

**IMPLEMENTASI METODE KLASIFIKASI  
DECISION TREE SLIQ  
PADA SISTEM EVALUASI MOBIL**

**SKRIPSI**

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

Oleh :

**ENGGAR WISMANING ASTITI  
0610960023 - 96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA FAKULTAS MATEMATIKA DAN  
ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2010**

UNIVERSITAS BRAWIJAYA



**LEMBAR PENGESAHAN SKRIPSI**  
**IMPLEMENTASI METODE KLASIFIKASI**  
***DECISION TREE SLIQ***  
**PADA SISTEM EVALUASI MOBIL**

Oleh:

**ENGGAR WISMANING ASTITI**  
**0610960023 - 96**

Setelah dipertahankan di depan Majelis Penguji  
Pada tanggal 12 Agustus 2011  
dan dinyatakan memenuhi syarat untuk memperoleh gelar  
**Sarjana Komputer dalam Bidang Ilmu Komputer**

Pembimbing I,

Pembimbing II,

Lailil Muflikhah S.Kom., MSc  
**NIP. 197411132005012001**

Edy Santoso, SSI., M.Kom  
**197404142003121004**

Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc  
**NIP.196908071994121001**

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Enggar Wismaning Astiti  
NIM : 0610960023-96  
Jurusan : Matematika  
Program Studi : Ilmu Komputer  
Penulis skripsi berjudul : Implementasi metode klasifikasi *decision tree* SLIQ pada sistem evaluasi mobil

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 12 Agustus 2011  
Yang menyatakan,

Enggar Wismaning Astiti  
NIM.0610960023-96

UNIVERSITAS BRAWIJAYA



# IMPLEMENTASI METODE KLASIFIKASI *DECISION TREE SLIQ* PADA SISTEM EVALUASI MOBIL

## ABSTRAK

Klasifikasi yang menyatakan mobil itu bagus atau tidak untuk dipilih biasanya masih dilakukan secara manual. Dalam memilih mobil ada beberapa kriteria yang sangat penting, yaitu harga pembelian, harga pemeliharaan, jumlah pintu, jumlah penumpang, kapasitas bagasi dan keamanan. Untuk itulah digunakan metode klasifikasi untuk membantu dalam melakukan evaluasi terhadap kriteria-kriteria mobil tersebut.

Proses pengklasifikasian data merupakan salah satu tugas (*task*) pada data mining. Ada banyak metode untuk membangun model klasifikasi, salah satunya adalah menggunakan metode *decision tree* dengan algoritma SLIQ (Supervised Learning in Quest). Algoritma ini dapat membangun *tree* yang rapat dan akurat karena SLIQ menggunakan strategi *breadth-first tree growing*, yaitu split pada semua *node* dari *tree* akan dievaluasi secara simultan setiap satu data dimasukkan, serta menggunakan teknik perhitungan gini indeks untuk pemilihan atribut terbaik.

Tahapan dalam mengimplementasikan algoritma SLIQ adalah menyusun kelas histogram yaitu jumlah frekuensi kelas terhadap nilai setiap atribut, penentuan split node dengan mencari nilai gini indeks atribut terkecil, pembentukan *tree* dan yang terakhir adalah pembentukan *rule*.

Penelitian ini membuktikan bahwa, *data mining* dengan metode klasifikasi *decision tree* SLIQ dapat diimplementasikan pada sistem evaluasi mobil. Tingkat akurasi yang dihasilkan oleh sistem ini mencapai 82 % pada jumlah data *training* 1678. Dan disimpulkan bahwa semakin bertambahnya data *training* maka kebenaran sistem dalam pemberian evaluasi akan semakin meningkat, hal ini disebabkan oleh semakin banyaknya *rule* yang dibentuk oleh sistem, sehingga memperbanyak informasi yang dihasilkan.

Kata kunci : sistem evaluasi mobil, *decision tree*, SLIQ

UNIVERSITAS BRAWIJAYA



# IMPLEMENTATION OF CLASIFICATION METHOD DECISION TREE SLIQ ON CAR EVALUATION SYSTEM

## ABSTRACT

Classification to state whether a car is good or not to be chosen usually is still conducted manually. In choosing a car, there are several important factors, that is the price, the maintenance cost, number of doors, passenger capacity, baggage capacity, and the safety. Therefore the method of classification is used to help doing the evaluation on those car criterias.

The data clasification process is one of the task on data mining.. There are a lot of methods to build classification method, one of them is by using *decision tree* with SLIQ algorithm (Supervised Learning in Quest). This algorithm can build tight and accurate tree because SLIQ uses breadth-first tree growing strategy that is, split on all nodes from every tree will be evaluated simultaneously to every data which is entered, and using gini index calculation technique to do the best attribute selection.

Steps in implementing SLIQ algorithm are compiling histogram, that is the number of class frequency for every attribute, deciding node split by finding out the gini index value of smallest attribute, tree formation, and the last is rule formation.

This study proves that, data mining with decision tree method which uses SLIQ algorithm could be implemented on the car evaluation system. The level of accuracy produced by this system could reach 82% for 1678 training data. And it is concluded that if the training data increases then the acuration of the system in the evaluation process also increases. This happens because the rule established in the system increases and it multiplied the resulting information.

Keywords : car evaluation system, decision tree, SLIQ

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Tuhan Yesus Kristus yang telah memberikan anugrah dan hikmatNya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Implementasi metode klasifikasi *decision tree* SLIQ pada sistem evaluasi mobil”. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Skripsi ini bertujuan untuk menerapkan *decision tree* sliq sebagai alternatif solusi pengklasifikasian data *car evaluation system* dengan menggunakan perangkat lunak.

Dalam penyusunan dan penyelesaian skripsi ini, Penulis mengucapkan terima kasih kepada :

1. Lailil Muflikhah, S.Kom., MSc selaku pembimbing utama dalam penulisan skripsi ini.
2. Edy Santoso, SSI., M.Kom selaku pembimbing pendamping dalam penulisan skripsi ini.
3. Nurul Hidayat, SPd., MSc selaku dosen pembimbing akademis.
4. Mardji, MT selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
5. Dr. Abdul Rouf Alghofari, M.Sc selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Brawijaya.
6. Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer.
7. Keluarga penulis tercinta, Ibu, Bapak dan adek terimakasih untuk kasih, doa, dukungan dan segalanya yang diberikan kepada penulis.
8. Keluarga Pakde Bambang Parikeno yang telah membantu doa dan biaya studi Penulis.
9. Bobby Rizky Handoko, semoga Tuhan senantiasa memberkati dan menuntun setiap langkahmu, terimakasih untuk kesabarannya dalam mendukung, menjaga dan membimbing penulis.

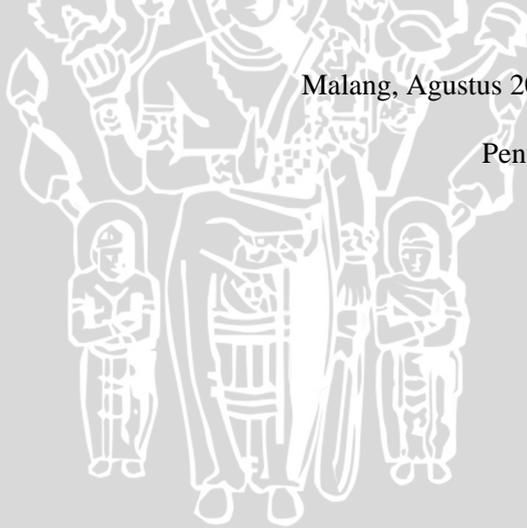
10. Dian, Rakanita, Mas Adi, Surya dan semua teman-teman ILKOM 2006, terimakasih atas kebersamaan, bantuan dan dukungannya selama ini.
11. Seluruh sahabat-sahabat “Youth GSJA Maranatha” , Kak Fery, Ce Lisa, Devina, Ko Ronny, Siska, Moris, Arum, Rino, Andrew dan Tifuk. Terimakasih untuk doa dan dukungan yang diberikan kepada penulis.
12. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih banyak kekurangan dalam laporan ini, oleh karena itu Penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi skripsi ini untuk kelanjutan penelitian serupa di masa mendatang.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan di masa mendatang.

Malang, Agustus 2011

Penulis



## DAFTAR ISI

	HALAMAN
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iii</b>
<b>LEMBAR PERNYATAAN</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>xi</b>
<b>DAFTAR ISI</b> .....	<b>xii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>DAFTAR TABEL</b> .....	<b>xix</b>
<b>DAFTAR SOURCECODE</b> .....	<b>xx</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xxi</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Data Mining .....	5
2.1.1 Metode Dalam Data Mining .....	6
2.1.2 Klasifikasi .....	7
2.1.3 <i>Decision tree</i> .....	10
2.1.3.1 Pembangunan Pohon ( <i>Tree Building</i> ).....	11
2.1.3.2 Algoritma <i>Decision tree</i> .....	13
2.2 <i>Supervised Learning in Quest (SLIQ)</i> .....	13
2.2.1 Memproses <i>split node</i> .....	14
2.2.2 Menyusun <i>Histogram Class</i> .....	15
2.4 Sistem Evaluasi Mobil.....	17
2.3 Pengujian Sistem .....	18
<b>BAB III METODOLOGI DAN PERANCANGAN SISTEM</b>	
3.1 Analisis Data .....	20

3.2 Analisis Perangkat Lunak.....	21
3.2.1 Deskripsi Sistem.....	21
3.2.2 Batasan Sistem.....	22
3.3 Perancangan Sistem.....	22
3.4.1 Perancangan Proses.....	22
3.4.2 Perancangan Basis Data.....	29
3.4 Contoh Perhitungan Manual.....	29
3.5 Perancangan Antar Muka.....	36
3.5.1 Form Menu Utama.....	36
3.5.2 Form Input Data.....	36
3.5.3 Form Proses Pembelajaran.....	37
3.5.4 Form Sistem Evaluasi Mobil.....	38
3.5.5 Form Pengujian Sistem.....	39
3.10 Perancangan Pengujian dan Analisis.....	42

**BAB IV IMPLEMENTASI DAN PEMBAHASAN**

4.1 Lingkungan Implementasi.....	43
4.1.1 Lingkungan Implementasi Perangkat Keras.....	43
4.1.2 Lingkungan Implementasi Perangkat Lunak.....	43
4.2 Implementasi Program.....	43
4.2.1 Struktur Data.....	43
4.2.2 Menyusun Kelas Histogram.....	44
4.2.3 Proses Penghitungan Gini Indeks.....	46
4.2.3 Proses Pemilihan Atribut Terbaik.....	47
4.2.3 Proses Pembentukan Pohon.....	49
4.2.3 Proses Pembentukan Rule.....	49
4.3 Penerapan Aplikasi.....	52
4.5 Pengujian dan Analisa Hasil.....	56

**BAB IV PENUTUP**

5.1 Kesimpulan.....	61
5.2 Saran.....	61

**DAFTAR PUSTAKA.....63**

**LAMPIRAN.....65**

## DAFTAR GAMBAR

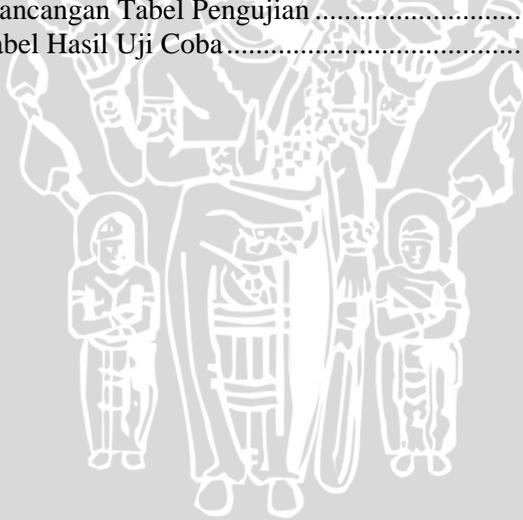
Gambar 2.1	<i>Supervised Learning</i> .....	6
Gambar 2.2	<i>UnSupervised Learning</i> .....	8
Gambar 2.3	Proses Pembangunan Model.....	9
Gambar 2.4	Proses Penggunaan Model.....	9
Gambar 2.5	Contoh <i>Decision Tree</i> .....	11
Gambar 2.6	Contoh <i>Histogram Class</i> untuk Atribut Katagorik16	
Gambar 2.7	Contoh <i>Histogram Class</i> untuk Atribut numerik..	16
Gambar 3.1	Alur Penelitian .....	19
Gambar 3.2	<i>Flowchart</i> Tahap Algoritma SLIQ .....	23
Gambar 3.3	<i>Flowchart</i> Tahap Pemebentukan <i>tree</i> .....	25
Gambar 3.4	<i>Flowchart</i> Tahap Penentuan <i>Split Node</i> .....	26
Gambar 3.5	<i>Flowchart</i> Proses Evaluasi Mobil.....	27
Gambar 3.6	Hasil <i>Tree</i> perhitungan manual.....	35
Gambar 3.7	<i>Form</i> Menu utama .....	36
Gambar 3.8	<i>Form</i> Menu Input Data .....	37
Gambar 3.9	<i>Form</i> Menu <i>training</i> .....	38
Gambar 3.10	<i>Form</i> Menu Evaluasi Mobil.....	39
Gambar 3.11	<i>Form</i> Menu Testing .....	40
Gambar 4.1	Proses Input Data.....	52
Gambar 4.2	Proses <i>Training</i> .....	53
Gambar 4.3	Proses Hitung Gini Indeks .....	54
Gambar 4.4	Proses Testing .....	55
Gambar 4.5	Proses Evaluasi Mobil .....	56
Gambar 4.6	Grafik Tingkat Akurasi Metode SLIQ.....	58
Gambar 4.7	Grafik Jumlah Rule Metode SLIQ.....	59

UNIVERSITAS BRAWIJAYA



## DAFTAR TABEL

Tabel 3.1	Format Tabel Data <i>Training</i> .....	28
Tabel 3.2	Format Tabel Testing dan Hasil <i>Rule</i> .....	28
Tabel 3.3	Contoh Data <i>Training</i> .....	29
Tabel 3.4	Histogram <i>class</i> atribut <i>person</i> dengan nilai batas 2 ..	29
Tabel 3.5	Histogram <i>class</i> atribut <i>person</i> dengan nilai batas 4 ..	30
Tabel 3.6	Histogram <i>class</i> atribut <i>person</i> dengan nilai batas 5 ..	30
Tabel 3.7	Histogram <i>class</i> atribut <i>buying</i> .....	30
Tabel 3.8	Tabel Hasil Gini Indeks.....	31
Tabel 3.9	Tabel Data <i>Lugboot</i> = <i>small</i> .....	32
Tabel 3.10	Tabel Data <i>Lugboot</i> = <i>med</i> .....	33
Tabel 3.11	Tabel Data <i>Lugboot</i> = <i>med</i> dan <i>Safety</i> = <i>low</i> .....	33
Tabel 3.12	Tabel Data <i>Lugboot</i> = <i>med</i> dan <i>Safety</i> = <i>med</i> .....	33
Tabel 3.13	Tabel Data <i>Lugboot</i> = <i>med</i> dan <i>Safety</i> = <i>high</i> .....	34
Tabel 3.14	Tabel Data <i>Lugboot</i> = <i>high</i> .....	34
Tabel 3.15	Rancangan Tabel Data Pembelajaran .....	41
Tabel 3.16	Rancangan Tabel Pengujian .....	41
Tabel 4.1	Tabel Hasil Uji Coba .....	57



UNIVERSITAS BRAWIJAYA



## DAFTAR SOURCECODE

Sourcecode 2.1	Algoritma Induksi .....	12
Sourcecode 2.2	Pemrosesan .....	14
Sourcecode 4.1	Struktur Data.....	43
Sourcecode 4.2	Proses Penghitungan Frekuensi Kelas dan Penyusunan Histogram .....	44
Sourcecode 4.3	Proses Penghitungan Gini Indeks .....	46
Sourcecode 4.4	Proses Penentuan Atribut terbaik.....	47
Sourcecode 4.5	Proses Pembentukan pohon .....	49
Sourcecode 4.6	Prosedur Pembentukan Rule .....	51



UNIVERSITAS BRAWIJAYA



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Banyak sekali jenis mobil yang ditawarkan di pasaran pada saat ini. Setiap produk mobil memiliki kelemahan dan keunggulan masing-masing yang ditinjau dari beberapa kriteria. Oleh karena itu memahami konsep dalam memilih sebuah mobil sangatlah penting terutama bagi pembeli pertama yang belum mengerti industri otomotif.

Pada jurnal penelitian yang berjudul “*Decision Model for Car Evaluation*” menjelaskan bahwa untuk melakukan klasifikasi yang menyatakan mobil itu bagus atau tidak untuk dipilih biasanya masih dilakukan secara manual dengan bantuan mekanik atau sales yang menyarankan untuk memilih mobil tertentu atau pendapat dari keluarga dan teman-teman yang sebelumnya berpengalaman dengan masalah mobil. Tetapi tidak sadar pembeli mungkin sering mengabaikan faktor-faktor yang sebenarnya sangat penting yaitu harga, kenyamanan dan keamanan dalam jangka panjang. Maka perlu diciptakan sebuah sistem evaluasi mobil yang dapat membantu mengatasi permasalahan tersebut ( Ronald, 2004).

