

**PENERAPAN ALGORITMA PUBLIC
DALAM PEMBANGUNAN *DECISION TREE*
UNTUK PEMBUATAN SISTEM REKOMENDASI
PEMBELIAN RUMAH**

SKRIPSI

oleh :
GITA PERMANA SANTOSO
0410960023-96

UNIVERSITAS BRAWIJAYA



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
2009**









**PENERAPAN ALGORITMA PUBLIC
DALAM PEMBANGUNAN *DECISION TREE*
UNTUK PEMBUATAN SISTEM REKOMENDASI
PEMBELIAN RUMAH**

SKRIPSI

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

oleh :

**GITA PERMANA SANTOSO
0410960023-96**



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

2009

i





LEMBAR PENGESAHAN SKRIPSI

**PENERAPAN ALGORITMA PUBLIC
DALAM PEMBANGUNAN *DECISION TREE*
UNTUK PEMBUATAN SISTEM REKOMENDASI
PEMBELIAN RUMAH**

oleh :

GITA PERMANA SANTOSO
0410960023-96

Telah dipertahankan di depan Majelis Penguji
pada tanggal 8 Januari 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana dalam bidang Ilmu Komputer

Pembimbing I

Wayan Firdaus M., SSi., MT
NIP. 132 158 724

Pembimbing II

Bayu Rahayudi, ST., MT
NIP. 132 318 424

**Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya**

Dr. Agus Suryanto, MSc
NIP. 132 126 049



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Gita Permana Santoso
NIM : 0410960023
Program Studi : Ilmu Komputer
Penulis Skripsi Berjudul : Penerapan Algoritma PUBLIC
Dalam Pembangunan *Decision Tree*
Untuk Pembuatan Sistem
Rekomendasi Pembelian Rumah

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, 8 Januari 2009
Yang menyatakan,

(Gita Permana Santoso)
NIM. 0410960023





PENERAPAN ALGORITMA PUBLIC
DALAM PEMBANGUNAN *DECISION TREE*
UNTUK PEMBUATAN SISTEM REKOMENDASI
PEMBELIAN RUMAH

ABSTRAK

Rumah merupakan kebutuhan primer manusia selain sandang dan pangan. Dan seiring kemajuan teknologi dan proses modernisasi, pandangan seseorang terhadap rumah pun semakin beraneka ragam. Hal inilah yang mendorong para pengembang untuk bersaing dalam menghasilkan produk rumah yang sesuai keinginan pembeli. Akan tetapi, karena para pengembang bergerak secara *individual*, rekomendasi yang diberikan menjadi cenderung *subjektif* dan kurang sesuai dengan keinginan pembeli.

Penelitian ini berusaha membangun sisten rekomendasi pembelian rumah yang *objektif* bagi pembeli dengan penerapan metode *Decision Tree* menggunakan algoritma PUBLIC pada proses klasifikasi data, dengan melibatkan sumber data dari gabungan para pengembang perumahan. Algoritma PUBLIC mengintegrasikan fase pemangkasan ke dalam fase pertumbuhan daripada melakukan tiap fase secara berurutan. Sehingga jumlah *node* yang diperluas pada fase pertumbuhan menjadi jauh berkurang, dan tentunya mempercepat waktu yang diperlukan.

Hasil pengujian menunjukkan bahwa dengan menggunakan 200 *record data training*, *error rate* yang dihasilkan sebesar 23.57%, dan semakin menurun seiring penambahan jumlah data *training*. Sehingga dapat diambil kesimpulan bahwa metode algoritma PUBLIC cukup baik untuk diterapkan dalam membangun sistem rekomendasi pembelian rumah.



repository.ub.ac

**PUBLIC ALGORITHM APPLICATION
IN *DECISION TREE* CLASSIFICATION
FOR HOUSING TRANSACTION RECOMMENDATION
SYSTEM**

ABSTRACT

Housing is human's basic needs besides clothes and foods. An along with technology development and modernization, people's point of view for buying house is more complicated. These conditions stimulate housing developers to compete in producing house to suit with the buyer's requirement. But, because the housing developers work individually, then the given recommendation is tends to be subjective and unsuitable with the what buyer's want.

This research heging to build recommendation system that objective for buyers using the Decision Tree method application with PUBLIC algorithm on data classification process, and with including the datasource from groups of housing developers. PUBLIC algorithm integrates the pruning phase into the building phase instead of performing them one after the other. Thus, the number of node that split in building phase is less and then will faster the times needed.

The test result shows that, with using 200 data training records, error rate yielded is 23.57%, and will decrease as the number of data training added. This is conclude that PUBLIC algorithm is good enough to be applicated for build housing transaction recommendation system.



KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahNya, Skripsi yang berjudul "Penerapan Algoritma PUBLIC Dalam Pembangunan *Decision Tree* Untuk Pembuatan Sistem Rekomendasi Pembelian Rumah" ini dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW, yang senantiasa memberikan cahaya petunjuk, dan atas keluarganya dan sahabat-sahabatnya.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Wayan Firdaus Mahmudy, SSI, MT selaku pembimbing utama, penasihat akademik dan Ketua Program Studi Ilmu Komputer Unibraw Malang.
2. Bayu Rahayudi, ST.,MT selaku pembimbing pendamping dalam penulisan skripsi.
3. Dr. Agus Suryanto, Msc selaku ketua jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengamalkan ilmunya kepada penulis.
5. Segenap staf dan karyawan di Jurusan Matematika Fakultas MIPA Universitas Brawijaya
6. Ayah, Ibu, Kakak dan Adik. Terima kasih atas cinta, kasih sayang, doa, dukungan dan semangat yang tiada henti.
7. Teman-teman Combo, Sigura-gura'29, Ilkomers. Terima kasih atas senyuman, semangat, dukungan, do'a dan hari-hari kita.

Pihak lain yang telah membantu terselesaikannya Skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, dan

mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, 20 Mei 2008

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
 BAB I PENDAHULUAN	
a. Latar Belakang	1
b. Rumusan Masalah	3
c. Tujuan Penelitian.....	3
d. Batasan Masalah.....	3
e. Manfaat	3
f. Metodologi Pemecahan Masalah	3
g. Sistematika Penulisan	4
 BAB II TINJAUAN PUSTAKA	
a. Data Mining	7
i. Pengertian Data Mining	7
ii. Proses Data Mining	7
iii. Metode Data Mining	8
b. Sistem Rekomendasi	9
c. Klasifikasi (Classification)	10
d. Metode Decision Tree.....	10
e. Algoritma PUBLIC.....	12
i. Fase Pertumbuhan Pohon.....	13
ii. Fase Pemangkasan Pohon.....	16
iii. Fase Pertumbuhan dan Pemangkasan dalam PUBLIC	20
 BAB III METODOLOGI PERANCANGAN SISTEM	
a. Analisis Sistem	23

i.	Deskripsi Sistem.....	23
ii.	Batasan Sistem	25
b.	Perancangan Basis Data	25
c.	Perancangan Sistem	31
i.	Perancangan proses	31
1.	Proses klasifikasi tipe rumah pada data <i>training</i> menggunakan algoritma PUBLIC	32
2.	Proses Rekomendasi Produk	38
ii.	Perancangan Antarmuka.....	38
d.	Perancangan Uji Coba.....	40
i.	Data yang digunakan dalam penelitian	40
1.	Data produk rumah.....	40
2.	Data kriteria pembeli.....	41
ii.	Pengujian sistem rekomendasi	41
1.	Pengujian tingkat kesalahan.....	41
2.	Pengujian waktu rekomendasi sistem.....	43
e.	Contoh Perhitungan Manual	43
 BAB IV HASIL DAN PEMBAHASAN		
a.	Perangkat Sistem.....	55
i.	Perangkat Lunak.....	55
ii.	Perangkat Keras.....	55
b.	Implementasi	55
i.	Implementasi Basis Data	55
ii.	Struktur Data	56
iii.	Pembentukan decision tree	57
iv.	Proses rekomendasi	61
c.	Penerapan Aplikasi	63
d.	Pengujian sistem rekomendasi	65
i.	Pengukuran tingkat kesalahan sistem.....	66
ii.	Pengukuran waktu rekomendasi sistem	67
e.	Analisa Hasil	68
 BAB V PENUTUP		
5.1	Kesimpulan	71
5.2	Saran.....	71
 DAFTAR PUSTAKA		
		73

