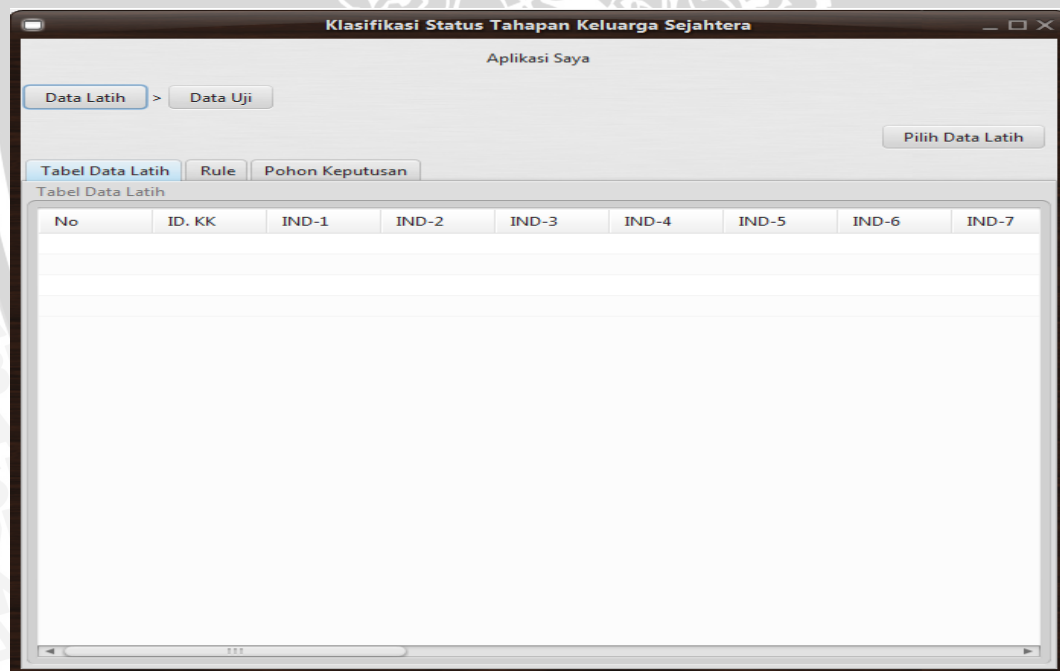
```

Source Code 4.6 Gambar Tree

4.3 Implementasi Antarmuka

Antar muka yang telah dirancang pada bab III di implementasikan dengan hasil yang akan ditunjukkan pada gambar – gambar dibawah ini. Antar muka sistem terdiri dari 2 layer, yakni Data Latih dan Data Uji, untuk berpindah antar layer menggunakan tombol Data Latih untuk menuju ke layer Data latih, dan menggunakan tombol Data Uji untuk berpindah ke layer Data Uji. Data Latih adalah layer dimana proses awal untuk memasukkan data pelatihan untuk membentuk tree terjadi. Layer Data Latih terdiri dari 3 tab yakni Tabel Data Latih, Rule, dan Pohoh Keputusan. Data Uji adalah layer yang digunakan untuk menguji atau mendapatkan class prediksi dari Rule yang didapat dari Pohon Keputusan yang terbentuk.

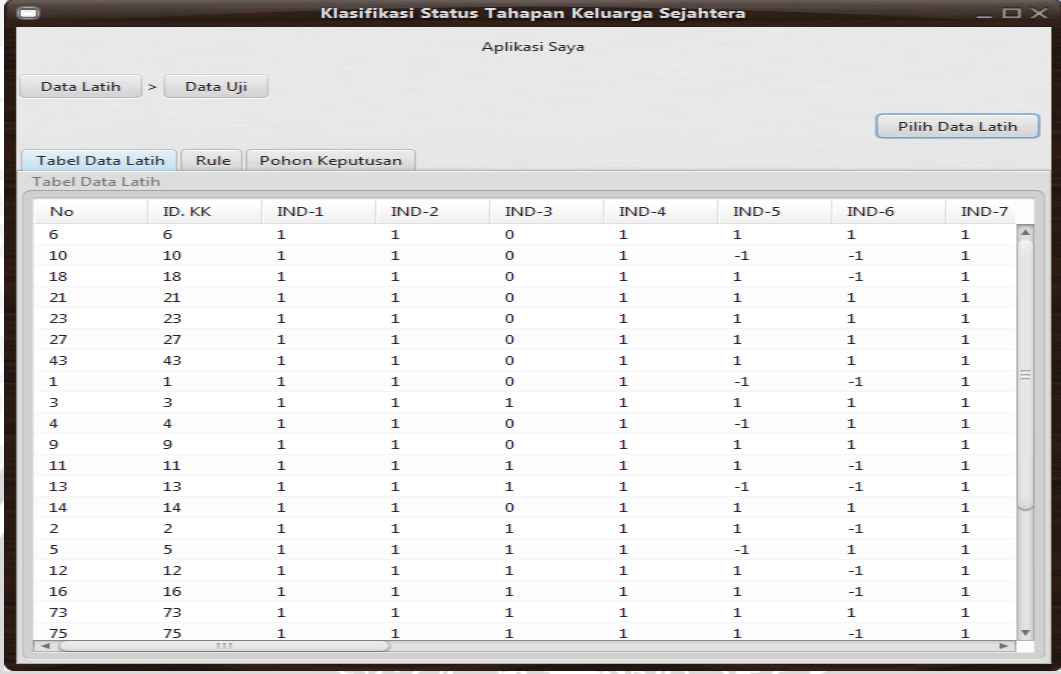
Antar muka tab Tabel Data Latih pada layer Data Latih digunakan untuk menampilkan data latih yang dimasukkan. Gambar 4.1 merupakan antar muka Tabel Data Latih.