Pada penelitian ini sistem evaluasi dibangun menggunakan teknik klasifikasi data. Proses pengklasifikasian data merupakan salah satu tugas (*task*) pada data mining. Data mining merupakan salah satu teknik pengolahan data dengan jumlah yang besar yang bertujuan untuk mendapatkan informasi mengelompokan data ke dalam sejumlah kelas. Proses klasifikasi sendiri dibagi ke dalam dua langkah, yaitu pembentukan model klasifikasi yang disebut *classifier* dan penggunaan model klasifikasi yang disebut klasifikasi, terhadap sekumpulan data (sampel) yang terdiri dari beberapa atribut dan kelas (katagori) yang disebut *data training*. Kemudian dari atribut-atribut data tersebut dibangun sebuah model klasifikasi yang berisi aturan- aturan (*rules*) untuk menentukan kelas dari data. Sedangkan pada proses penggunaan model klasifikasi, model klasifikasi berupa sekumpulan aturan (*rules*) yang terbentuk akan digunakan untuk memprediksi kelas dari data baru yang kelasnya belum diketahui.

Pada penelitian sebelumnya sistem evaluasi mobil pernah di implementasikan menggunakan algoritma *decision tree* C4.5 dan

metode pemangkasan *pessimistic eror pruning*. Tingkat akurasi yang dihasilkan mencapai 71,99 % (Bagus Hari M., 2009).

Selain C4.5 terdapat algoritma *decision tree* yang lain, yaitu SLIQ (*Supervised Learning in Quest*). Algoritma SLIQ merupakan algoritma *decision tree* perbaikan dari algoritma C4.5 yang dikembangkan oleh tim proyek IBM's Quest. Kemunculan algoritma ini merupakan jawaban dari kekurangan algoritma-algoritma sebelumnya yang memiliki keterbatasan memori untuk *dataset* dalam jumlah yang besar. *Decision tree* dalam SLIQ menggunakan teknik yang dapat mempersingkat *learning time* dengan tetap mempertahankan tingkat akurasi yang tinggi. Algoritma ini dapat membangun *tree* yang rapat dan akurat karena SLIQ menggunakan strategi *breadth-first tree growing*, yaitu split pada semua *node* dari *tree* akan dievaluasi secara simultan setiap satu data dimasukkan, serta menggunakan teknik perhitungan gini indeks untuk pemilihan atribut terbaik (Manish Mehta, 1996).

Berdasarkan penjelasan diatas maka dalam skripsi ini akan dilakukan penelitian untuk mengimplementasikan metode klasifikasi *decision tree* menggunakan algoritma SLIQ untuk pembangunan sistem evaluasi mobil. Pengimplementasian tersebut juga diharapkan untuk mengetahui tingkat akurasi sistem dan jumlah *rule* yang dihasilkan.

## 1.2 Rumusan Masalah

Rumusan masalah pada skripsi ini, adalah :

1. Bagaimana implementasi metode klasifikasi *decision tree* SLIQ pada sistem evaluasi mobil.
2. Bagaimana tingkat akurasi sistem evaluasi mobil yang dihasilkan terhadap sejumlah *data test*.
3. Bagaimana pengaruh penambahan jumlah data training terhadap tingkat akurasi apakah semakin meningkat pula.

## 1.3 Batasan Masalah

Dalam pengerjaan tugas akhir ini batasan masalah yang diberikan untuk menghindari meluasnya permasalahan adalah sebagai berikut :

1. Dataset yang digunakan merupakan data nyata "*Car evaluation*" yang di ambil dari homepage UCI *Machine Learning Repository*.

2. Perhitungan data yang dilakukan tanpa missing value.

#### **1.4 Tujuan**

Tujuan yang hendak dicapai penulis dalam tugas akhir ini adalah :

1. Mengimplementasikan metode klasifikasi *decision tree* SLIQ pada sistem evaluasi mobil.
2. Mengetahui tingkat akurasi pada sistem evaluasi mobil yang dihasilkan oleh metode tersebut terhadap sejumlah data latih.
3. Mengetahui apakah dengan bertambahnya jumlah data *training* maka tingkat akurasi semakin meningkat pula.

#### **1.5 Manfaat**

Manfaat yang dapat diperoleh dari skripsi ini adalah dapat mengklasifikasikan kelas mobil antara mobil yang dapat diterima dengan baik sampai tidak diterima berdasarkan data *car evaluation* menggunakan metode klasifikasi *decision tree* SLIQ.

#### **1.6 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini dibagi menjadi lima bab dengan masing – masing bab diuraikan sebagai berikut :

1. BAB I PENDAHULUAN  
Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA  
Menguraikan teori – teori yang berhubungan dengan *sistem* evaluasi mobil menggunakan metode klasifikasi *decision tree* SLIQ.
3. BAB III METODE PENELITIAN  
Pada bab ini akan dijelaskan mengenai metode–metode yang digunakan dan langkah– langkah yang harus dilakukan dalam penelitian ini.
4. BAB IV HASIL DAN PEMBAHASAN  
Pada Bab ini akan dijelaskan mengenai proses implementasi sistem, aplikasi, pengujian, dan analisis sistem perangkat lunak yang dibangun.

## 5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian dan saran yang diharapkan bermanfaat untuk pengembangan skripsi ini selanjutnya.

UNIVERSITAS BRAWIJAYA



## BAB II

### DASAR TEORI

Pada bab ini akan membahas tentang dasar teori yang menunjang penelitian, yaitu mengenai *data mining* meliputi definisi *data mining*, proses dalam *data mining*, dan metode dalam *data mining*. Selain membahas tentang data mining pada bab ini juga membahas tentang klasifikasi, *decision tree* dan algoritma *SLIQ*.

#### 2.1 Data Mining

Terdapat beberapa definisi mengenai data mining. Menurut Giri (2003), definisi sederhana dari data mining adalah ekstraksi informasi atau pola yang penting dan menarik dari data yang ada di database skala besar. Dalam jurnal ilmiah, data mining adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui, secara manual dari sekumpulan data.

Sedangkan menurut Menurut Hand (2001), Data mining merupakan sebuah proses menganalisis sekumpulan data hasil penelitian, dengan tujuan untuk menemukan hubungan antar data, dan untuk meringkas data sehingga data menjadi mudah dimengerti dan berguna bagi pemilik data.

Data mining memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelegent*), *machine learning*, statistik dan database (Pramudiono, 2003).

Proses- proses yang terdapat pada data mining, yaitu (Kantardzic, 2003) :

1. Menentukan permasalahan dan menformulasikan hipotesis. Pada praktiknya antara ahli data mining dengan ahli aplikasi.
2. Mengumpulkan data  
Proses ini mengenai tentang bagaimana data dibangkitkan (*generate*) dan dikumpulkan (*collect*).
3. *Preprocessing* Data  
Terdapat dua sub fase dalam *preprocessing* data, yaitu :
  1. *Data Preparation* ⇔ Proses penyiapan data sebelum data diolah (dimodelkan).
  2. *Data-dimensionality reduction* ⇔ Proses peringkasan data
  4. Memperkirakan model.
  5. Menginterpretasikan model dan memberikan kesimpulan

### 2.1.1 Metode dalam data mining

Menurut Kantardzic (2003), berdasarkan tujuannya data mining dapat dibagi menjadi dua yaitu prediksi dan deskripsi. Data mining prediksi bertujuan untuk menghasilkan model yang dapat digunakan untuk melakukan klasifikasi, prediksi, perkiraan (estimation), dan pekerjaan lainnya. Data mining prediksi menggunakan variabel-variabel dari sekumpulan data yang sudah diketahui untuk memprediksi nilai-nilai yang tidak diketahui. Metode-metode yang digunakan dalam data mining prediksi adalah klasifikasi dan regresi. Sedangkan tujuan dari data mining deskripsi adalah untuk mendapatkan pemahaman terhadap sistem yang sudah dianalisa dengan cara menemukan pola dan hubungan di dalam sekumpulan dataset yang besar. Metode-metode yang digunakan dalam data mining deskripsi adalah *clustering* dan *summarization*.

Metode-metode data mining yang lainnya antara lain *dependency modelling*, yaitu metode untuk menemukan model lokal yang dapat menjelaskan ketergantungan antara variabel-variabel dan sekumpulan data dan *change and deviation detection*, yaitu menemukan perubahan yang signifikan di dalam sekumpulan data.

Dari metode-metode yang telah disebutkan di atas, digunakan metode klasifikasi (*classification*) yang merupakan salah satu dari data mining prediksi. Untuk pembahasan lebih lanjut mengenai klasifikasi akan dijelaskan pada subbab berikutnya.

Menurut Zaiane (1999), metode *data mining* dibagi ke dalam dua macam, yaitu :

#### 1. *Supervised Learning* (proses belajar terawasi)

Pada pembelajaran ini target tersedia. Dengan kata lain, diketahui label atau nama dan nomor kelas pada data *training* (supervisi), kemudian data baru diklasifikasikan berdasarkan data *training* tersebut. Contoh : klasifikasi. Gambar 2.1 merupakan ilustrasi dari proses *supervised learning*



Gambar 2.1 *Supervised Learning*

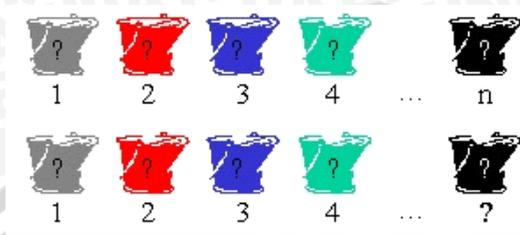
Pada Gambar 2.1 masing-masing gambar sudah memiliki label warna, maka jika ada gambar baru dapat dikelompokkan sesuai dengan label gambar-gambar tersebut.

Kebanyakan metode *supervised data mining* terdiri dari metode membangun model (*building*) dan mengevaluasi model (*evaluating*). Pertama-tama, algoritma *supervised* diberikan sebuah *training set* dari data, yang mana di dalamnya terdiri dari variabel-variabel prediktor (atribut) dan variabel-variabel target (atribut kelas). Nilai dari variabel target yaitu nilai-nilai yang terklasifikasi (kelas). Sebagai contoh : Misalnya akan diklasifikasikan golongan pendapatan (*income bracket*) berdasarkan pada umur, jenis kelamin, dan pekerjaan. Maka, algoritma klasifikasi akan membutuhkan sekumpulan *record-record* yang berisi informasi lengkap dari setiap *field* (kolom), termasuk *field* target (kolom target). Dalam hal ini, variabel prediktor yaitu variabel yang mengisi nilai atribut dari kolom-kolom umur, jenis kelamin, dan pekerjaan. Sedangkan variabel target yaitu nilai-nilai yang mengisi kolom *income bracket*. Dengan kata lain, *record-record* pada *data training* harus telah terklasifikasi sebelumnya. Model dari *data mining* sementara, dibangun dengan menggunakan sampel-sampel *training* yang tersedia dalam *dataset training*.

Langkah berikutnya dalam metode *supervised*, yaitu menguji bagaimana model *data mining* sementara diaplikasikan pada sekumpulan *dataset testing*. Pada *dataset testing*, nilai-nilai dari variabel target disembunyikan sementara dari model *data mining* sementara yang melakukan klasifikasi berdasarkan struktur dan pola yang dipelajari dari *dataset training*. Kemudian efisiensi dari pengklasifikasian dievaluasi dengan cara membandingkan nilai kelas hasil pengklasifikasian menggunakan model *data mining* sementara.

## 2. *Unsupervised Learning* (proses belajar tak terawasi)

Pada proses pembelajaran ini, tidak membutuhkan target untuk keluarannya. Dengan kata lain, tidak terdapat label atau nama kelas pada data *training*, atau bahkan tidak diketahui nomor kelasnya. Data *training* dikelompokkan berdasarkan ukuran kesamaan (*similarity*). Oleh karena itu tidak ada perbandingan yang dilakukan dengan respon ideal yang ditetapkan sebelumnya. Rangkaian pelatihan hanya berisi vektor masukan saja. Contoh : *Clustering*. *Unsupervised learning* dapat ditunjukkan pada Gambar 2.2.



**Gambar 2.2** *Unsupervised Learning*

Pada Gambar 2.2, masing-masing gambar belum memiliki label atau nama, maka perlu dikelompokkan berdasarkan kriteria tertentu.

### 2.1.2 Klasifikasi

Klasifikasi merupakan salah satu tugas *data mining* yang menggunakan metode *supervised*. Klasifikasi dapat didefinisikan sebagai salah satu proses pengelompokkan data (membangun sebuah model) berdasarkan pada *training set* dan kemudian menggunakan model tersebut untuk mengklasifikasikan data baru (*Anonymous*, 2006).

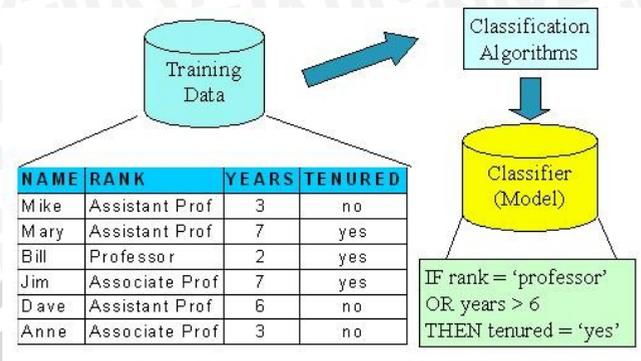
Terdapat dua buah langkah utama dalam klasifikasi, yaitu :

1. Pembangunan model (*model construction*)

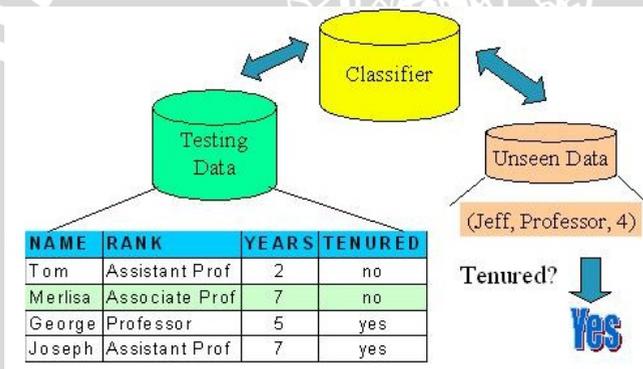
Inti dari proses ini yaitu mendeskripsikan sekumpulan kelas-kelas yang sudah ditentukan sebelumnya (*predetermined class*). Di dalam proses ini, setiap *tuple/sample* diasumsikan termasuk ke dalam sebuah kelas yang sudah terdefinisi sebelumnya (*predefined class*), sebagaimana yang telah ditentukan oleh atribut kelas. Sekumpulan *tuple/sample* yang digunakan dalam pembangunan model ini disebut dengan *training set*. Kemudian, model yang terbentuk dapat direpresentasikan sebagai *classification rules*, *decision tree*, atau sebuah formula matematika.

2. Penggunaan model (*model usage*)

Penggunaan model yaitu untuk mengklasifikasikan objek-objek yang belum diketahui nilai kelasnya (*future or unknown objects*). Sebelum model digunakan untuk mengklasifikasikan objek, akan dilakukan terlebih dahulu perkiraan keakuratan dari model yang terbentuk. Proses klasifikasi diilustrasikan oleh Gambar 2.3 dan 2.4.



**Gambar 2.3** Proses pembangunan model



**Gambar 2.4** Proses penggunaan model

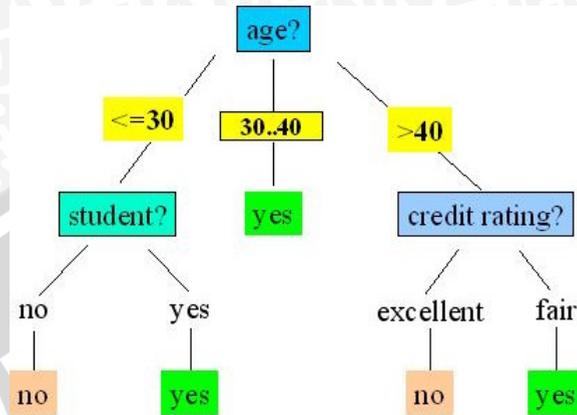
Gambar 2.3, menunjukkan proses pembangunan model, dimana terdapat sejumlah *data training* yang kemudian *data training* tersebut mengalami proses pembelajaran yang dilakukan oleh algoritma klasifikasi. Kemudian dari hasil proses pembelajaran tersebut, terbentuklah *classifier* atau model klasifikasi. Sedangkan Gambar 2.4, menunjukkan proses penggunaan model. Setelah model klasifikasi dihasilkan, model tersebut diuji keakuratannya. Model yang memiliki nilai akurasi yang layaklah yang kemudian digunakan untuk mengklasifikasikan data yang nilai kelasnya belum diketahui.