DAFTAR GAMBAR

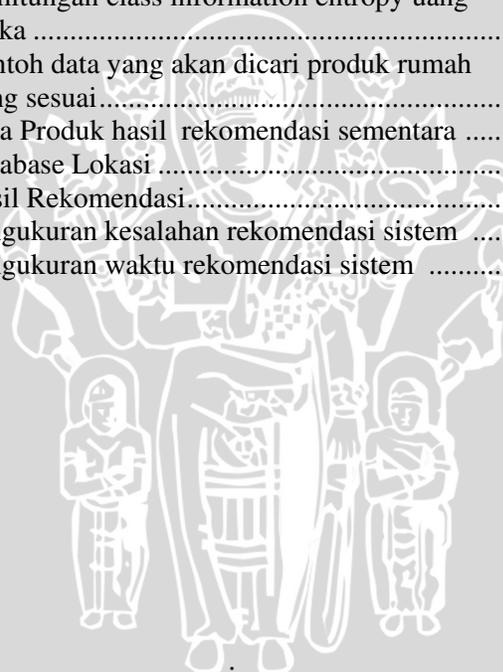
	Halaman
Gambar 2.1 Proses data mining.....	8
Gambar 2.2 Contoh penerapan metode <i>decision tree</i> untuk klasifikasi tipe rumah	11
Gambar 2.3 Contoh Aplikasi Persetujuan Pinjaman	14
Gambar 2.4 Algoritma pemangkasan PUBLIC	21
Gambar 2.5 Algoritma perhitungan lowerbound	21
Gambar 3.1 Langkah-langkah penelitian	24
Gambar 3.2 Entity Relationship Diagram (ERD)	30
Gambar 3.3 Flowchart pembentukan aturan klasifikasi tipe rumah	33
Gambar 3.4 Flowchart prosedur penentuan cutpoint terbaik	34
Gambar 3.5 Flowchart prosedur ComputeCost&PrunePUBLIC.....	36
Gambar 3.6 Flowchart prosedur perhitungan Lowerbound...	37
Gambar 3.7 Flowchart prosedur proses rekomendasi produk	39
Gambar 3.8 Diagram aplikasi sistem	38
Gambar 3.9 Decision tree untuk klasifikasi type rumah	54
Gambar 4.1 Struktur data <i>record</i> tiap kriteria	56
Gambar 4.2 Penggunaan <i>query</i> pada pemanggilan data produk dari <i>database</i>	57
Gambar 4.3 Fungsi EntAtribut()	58
Gambar 4.4 Fungsi EntCutpoint().....	58
Gambar 4.5 Potongan pemanggilan fungsi Cost_Node()	59
Gambar 4.6 Fungsi Cost_Node()	60
Gambar 4.7 Contoh penyimpanan <i>rule</i> ke dalam <i>database</i> pada prosedur Database()	61
Gambar 4.8 Potongan prosedur rekomendasi.....	63
Gambar 4.9 Tampilan menu input produk rumah	64
Gambar 4.10 Tampilan menu klasifikasi data	64
Gambar 4.11 Tampilan menu rekomendasi	65
Gambar 4.12 Tampilan menu pengujian sistem	66
Gambar 4.13 Grafik pengukuran kesalahan rekomendasi	69
Gambar 4.14 Grafik pengukuran waktu rekomendasi	70



DAFTAR TABEL

Halaman

Tabel 2.1	Himpunan Data Nyata.....	13
Tabel 2.2	Waktu Eksekusi Data Nyata	13
Tabel 3.1	Tabel Kriteria (t_Kriteria)	26
Tabel 3.2	Tabel rumah (t_Rumah)	27
Tabel 3.3	Tabel pengembang (t_Pengembang)	28
Tabel 3.4	Tabel lokasi (t_lokasi).....	28
Tabel 3.5	Tabel klasifikasi (t_Klasifikasi)	29
Tabel 3.6	Tipe atribut dan kategori pada kriteria pembeli .	41
Tabel 3.7	Tabel Jumlah Data Training.....	42
Tabel 3.8	Rancangan Tabel Catatan Tingkat Kesalahan.....	42
Tabel 3.9	Pengukuran waktu rekomendasi sistem	43
Tabel 3.10	Contoh data sampel.....	44
Tabel 3.11	Tabel Distribusi Kelas.....	45
Tabel 3.12	Perhitungan class information entropy uang muka	46
Tabel 3.13	Contoh data yang akan dicari produk rumah yang sesuai.....	52
Tabel 3.14	Data Produk hasil rekomendasi sementara	53
Tabel 3.15	Database Lokasi	53
Tabel 3.16	Hasil Rekomendasi.....	53
Tabel 4.1	Pengukuran kesalahan rekomendasi sistem	67
Tabel 4.2	Pengukuran waktu rekomendasi sistem	68





BAB I PENDAHULUAN

1.1 Latar Belakang

Rumah pada hakekatnya merupakan kebutuhan dasar (*basic needs*) manusia selain sandang dan pangan, juga pendidikan dan kesehatan. Dan seiring dengan kemajuan teknologi dan proses modernisasi, serta peningkatan kesejahteraan, maka sikap dan pandangan seseorang terhadap rumah ikut berubah pula. Antara lain adalah 'Aspek Prestise', yang menganggap rumah sebagai media untuk menaikkan status sosialnya, 'Aspek Komoditi' dengan pertimbangan menjadikan rumah sekaligus sebagai barang investasi, dan anggapan sebagai 'Aspek keterpaksaan', yakni untuk meningkatkan rasa aman terhadap kerawanan kriminalitas yang marak terjadi di kota besar. Rumah bukan lagi menjadi sekedar sebagai tempat berteduh dan melindungi diri penghuninya dari kondisi alam dan bahaya dari luar, namun sudah berkembang sebagai sarana yang dapat menunjukkan jati diri dan pribadi penghuninya. Dan pentingnya keberadaan rumah inilah yang kemudian diterjemahkan para pengusaha ke dalam bisnis *property* perumahan.