Gambar 4.1 Antar muka Tabel Data Latih

Saat pertama kali program dijalankan tab Tabel Data Latih hanya menampilkan judul tiap kolom yang ada. Untuk memasukkan data latih yang akan digunakan adalah mengklik tombol Pilih Data Latih dan memilih data yang

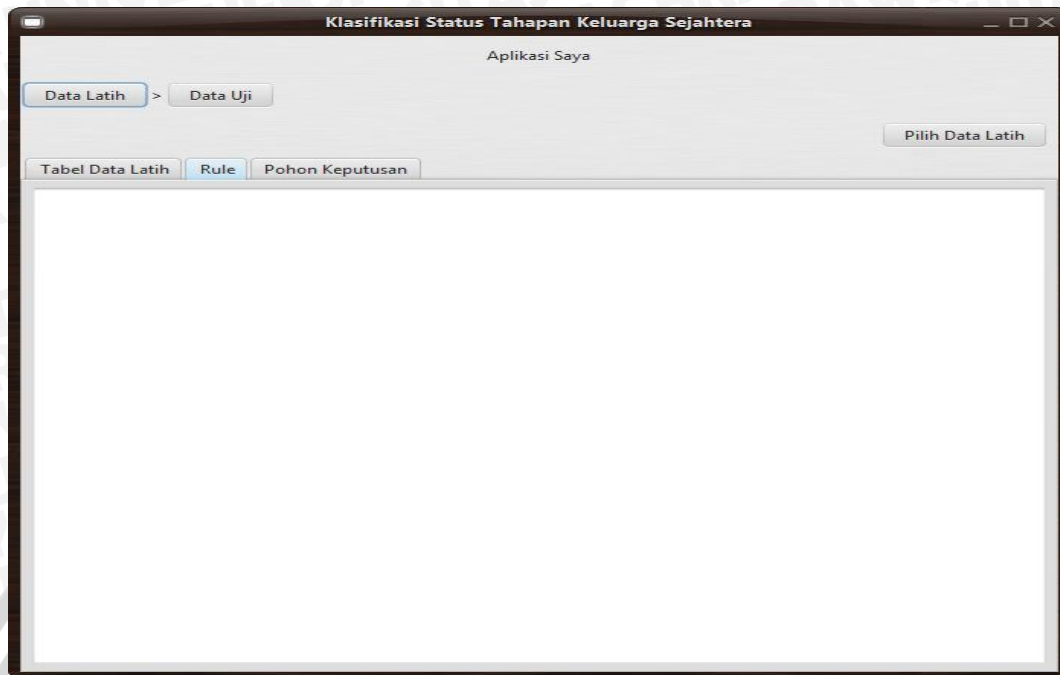
diinginkan, setelah data terpilih maka data ditampilkan pada tab Tabel Data Latih. Gambar 4.2 merupakan antar muka Tabel Data Latih setelah data dimasukkan.



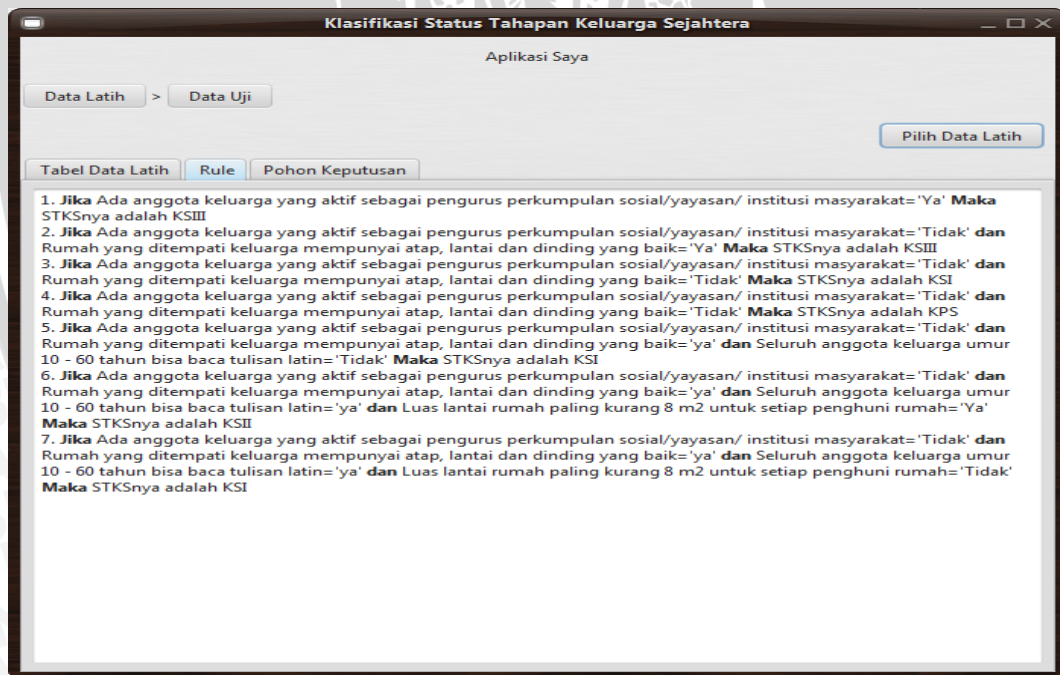
No	ID. KK	IND-1	IND-2	IND-3	IND-4	IND-5	IND-6	IND-7
6	6	1	1	0	1	1	1	1
10	10	1	1	0	1	-1	-1	1
18	18	1	1	0	1	1	-1	1
21	21	1	1	0	1	1	1	1
23	23	1	1	0	1	1	1	1
27	27	1	1	0	1	1	1	1
43	43	1	1	0	1	1	1	1
1	1	1	1	0	1	-1	-1	1
3	3	1	1	1	1	1	1	1
4	4	1	1	0	1	-1	1	1
9	9	1	1	0	1	1	1	1
11	11	1	1	1	1	1	-1	1
13	13	1	1	1	1	-1	-1	1
14	14	1	1	0	1	1	1	1
2	2	1	1	1	1	1	-1	1
5	5	1	1	1	1	-1	1	1
12	12	1	1	1	1	1	-1	1
16	16	1	1	1	1	1	-1	1
73	73	1	1	1	1	1	1	1
75	75	1	1	1	1	1	-1	1

Gambar 4.2 Antar muka Tabel Data Latih setelah memilih data

Gambar 4.2 adalah gambar antar muka Tabel Data Latih setelah memilih data latih. Data latih yang dipilih ditampilkan pada tabel serta diproses untuk mendapatkan pohon keputusan dan *rule*. Antar muka Rule digunakan untuk menampilkan aturan yang didapat dari pohon keputusan yang terbentuk. Gambar 4.3 merupakan antar muka Rule.