Terdapat banyak teknik untuk membangun model klasifikasi, salah satu yaitu *decision tree*. Penjelasan mengenai *decision tree* lebih detailnya terdapat pada subbab berikutnya.

### 2.1.3 *Decision tree*

*Decision tree* merupakan salah satu teknik yang digunakan dalam klasifikasi, yang berbentuk struktur *flowchart* yang menyerupai *tree* (pohon) (Han., dkk, 2001). *Decision tree* dibangun oleh dua macam simpul, yaitu simpul (*node*) internal dan simpul daun (*leaf*), serta cabang. Setiap simpul internal (simpul yang memiliki cabang) merepresentasikan *attribute test*, cabang dari sebuah simpul internal berkorespondensi dengan semua hasil yang mungkin dari tes pada simpul. Sedangkan, simpul daun (*leaf*) merepresentasikan nilai dari kelas (Kantardzic, 2003).

Contoh *decision tree* untuk klasifikasi membeli komputer diilustrasikan pada Gambar 2.5. Simpul *age* merupakan simpul akar sekaligus simpul internal (simpul yang berwarna biru) dimana simpul memiliki nilai tes 'age=?'. Cabang-cabang (berwarna putih) yang bernilai  $\leq 30$ ,  $30 \dots 40$ , dan  $> 40$  merupakan cabang yang berkorespondensi dengan hasil dari tes yang dilakukan pada simpul internal tes 'age=?'. Simpul (*node*) *leaf* ditunjukkan oleh simpul berwarna biru muda dan *pink*, yang berarti nilai dari kelas (atribut target). Contoh *decision tree* pada Gambar 2.6 menggambarkan bahwa semua sampel yang memiliki nilai  $age \leq 30$  dan merupakan seorang pelajar (*student*='Yes') atau semua orang yang memiliki nilai  $age > 40$  dan *credit rating*=*fair* akan masuk ke dalam kelas membeli komputer (kelas *buy*='YES'). Begitu pula untuk semua orang yang memiliki umur antara 30 tahun sampai 40 tahun, akan masuk ke dalam kelas 'YES' membeli komputer. Sedangkan untuk orang yang memiliki nilai  $age \leq 30$  dan bukan pelajar (*student*='No') atau orang yang memiliki umur  $> 40$  dan *credit rating*=*exellent*, masuk ke dalam kelas tidak membeli komputer (kelas *buy*='NO') (Anonymous, 2006).



**Gambar 2.5** Contoh *decision tree*

*Decision tree* menangani dua macam tipe atribut, yaitu (Anonymous. 2004) :

1. Numeris atau kontinyu : Domain memiliki nilai tak hingga, direpresentasikan dalam bentuk *real number* (contoh : umur, gaji).
2. Nominal atau kategoris : domain memiliki nilai terhingga (dalam bentuk *finite set*) (contoh : pekerjaan, status pernikahan, ras).

Terdapat tiga hal pada metode *decision tree*, yaitu proses pembangunan *tree*, proses pemangkasan *tree* dan model *decision tree*. Penjelasan dari dua proses tersebut terdapat pada subbab selanjutnya.

### 2.1.3.1 Pembangunan Pohon (*Tree building*)

Dalam tahap pertumbuhan, *tree* dibangun dengan cara membagi data secara rekursif hingga tiap bagian terdiri dari data yang berasal dari kelas yang sama (Tan., dkk, 2004). Pembangunan *tree* dilakukan dengan sebuah algoritma induksi *decision tree*. Rangka algoritma induksi *decision tree* disebut *TreeGrowth* yang terlihat pada algoritma di bawah ini. Masukan terdiri dari *record* pelatihan (*training record*) E dan atribut F. Cara kerja algoritma yaitu dengan memilih atribut yang terbaik untuk memisahkan data secara rekursif dan mengembangkan *node leaf* pada *tree* sampai ditemui kriteria untuk berhenti (Kohavi dan Quinlan, 1999). Algoritma induksi *decision tree* dapat dilihat pada *Sourcecode 2.1*:

*TreeGrowth(E,F)*

1. **if** *stopping\_cond(E,F) = true* **then**
2.     *leaf = createNode ()*
3.     *leaf.label = Clasify(E)*
4.     **return** *leaf*.
5. **else**
6.     *root = createNode ()*
7.     *root.test\_cond = find\_best\_split(E,F)*
8.     *let V = {v/ v is a possible outcome of root.test\_cond}*
9.     **for each**  $v \in V$  **do**
10.          $E_v = \{e \mid \text{root.test\_cond}(e) \text{ and } e \in E\}$
11.         *child = TreeGrowth(E<sub>v</sub>,F)*
12.         add *child* as descendent of *root* and *label* the edge (*root*→*child*) as *v*
13.     **end for**
14. **end if**
15. **return** *root*

**Sourcecode 2.1** Algoritma Induksi

Penjelasan dari rangka algoritma induksi *decision tree* tersebut adalah sebagai berikut:

1. Fungsi **createNode ()** memperluas pohon keputusan (*decision tree*) dengan membuat *node -node* baru. *Node* yang terdapat pada *decision tree* mempunyai sebuah *test condition (attribute test)* yang ditandai dengan ***node.test\_cond***, atau sebuah label kelas yang ditandai dengan ***node.label***.
2. Fungsi **find\_best\_split()** menentukan atribut mana yang harus dipilih sebagai suatu *test condition* untuk membagi data pelatihan (*training record*). Pemilihan *test condition* bergantung pada perhitungan kemurnian (*impurity*) yang digunakan untuk menentukan *split* terbaik (*attribute selection measure*), misalkan : entropy, info Gain, dan Gini index (Anonymous, 2006).
3. Fungsi **Clasify()** menentukan label kelas untuk ditempatkan pada *node leaf*. Untuk setiap *node leaf t*, umpamakan  $p(i | t)$  adalah fraksi dari tiap *record* pelatihan dari kelas *i* pada *node t*. Pada umumnya, *node leaf* menempatkan kelas yang mempunyai

- jumlah mayoritas pada *record* pelatihan,  $leaf.label = \underset{i}{\operatorname{argmax}} p(i|t)$ , dimana operator *argmax* mengembalikan nilai maksimum dari  $p(i|t)$ . Selain menyediakan informasi yang dibutuhkan untuk label kelas dari suatu *node leaf*, fraksi  $p(i|t)$  dapat juga digunakan untuk memperkirakan kemungkinan (probabilitas) suatu *record* ditempatkan dalam *node leaf*  $t$  pada kelas  $i$ .
4. Fungsi **stopping\_cond()** digunakan untuk menghentikan proses pembentukan pohon dengan menguji apakah semua *record* mempunyai label kelas yang sama atau nilai atribut yang sama. Cara lain untuk menghentikan fungsi rekursi pembentukan pohon adalah dengan menguji apakah jumlah atributnya telah habis, jika demikian maka label kelas akan ditentukan dengan perhitungan kelas terbanyak (*majority class*).

### 2.1.3.2 Algoritma Decision Tree

Terdapat beberapa model decision *tree*, yaitu Hunt's Algorithm (*one of the earliest*), CART, CLS, CHAID, ID3, C4.5, SLIQ, SPRINT, namun algoritma yang telah mengalami modifikasi sehingga dapat digunakan untuk dataset yang besar dan mempersingkat learning time adalah SLIQ (*Supervised Learning In Quest*) karena menggunakan teknik *pre-sorting* dan *breadth first growth* pada fase pembentukan pohon. Penjelasan mengenai algoritma SLIQ terdapat pada subbab berikutnya.

## 2.2 *Supervised Learning in Quest (SLIQ)*

Menurut Manish (1996), algoritma SLIQ ini dikembangkan oleh tim proyek IBM's Quest pada sekitar tahun 1996. Algoritma SLIQ menggunakan modifikasi dari *tree classifier* sehingga bisa dipakai juga untuk dataset yang besar. SLIQ bisa dipakai untuk atribut dengan tipe numerik dan kategorikal. *Decision tree* dalam SLIQ menggunakan teknik novel untuk mempersingkat *learning time* dengan tetap mempertahankan tingkat akurasi yang tinggi. Algoritma ini juga tidak mempunyai batasan jumlah *training* data atau jumlah atribut yang digunakan. Dengan demikian SLIQ mempunyai potensi menghasilkan klasifikasi yang lebih akurat pada *training dataset* yang lebih besar, yang tidak dapat dilakukan oleh algoritma-

algoritma sebelumnya. Kesimpulannya, SLIQ bisa dieksekusi dengan lebih cepat dengan *tree* yang dihasilkan lebih kecil.

SLIQ menggunakan teknik pre-sorting di fase *tree-growth* untuk mengurangi cost evaluasi atribut numerik. Prosedur sorting ini diintegrasikan dengan strategi *breadth-first tree growing* untuk memungkinkan SLIQ melakukan klasifikasi pada dataset yang ada dalam *disk*.

Terdapat beberapa proses untuk membangun pohon dalam algoritma SLIQ yaitu proses *Breadth-First Growth*, proses *split node*, menyusun histogram *class*. Penjelasan lebih lanjut terdapat pada subbab berikutnya.

### 2.2.1 Memproses Split Node

Pada algoritma SLIQ proses *split node* memakai strategi *breadth-first*. Konsekuensinya split pada semua *node* dari *tree* akan dievaluasi secara simultan setiap satu data dimasukkan. *Pseudocode* proses split dituliskan pada *Sourcecode 2.2* (Manish,1996) :

```
EvaluateSplit()
  for each attribute A do
    traverse attribute list of A
    for each vaue v in the attribute list do
      find the corresponding entry in the class list, and
      hence the corresponding class and the node node
(say l)
      update yhe class histogram in the node l
      if A is a numeric attribute then
        compute spitting index for node ( $A \leq v$ )for node l

      if A is a categorical attribute then
        for each node of tree do
          find subset of A with best split
```

**Sourcecode 2.2** Pemrosesan Split Node

Untuk mengevaluasi atribut sebagai alternatif split terbaik, SLIQ menggunakan perhitungan gini indeks.

Gini indeks adalah ukuran dispersi statistik dikembangkan oleh statistik dan sosiolog Italia Corrado gini dan diterbitkan pada tahun 1912 pada bukunya yang berjudul “*Variability and Mutability*”. Gini

indeks adalah ukuran ketimpangan distribusi, nilai 0 mengekspresikan kesetaraan total dan nilai 1 ketidaksetaraan maksimal. Sebuah gini indeks yang rendah menunjukkan distribusi yang lebih setara, dengan 0 yang sesuai untuk menyelesaikan persamaan, sedangkan gini indeks yang lebih tinggi mengindikasikan distribusi yang lebih merata, dengan 1 yang sesuai untuk menyelesaikan ketimpangan.

Untuk menghitung gini *splitting-index* dari sebuah atribut dari suatu *node*, maka dibutuhkan distribusi frekuensi dari nilai kelas yang ada pada data partition yang berkorespondensi dengan *node*. Jika set data  $T$  berisi contoh-contoh dari kelas  $n$ ,  $gini(T)$  didefinisikan pada persamaan 2.1.

$$gini(T1) = 1 - \sum p_j^2 \quad (2.1)$$

Keterangan :

$p_j$  : frekuensi relatif dari kelas  $j$  pada  $T1$ .

Setelah  $T$  membelah menjadi  $n$  subset  $T1 \dots Tn$  gini indeks data split didefinisikan seperti pada persamaan 2.2.

$$gini(T)Split = \frac{T1}{T} \cdot gini(T1) + \dots + \frac{Tn}{T} \cdot gini(Tn) \quad (2.2)$$

Keterangan :

$n$  : jumlah nilai atribut  $T$

Untuk menemukan titik pemecahan terbaik untuk *node*, daftar atribut *node* diperikasa untuk mengevaluasi split untuk atribut. Atribut yang berisi titik split dengan gini indeks nilai terendah digunakan untuk membelah *node record*. Hal ini karena dalam pencarian split node dicari distribusi yang lebih setara, dengan 0 yang menyatakan bahwa node sudah setara atau sudah pure.

### 2.2.3 Menyusun Histogram Class

Distribusi diakumulasikan dalam sebuah *histogram class* yang terhubung pada setiap *leaf node*. Untuk atribut numerik, histogram berbentuk daftar pasangan yang terdiri dari <batas, kelas, frekuensi>. Sedangkan untuk atribut kategorikal, histogram ini berbentuk daftar dari 3 hal yaitu <nilai atribut, kelas, frekuensi>.

Jika A adalah atribut *numerik*, dilakukan perhitungan gini dalam waktu yang bersamaan *splitting-index* untuk  $test A \leq v$  untuk *node* ini. Jika A adalah atribut kategorikal, maka tunggu sampai semua daftar atribut selesai dilakukan *scanning* dan kemudian cari subset dari A dengan split terbaik. Dalam satu traversal dari sebuah daftar atribut, split terbaik yang dipakai pada atribut ini diketahui oleh semua *node node s*. Split terbaik ini lalu disimpan dengan setiap *node node s*.

Penyusunan *histogram class* pada atribut katagorik dengan menghitung jumlah kelas untuk masing nilai atribut. Dan untuk atribut numeric disusun berdasarkan masing - masing batas  $v, A \leq v$  dan  $A > v$ , batas  $v$  split yang dipilih adalah batas  $v$  yang memiliki gini indeks terkecil (Joshi,1999).

Contoh *histogram class* dapat dilihat pada gambar 2.6 dan gambar 2.7.

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Gambar 2.6 Contoh *Histogram class* untuk atribut katagorik

	Cheat		No		Yes		No		No		No	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
S	Taxable Income											
IS	60	70	75	85	90	95	100	120	125	172	230	
	55	65	72	80	87	92	97	110	122	172	230	
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0	0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.393	0.343	0.375	0.400	0.420	

Gambar 2.7 Contoh *Histogram class* untuk atribut numerik

Dalam pembentukan *tree*, proses penyusunan histogram dan split node dilakukan berulang-ulang sampai setiap *leaf node* menjadi *pure*

*node* (berisi satu *record* yang termasuk dalam satu kelas saja) dan tidak bisa dilakukan *split node* lagi.

## 2.4 Sistem Evaluasi Mobil

Ketika seseorang berkeinginan untuk membeli mobil, maka banyak aspek yang mempengaruhi keputusannya dalam memilih tipe mobil. Faktor seperti harga beli, harga pemeliharaan, kenyamanan dan keamanan penting untuk dipertimbangkan. Sistem evaluasi mobil dilakukan dengan menggunakan database *Car Evaluation* yang merupakan informasi struktural yang sangat penting bahwa seseorang harus melihat fitur dan klasifikasi setiap mobil yang dapat membantu pengambilan keputusan. Karena sifatnya yang mendasar, evaluasi mobil ini sangat informatif khususnya membantu untuk pelatihan praktis dan penemuan metode struktur dalam mendukung pencarian keputusan. Database ini mengklasifikasikan spesifikasi mobil menurut harga, nyaman dan keamanan (Ronald, 2004).

Kesimpulannya sistem evaluasi mobil membantu pengguna untuk mendapatkan petunjuk melalui kelas yang dihasilkan melalui kriteria mobil tersebut.

Keputusan hirarki sistem evaluasi mobil pertama kali di buat oleh M. Bohanec dan V. Rajkovic. Model evaluasi yang dilakukan sesuai dengan konsep berikut :

.Harga	Harga Keseluruhan
.. <i>Buying</i>	Tingkat Harga Beli( <i>low, med, high, vhigh</i> )
.. <i>Maint</i>	Tingkat Harga Pemeliharaan( <i>low, med, high, vhigh</i> )
. <i>Tech</i>	Karakteristik Teknik
.. <i>Comfort</i>	Kenyamanan
... <i>Door</i>	Jumlah Pintu(2,3,4,5/lebih)
... <i>Persons</i>	Jumlah Penumpang(2,4,5/lebih)
... <i>Lugboot</i>	Kapasitas Bagasi ( <i>small, med, big</i> )
.. <i>Safety</i>	Tingkat Keamanan Mobil( <i>low, med, high</i> )

Kelas hasil evaluasi dinyatakan dengan *unaccept (unacc)*, *accept (acc)*, *good* dan *very good (vgood)*. *Unaccept* menyatakan mobil dengan kriteria yang tidak diterima oleh konsumen, *accept* menyatakan mobil dengan kriteria yang mendapat penerimaan oleh

konsumen, *good* menyatakan kriteria mobil tidak hanya diterima tetapi juga dinilai bagus oleh konsumen, dan *very good* menyatakan bahwa kriteria mobil yang mendapat penerimaan dan penilaian sangat bagus oleh konsumen.

## 2.4 Pengujian Sistem

Untuk menguji performansi sebuah model yang dibangun oleh algoritma klasifikasi dapat dilakukan dengan menghitung jumlah dari *test record* yang diprediksi secara benar (akurasi) oleh model tersebut.