Tiap pengembang perumahan berusaha memberikan kelebihan ataupun keunikan-keunikan, baik dalam hal lokasi yang strategis, harga yang terjangkau, fasilitas, dsb. Tentunya didukung pula oleh sistem pemasaran yang baik. Akan tetapi, karena pemasaran dilakukan oleh masing-masing pengembang perumahan, terkadang hasil rekomendasi yang diberikan cenderung menguntungkan pihak pengembang dan kurang sesuai dengan keinginan pembeli. Masalah inilah yang kemudian melatarbelakangi pembuatan sistem rekomendasi yang objektif bagi pembeli, dengan melibatkan sumber data dari gabungan para pengembang perumahan. Sistem rekomendasi yang dapat memberikan saran mengenai produk rumah yang paling sesuai dengan kriteria yang diinginkan pembeli.

Sebuah rekomendasi dapat dihasilkan dengan melakukan analisis terhadap berbagai informasi yang diperoleh. Dalam kasus ini, data yang berperan penting dalam menghasilkan suatu rekomendasi adalah data hasil produk dari tiap pengembang perumahan. Data hasil produk dari tiap pengembang perumahan, yang meliputi tipe rumah, luas bangunan, lokasi, harga yang menyertainya akan

diklasifikasikan dan diolah menjadi sebuah sistem rekomendasi dengan menggunakan *data mining*.

Data mining adalah sebagai pemodelan atau penemuan pola-pola yang tersembunyi dengan memanfaatkan data dalam volume yang besar (Moertini, 2002). *Data mining* memiliki kemampuan untuk memberikan rekomendasi terhadap suatu kasus, dengan memprediksi dan mempelajari data-data sebelumnya yang telah ada. Dalam *data mining* sendiri terdapat beberapa teknik yang berbeda, salah satunya ialah teknik klasifikasi (*classification*). Teknik klasifikasi pada *data mining* digunakan untuk mengelompokkan data ke dalam kelas-kelas yang telah ditentukan sebelumnya. Selain itu, teknik klasifikasi ini juga memiliki kemampuan untuk memprediksi data baru dengan mengklasifikasikan data baru tersebut berdasarkan data yang sudah ada. Dalam pembuatan sistem rekomendasi ini digunakan teknik klasifikasi dengan *decision tree* (pohon keputusan).

Decision tree merupakan salah satu fungsional dari *data mining* yang menggunakan representasi *tree* untuk menentukan aturan-aturan klasifikasinya (Han dan Khamber, 2001). Kelebihan *decision tree*, selain mudah dalam membangun model *tree* dan mudah menggunakannya, *decision tree* juga dapat membagi data dalam tiap-tiap titik cabang tanpa kehilangan banyak data (Berson dan Stephen, 1997). Salah satu metode *decision tree* yang dapat digunakan untuk membuat sistem rekomendasi ialah algoritma PUBLIC.

Menurut penelitian sebelumnya yang dilakukan Rastogi dan Sim, algoritma PUBLIC (*PrUning and BuiLding Integrated in Classification*) mampu mempercepat waktu yang diperlukan, namun tetap menghasilkan pohon dengan tingkat akurasi yang sama dengan pohon yang dihasilkan dari 2 fase yang terpisah (dengan menggunakan data dalam bidang kedokteran, yang akan dijelaskan lebih detail dalam Bab 2). Sehingga diharapkan dengan penggunaan algoritma PUBLIC, dapat dihasilkan sistem rekomendasi pembelian rumah yang akurat. Sistem rekomendasi yang dapat membantu semua pihak yang berkepentingan, baik konsumen yang ingin membeli rumah sesuai kriteria mereka ataupun konsumen yang belum mempunyai gambaran mengenai kriteria yang diinginkan, pihak asosiasi perumahan dalam upaya meningkatkan pelayanan pada konsumen, serta secara tidak langsung juga meningkatkan omzet penjualan para pengembang perumahan.

1.2 Rumusan Masalah

Rumusan masalah yang akan dijadikan obyek penelitian, antara lain:

1. Bagaimana implementasi algoritma PUBLIC dalam pembangunan *decision tree* pada proses klasifikasi data produk berdasarkan tipe rumah.
2. Bagaimana melakukan pengujian dan analisis hasil implementasi algoritma PUBLIC.

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah tersebut, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma PUBLIC dalam pembangunan *decision tree* pada proses klasifikasi data produk berdasarkan tipe rumah.
2. Menganalisis akurasi dan waktu penelusuran pada *decision tree* yang dihasilkan algoritma PUBLIC, untuk sistem rekomendasi pembelian rumah.