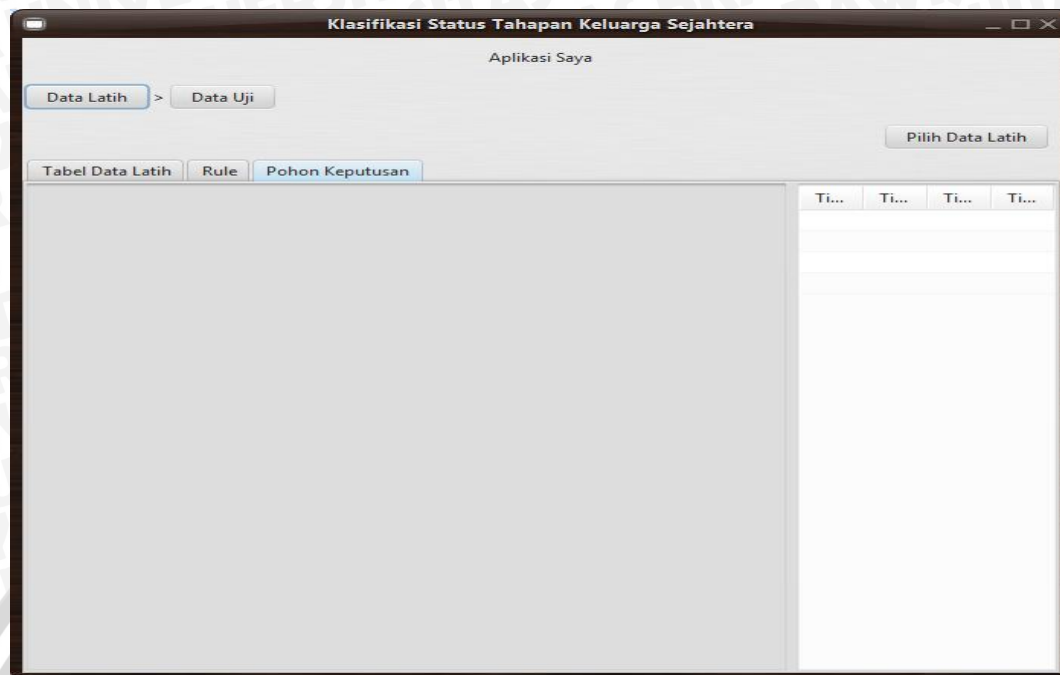


Gambar 4.3 Antar muka Rule



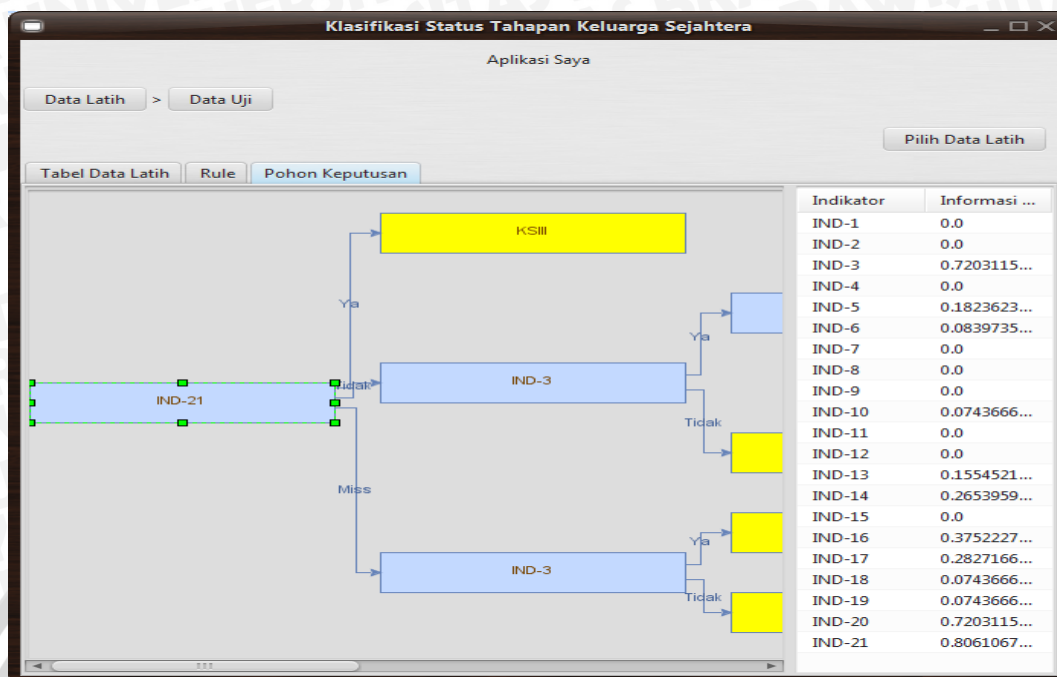
Gambar 4.4 Antar muka Rule setelah data latih dimasukkan

Antar muka Rule pada awalnya tidak menampilkan apa – apa saat program dijalankan, baru setelah data latih dimasukkan antar muka Rule menampilkan aturan – aturan yang diperoleh dari pohon keputusan yang terbentuk seperti yang ditampilkan pada gambar 4.4.