Dalam melakukan evaluasi, data dibagi menjadi dua bagian, yang pertama digunakan sebagai data latih dan sisanya digunakan sebagai data uji. Apabila tidak menggunakan data latih dan data uji secara terpisah tingkat akurasi yang diperoleh akan berlebihan. Prosentasi data uji ini biasanya 5% sampai dengan 33%. (Anonymous,1999). Data uji ini dapat dikatakan juga sebagai data sampel dari seluruh data.

Akurasi dinyatakan pada persamaan 2.11 (Tan.,dkk,2004).

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \quad (2.11)$$

Jumlah prediksi benar, yaitu jumlah *record data testing* yang memiliki hasil prediksi menggunakan model klasifikasi yang dibangun dari data *training*, sama dengan nilai kelas *data testing* semula. Jumlah prediks salah, yaitu jumlah *record* data yang memiliki hasil prediksi kelas menggunakan model klasifikasi yang dibangun dari *data training*, tidak sama dengan nilai kelas data awal. Sedangkan jumlah total prediksi, yaitu jumlah keseluruhan *record* yang diklasifikasikan menggunakan model klasifikasi yang dibangun dari data *training* (jumlah prediksi benar + jumlah prediksi salah).

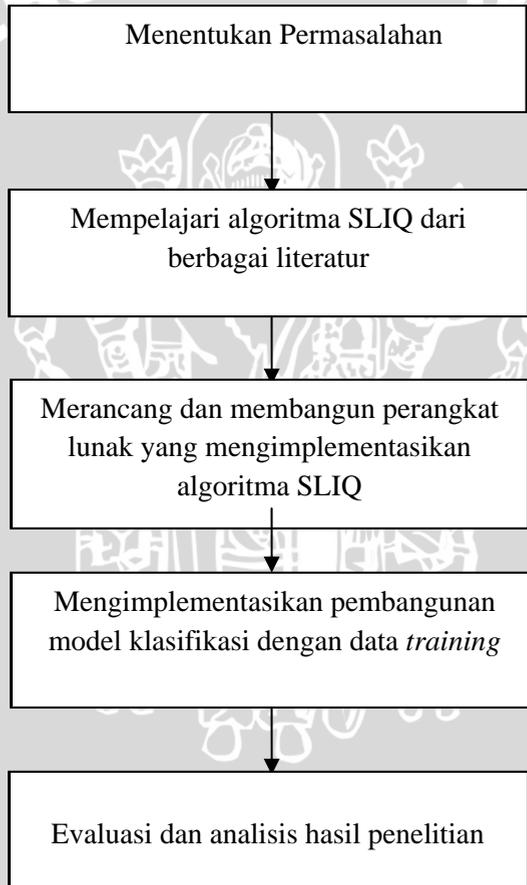
Algoritma klasifikasi berusaha untuk mencari model yang mempunyai akurasi yang tinggi atau *error rate* rendah ketika model diterapkan pada *test set*.

### BAB III

#### METODOLOGI DAN PERANCANGAN

Bab ini menjelaskan metode metode dan langkah-langkah yang digunakan untuk menyusun penelitian dan rancangan dari sistem yang dibuat sebagai perangkat uji coba.

Adapun langkah-langkah yang dilakukan dalam penelitian dapat ditunjukkan pada Gambar 3.1



**Gambar 3.1** Alur Penelitian

Keterangan gambar :

1. Menentukan permasalahan, yaitu bagaimana mengimplementasikan *decision tree* SLIQ untuk membuat model klasifikasi data sistem evaluasi mobil.
2. Mempelajari SLIQ dari berbagai literature. Penelitian ini juga berdasarkan pada penelitian yang dilakukan oleh Fatalla Ronald yang berjudul *Decision Model for Car Evaluation*. Tetapi pada penelitian ini penulis mencoba untuk menerapkan dengan menggunakan algoritma SLIQ.
3. Merancang dan membangun sebuah perangkat lunak yang mengimplementasikan algoritma SLIQ untuk proses model klasifikasi data sistem evaluasi mobil.
4. Mengimplemetasikan algoritma SLIQ untuk pembangunan model klasifikasi data sistem evaluasi mobil.
5. Mengevaluasi hasil dari implementasi, yaitu akurasi dari model klasifikasi (model *tree*) yang dihasilkan menggunakan algoritma SLIQ setelah diujikan dengan data *training*.

### 3.1 Analisis Data

Salah satu tujuan penelitian ini adalah menganalisis tingkat keakurasian pada metode yang digunakan. Dari hasil yang diperoleh, diharapkan dapat memberikan penjelasan mengenai besar tingkat keakurasian yang dimiliki oleh subyek penelitian. Berdasarkan hal tersebut, dapat dikatakan bahwa penelitian ini bersifat prediktif. Penelitian yang bersifat prediktif ialah penelitian yang menggunakan ciri-ciri tertentu dari sebuah data yang dimaksudkan untuk melakukan prediksi atau menjawab pertanyaan yang dipelajari (Tan.,dkk, 2004).

Data yang digunakan pada penelitian ini yaitu kumpulan data tentang evaluasi mobil yang diambil dari situs *UCI Machine Learning Repository* ([www.ics.uci.edu/~mllearn/MLRepository.html](http://www.ics.uci.edu/~mllearn/MLRepository.html)) yang menyediakan data-data nyata untuk digunakan menguji model klasifikasi. Data disimpan pada sebuah tabel. Data memiliki 7 atribut (termasuk atribut kelas) yang terdiri dari 4 atribut katogorik *buying*, *maint*, *lugboot*, *safety* dan 2 atribut numerik *doors* dan *persons*, serta *class* terdiri dari *unaccept*, *accept*, *good*, *very\_good*.

Data kemudian diambil sebanyak 50 *record* untuk digunakan sebagai data *testing*. Kemudian sisanya dibagi menjadi 5 bagian *data training*, dengan komposisi 20%, 40%, 60%, 80%, dan 100 % dari

data keseluruhan dan masing memiliki sebaran kelas yang sama dengan sebaran data keseluruhan.

### 3.2 Analisis Perangkat Lunak

Pada Sub bab ini akan dijelaskan mengenai perangkat lunak yang akan dibangun.

#### 3.2.1 Deskripsi Sistem

Perangkat lunak yang dibangun merupakan sebuah perangkat lunak yang mengimplementasikan dua algoritma *decision tree* yaitu algoritma SLIQ untuk membangun model klasifikasi sistem evaluasi mobil. Perangkat lunak yang dibangun kemudian digunakan untuk proses uji coba tingkat akurasi dari model klasifikasi yang dihasilkan algoritma SLIQ. Hasil dari proses uji coba tingkat akurasi dari model klasifikasi untuk sistem evaluasi mobil, selanjutnya akan dianalisis.

Adapun langkah-langkah ketika *user* menggunakan perangkat lunak yang dibangun, yaitu :

1. *User* menginputkan tabel data *training* dan tabel data *testing* yang disimpan dalam *database*.
2. Sistem mengambil informasi dari tabel data *training* dan data *testing* berupa nama tabel dan jumlah *record*.
3. Sistem melakukan pembangunan model klasifikasi untuk sistem evaluasi mobil menggunakan algoritma SLIQ.
4. Sistem menghasilkan model klasifikasi berupa berupa model *tree* dalam bentuk *rule*.
5. Sistem akan menampilkan hasil evaluasi mobil dari *rule* yang terbentuk.
6. Sistem mengklasifikasikan data *testing* dengan *rule* yang terbentuk untuk mengetahui hasil evaluasi mobil.

#### 3.2.2 Batasan sistem

Adapun batasan pada sistem yang dibangun :

1. Database yang digunakan hanya untuk menyimpan tabel, tidak ada tabel yang saling berelasi.
2. Data tidak memiliki missing value artinya data lengkap, tidak ada nilai atribut yang hilang.

### 3.3 Perancangan Sistem

Pada Sub bab ini akan dijelaskan mengenai berbagai proses yang terjadi pada implementasi pengklasifikasian data dengan *decision tree* SLIQ untuk sistem evaluasi mobil.

#### 3.3.1 Perancangan Proses

Secara umum perancangan proses pada sistem evaluasi mobil ini meliputi pemilihan *request* data “*car evaluation*”, proses pembelajaran dari data “*car evaluation*” dan pengklasifikasian data.

Dengan menggunakan metode SLIQ sistem akan membangun pohon keputusan dengan mengklasifikasikan data *car evaluation* berdasarkan kelasnya. Proses tersebut akan menghasilkan aturan klasifikasi evaluasi mobil. Dalam proses evaluasi mobil, *user* akan memasukan data kriteria mobil yaitu *buying*, *maint*, *doors*, *persons*, *lug\_boot* dan *safety*. Dari masukan tersebut sistem akan menentukan kelas evaluasi yang sesuai dengan membaca aturan klasifikasi yang telah terbentuk. Output dari *sistem* evaluasi adalah jenis kelas evaluasi dari kriteria mobil tersebut.

Proses pertama dilakukan oleh sistem yaitu proses *request* data, dimana sistem akan mendapatkan data untuk diolah, yaitu *data training* dan data *testing* beserta informasinya. Adapun informasi *data training* yang didapatkan yaitu berupa jumlah *record* tabel *data training*. Sedangkan informasi tabel data *testing* yang didapatkan yaitu nama tabel *testing* dan jumlah *record*.

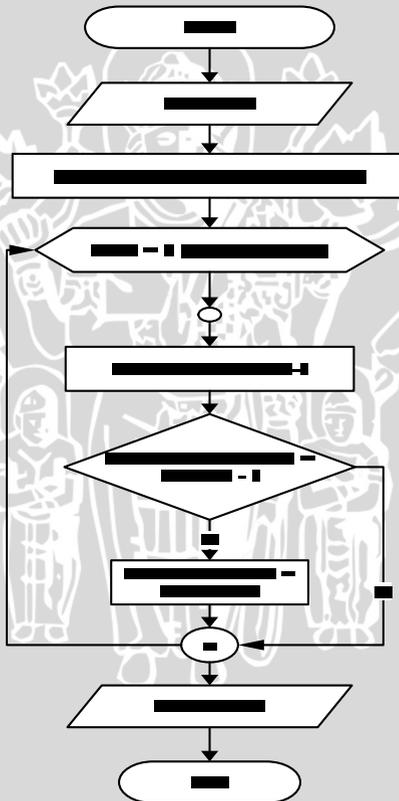
Pada saat *user* pertama kali menjalankan perangkat lunak, *user* akan memilih nama tabel data *training* yang berisi keseluruhan data *car evaluation*. Kemudian respon dari sistem yaitu sistem melakukan koneksi ke DBMS agar aplikasi dapat mengakses tabel yang disimpan dalam *database*. Kemudian sistem menghitung jumlah *record* dari tabel data *training*. Kemudian sistem menyimpan informasi tabel data *training*. Proses terakhir pada *request* data yaitu, sistem menampilkan semua informasi data *training* yang telah disimpan, kepada *user*. Begitu pula saat pengujian sistem maka *user* akan memilih nama tabel *data testing* yang akan digunakan untuk menguji sistem.

Untuk proses pembelajaran dan pengklasifikasian masing-masing metode akan dijelaskan pada subbab berikutnya.

1. Susunan tahap-tahap algoritma SLIQ membentuk aturan klasifikasi adalah sebagai berikut :

- Pembentukan *decision tree* dimulai dengan membaca database *car evaluation*, proses sorting atribut numeric *doors* dilakukan langsung dari database.
- Penentuan *split node*.
- Pembentukan *tree*.
- Setelah proses klasifikasi selesai, sistem menghasilkan aturan klasifikasi dan menyimpan dalam *database*.

Untuk lebih jelasnya dapat dilihat pada Gambar 3.2. *flowchart* tahap pembentukan aturan klasifikasi pada data *training* algoritma SLIQ.

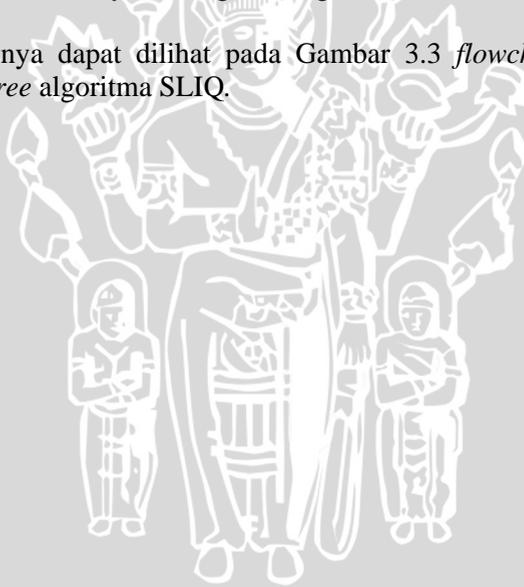


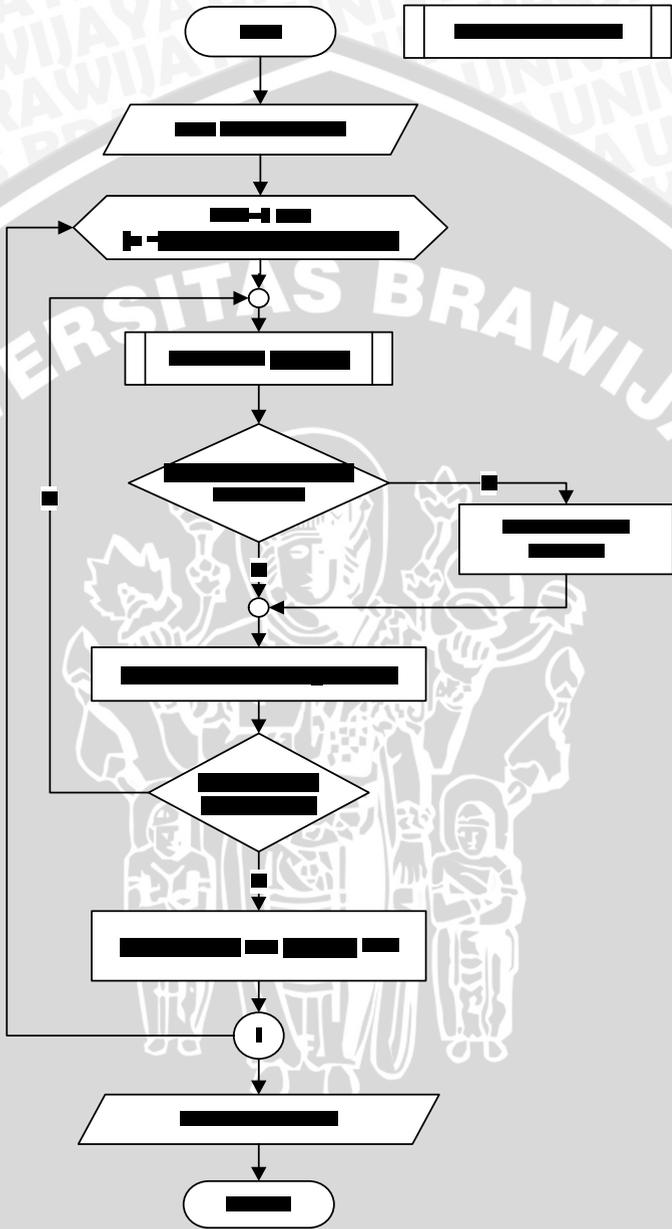
Gambar 3.2 *Flowchart* Tahap Algoritma SLIQ

2. Tahap pembentukan *tree*

- Pertama kali yang dilakukan adalah memanggil *procedure* penentuan split *node*, untuk mencari atribut terpilih sebagai *node* akar.
- Melakukan perulangan terhadap jumlah kriteria atau *leaf* dari split *node*.
- Memanggil *procedure* untuk mencari split *node* dari masing-masing nilai atribut.
- Jika ditemukan split *node* atau kelas, maka menambahkan hasilnya pada *tree*. Jika tidak maka hasilnya adalah kelas terbesar dari data terakhir.
- Jika *leaf node* berupa kelas maka tidak dilakukan *split node* lagi, jika tidak maka akan dilakukan perulangan kembali terhadap jumlah *leaf split node* dan seterusnya sampai semua *leaf* menemukan kelasnya masing-masing.

Untuk lebih jelasnya dapat dilihat pada Gambar 3.3 *flowchart* tahap pembentukan *tree* algoritma SLIQ.