1.4 Batasan Masalah

Batasan masalah pada skripsi ini ialah:

1. Jenis produk rumah dan nama tiap pengembang perumahan telah ditentukan sebelumnya pada tabel rumah dan tabel pengembang.
2. Data yang dijadikan acuan sebagai data *training* adalah data produk rumah yang dihasilkan dari para pengembang perumahan.
3. Data yang dijadikan acuan sebagai data *uji* adalah data kriteria rumah yang diinginkan pembeli.

1.5 Manfaat

Manfaat yang bisa diambil dari penulisan skripsi ini adalah menghasilkan suatu sistem rekomendasi pembelian rumah bagi pembeli dengan penerapan algoritma PUBLIC, yang dapat mempercepat proses klasifikasi data.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan skripsi ini adalah:

1. Studi Literatur
Mempelajari teori-teori yang berhubungan dengan algoritma PUBLIC pada metode decision tree dari berbagai referensi.
2. Pendefinisian dan analisis masalah
Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi sistem
Membuat perancangan perangkat lunak dan mengimplementasikan hasil rancangan tersebut yaitu membuat perangkat lunak sistem rekomendasi pembelian rumah.
4. Uji coba dan analisis hasil implementasi
Menguji perangkat lunak, dan menganalisis hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Pembuatan skripsi ini dilakukan dengan pembagian bab sebagai berikut:

BAB I : PENDAHULUAN

Bab I membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan skripsi.

BAB II : TINJAUAN PUSTAKA

Bab II mencantumkan beberapa tinjauan pustaka yang berkaitan dengan penelitian, diantaranya : *Data Mining*, Sistem Rekomendasi, Klasifikasi, *Decision Tree*, Algoritma *PUBLIC*.

BAB III : METODOLOGI DAN PERANCANGAN

Bab III menerangkan beberapa hal mengenai metode dan perancangan yang akan dilakukan dalam membangun sistem ini. Beberapa poin yang dibahas ialah: Analisis Sistem, Perancangan Basis Data, Perancangan Sistem, Perancangan Uji Coba, dan Contoh Perhitungan Manual dari suatu kasus.