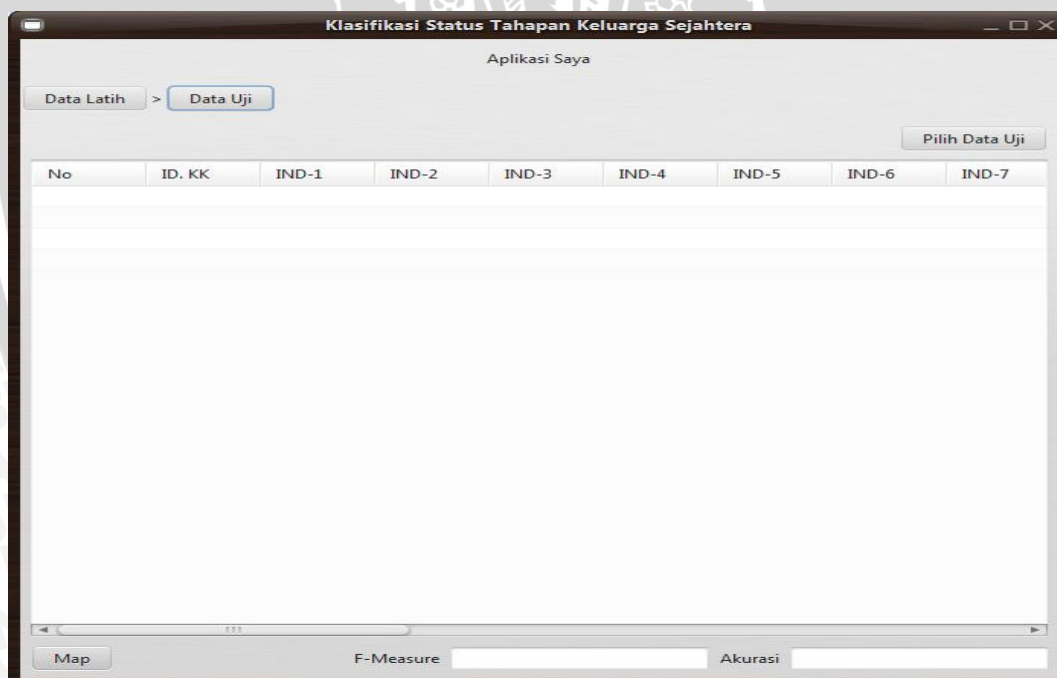


Gambar 4.5 Antar muka Pohon Keputusan

Antar muka Pohon Keputusan terdiri dari 2 bagian yakni pohon keputusan yang dibentuk menggunakan graph dan tabel nilai perhitungan information gain pada suatu node yang dipilih (diklik) user. Awal mula program dijalankan antar muka Pohon Keputusan tidak menampilkan apa – apa, pohon keputusan dan nilai information gain akan tampil setelah data latih dimasukkan. Gambar 4.6 menunjukkan gambar antar muka Pohon Keputusan setelah data latih dimasukkan.



Gambar 4.6 Antar muka Pohon Keputusan setelah data latih dimasukkan.



Gambar 4.7 Antar muka Data Uji

Gambar 4.7 merupakan gambar antar muka Data Uji yang digunakan untuk pengujian setelah proses pelatihan selesai. Pada awal program dijalankan antar muka Data Uji hanya menampilkan judul tiap kolom, untuk menjalankan proses pengujian maka perlu mengklik Pilih Data Uji untuk memilih data uji yang

ingin digunakan. Gambar 4.8 menampilkan gambar antar muka Data Uji setelah user memilih data uji yang diinginkan.