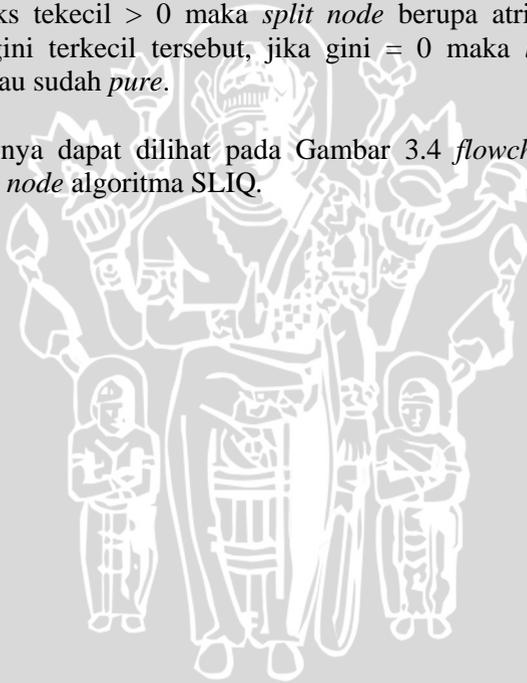


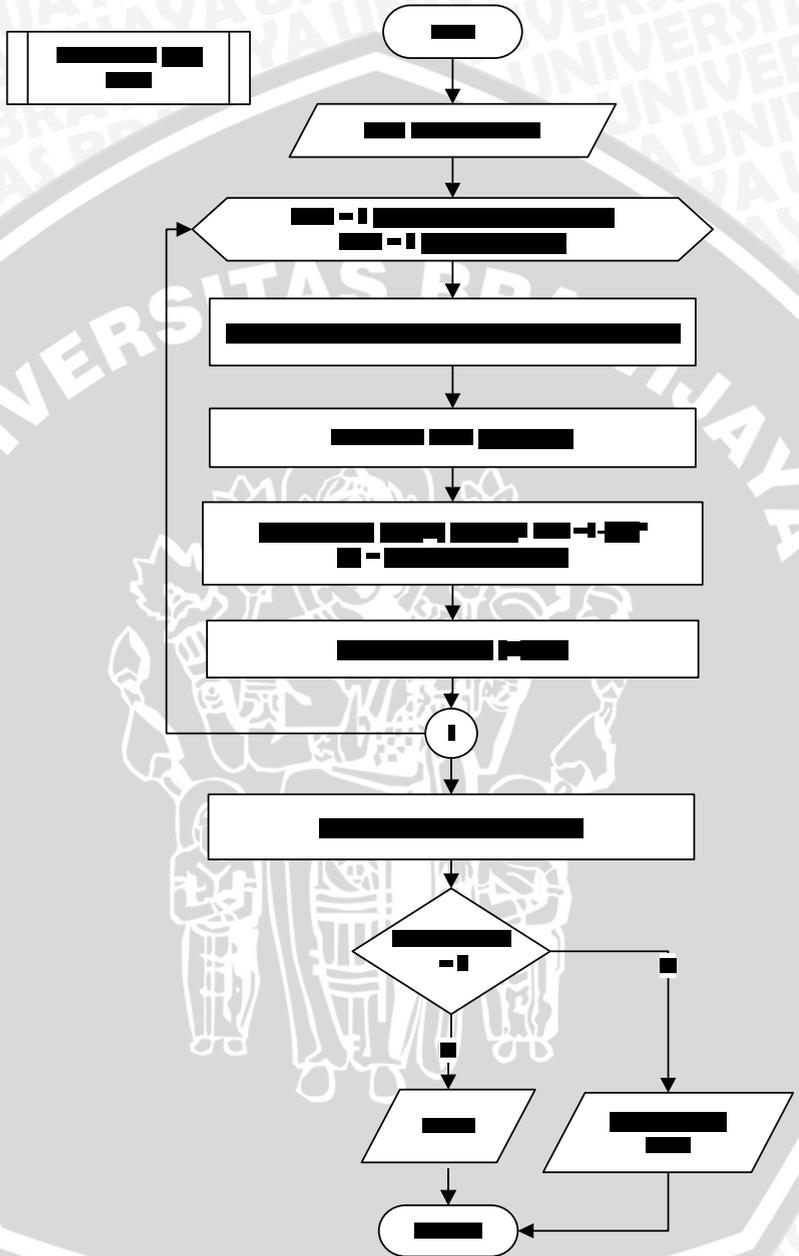
**Gambar 3.3** Flowchart Tahap Pembentukan Tree

### 3. Tahap penentuan *split node*

- Melakukan perulangan terhadap jumlah nilai dan jumlah kelas pada atribut yang akan dihitung gini indeksnya.
- Mengambil frekuensi kelas terhadap nilai dari atribut  
Untuk atribut numeris histogram terdiri dari <batas nilai, kelas, frekuensi>, sedangkan untuk atribut kategoris histogram terdiri dari <nilai atribut, kelas, frekuensi>
- Menghitung *gini-splitting index* tiap atribut dengan persamaan 2.1. menghitung gini tiap *leaf* terlebih dahulu kemudian dicari gini *split* keseluruhan.
- Kemudian, mencari gini indeks terkecil yang dimiliki oleh setiap atribut.
- Jika gini indeks terkecil  $> 0$  maka *split node* berupa atribut dengan nilai gini terkecil tersebut, jika gini = 0 maka *leaf* berupa kelas atau sudah *pure*.

Untuk lebih jelasnya dapat dilihat pada Gambar 3.4 *flowchart* tahap penentuan *split node* algoritma SLIQ.





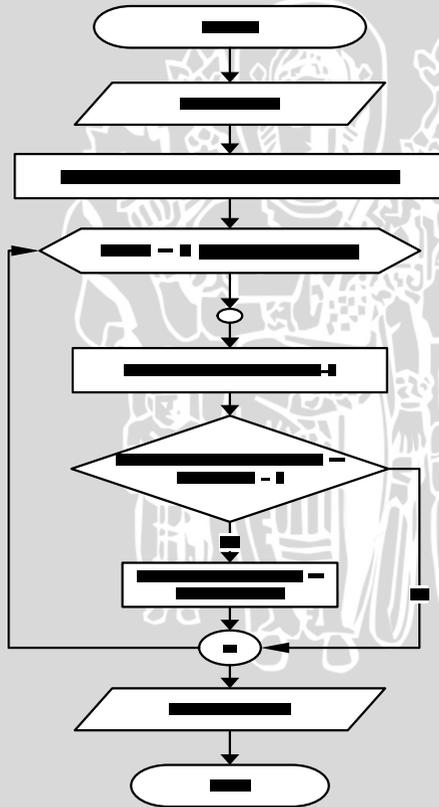
**Gambar 3.4** Flowchart tahap penentuan split node

4. Proses evaluasi kriteria mobil

Dalam proses evaluasi mobil yang dilakukan adalah mengevaluasi mobil berdasarkan kriteria mobil yang diinputkan *user*. Tahapan-tahapan prosesnya adalah sebagai berikut :

- *User* menginputkan data kriteria mobil (*buying, maint, person, door, lug-boot, safety*).
- Sistem akan mengevaluasi criteria tersebut dengan membaca aturan klasifikasi yang telah terbentuk.
- Proses akan dilakukan sampai didapatkan hasil evaluasi.

Proses evaluasi mobil untuk lebih jelasnya dapat dilihat pada gambar 3.5.



**Gambar 3.5** Flowchart Proses Evaluasi Mobil

### 3.4 Perancangan Basis Data

Berdasarkan pada perancangan proses, tidak ada tabel yang saling berelasi. Tabel digunakan hanya untuk menyimpan data, baik data *training*, data *testing* dan *rule*. Struktur tabel data *testing* dan *hasil rule sama*, yang membedakan hanya isi *record* dari tiap-tiap data. Format tabel data *training*, ditunjukkan pada Tabel 3.1 dan format tabel *Testing* dan *hasil rule* ditunjukkan pada tabel 3.2. Format tabel *Testing* dan *hasil rule* memiliki kesamaan tipe data.

Tabel 3.1 Format tabel data *training*

Nama Kolom	Type Data	Batasan	Keterangan
<b>TrainingID</b>	AutoNumber	PK	
<i>Buying</i>	Text		Nama kriteria
<i>Maint</i>	Text		Nama kriteria
<i>Doors</i>	Number		Nama kriteria
<i>Persons</i>	Number		Nama kriteria
<i>Lugboot</i>	Text		Nama kriteria
<i>Safety</i>	Text		Nama kriteria
<i>Class</i>	Text		Nama kriteria

Tabel 3.2 Format tabel *testing* dan *hasil rule*

Nama Kolom	Type Data	Batasan	Keterangan
<b>No</b>	AutoNumber	PK	
<i>Buying</i>	Text		Nama kriteria
<i>Maint</i>	Text		Nama kriteria
<i>Doors</i>	Text		Nama kriteria
<i>Persons</i>	Text		Nama kriteria
<i>Lugboot</i>	Text		Nama kriteria
<i>Safety</i>	Text		Nama kriteria
<i>Class</i>	Text		Nama kriteria

### 3.4 Contoh Perhitungan Manual

Pada subbab ini akan dijelaskan perhitungan manual proses klasifikasi data menggunakan algoritma SLIQ.

Tabel 3.3 merupakan contoh data *training* yang akan di gunakan untuk proses pembentukan pohon.

Tabel 3.3 Contoh Data *Training*

No	Buying	Maint	Persons	Doors	lugboot	safety	class
	vhigh	med	2	2	big	Med	acc
2	vhigh	med	2	3	big	High	acc
3	high	med	4	2	big	Med	acc
4	high	med	4	4	big	High	acc
5	med	vhigh	2	5	big	High	acc
6	med	vhigh	4	5	small	Low	unacc
7	med	med	4	4	med	Med	good
8	med	med	4	4	med	High	vgood
9	med	med	4	3	big	Low	unacc
10	med	med	4	3	big	Med	acc
11	med	med	4	4	big	High	vgood
12	vhigh	vhigh	2	5	big	Low	unacc
13	vhigh	vhigh	2	2	big	Med	unacc
14	low	low	5	6	small	High	good
15	low	low	6	5	med	Low	unacc
16	low	low	5	6	med	Med	good
17	low	low	7	4	med	High	vgood
18	low	low	7	3	big	Low	unacc
19	low	low	8	4	big	Med	good
20	low	low	6	4	big	High	vgood

1. Membuat histogram *class*

Untuk atribut bertipe numeris semua nilai atribut dibuat histogram *class*-nya. Misal histogram *class* pada atribut *person* kelas dengan nilai batas 2,3 dan 4 ditunjukkan pada tabel 3.3, 3.4 dan 3.5.

Tabel 3.4 Histogram *class* atribut *person* dengan nilai batas 2

Person	Left	Right
(2)	<=2	>2
Unacc	2	4
Acc	3	4

Good	0	3
vgood	0	4
Jumlah	5	15

Tabel 3.5 Histogram *class* atribut *person* dengan nilai batas 4

<i>Persons</i> (4)	Left <=4	Right >4
unacc	7	2
Acc	4	0
Good	0	3
vgood	2	2
Jumlah	13	7

Tabel 3.6 Histogram *class* atribut *person* dengan nilai batas 5

<i>Person</i> (5)	Left <=5	Right >5
unacc	4	2
Acc	7	0
Good	2	1
vgood	2	3
Jumlah	15	5

Untuk atribut bertipe katagorik, misal histogram *class* pada atribut *buying* ditunjukkan pada tabel 3.7

Tabel 3.7 Histogram *class* atribut *buying*

<i>Buying</i>	low	med	high	vhigh
Unacc	2	2	0	2
Acc	0	3	2	2
good	3	0	0	0
vgood	2	2	0	0

## 2. Menghitung gini *index* untuk memilih atribut terbaik

Untuk atribut *person* dengan batas nilai <=4. Gini index dihitung menggunakan persamaan 2.1.

$$N_1 = 1 - \left(\frac{7}{13}\right)^2 - \left(\frac{4}{13}\right)^2 - \left(\frac{0}{13}\right)^2 - \left(\frac{2}{13}\right)^2 = 0,5917$$

$$N_2 = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{0}{5}\right)^2 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0,6530$$

$$\text{Gini}(T) = \frac{13}{20} \cdot 0,5917 + \frac{5}{20} \cdot 0,6530 = 0,61319$$

Perhitungan juga dilakukan pada batas 2 dan 5, kemudian dipilih batas dengan nilai gini terkecil dan untuk atribut *persons* didapatkan batas 4 yang memiliki nilai gini indeks terkecil. Jadi titik split data yang digunakan jika *persons* terpilih sebagai *node* adalah *persons*  $\leq 4$  dan *persons*  $>4$ . Demikian juga untuk atribut *door* dihitung dengan cara yang sama seperti atribut *persons* dengan batas 2, 3, 4 dan 5.

Perhitungan gini *index* untuk atribut katagorik dengan menggunakan persamaan 2.1 misal untuk atribut *buying* :

$$N_1(\text{low}) = 1 - \left(\frac{2}{7}\right)^2 - \left(\frac{0}{7}\right)^2 - \left(\frac{3}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0,65$$

$$N_2(\text{med}) = 1 - \left(\frac{2}{7}\right)^2 - \left(\frac{3}{7}\right)^2 - \left(\frac{0}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0,65$$

$$N_3(\text{high}) = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

$$N_4(\text{vhigh}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0,5$$

$$\text{Gini}(T) = \frac{7}{20} \cdot 0,65 + \frac{7}{20} \cdot 0 + \frac{2}{20} \cdot 0 + \frac{4}{20} \cdot 0,5 = 0,5571$$

Untuk atribut katagorik *maint*, *lugboot* dan *safety* perhitungan juga dilakukan dengan cara yang sama. Pada perulangan pertama menghasilkan gini index seperti pada tabel 3.8

Tabel 3.8 Hasil Gini indeks

Atribut	Gini Index
<i>Buying</i>	0.55714

<i>Maint</i>	0.52579
<i>Doors</i>	0.61667
<i>Persons</i>	0.61319
<i>Lugboot</i>	0.30631
<i>Safety</i>	0.43750

### 3. Mencari atribut dengan gini indeks terkecil

Setelah gini *index* semua atribut selesai dihitung, maka dipilih atribut dengan gini *index* terkecil. Pada contoh ini atribut *Lugboot* yang memiliki nilai terkecil, jadi atribut *Lugboot* digunakan sebagai *node* akar. Kemudian menentukan setiap subset dari masing-masing split berdasarkan kriterianya yaitu *lugboot* = small, *lugboot* = med, dan *lugboot* = *maint*.

Sistem akan memproses langkah 1 sampai langkah 3 secara simultan sampai setiap *node* sudah pure atau terbentuk *tree* dengan *leaf* hanya dalam satu *class*. Jika terdapat *node* yang belum pure dan atribut sudah habis maka kelas akan diambil dari kelas terbesar, tetapi jika semua jumlah kelas sama maka diambil kelas pertama yang ada pada *node* tersebut.

Cabang pertama yang diproses adalah *lugboot* = *small*, subset datanya dapat dilihat pada tabel 3.9.

Tabel 3.9 Tabel Data *Lugboot* = *small*

No	<i>Buying</i>	<i>Maint</i>	<i>Person</i>	<i>Doors</i>	<i>lugboot</i>	<i>Safety</i>	<i>class</i>
6	med	Vhigh	4	5	small	Low	unacc
14	Low	Low	5	6	small	High	good

Dari data subset *lugboot* = *small*, kemudian dicari *split node* dengan cara yang sama seperti langkah 1 sampai 3, dan di dapatkan atribut *doors* yang terpilih. Kemudian diproses seperti langkah 1 dan 2 untuk atribut numerik dan didapatkan batas 5 ( $\leq 5$  dan  $> 5$ ) yang terpilih sebagai titik *split* atribut *doors*. Dapat dilihat pada tabel 3.8 bahwa *doors* yang bernilai  $\leq 5$  memiliki 1 data dengan kelas unacc, jadi *leaf* ini sudah pure dan begitu juga dengan *doors*  $> 5$  juga memiliki 1 data dengan kelas good, hal ini juga berarti bahwa *leaf* *doors*  $> 5$  sudah pure.

Jadi selanjutnya memproses cabang kedua dari *node lugboot* yaitu *lugboot = med*, subset datanya dapat dilihat pada tabel 3.10. Setelah diproses seperti langkah 1 sampai 3 didapatkan atribut *safety* yang terpilih.

Tabel 3.10 Tabel Data *Lugboot = med*

No	Buying	Maint	Persons	Doors	lugboot	Safety	class
7	Med	med	4	4	med	Med	good
8	Med	med	4	4	med	High	vgood
15	Low	low	6	5	med	Low	unacc
16	Low	low	5	6	med	Med	good
17	Low	low	7	4	med	High	vgood

*Safety* memiliki 3 kriteria yaitu *low*, *med* dan *high* dicari subsetnya masing-masing. Ditunjukkan pada gambar 3.11 bahwa *leaf safety = low* memiliki 1 data dengan kelas yaitu *unacc*, jadi *leaf* ini sudah pure.

Tabel 3.11 Tabel Data *Lugboot = med* dan *Safety = low*

No	Buying	Maint	Persons	Doors	lugboot	Safety	class
15	Low	Low	6	5	med	Low	unacc

Begitu juga dengan *leaf safety = med* sudah pure memiliki 2 data dengan kelas yang sama yaitu *good*, ditunjukkan pada tabel 3.12.

Tabel 3.12 Tabel Data *Lugboot = med* dan *Safety = med*

No	Buying	Maint	Persons	Doors	lugboot	Safety	class
7	Med	Med	4	4	med	Med	good
16	Low	Low	5	6	med	Med	good

*Leaf safety = high* juga sudah pure memiliki 2 data dengan kelas yang sama yaitu *vgood*, ditunjukkan pada tabel 3.13.