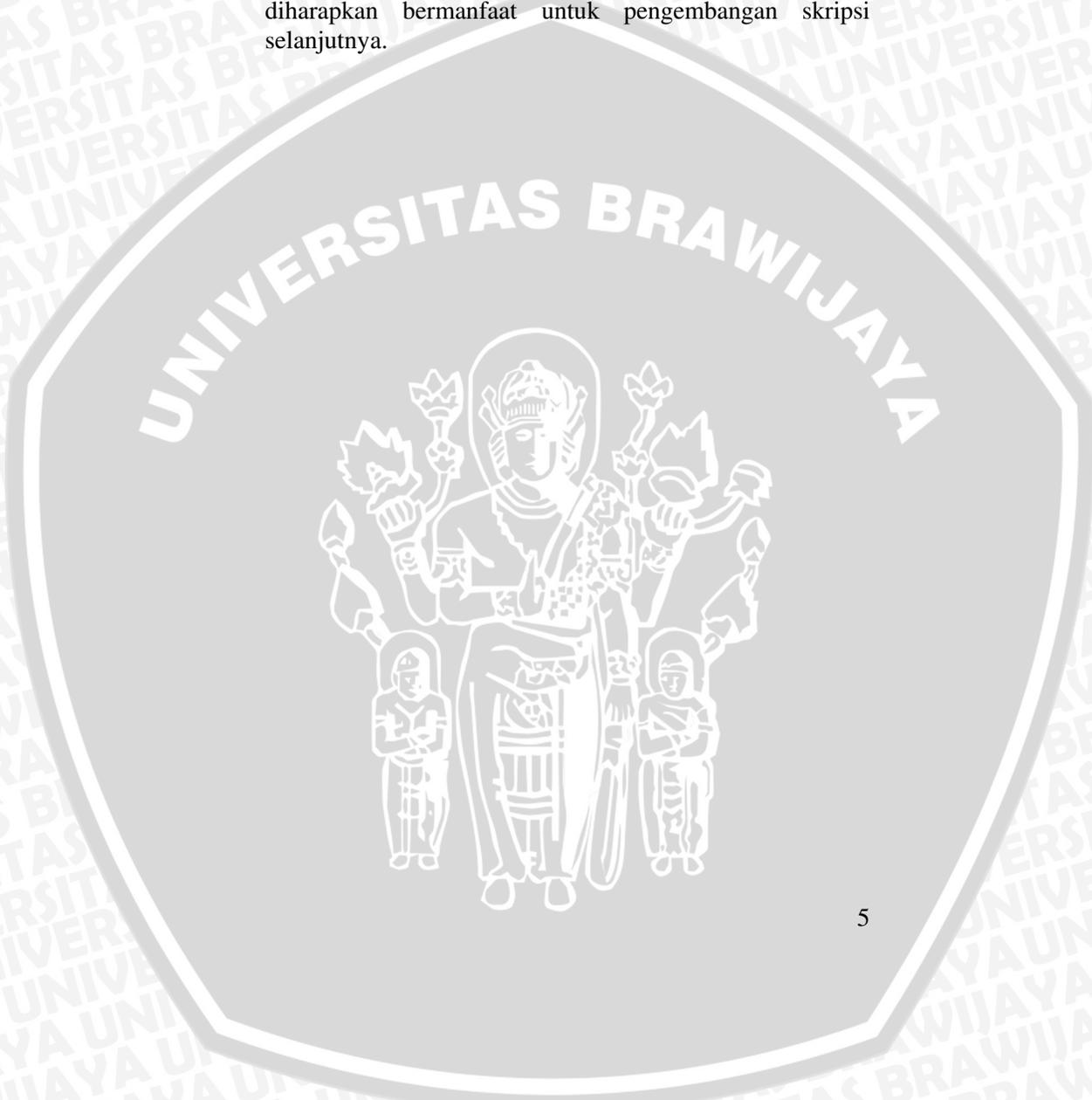
BAB IV : HASIL DAN PEMBAHASAN

Bab IV menerangkan proses implementasi dari rancangan penelitian yang dijelaskan pada bab III. Implementasi yang

dijelaskan terdiri dari implementasi basis data dan implementasi program. Selain itu, bab IV juga menjelaskan penerapan aplikasi, analisis hasil percobaan mengenai tingkat kesalahan berdasarkan jumlah data pembelajaran yang digunakan dalam proses klasifikasi untuk menghasilkan rekomendasi, serta analisis dan penjelasannya.

BAB V : KESIMPULAN DAN SARAN

Bab V berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan skripsi selanjutnya.





BAB II TINJAUAN PUSTAKA

2.1 *Data mining*

2.1.1 Pengertian *data mining*

Data Mining adalah proses mencari informasi atau mengambil suatu keputusan dari database yang besar. Menurut Zalilia (2007), sebelum mengerti makna *data mining*, diperlukan untuk memahami makna kata *mining*. Dalam bahasa Inggris, *to mining* dapat diartikan sebagai kegiatan menambang. *To mining* dapat juga berarti mengambil ekstrak atau sari. Sehingga secara sederhana dapat dikatakan bahwa *data mining* adalah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di *database* yang besar.

Data mining juga dapat didefinisikan sebagai pemodelan atau penemuan pola-pola yang tersembunyi dengan memanfaatkan data dalam volume yang besar (Moertini, 2002). Sedangkan menurut Guidici (2003) *data mining* adalah proses seleksi, eksplorasi, dan pemodelan dari data dalam jumlah besar untuk menemukan aturan atau relasi yang pada awalnya tidak diketahui dengan tujuan memperoleh hasil yang jelas dan bermanfaat untuk pemilik database. Dan menurut Pramudiono (2003) *data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual.

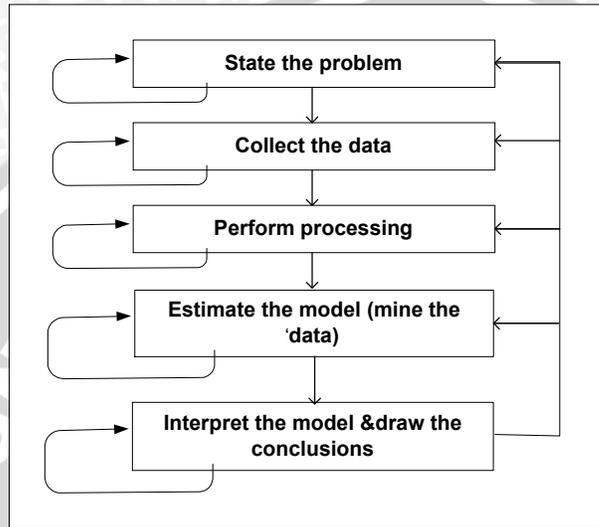
2.1.2 Proses *data mining*

Ada beberapa tahap atau proses yang harus dilalui dalam mengambil suatu keputusan. Dimulai dari merumuskan sumber masalah, mengolahnya, dan mengambil keputusan dari kesimpulan yang didapat. Menurut Kantardzic (2003), ada beberapa prosedur umum eksperimental dengan menggunakan *data mining*. Antara lain:

- a) Pertama, ialah merumuskan permasalahan, dimana pada tahap ini ditetapkan sebuah rumusan masalah serta variabel-variabel yang terlibat.
- b) Prosedur yang kedua ialah mengumpulkan data. Pada prosedur ini, konsentrasi ditujukan mengenai proses pembuatan atau pengumpulan data.