No	ID. KK	IND-1	IND-2	IND-3	IND-4	IND-5	IND-6	IND-7
121	121	1	1	1	1	-1	1	1
122	122	1	1	1	1	1	-1	1
123	123	1	1	1	1	-1	-1	1
124	124	1	1	1	1	-1	-1	1
125	125	1	1	1	1	1	-1	1
126	126	1	1	1	1	1	-1	1
127	127	1	1	1	1	-1	-1	1
128	128	1	1	1	1	1	1	1
129	129	1	1	0	1	1	1	1
130	130	1	1	1	1	-1	1	1
131	131	1	1	1	1	-1	-1	1
132	132	1	1	1	1	1	1	1
133	133	1	1	1	1	1	1	1
134	134	1	1	1	1	1	-1	1
135	135	1	1	1	1	1	1	1
136	136	1	1	1	1	1	1	1
137	137	1	1	1	1	1	1	1
138	138	1	1	1	1	1	1	1
139	139	1	1	1	1	-1	-1	1
140	140	1	1	1	1	1	1	1
141	141	1	1	1	1	1	-1	1

Gambar 4.8 Antar muka Data Uji setelah memasukkan data uji

Data uji yang telah dipilih ditampilkan pada antar muka Data Uji dan pada kolom paling kanan terdapat tambahan kolom yakni kolom STATUS PREDIKSI untuk menampilkan status tks berdasarkan *rule* yang didapat dari pohon keputusan. Bagian bawah menampilkan nilai F-Measure rata – rata dari tiap class dan nilai akurasi dari sistem.

BAB V

PENGUJIAN DAN ANALISIS

Bab ini membahas tentang tahapan pengujian dan analisis implementasi *Decision Tree* dengan Algoritma *ID3* untuk klasifikasi status tahapan keluarga sejahtera. Proses pengujian dilakukan dengan mengevaluasi hasil uji coba sesuai dengan perancangan uji coba yang telah ditentukan sebelumnya.

5.1 Hasil Uji Coba Skenario 1

Skenario 1 menguji 2 jenis data yakni data seimbang dan tidak seimbang. Pengujian dilakukan untuk membandingkan aturan yang terbentuk dan akurasi dari tree. Maksud dari data seimbang dan data tidak seimbang adalah jumlah class yang ada pada dataset dimana untuk data seimbang jumlah data pada tiap class disamakan, sedangkan pada data tidak seimbang jumlah data pada tiap class berbeda – beda. Pada skenario ini terdapat proses pengolahan data terlebih dahulu, dikarenakan jumlah data class KS III+ sangat sedikit maka class KS III+ tidak diikuti sertakan (dihapus dari dataset). Pengujian dilakukan dengan 25 data uji untuk tiap kali pengujian, yang berbeda hanya jumlah data latih, hal ini dilakukan agar mempermudah pengambilan kesimpulan pada proses analisa. Hasil dari uji coba skenario 1 ditunjukkan pada tabel 5.1.

Tabel 5.1 Hasil Uji Coba Sistem Skenario 1

Data Latih	Aturan yang Terbentuk		Akurasi	
	Data Seimbang	Data Tidak Seimbang	Data Seimbang	Data Tidak Seimbang
28 data	8	8	88%	72%
52 data	12	14	76%	68%
76 data	12	11	80%	76%
100 data	12	11	84%	84%

Contoh hasil rule yang didapatkan dari 28 data latih dengan jenis data seimbang :

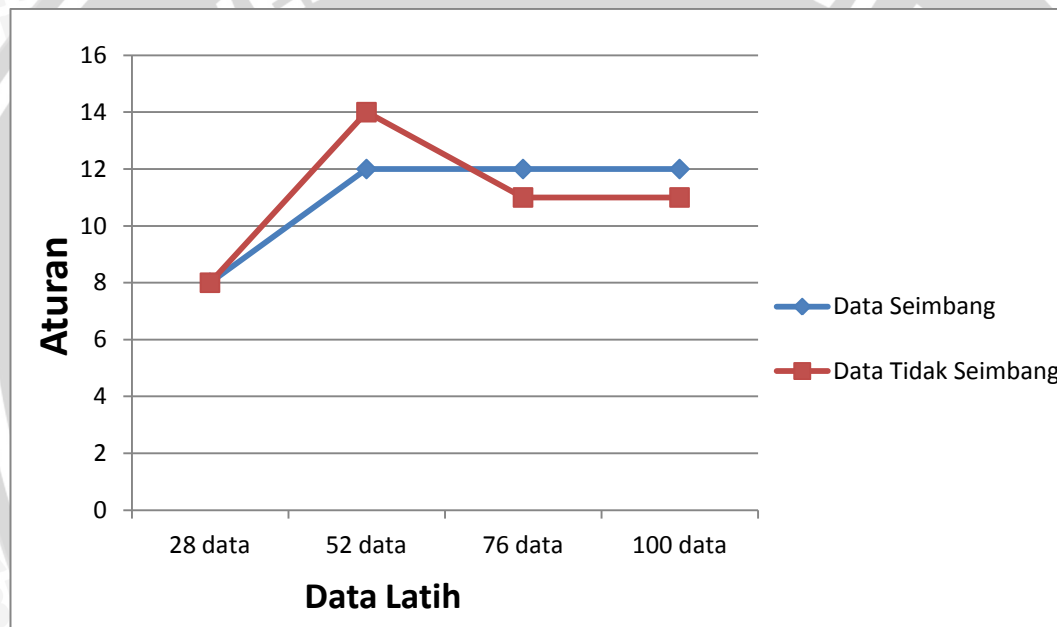
1. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='Tidak' **dan** Keluarga secara teratur dengan suka rela memberikan sumbangan materiil untuk kegiatan sosial='Tidak' **Maka** STKSnya adalah KPS.
2. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='Tidak' **dan** Keluarga secara teratur dengan suka rela memberikan sumbangan materiil untuk kegiatan sosial='Kosong' **Maka** STKSnya adalah KSI.
3. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='Tidak' **Maka** STKSnya adalah KSII.
4. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='ya' **dan** Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang='Tidak' **dan** Luas lantai rumah paling kurang 8 m2 untuk setiap penghuni rumah='Tidak' **Maka** STKSnya adalah KSI.
5. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='ya' **dan** Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang='Tidak' **dan** Luas lantai rumah paling kurang 8 m2 untuk setiap penghuni rumah='ya' **dan** Seluruh anggota keluarga umur 10 - 60 tahun bisa baca tulisan latin='Ya' **Maka** STKSnya adalah KSII.
6. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='ya' **dan** Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang='Tidak' **dan** Luas lantai rumah paling kurang 8 m2 untuk setiap penghuni rumah='ya' **dan** Seluruh anggota keluarga umur 10 - 60 tahun bisa baca tulisan latin='Tidak' **Maka** STKSnya adalah KSI.
7. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='ya' **dan** Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang='ya' **dan** Seluruh anggota keluarga umur 10 - 60 tahun bisa baca tulisan latin='Ya' **Maka** STKSnya adalah KSIII.
8. **Jika** Rumah yang ditempati keluarga mempunyai atap, lantai dan dinding yang

baik='ya' **dan** Kebiasaan keluarga makan bersama paling kurang seminggu sekali dimanfaatkan untuk berkomunikasi='ya' **dan** Sebagian penghasilan keluarga ditabung dalam bentuk uang atau barang='ya' **dan** Seluruh anggota keluarga umur 10 - 60 tahun bisa baca tulisan latin='Tidak' **Maka** STKSnya adalah KSI.

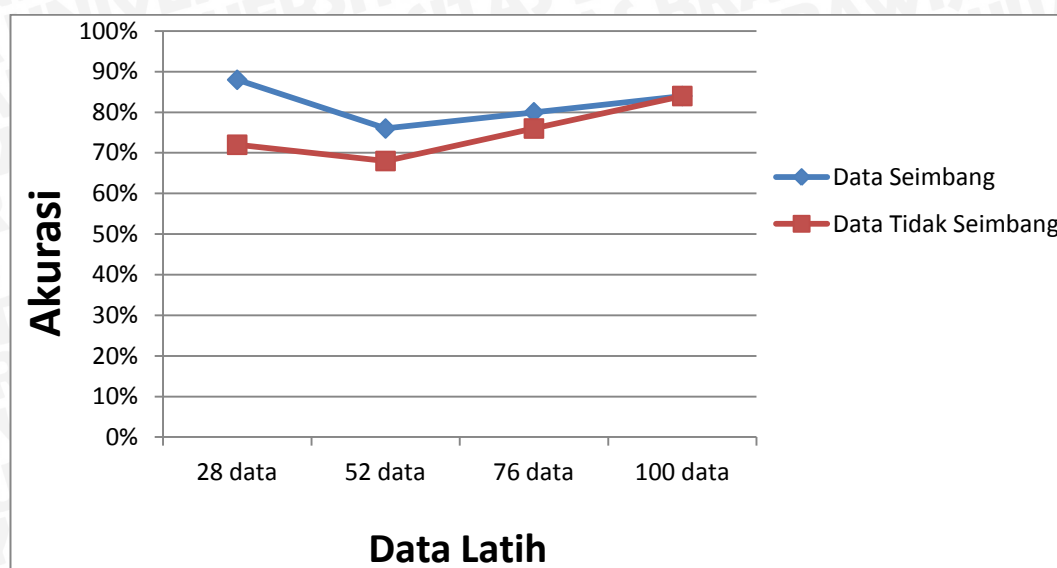
Dari hasil uji coba skenario 1 didapat aturan yang terbentuk pada data seimbang sebanyak 8 aturan untuk 28 data latih, dan 12 aturan untuk 52, 76, dan 100 data latih. Aturan yang terbentuk pada data tidak seimbang sebanyak 8 aturan untuk 28 data latih, 14 aturan untuk 52 data latih, dan 11 aturan untuk 76 dan 100 data latih. Perbedaan jumlah aturan yang terbentuk dikarenakan variasi data yang berbeda sehingga hasil perhitungan information gain berbeda dan tree yang terbentuk berbeda pula. Semakin banyak variasi data pada data latih akan menyebabkan tree yang terbentuk semakin besar, begitu pula aturannya.

Akurasi yang didapat dari hasil uji coba menunjukkan 28 data latih dengan jenis data yakni data seimbang yang mendapatkan hasil terbaik dengan tingkat akurasi sebesar 88%. Berdasarkan hasil uji coba ini dan pengamatan pada data latih yang digunakan bisa di ambil kesimpulan kenapa 28 data latih bisa mendapatkan hasil akurasi yang terbaik yakni dikarenakan variasi data pada 28 data latih lebih sedikit dibandingkan yang lain sehingga lebih mencirikan classnya serta tingkat kemiripan variasi data pada 28 data latih dengan data ujinya lebih besar dibandingkan data latih yang lain.

Akurasi yang didapat dari data seimbang lebih baik daripada data tidak seimbang. Rata – rata akurasi pada data seimbang mencapai 82%, sedangkan rata – rata akurasi pada data tidak seimbang hanya 75%. Data seimbang memiliki rata – rata akurasi yang lebih baik dikarenakan tiap class pada data seimbang memiliki jumlah yang sama, sedangkan pada data tidak seimbang jumlah data tiap class berbeda, sehingga terdapat class yang kekurangan variasi data dan ada class yang variasi datanya terlalu banyak sehingga tidak mencirikan class tersebut. Perbandingan jumlah Aturan yang Terbentuk dan Akurasi antara 2 jenis data dapat dilihat pada gambar 5.1 dan 5.2.