Tabel 3.13 Tabel Data *Lugboot = med* dan *Safety = high*

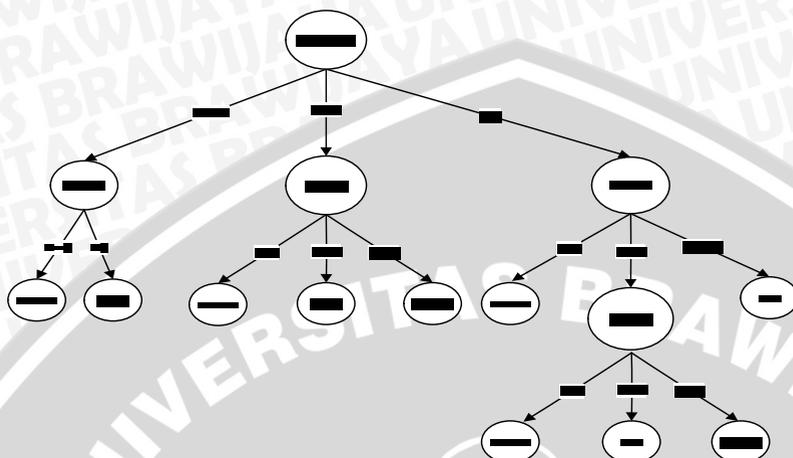
No	Buying	Maint	Persons	Doors	lugboot	Safety	class
8	Med	med	4	4	Med	High	vgood
17	Low	low	7	4	Med	High	vgood

Kemudian untuk cabang ketiga dari *lugboot* yaitu *big* di cari subset datanya dan dapat dilihat pada tabel 3.14.

Tabel 3.14 Tabel Data *Lugboot = big*

No	Buying	Maint	Persons	Doors	lugboot	safety	class
1	vhigh	med	2	2	Big	Med	Acc
2	vhigh	med	2	3	Big	High	Acc
3	High	med	4	2	Big	Med	Acc
4	High	med	4	4	Big	High	Acc
5	Med	vhigh	2	5	Big	High	Acc
9	Med	med	4	3	Big	Low	unacc
10	Med	med	4	3	Big	Med	Acc
11	Med	med	4	4	Big	High	vgood
12	vhigh	vhigh	2	5	Big	Low	unacc
13	vhigh	vhigh	2	2	Big	Med	unacc
18	Low	low	7	3	Big	Low	unacc
19	Low	low	8	4	Big	Med	good
20	Low	low	6	4	Big	High	vgood

Setelah diproses seperti pada langkah 1 sampai 3 maka dihasilkan bahwa atribut *maint* yang terpilih pada cabang *lugboot = big*. Selanjutnya diproses kembali seperti pada *subtree safety* serta langkah 1, 2 dan 3 sampai semua node sudah pure. Dan dari proses tersebut terbentuk *Tree* yang ditunjukkan pada gambar 3.6.



**Gambar 3.6** Hasil *Tree* Perhitungan Manual

4. Proses ekstraksi *decision tree* ke dalam *rule*

*Decision tree* sementara yang ditunjukkan oleh gambar 3.6, maka *rule* yang terbentuk sementara adalah sebagai berikut :

1. **IF** *Lugboot* = small **THEN** *Doors*  $\leq$  5 **THEN** *class* = unacc
2. **IF** *Lugboot* = small **THEN** *Doors* > 5 **THEN** *class* = good
3. **ELSE IF** *Lugboot* = med **AND** *Safety* = low **THEN** *class* = unacc
4. **ELSE IF** *Lugboot* = med **AND** *Safety* = med **THEN** *class* = good
5. **ELSE IF** *Lugboot* = med **AND** *Safety* = high **THEN** *class* = vgood
6. **ELSE IF** *Lugboot* = med **AND** *Maint* = low **THEN** *class* = unacc
7. **ELSE IF** *Lugboot* = med **AND** *Maint* = med **AND** *Safety* = low **THEN** *class* = unacc
8. **ELSE IF** *Lugboot* = med **AND** *Maint* = med **AND** *Safety* = med **THEN** *class* = acc
9. **ELSE IF** *Lugboot* = med **AND** *Maint* = med **AND** *Safety* = high **THEN** *class* = vgood
10. **ELSE IF** *Lugboot* = med **AND** *Maint* = high **THEN** *class* = vgood

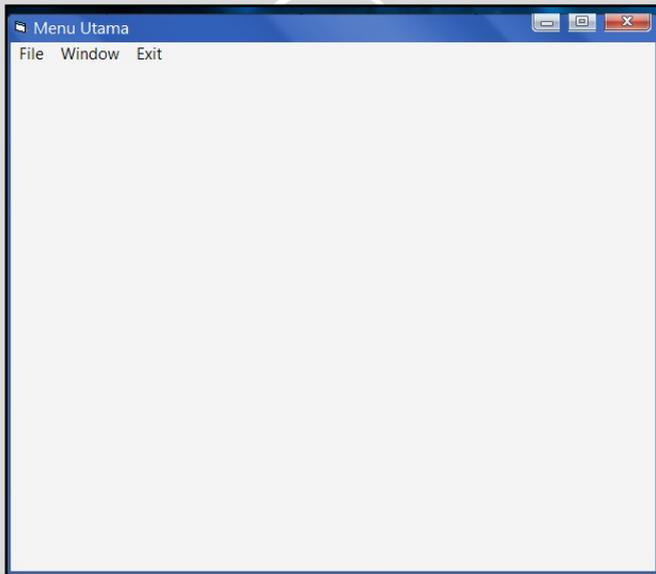
### 3.4 Perancangan Antarmuka

Perancangan aplikasi sistem terdiri dari empat menu utama, diantaranya menu input data, menu proses pembelajaran, menu tes evaluasi dan menu pengujian sistem.

#### 3.3.1 Form Menu Utama

Rancangan menu utama sistem ditunjukkan pada gambar 3.7. Keterangan gambar :

1. *File* berisi menu input data, *training*, *testing* dan evaluasi mobil.
2. *Window* untuk menampilkan form yang aktif
3. *Exit* adalah menu untuk keluar dari sistem



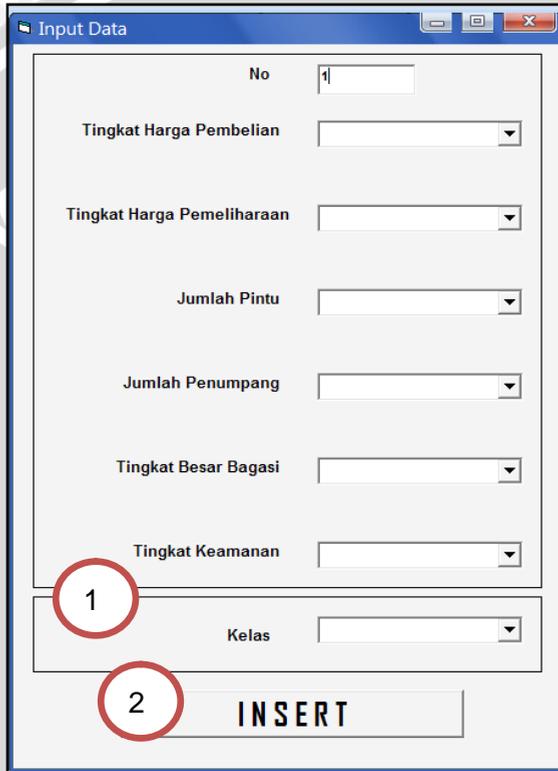
Gambar 3.7 Form Menu Utama

#### 3.6.2 Form Input Data

Menu ini dirancang untuk memproses seluruh input data *car evaluation* yang telah ada maupun data terbaru untuk digunakan sebagai *testing* sistem. Rancangan menu input data dapat dilihat pada gambar 3.8.

Keterangan gambar :

1. Untuk menginputkan data baru *testing*.
2. Button untuk memerintahkan penyimpanan data baru pada database.



The image shows a web browser window with a form titled "Input Data". The form contains several input fields, each with a label and a corresponding input element:

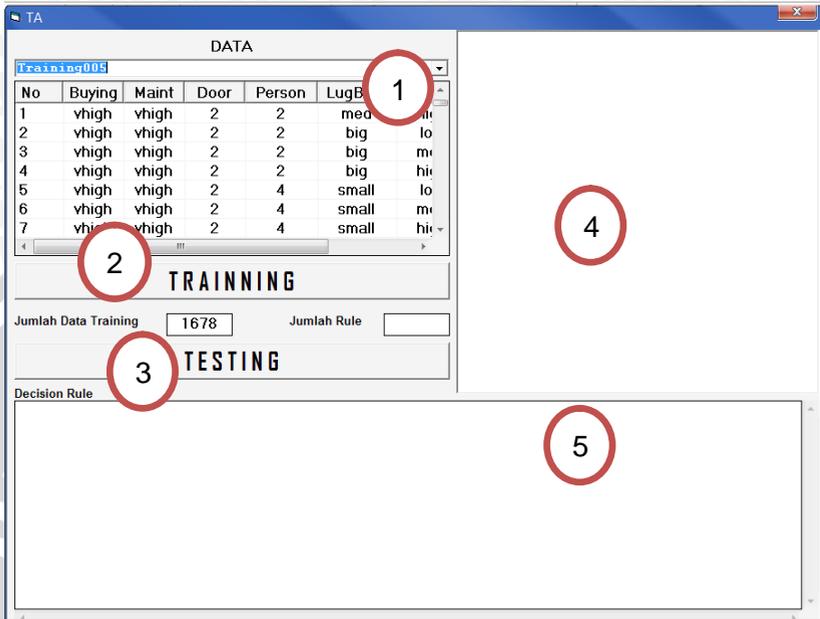
- No**: A text input field containing the number "1".
- Tingkat Harga Pembelian**: A dropdown menu.
- Tingkat Harga Pemeliharaan**: A dropdown menu.
- Jumlah Pintu**: A dropdown menu.
- Jumlah Penumpang**: A dropdown menu.
- Tingkat Besar Bagasi**: A dropdown menu.
- Tingkat Keamanan**: A dropdown menu.
- Kelas**: A dropdown menu.

At the bottom of the form is a large button labeled "INSERT". Two red circles with numbers "1" and "2" are overlaid on the image. Circle "1" is positioned over the "No" input field, and circle "2" is positioned over the "INSERT" button.

Gambar 3.8 Form Menu Input Data

### 3.6.3 Form Proses Pembelajaran

Menu ini dirancang untuk memproses seluruh data untuk melakukan proses pembentukan *tree*. Rancangan menu *training* dapat dilihat pada gambar 3.9.



**Gambar 3.9** Form Menu *training*/ proses pembelajaran

Keterangan gambar :

1. Untuk memilih data *training* dan menampilkannya.
2. *Button* untuk memulai proses data *training*
3. *Button* untuk membuka *form testing*
4. Menampilkan bentuk *tree*
5. Menampilkan *rule*

### 3.6.4 Form Sistem Evaluasi Mobil

Menu ini dirancang untuk melakukan evaluasi terhadap kriteria suatu mobil yang diinputkan oleh *user*. Rancangan menu evaluasi mobil dapat dilihat pada gambar 3.10.

Keterangan gambar :

1. Memasukan data kriteria mobil
2. Memulai proses evaluasi
3. Menampilkan hasil pencarian kelas

**Gambar 3.10** Form Menu Evaluasi Mobil

### 3.3.5 Form pengujian sistem

Menu ini terdiri dari pengujian tingkat kebenaran sistem pada saat pemberian evaluasi untuk mengukur tingkat kebenaran / akurasi. Rancangan menu pengujian dapat dilihat pada gambar 3.11.

Keterangan gambar :

1. Memilih data *testing* dan menampilkannya
2. Memulai proses pengujian
3. Menampilkan hasil pengujian jumlah benar dan akurasi



Tabel 3.15 Rancangan Tabel Data Pembelajaran

<b>Data Training</b>	<b>Komposisi</b>	<b>Jumlah Data Training</b>
<i>Training001</i>		
<i>Training002</i>		
<i>Training003</i>		
<i>Training004</i>		
<i>Training005</i>		

Pada tabel 3.16 merupakan rancangan tabel yang digunakan untuk pengujian rekomendasi sistem pada sejumlah data tes dengan jumlah data pembelajaran yang berbeda – beda.

Tabel 3.16 Rancangan Tabel Pengujian

<b>Data Training</b>	<b>Jumlah Aturan</b>	<b>Jumlah Benar</b>	<b>Akurasi (%)</b>

Keterangan :

- Data Pembelajaran ke- : Data yang digunakan untuk pembelajaran
- Jumlah aturan (*Rule*) : Jumlah aturan yang dihasilkan dari pembentukan *tree*
- Jumlah Benar : Jumlah data benar dalam proses pengujian
- Akurasi : Tingkat kebenaran sistem dalam persen

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1 Lingkungan Implementasi

Lingkungan implementasi yang dijelaskan pada subbab ini adalah lingkungan implementasi perangkat lunak dan perangkat keras.

##### 4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan pada pengembangan sistem evaluasi mobil dengan menggunakan algoritma SLIQ ini adalah :

1. Notebook Intel Core 2 Duo Processor T5750 ( 2 MB L2 cache, 2 GHz, 667 MHz FSB)
2. RAM 1 GB DDR2
3. 120 Gigabyte SATA HDD
4. 14.1" Wide crystal view
5. Optical mouse

##### 4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat Lunak yang digunakan pada pengembangan sistem evaluasi mobil dengan menggunakan algoritma SLIQ ini adalah :

1. Sistem operasi Window XP SP2
2. Visual Basic 6
3. Microsoft Access 2003
4. Microsoft Office Excel 2003
5. Notepad

#### 4.2 Implementasi Program

Pada sub bab akan dijelaskan implementasi program sistem evaluasi mobil menggunakan decision *tree* SLIQ.

##### 4.2.1 Struktur Data

Struktur data pada program ini dapat dilihat pada sourcecode 4.1.

Variabel	Keterangan
Public GlobalString(14) As String	Untuk menyimpan nama

<pre>Type DataCarEvaluation   No As Integer   Buying As String   Maint As String   Door As String   Person As String   Lugboot As String   Safety As String   Class As String End Type</pre>	<p>atribut, nilai atribut, kelas pada proses pembentukan <i>tree</i></p> <p><i>Record</i> untuk memproses penyimpanan input data ke database dan penyimpanan <i>rule</i> ke database</p>
<pre>Type Test   No(150) As Integer   K1(150) As String   K2(150) As String   K3(150) As String   K4(150) As String   K5(150) As String   K6(150) As String   K7(150) As String   K8(150) As String End Type</pre>	<p><i>Record</i> untuk menyimpan pengambilan data testing dari database untuk pengujian sistem</p>
<pre>Dim DataTemp(6, 4) As String Dim TipeTEMP(10) As String</pre>	<p><i>Array</i> untuk penyimpanan nilai atribut sementara</p> <p><i>Array</i> untuk menyimpan <i>node</i> sementara</p>

**SourceCode 4.1** Struktur Data

### 4.2.2 Menyusun Kelas Histogram

Proses pengaksesan data dari database untuk menghitung frekuensi kelas dan menyusun kelas histogram dapat dilihat pada *sourcecode* 4.2.

```
Private Sub Hitung()
  Dim i, j As Integer
  Dim N1, N2, N3, N4 As Single
  Dim GiniIndex As Single
```

```

For i = 1 To 28
    CB(i) = Empty
    CB1(i) = Empty
    CB2(i) = Empty
    CB3(i) = Empty
    CB4(i) = Empty
Next

If Len(GlobalString(1)) = 0 Then
    For i = 1 To 4
        For j = 1 To 4
            SQLCMD = "SELECT COUNT(Buying) FROM " & TabelName &
                " WHERE CLASS='" & Class(j) & "' AND Buying='" &
                Trim(Data(i)) & "'"
            If Len(GlobalString(2)) > 0 Then SQLCMD = SQLCMD & "
                AND Maint='" & GlobalString(2) & "'"
            If Len(GlobalString(3)) > 0 Then SQLCMD = SQLCMD & "
                AND Doors" & GlobalString(3)
            If Len(GlobalString(4)) > 0 Then SQLCMD = SQLCMD & "
                AND Persons" & GlobalString(4)
            If Len(GlobalString(5)) > 0 Then SQLCMD = SQLCMD & "
                AND Lugboot='" & GlobalString(5) & "'"
            If Len(GlobalString(6)) > 0 Then SQLCMD = SQLCMD & "
                AND Safety='" & GlobalString(6) & "'"
            Set ADOrs = ADOConn.Execute(SQLCMD)
            If j = 1 Then CB1(i) = ADOrs(0)
            If j = 2 Then CB2(i) = ADOrs(0)
            If j = 3 Then CB3(i) = ADOrs(0)
            If j = 4 Then CB4(i) = ADOrs(0)

            Set ADOrs = Nothing
        Next
        CB(i) = Val(CB1(i)) + Val(CB2(i)) + Val(CB3(i)) +
            Val(CB4(i))
    Next
Next

```

**SourceCode 4.2** Proses penghitungan frekuensi kelas dan penyusunan histogram

Proses penyusunan histogram dilakukan dengan cara mengambil jumlah frekuensi kelas pada masing-masing nilai atribut dan mengisikannya pada tabel histogram. GlobalString(1) untuk nilai atribut buying, globalstring(2) untuk nilai atribut maint, globalstring(3) untuk nilai atribut doors, globalstring(4) untuk atribut persons, globalstring(5) untuk atribut lugboot, globalstring(6) untuk atribut safety.