- c) Prosedur yang ketiga yaitu *preprocessing data*, yaitu prosedur untuk menyeleksi data yang akan digunakan dalam proses.
- d) Prosedur selanjutnya ialah estimasi model yang dapat disebut sebagai proses utama pada prosedur ini, sebab implementasi dari teknik *data mining* dilakukan pada prosedur ini.
- e) Dan yang terakhir ialah menafsirkan informasi yang dihasilkan dari proses sebelumnya.

Blok diagram dari prosedur umum dengan menggunakan *data mining* dapat dilihat pada Gambar 2.1



Gambar 2.1. Proses *data mining* (Kantardzic, 2003)

Ada beberapa teknik yang dapat digunakan pada *data mining*. Menurut Larose (2005), teknik *data mining* yang utama ialah *description*, *estimation*, *prediction*, *classification*, *clustering*, dan *association*. Sedangkan menurut Kantardzic (2003), teknik-teknik tersebut ialah *classification*, *regression*, *clustering*, *summarization*, *dependency modeling*, dan *change and deviation detection*.

2.1.3 Metode *data mining*

Menurut Kumar, dkk (2004), ada beberapa metode yang digunakan dalam *data mining* diantaranya adalah *predictive*

modelling, association analysis, clustering, dan anomaly detection. *Predictive modelling* digunakan dalam membangun model untuk memprediksi suatu nilai yang mempunyai kriteria atau ciri-ciri tertentu. *Association analysis* bertujuan untuk menghasilkan sejumlah *rule* yang menjelaskan sejumlah data yang terhubung kuat antara satu dengan yang lainnya. *Clustering* bertujuan untuk mengelompokkan data yang homogen atau sejenis sehingga data yang berada di *cluster* yang sama mempunyai banyak kesamaan dibandingkan dengan data yang ada di *cluster* yang berbeda. *Anomaly detection* bertujuan untuk menemukan anomali atau *outlier*, yaitu data yang sangat berbeda dengan data-data yang lainnya.

Teknik yang digunakan dalam penelitian ini adalah teknik klasifikasi yang merupakan salah satu teknik dari metode *predictive modelling*. Untuk pembahasan lebih lanjut mengenai klasifikasi akan dijelaskan pada subbab 2.3 Klasifikasi.

2.2 Sistem Rekomendasi

Sistem rekomendasi merupakan suatu usaha memberikan sebuah masukan pada *user* berupa saran atau rekomendasi terhadap sesuatu. Sistem rekomendasi dapat juga dikatakan sebagai teknologi pemilihan informasi personal yang digunakan untuk memprediksi apakah seorang *user* tertentu akan tertarik dengan sesuatu (Han.dkk, 2005). Dalam membangun sebuah sistem rekomendasi, terdapat dua metode pengumpulan informasi (*information filtering*) sebagai bahan rekomendasi, yaitu:

- 1) dengan menganalisa isi informasi, sebagai contoh pada *content-based filtering*, dan
- 2) dengan mengacu pada kebiasaan atau perilaku *user* (pengguna) lain yang mengakses sistem, contohnya pada *colaborative filtering* (Haruechayasak.dkk, 2004).

Colaborative Filtering (CF) adalah metode yang dapat secara otomatis memberikan prediksi (*filtering*) tentang ketertarikan *user* terhadap sesuatu dengan mengumpulkan dan mempelajari informasi dari banyak *user* (*Collaboratif*). Rekomendasi yang dihasilkan biasanya didasarkan pada kecocokan pada keadaan sebenarnya antara profil seorang *user* dengan data (*record*) yang sama atau terdekat yang diperoleh dari sistem pada waktu berbeda dari *user* lain (Gunduz.dkk, 2004).

2.3 Klasifikasi (*Classification*)

Klasifikasi adalah salah satu teknik dalam *data mining* yang bertujuan mengelompokkan data ke dalam kelas-kelas yang sudah ditentukan sebelumnya. Klasifikasi dapat juga didefinisikan sebagai proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk memperkirakan kelas dari suatu objek yang labelnya tidak diketahui (Pramudiono, 2003). Sedangkan menurut Moertini (2002), klasifikasi dapat dikatakan sebagai fungsi pembelajaran yang memetakan (mengklasifikasi) sebuah unsur (item) data ke dalam salah satu dari beberapa kelas yang sudah didefinisikan sebelumnya.