Gambar 5.1 Grafik Jumlah Aturan yang Terbentuk Skenario 1



Gambar 5.2 Grafik Akurasi Skenario 1

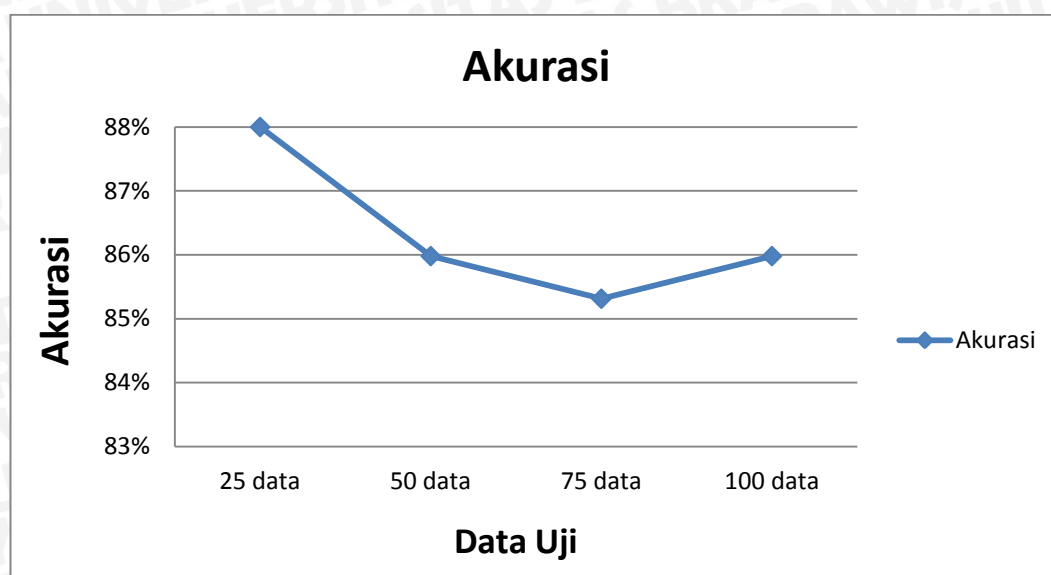
5.2 Hasil Uji Coba Skenario 2

Skenario 2 ditujukan untuk mengetahui apakah aturan yang terbaik yang diperoleh dari skenario 1 merupakan aturan yang terbaik saat diterapkan ke beberapa data uji yang berbeda. Pengujian dilakukan empat kali pengujian yakni dengan 25 data uji, 50 data uji, 75 data uji, serta 100 data uji. Hasil dari uji coba skenario 2 ditampilkan pada tabel 5.2.

Tabel 5.2 Hasil Uji Coba Sistem Skenario 2

Data Uji	Akurasi
25 data	88%
50 data	86%
75 data	85,34%
100 data	86%

Berdasarkan hasil uji coba skenario 1 didapat akurasi yang terbaik adalah saat data latih menggunakan data seimbang dengan jumlah 28 data latih. Data latih tersebut kemudian diujikan 4 kali yakni dengan 25 data uji, 50 data uji, 75 data uji, dan 100 data uji. Hasil dari uji coba menunjukkan nilai akurasi yang terbaik didapat saat data latih diuji menggunakan 25 data uji yakni 88%. Perbandingan hasil akurasi dapat dilihat pada gambar 5.3.



Gambar 5.3 Grafik Akurasi Skenario 2

Hasil uji coba menunjukkan nilai akurasi yang tidak jauh berbeda pada tiap pengujian, sehingga bisa disimpulkan bahwa data latih yang digunakan adalah data latih paling optimal. Hasil dari pengamatan data didapatkan penyebab terjadinya penurunan akurasi pada pengujian dengan 50 data uji dan 75 data uji dikarenakan jumlah class KS I pada 50 dan 75 data uji lebih banyak daripada 25 data uji, sedangkan banyaknya kesalahan prediksi terjadi pada class KS I sehingga akurasinya menurun.

5.3 Hasil Uji Coba Skenario 3

Skenario 3 menguji validitas yakni melakukan pelatihan dan pengujian menggunakan data yang sama. Hasil pengujian skenario 3 ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Uji Coba Sistem Skenario 3

Data Uji	Akurasi
8 data	100%
16 data	100%
24 data	100%
32 data	100%

Hasil uji coba menunjukkan bahwa penerapan algoritma ID3 sudah benar dimana dapat dilihat dari hasil pengujian menunjukkan akurasi 100%.

5.4 Hasil Uji Coba Skenario 4

Skenario 4 menguji data tahapan keluarga sejahtera namun dengan mengganti semua data missing value dengan data ya. Hasil uji coba skenario 4 ditunjukkan pada tabel 5.4.

Tabel 5.4 Hasil Uji Coba Sistem Skenario 4

Data Latih	Aturan yang Terbentuk		Akurasi	
	Data Seimbang	Data Tidak Seimbang	Data Seimbang	Data Tidak Seimbang
28 data	8	9	88%	88%
52 data	11	13	88%	80%
76 data	11	10	88%	84%
100 data	11	10	88%	84%

Hasil uji coba skenario 4 jika dibandingkan dengan skenario 1 lebih baik skenario 4. Akurasi tertinggi sama yakni 88% namun rata – rata akurasi yang didapatkan lebih baik. Rata – rata akurasi skenario 1 pada data seimbang mencapai 82% dan pada data tidak seimbang mencapai 75%, sedangkan pada skenario 4 rata – rata akurasi pada data seimbang mencapai 88% dan pada data tidak seimbang mencapai 84%.