### 4.2.3 Proses penghitungan gini indeks

Penghitungan gini indeks dapat dijelaskan *sourcecode* 4.3.

```
`N1 = 0: N2 = 0: N3 = 0: N4 = 0

If Val(CB(1)) > 0 Then N1=1-(CB1(1)/CB(1))^2-(CB2(1)/
CB(1))^2-(CB3(1)/CB(1))^2-(CB4(1)/CB(1))^2
If Val(CB(2)) > 0 Then N2 = 1 - (CB1(2) / CB(2)) ^ 2 -
(CB2(2) / CB(2)) ^ 2 - (CB3(2) / CB(2)) ^ 2 - (CB4(2) /
CB(2)) ^ 2
If Val(CB(3)) > 0 Then N3 = 1 - (CB1(3) / CB(3)) ^ 2 -
(CB2(3) / CB(3)) ^ 2 - (CB3(3) / CB(3)) ^ 2 - (CB4(3) /
CB(3)) ^ 2
If Val(CB(4)) > 0 Then N4 = 1 - (CB1(4) / CB(4)) ^ 2 -
(CB2(4) / CB(4)) ^ 2 - (CB3(4) / CB(4)) ^ 2 - (CB4(4) /
CB(4)) ^ 2
GiniIndex = 0
If Val(CB(1))+Val(CB(2))+Val(CB(3))+Val(CB(4))>0 Then
    GiniIndex = GiniIndex + Val(CB(1)) * N1 /
    (Val(CB(1)) + Val(CB(2)) + Val(CB(3)) + Val(CB(4)))
    GiniIndex = GiniIndex + Val(CB(2)) * N2 /
    (Val(CB(1)) + Val(CB(2)) + Val(CB(3)) + Val(CB(4)))
    GiniIndex = GiniIndex + Val(CB(3)) * N3 /
    (Val(CB(1)) + Val(CB(2)) + Val(CB(3)) + Val(CB(4)))
    GiniIndex = GiniIndex + Val(CB(4)) * N4 /
    (Val(CB(1)) + Val(CB(2)) + Val(CB(3)) + Val(CB(4)))
End If
Hasil(1) = Format(GiniIndex, "#0.00000")
TGini = TGini & "Buying :" & Format(GiniIndex,
"#0.00000") & vbCrLf
Else
    Frame(1).Caption = "CLASS - Buying : " &
GlobalString(1)
    Frame(1).ForeColor = vbRed
    End If
```

#### **SourceCode 4.3** Proses penghitungan gini indeks

Proses penghitungan gini indeks mengambil nilai dari tabel histogram dan menghitung gini indeks berdasarkan persamaan 2.1 dan 2.2. Setiap nilai gini indeks atribut disimpan pada array Hasil.

#### 4.2.4 Proses pemilihan atribut terbaik

Setelah proses penghitungan gini indeks masing-masing atribut, selanjutnya adalah proses pemilihan nilai gini indeks yang terkecil, dimana atribut yang memiliki nilai gini indeks terkecil akan terpilih sebagai atribut terbaik dan disimpan dalam globalstring(11), tetapi jika nilai gini indeks = 0 maka kelas sudah *pure* dan kelas disimpan pada globalString(11). Globalstring(12) untuk memberi tanda 1 jika ada atribut yang terpilih dan 0 untuk kondisi sudah *pure*. Globalstring(11) akan diisi \* jika tidak ditemukan atribut maupun kelas yang sudah *pure*. Globalstring(13) untuk menyimpan indeks *label* awal atribut terpilih dan Globalstring(14) untuk menyimpan indeks *label* akhir suatu atribut terpilih. Proses pemilihan atribut terbaik dapat dilihat pada *sourcecode* 4.4. Hasil dari proses ini digunakan pada proses pembentukan *tree*.

```
For I = 1 To 6
    If Format(Hasil(i), "#0.00000") < GiniIndex Then
        n = i
        GiniIndex = Hasil(i)
    End If
Next
If n > 0 Then
    'OUTPUT HASIL GINI INDEX TERKECIL
    Select Case n
        Case 1:
            GlobalString(11) = ", AND Buying"
            GlobalString(12) = 1
            GlobalString(13) = 1
            GlobalString(14) = 4
            If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Buying" & vbCrLf
        Case 2:
            GlobalString(11) = ", AND Maint"
            GlobalString(12) = 1
            GlobalString(13) = 5
            GlobalString(14) = 8
            If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Maint" & vbCrLf
        Case 3:
            GlobalString(11) = ", AND Doors"
            GlobalString(12) = 1
            If T1 = 1 Then
                GlobalString(13) = 9
                GlobalString(14) = 10
            End If
        End Select
End If
End Sub
```

```

End If
If T1 = 2 Then
    GlobalString(13) = 11
    GlobalString(14) = 12
End If
If T1 = 3 Then
    GlobalString(13) = 13
    GlobalString(14) = 14
End If
If T1 = 4 Then
    GlobalString(13) = 15
    GlobalString(14) = 16
End If
If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Doors" & vbCrLf
    Case 4:
        GlobalString(11) = ", AND Persons"
        GlobalString(12) = 1
        If T2 = 1 Then
            GlobalString(13) = 17
            GlobalString(14) = 18
        End If
        If T2 = 2 Then
            GlobalString(13) = 19
            GlobalString(14) = 20
        End If
        If T2 = 3 Then
            GlobalString(13) = 21
            GlobalString(14) = 22
        End If
        If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Persons" & vbCrLf
            Case 5:
                GlobalString(11) = ", AND Lugboot"
                GlobalString(12) = 1
                GlobalString(13) = 23
                GlobalString(14) = 25
                If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Lugboot" & vbCrLf
                    Case 6:
                        GlobalString(11) = ", AND Safety"
                        GlobalString(12) = 1
                        GlobalString(13) = 26
                        GlobalString(14) = 28
                        If GiniIndex <> 0 Then TGini = TGini &
"Terpilih Safety" & vbCrLf
                            End Select
                        // Jika node sudah menemukan kelasnya

```

```

If GiniIndex = 0 Then
    GlobalString(11) = " - unacc"
    If Val(CB1(1)) + Val(CB1(2)) + Val(CB1(3)) +
Val(CB1(4)) > 0 Then GlobalString(11) = " - unacc"
    If Val(CB2(1)) + Val(CB2(2)) + Val(CB2(3)) +
Val(CB2(4)) > 0 Then GlobalString(11) = " - acc"
    If Val(CB3(1)) + Val(CB3(2)) + Val(CB3(3)) +
Val(CB3(4)) > 0 Then GlobalString(11) = " - good"
    If Val(CB4(1)) + Val(CB4(2)) + Val(CB4(3)) +
Val(CB4(4)) > 0 Then GlobalString(11) = " - vgood"

-----

End if
Else
    GlobalString(11) = "*"
    GlobalString(12) = 0
    GlobalString(13) = 0
    GlobalString(14) = 0
End if

```

**SourceCode 4.4** Proses penentuan atribut terbaik

#### 4.2.5 Proses Pembentukan Pohon

Porses pembentukan pohon dimulai dengan memanggil *procedure* Hitung() untuk mencari *Split node*. Nilai globalstring(13) dan globalstring(14) dihasilkan dari prosedur Hitung() yang berisi indeks label atribut yang terpilih sebagai *node*, dan sekaligus menunjukkan jumlah kriteria atribut terpilih untuk mencari *node child* berikutnya. Globalstring(urutan(1)) menyimpan nilai atribut untuk mencari atribut terpilih. Jika globalstring(12) = 1 maka *node* sudah *pure* dan proses berhenti, jika belum *pure* maka dilakukan pencarian *split node* selanjutnya. Proses ini akan membentuk suatu pohon dan *decision rule* dan dijelaskan pada *sourcecode 4.5*

```

Private Sub Form_Load()
    Dim nMulai(6) As Byte
    Dim nAkhir(6) As Byte
    Dim Urutan(6) As Byte
    Dim i, j, k, l, m, n, o, p, q, r, s, t As Integer
    Dim RuleTEMP, Kelas, Child As String
    Call DeleteRule1
    Menu_Utama.TV1.Nodes.Clear
    Urutan(1) = 0
    For i = 1 To 11

```

```

GlobalString(i) = Empty
Next
Call Kosong(1)
Call KosongRll
TGini = TGini & "Level 0" & vbCrLf
Call Hitung
Menu_Utama.TV1.Nodes.Add, "LEVEL1", GlobalString(11)
nMulai(1) = Val(GlobalString(13))
nAkhir(1) = Val(GlobalString(14))
Menu_Utama.TRule = ""

For i = 1 To nAkhir(1) - nMulai(1) + 1
    Menu_Utama.TV1.Nodes.Add "LEVEL 1", tvwChild, "LEVEL
    2" & i, Data(nMulai(1) + i - 1)
    Urutan(1) = TIPE(Mid(Menu_Utama.TV1.Nodes("LEVEL 2"
    & i).Parent, 7))
    GlobalString(Urutan(1)) =
    Trim(Menu_Utama.TV1.Nodes("LEVEL 2" & i).Text)
    GlobalString(7) = DataTemp(1, 1)
    GlobalString(8) = DataTemp(1, 2)
    GlobalString(9) = DataTemp(1, 3)
    GlobalString(10) = DataTemp(1, 4)
    GlobalString(11) = Empty
    GlobalString(12) = Empty
    Call Hitung
    TipeTEMP(10) = GlobalString(Urutan(1))

    If Len(GlobalString(11)) > 0 Then
        If GlobalString(11) = "*" Then
            GlobalString(Urutan(1)) = Empty
            GlobalString(Urutan(2)) = Empty
            GlobalString(7) = DataTemp(5, 1)
            GlobalString(8) = DataTemp(5, 2)
            GlobalString(9) = DataTemp(5, 3)
            GlobalString(10) = DataTemp(5, 4)
            GlobalString(11) = Empty
            GlobalString(12) = Empty
            Call Hitung
            GlobalString(12) = 0
        Menu_Utama.TV1.Nodes("LEVEL 2" & i).text =
        Menu_Utama.TV1.Nodes("LEVEL 2" & i).text &
        KelasTerbesar(nMulai(1) + i - 1)
    Else
        Menu_Utama.TV1.Nodes("LEVEL 2" & i).Text =
        Menu_Utama.TV1.Nodes("LEVEL 2" & i).Text &
        GlobalString(11)
        nMulai(2) = Val(GlobalString(13))
        nAkhir(2) = Val(GlobalString(14))
    
```

```

End If
GlobalString(Urutan(1)) = TipeTEMP(10)

// Jika leaf belum pure, dilakukan proses split node
If Val(GlobalString(12)) = 1 Then

    DataTemp(1, i) = Mid(GlobalString(11), 7)

    -----

```

#### **SourceCode 4.5** Proses pembentukan pohon

### **4.2.6 Proses Pembentukan Rule**

Proses pembentukan *rule* sebenarnya menjadi satu kesatuan dengan *looping* pembentukan pohon. Pada saat mengekstrak pohon menjadi *rule*, sistem ini juga melakukan penyimpanan *rule* tersebut dalam database dengan memanggil prosedur *InsertRule1*. Proses pembentukan *rule* dapat dilihat pada *sourcecode* 4.6.

```

Rl.Rule = Empty
Call KosongR11
If UCase(Mid(Menu_Utama.TV1.Nodes("LEVEL 1").text,
7)) = "PERSONS" UCase(Mid(Menu_Utama.TV1.Nodes
("LEVEL 1").text, 7)) = "DOORS" Then
    Dt = UCase(Mid(Menu_Utama.TV1.Nodes("LEVEL
1").text, 7))
    Krt = UCase(Mid(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text, 1, InStr(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text, " - ")))
    Rl.Rule = "IF " & Dt & " " & Krt
    Call Find
Else
    Dt = UCase(Mid(Menu_Utama.TV1.Nodes("LEVEL
1").text, 7))
    Krt = UCase(Mid(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text, 1, InStr(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text, " - ")))
    Rl.Rule = "IF " & Dt & " =" & Krt
    Call Find
End If
R11.Class=UCase(Right(Menu_Utama.TV1.Nodes("LEVEL
2" & i).text, Len(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text) - InStr(Menu_Utama.TV1.Nodes("LEVEL 2" &
i).text, " - ") - 2))

```

```
Rl.Rule = Rl.Rule & "THEN " & Rl1.Class  
Menu_Utama.TRRule = Menu_Utama.TRRule & Rl.Rule &  
vbCrLf  
Call InsertRule1 //penyimpanan pada database  
End If  
Next  
next
```

*SourceCode 4.6* Proses pembentukan Rule

### 4.3 Penerapan Aplikasi

Aplikasi yang dibangun ini digunakan untuk melakukan proses input data, proses pembelajaran, pengujian dan proses evaluasi mobil.

No	1
Tingkat Harga Pembelian	low
Tingkat Harga Pemeliharaan	med
Jumlah Pintu	3
Jumlah Penumpang	4
Tingkat Besar Bagasi	small
Tingkat Keamanan	low
Kelas	unacc

**INSERT**

**Gambar 4.1** Proses Input Data

Menu pada gambar 4.1 digunakan untuk proses penginputan data *testing* baru, yang akan dimasukkan ke dalam database data *testing*. Pada proses input data, *user* akan memasukkan data kriteria dari suatu mobil, yaitu tingkat pembelian untuk *buying*, tingkat harga pemeliharaan untuk *maint*, jumlah pintu untuk *doors*, jumlah penumpang untuk *persons*, tingkat besar bagasi untuk *lugboot* dan tingkat keamanan untuk *safety*.

Pada proses pembelajaran, *user* pertama kali harus memilih data *training* seperti yang ditunjukkan pada Gambar 4.2. Pada penerapan ini data *training* yang digunakan adalah *Training005*, yang berisi keseluruhan data *training*. Setelah proses pembelajaran selesai maka hasil *rule* dan pohon ditampilkan seperti pada gambar 4.2.

**DATA**

Training005

No	Buying	Maint	Door	Person	LugBoot	Sat
1	vhigh	vhigh	2	2	med	hi
2	vhigh	vhigh	2	2	big	lo
3	vhigh	vhigh	2	2	big	me
4	vhigh	vhigh	2	2	big	hi
5	vhigh	vhigh	2	4	small	lo
6	vhigh	vhigh	2	4	small	me
7	vhigh	vhigh	2	4	small	hi

**TRAINING**

Jumlah Data Training: 1678      Jumlah Rule: 93

**TESTING**

**Decision Rule**

```

IF SAFETY = LOW THEN UNACC
IF SAFETY = MED AND PERSONS <= 2 AND LUGBOOT = SMALL THEN UNACC
IF SAFETY = MED AND PERSONS <= 2 AND LUGBOOT = MED THEN UNACC
IF SAFETY = MED AND PERSONS <= 2 AND LUGBOOT = BIG THEN UNACC
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND LUGBOOT = AND DOORS <= 2 THEN ACC
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND LUGBOOT = AND DOORS > 2 THEN GOOD
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND LUGBOOT = MED THEN ACC
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND LUGBOOT = BIG THEN GOOD
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND DOORS AND LUGBOOT = SMALL THEN ACC
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND DOORS AND LUGBOOT = MED THEN GOOD
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND DOORS AND LUGBOOT = BIG THEN GOOD
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = AND DOORS > 3 THEN ACC
IF SAFETY = MED AND PERSONS > 2 AND BUYING = AND MAINT = HIGH THEN ACC
    
```

Gambar 4.2 Proses Training

Tetapi sistem terlebih dahulu menampilkan form Hitung Gini Index yang digunakan untuk menyusun histogram *class* pada setiap proses *split node* sampai data kelas ditemukan, form ditunjukkan pada gambar 4.3, sebelah kanan adalah catatan nilai gini indeks untuk masing-masing atribut pada setiap prosesnya.

The screenshot shows a software window titled "Hitung Gini Index" with several sections for different attributes. Each section contains a table for class splits and a calculated Gini Index value.

**CLASS - Buying : vhigh**

	unacc	acc	good	vgood
low				
med				
high				
vhigh				

Gini Index CLASS terhadap Buying

**CLASS - Maint : vhigh**

	unacc	acc	good	vgood
low	0	2	0	0
med	0	1	0	0
high	2	0	0	0
vhigh	2	0	0	0

Gini Index CLASS terhadap Doors : 0.43445

**CLASS - Doors : > 4**

	unacc	acc	good	vgood
<= 2				
> 2				

Gini Index CLASS terhadap Doors : 0.43417

**CLASS - Persons : > 2**

	unacc	acc	good	vgood
<= 2				
> 2				

Gini Index CLASS terhadap Doors : 0.43327

**CLASS - LugBoot : big**

	unacc	acc	good	vgood
small				
med				
big				

Gini Index CLASS terhadap Doors : 0.43459

**CLASS - Safety : high**

	unacc	acc	good	vgood
low				
med				
high				

Gini Index CLASS terhadap Doors : 0.31169

**CLASS - Persons : <= 4**

	unacc	acc	good	vgood
<= 4				
> 4				

Gini Index CLASS terhadap Doors : 0.39394

**CLASS - Persons : <= 5**

	unacc	acc	good	vgood
<= 5				
> 5				

Gini Index CLASS terhadap Doors : 0.39669

**CLASS - LugBoot : big**

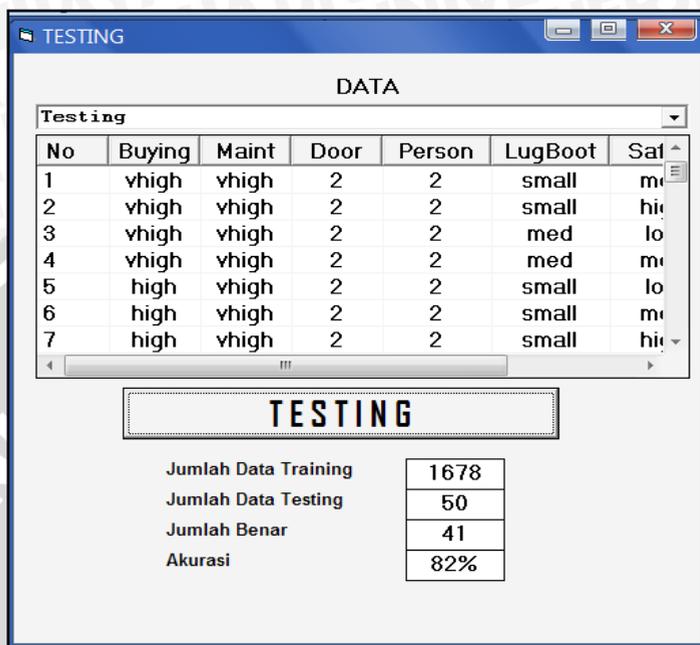
	unacc	acc	good	vgood
small				
med				
big				

Gini Index CLASS terhadap Safety

**Level 0**

- Buying : 0.430
- Maint : 0.441
- Doors : 0.450
- Persons: 0.382
- LugBoot: 0.447
- Safety: 0.3774
- Terpilih Safe
- Buying : 0.000
- Maint : 0.000
- Doors : 0.000
- Persons: 0.000
- LugBoot: 0.000
- Kelas UNACC
- Buying : 0.468
- Maint : 0.483
- Doors : 0.506
- Persons: 0.390
- LugBoot: 0.477
- Terpilih Pers
- Buying : 0.000
- Maint : 0.000
- Doors : 0.000
- LugBoot: -0.02
- Terpilih Lugh
- Buying : 0.000
- Maint : 0.000
- Doors : 0.000
- Kelas UNACC
- Buying : 0.000
- Maint : 0.000
- Doors : 0.000
- Kelas UNACC
- Buying : 0.000

Gambar 4.3 Proses Hitung Gini Indeks



**Gambar 4.4** Proses *Testing*

Setelah proses pembentukan pohon dilakukan maka selanjutnya adalah proses *testing* atau pengujian. Untuk melakukan *testing*, *user* harus memilih tabel *testing*. Setelah proses *testing* selesai maka jumlah benar dan tingkat akurasi akan ditampilkan, seperti ditunjukkan pada Gambar 4.4 terlihat jumlah data testing 50 *record*, jumlah data benar 41 dan akurasi 82%.

Untuk melakukan proses evaluasi mobil *user* harus memasukkan data seperti ditunjukkan pada gambar 4.5, kemudian kelas dari data tersebut akan ditampilkan berdasarkan pencarian *rule* hasil proses *training* terakhir.

Sistem Evaluasi Mobil

Tingkat Harga Pembelian: med

Tingkat Harga: med

Jumlah Pintu: 4

Jumlah Penumpang: 2

Tingkat Besar Bagasi: big

Tingkat Keamanan: high

**EVALUATE**

Kelas: UNACC

**Gambar 4.5** Proses Evaluasi Mobil

#### 4.4 Pengujian dan Analisa Hasil

Untuk mengetahui kinerja metode *decision tree* SLIQ dalam memecahkan permasalahan, dilakukan uji coba sebanyak 5 kali dan didapatkan hasil pengujian tingkat kebenaran pada table 4.1.

Tabel 4.1 merupakan tabel tingkat akurasi sejumlah data pembelajaran yang telah ditetapkan sebelumnya dan data test yang memiliki jumlah yang sama yaitu *50 record*. Data *training* memiliki sebaran kelas yang sama seperti pada sebaran data *training* keseluruhan. Data *Training001* dengan jumlah data 331 (20 % dari data keseluruhan) memiliki tingkat akurasi 68 %, data *Training002* dengan jumlah data 645 (40 % dari data keseluruhan) memiliki

akurasi 68 %, data *Training*003 dengan jumlah data 989 (60 % dari data keseluruhan) memiliki tingkat akurasi 74 %, data *training*004 dengan jumlah data 1329 (60 % dari data keseluruhan) memiliki tingkat akurasi 74% dan pada data *Training*005 dengan jumlah data 1678 (data keseluruhan) memiliki akurasi 82 %, paling tinggi diantara semua data *training*.

**Tabel 4.1** Tabel hasil uji coba

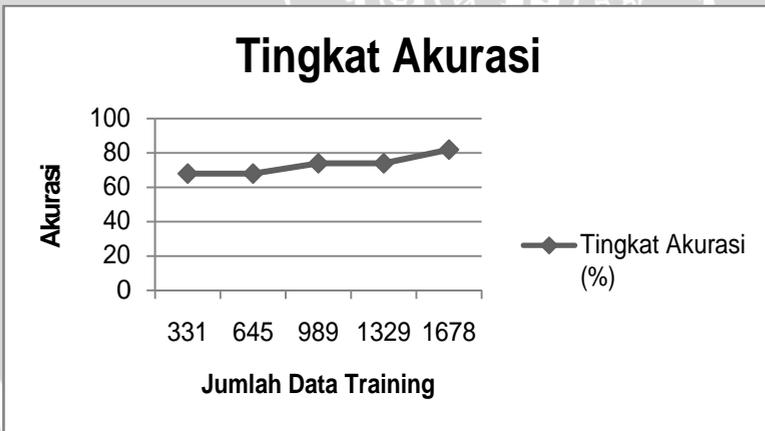
Nama	Jumlah Data <i>Training</i>	Komposisi	Jumlah <i>Rule</i>	Jumlah Benar	Akurasi (%)
Training 001	331	Unacc = 236 Acc = 73 Good = 11 Vgood = 12	12	24	68 %
Training 002	645	Unacc = 453 Acc = 145 Good = 23 Vgood = 24	43	24	68 %
Training 003	989	Unacc = 699 Acc = 219 Good = 32 Vgood = 38	50	37	74 %
Training 004	1329	Unacc = 940 Acc = 292 Good = 44 Vgood = 53	77	37	74 %
Training 005	1678	Unacc = 1210 Acc = 384 Good = 65 Vgood = 69	93	41	82 %

Berdasarkan pada evaluasi uji coba, dapat disimpulkan bahwa secara umum, grafik menunjukkan peningkatan tingkat akurasi terhadap penambahan jumlah *record* data *training*. Maka dapat diambil kesimpulan bahwa semakin banyak jumlah data

pembelajaran yang digunakan, maka klasifikasi yang diberikan akan semakin benar begitu juga dengan tingkat akurasi yang semakin meningkat. Dan pada data pembelajaran yang terbanyak yaitu, 1678 akurasi mencapai hasil yang baik yaitu 82 %. Grafik tingkat akurasi dapat dilihat pada gambar 4.6.

Demikian juga dengan *rule* yang dihasilkan, semakin banyak data pembelajaran maka *rule* yang dihasilkan semakin banyak, oleh karena itu tingkat kebenarannya dalam mengklasifikasikan data semakin meningkat pula. Grafik Jumlah *rule* dapat dilihat pada gambar 4.7.

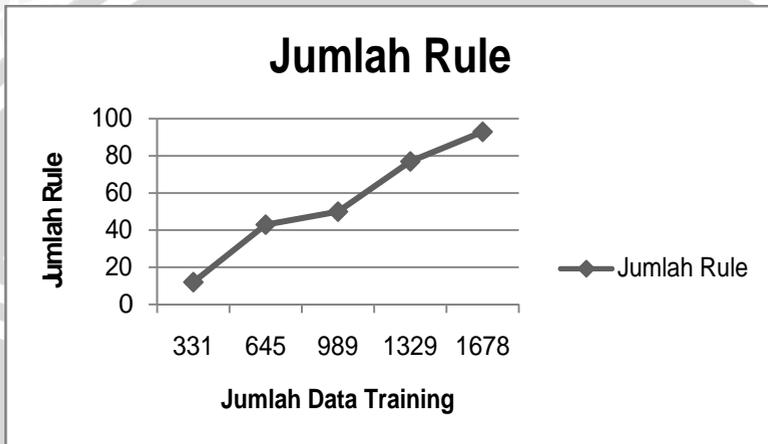
Dan kelamahan dari sistem ini adalah pada jumlah data pembelajaran yang digunakan sedikit tingkat akurasi yang dihasilkan kurang bagus, terlihat pada data *training* dengan jumlah 331 dan 645 tingkat akurasi yang dihasilkan 68% hal ini karena informasi yang dikumpulkan oleh sistem sedikit pula.



**Gambar 4.6** Grafik Tingkat akurasi metode SLIQ

Pada Grafik 4.6, dapat dilihat bahwa jumlah data pembelajaran mempengaruhi tingkat kebenaran yang dihasilkan. Semakin meningkat data pembelajaran maka semakin meningkat juga tingkat akurasinya.

Dan pada grafik 4.7 terlihat pula semakin banyak data pembelajaran, maka semakin banyak juga *rule* yang terbentuk dan semakin banyak pula informasi yang dikumpulkan oleh sistem. Hal ini tentunya akan meningkatkan akurasi hasil evaluasi mobil yang diberikan kepada *user*.



**Grafik 4.7** Grafik Jumlah *rule* metode SLIQ

UNIVERSITAS BRAWIJAYA



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari tugas akhir ini adalah :

1. *Data mining* dengan metode *decision tree* menggunakan algoritma SLIQ dapat diimplementasikan untuk sistem evaluasi mobil, dengan menggunakan data *car evaluation*. Dengan tahapan yaitu menyusun kelas histogram, menghitung gini indeks setiap atribut, menentukan *split node* dengan mencari atribut yang memiliki gini indeks terkecil, pembentukan tree dan pembentukan *rule*.
2. Tingkat akurasi sistem pada data *training* 1678 mencapai 82 %.
3. Semakin bertambahnya data pembelajaran maka tingkat akurasi sistem dalam pemberian evaluasi akan semakin meningkat, hal ini disebabkan oleh semakin banyaknya *rule* yang dibentuk oleh sistem, sehingga memperbanyak informasi yang dihasilkan.

#### 5.2 Saran

Sistem yang dibangun masih belum sempurna. Hal ini dapat bermanfaat untuk mengembangkan aplikasi ini adalah :

1. Untuk penelitian algoritma SLIQ selanjutnya diharapkan untuk menambah jumlah atribut, agar pembentukan *tree* lebih beragam.
2. Membangun sistem evaluasi mobil dengan ditambahkan proses penentuan nilai kriteria setiap atribut.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Anonymous* . 2006. *Comp 290-90 Seminar GNET 214 BCB Module*.  
[www.cs.unc.edu/Courses/comp290-09-s06/Lecturenotes/classification1.ppt](http://www.cs.unc.edu/Courses/comp290-09-s06/Lecturenotes/classification1.ppt). Diakses tanggal 2 Agustus 2010.
- Anonymous*. 1999. *Introduction to Data Mining and Knowledge Discovery Third Ed.*, two Crows corporation internet :  
<http://twocrows.com>. Diakses tanggal 5 Agustus 2006.
- Bohanec, Marko. 1994. *Data Car Evaluation*. [www.is.edu/](http://www.is.edu/). Diakses tanggal 25 Juni 2010.
- Dunham, Margaret H. 2002. *Data Mining Introductory and Advanced Topics Part I*. <http://enr.smu.edu/~mhd/dmbook/part1.ppt>. Diakses pada tanggal 2 Agustus 2006.
- Fatalla, Ronald.2004. *Decision Model for Car Evaluation*.  
[http://www.hi.is/~benedikt/Courses/Car\\_Evaluation\\_Method.pdf](http://www.hi.is/~benedikt/Courses/Car_Evaluation_Method.pdf)  
. Diakses tanggal 22 Juli 2010.
- Giri S, Yudho. 2003. *Data Mining Menggali informasi yang terpendam*. <http://ikc.vip.net.id/populer/yudho-datamining.pdf>.  
Diakses pada tanggal 26 Juli 2010.
- Han, Jiawei and Khamber, Micheline. 2001. *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco.
- Hari M, Bagus. 2009. Analisis perbandingan metode perhitungan error untuk pemangkasan pohon yang menggunakan aturan post-pruning pada decision tree C4.5. Universitas Brawijaya. Malang.
- Kantardzic, Mehmed. 2003. *Data mining : Concepts, Models, Method and Algorithm*. John Wiley & Sons. New York.
- Larose, Daniel T. 2005. *Discovering Knowledge in Data An Introduction to Data Mining*, John Wiley & Sons,Inc., Hoboken, New Jersey.

Mannila, Smyth, and Hand, David. 2001. *Principle of Data Mining*. MIT Press, Cambridge.

Mehta, Manish; Rakesh Agrawal; Jorma Rissanen. 1996. *SLIQ: A Fast Scalable Classifier for Data Mining*. <http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0405/KDD/paper/MAR96.pdf>. Diakses tanggal 27 Agustus 2010.

Pramudiono, Iko. 2003. *Pengantar Data mining : Menambang Permata Pengetahuan di gunung Data*. <http://ilmukomputer.com/2006/08/25/pengantar-data-mining/iko-datamining.pdf>. Diakses tanggal 25 Juli 2010.

Tan, Steinbach, Kumar, 2004. *Introduction to Data Mining*. Addison-Wesley Companion Book Site. New York.

Vipin, Kumar and Joshi, Mahesh. 1999. *High Performance Data Mining*. Univ of Mineshota. Mineshota.

Zaiane, Osmar R. 1999. *Principles of Knowledge Discovery in Databases*. University of Alberta. <http://www.cs.ualberta.ca/~zaiane/courses/cmpu690/slides/Chapter8>. Diakses tanggal 25 Juli 2010.

## LAMPIRAN

1. Title: Car Evaluation Database

2. Sources:

(a) Creator: Marko Bohanec

(b) Donors: Marko Bohanec

(marko.bohanec@ijs.si)

Blaz Zupan (blaz.zupan@ijs.si)

(c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is

derived, was first presented in

M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988.

Within machine-learning, this dataset was used for the evaluation

of HINT (Hierarchy INDuction Tool), which was proved to be able to

completely reconstruct the original hierarchical model. This,

together with a comparison with C4.5, is presented in

B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by

function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:

Car Evaluation Database was derived from a simple hierarchical

decision model originally developed for the demonstration of DEX

(M. Bohanec, V. Rajkovic: Expert system for decision

making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates

cars according to the following concept structure:

CAR	car acceptability
. PRICE	overall price
. . buying	buying price
. . maint	price of the maintenance
. TECH	technical characteristics
. . COMFORT	comfort
. . . doors	number of doors
. . . persons	capacity in terms of persons to carry
. . . lug_boot	the size of luggage boot
. . safety	estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts:

PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see <http://www-ai.ijs.si/BlazZupan/car.html>).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug\_boot, safety.

Because of known underlying concept structure, this database may be

particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 1728  
(instances completely cover the attribute space)

6. Number of Attributes: 6

7. Attribute Values:

buying	v-high, high, med, low
maint	v-high, high, med, low
doors	2, 3, 4, 5-more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	( 3.993 %)
v-good	65	( 3.762 %)

| names file (C4.5 format) for car evaluation domain

| class values

unacc, acc, good, vgood

| attributes

buying: vhigh, high, med, low.

maint: vhigh, high, med, low.

doors: 2, 3, 4, 5more.

persons: 2, 4, more.

lug\_boot: small, med, big.

safety: low, med, high.

UNIVERSITAS BRAWIJAYA



# UNIVERSITAS BRAWIJAYA



This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.