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari skripsi ini adalah :

1. Decision Tree dengan algoritma Iterative Dichotomiser 3 merupakan metode yang mampu digunakan secara baik dalam pengklasifikasian status tahapan keluarga sejahtera dengan penanganan data kosong (*missing value*) di jadikan/dianggap suatu nilai yang baru/tersendiri berdasarkan tahapan proses yang dilalui, dimana tahapan proses yang dilalui antara lain proses pembacaan data, perhitungan entropy, perhitungan information gain, mendapatkan nilai information gain tertinggi, membentuk tree, menelusuri tree untuk memperoleh aturan yang diakhiri dengan pengujian berdasarkan skenario uji coba.
2. Jumlah aturan terbanyak yang didapatkan adalah 14 aturan yakni pada 52 data latih yang tidak seimbang, sedangkan yang paling sedikit adalah 8 aturan yang didapatkan pada 28 data latih seimbang dan tidak seimbang. Nilai akurasi tertinggi sebesar 88% yakni pada 28 data latih seimbang, sedangkan akurasi terendah sebesar 68% pada 52 data latih tidak seimbang.
3. Variasi data sangat berpengaruh terhadap tree yang terbentuk serta akurasi yang didapat. Jumlah data latih tidak mempengaruhi bentuk tree jika variasi data tidak berbeda.

6.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian skripsi ini adalah :

1. Penggunaan metode tambahan untuk penanganan *missing value* agar hasil yang didapatkan lebih baik.
2. Penambahan metode untuk mengatasi ketidak seimbangan data latih karena data dilapangan tidak selalu seimbang.

DAFTAR PUSTAKA

- [BKB-12] Badan Kependudukan dan Keluarga Berencana Nasional. 2012. "Profil Hasil Pendataan Keluarga Tahun 2012", Direktorat Pelaporan dan Statistik, Jakarta.
- [BHA-09] Bahety, Anand. 2009. "Extension and Evaluation of ID3 – Decision Tree Algorithm", University of Maryland, College Park.
- [RGD-14] Dona Rashmi, G., Uma Maheswari, K., DR. Narayami, V. 2014, "Analytical Research in the Geographical Area for classifying Childhood Obesity Using ID3", IPASJ, Volume 2 Issue 5.
- [HAN-01] Han, Jiawei dan Micheline Khamber. 2001. "Data Mining : Concepts and Technique", Morgan Kaufmann Publisher, San Francisco, USA.
- [KOH-99] Kohavi, R., Quinlan. 1999. "Decision Tree Discovery", AAAI and The MIT, Pres, 1-16.
- [NOC-12] Kumar, Nilmar., et al. 2012, "An application of ID3 Decision Tree Algorithm in land capability classification", Agropedology, Nagpur..
- [LIG-05] Liang, G. 2005. "A Comparative Study of Three Decision Tree Algorithms: ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3", Informatics & Economics Erasmus University Rotterdam, The Netherlands.
- [LIN-06] Ling, Bai. & Wei, Guo Zhu. 2006, *Journal of Cancer Molecules* 2, No.4, hal. 141-153, Guilin, China.
- [IUM-14] Maulina, Inas Ulvy. 2014, "Klasifikasi Kanker Payudara Menggunakan Decision Tree dengan Algoritma Iterative Dichotomiser 3 (ID3)", Universitas Brawijaya, Malang.
- [MIT-97] Mitchell, Tom. 1997. "Machine Learning", Singapore, McGraw-Hill.

- [RSS-13] R, Sathiyaraj., V, Sujatha. 2013. “Research Scholars Evaluation Based on Guides View Using ID3”, IJRET, Volume 2 Issue 10.
- [KBR-13] Rinarto, Kingkin Bangkit. 2013. “ Fuzzy Decision Tree dengan Algoritma Iterative Dichotomiser 3 (ID3) untuk Klasifikasi Gangguan Hati”, Universitas Brawijaya, Malang.
- [SUK-11] Suknovic, M., et al. 2011, “Reusable Component in Decision Tree Induction Algorithms”, Springer-Verlag.
- [TAN-06] Tan P. N. 2006. “Introduction to Data Mining”, Addison Wesley.
- [WYD-09] Wahyudin. 2009. “Metode *Iterative Dichotomizer 3 (ID3)* Untuk Penyeleksian Penerimaan Mahasiswa Baru”, IC²T, Volume 2 Nomor 2, Bandung.



Biodata Penulis



Nama : Selly Johansyah Isamuddin

NIM : 105090600111037

Program Studi : Ilmu Komputer / Informatika

Bidang Minat : Komputasi Cerdas

Email : selly.johansyah.i@gmail.com

No. HP : +6285730026857

Alamat : Dsn. Medan Bhakti RT 01 RW 04 Ds. Sumobito Jombang

Riwayat Pendidikan

- 1998-2004 : MI Khoiriyah Sumobito
- 2004-2007 : SMPN 1 Sumobito
- 2007-2010 : SMAN 1 Jombang
- 2010-2015 : Ilmu Komputer, Universitas Brawijaya