Sasaran dari permasalahan-permasalahan klasifikasi ialah untuk mengidentifikasi karakteristik yang menunjukkan pada kelompok mana setiap kasus harus ditempatkan. Hal ini dapat digunakan untuk memahami data yang sudah ada, atau juga untuk memprediksi data yang baru. Proses klasifikasi pada *decision tree* biasanya dibagi menjadi dua fase, yaitu *learning* dan *test*. Dalam fase *learning*, sebagian data yang telah diketahui kelasnya digunakan untuk membentuk aturan *tree*, sedangkan pada fase *test*, aturan *tree* yang telah terbentuk digunakan untuk menganalisa data baru dan menguji tingkat akurasi datanya. Apabila tingkat akurasinya mencukupi, maka model tersebut dapat digunakan untuk memprediksi data baru.

Metode yang akan digunakan dalam penelitian ini adalah metode *decision tree* dengan algoritma PUBLIC. Metode-metode lain yang pada umumnya digunakan dalam proses klasifikasi adalah *neural network*, *genetic algorithm*, *fuzzy*, *case-based reasoning*, dan *k-nearest neighbor*.

2.4 Metode *Decision tree*

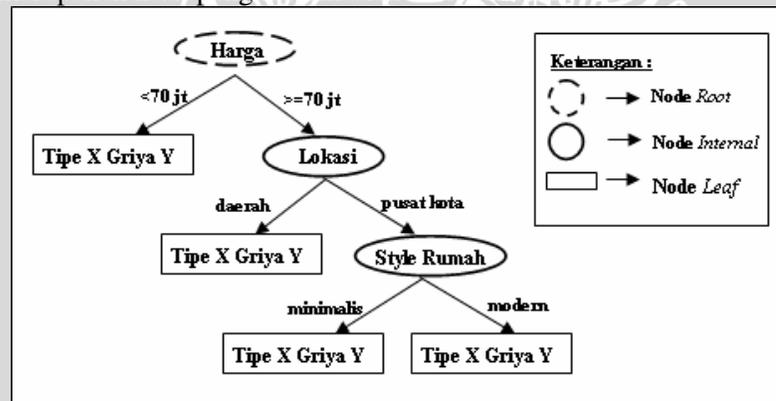
Sebagaimana yang telah dijelaskan bahwa proses klasifikasi data dalam suatu permasalahan dapat diselesaikan dengan menggunakan metode yang berbeda-beda, tetapi *decision tree* merupakan metode yang paling populer karena mudah untuk diinterpretasikan oleh manusia. Metode *decision tree* menggunakan representasi *tree* untuk menentukan aturan-aturan klasifikasi. Metode *decision tree* dapat juga dikatakan sebagai *flowchart* seperti struktur *tree*, dimana tiap *node internal* menunjukkan sebuah tes pada sebuah atribut, tiap cabang menunjukkan hasil dari tes, dan setiap *node leaf*

menunjukkan kelas-kelas atau distribusi kelas (Han dan Khamber, 2001). Metode *decision tree* memiliki dua tipe, yaitu *classification tree*, dan *regression tree*.

Proses dari *classification tree* adalah memberi label dan memasukkan *record-record* ke dalam kelas yang telah disediakan. Sedangkan proses *regression tree* ialah membuat estimasi nilai dari sebuah variabel target yang berdasar pada nilai numerik. Tipe metode *decision tree* yang akan digunakan dalam penelitian ini adalah *classification tree*, karena penelitian ini hanya mengklasifikasikan data-data ke dalam kelas-kelas yang telah tersedia.

Metode *decision tree* termasuk dalam *supervised learning* yaitu setiap masukan memiliki keluaran yang nantinya akan dijadikan sebagai acuan pembelajaran (*learning*). Maka dalam proses pembelajarannya membutuhkan kumpulan data yang banyak dengan kemungkinan variasi yang berbeda-beda pada tiap-tiap kelas yang telah didefinisikan. Hal ini disebabkan metode *decision tree* menganalisa dari data yang telah ada, apabila data tersebut secara sistematis kekurangan data pembelajarannya maka hasil klasifikasi akan menjadi kurang optimal.

Suatu *tree* direpresentasikan dengan tiga tipe *node* yang berbeda, diantaranya adalah *node root*, *node internal*, dan *node leaf* atau *terminal*. Contoh representasi tipe *node* yang berbeda dapat dilihat pada Gambar 2.2. Dimana yang bertindak sebagai *node root* adalah kriteria harga, sebagai *node internal* adalah kriteria lokasi dan biaya rumah, dan sebagai *node leaf* adalah produk-produk rumah yang merupakan hasil pengklasifikasian dari semua kriteria.



Gambar 2.2 Contoh penerapan metode *decision tree* untuk klasifikasi tipe rumah (Rastogi dan Shim, 1998)

repository.ub.ac

Terdapat beberapa algoritma pada metode *decision tree*, seperti CLS, ID3, C4.5, CART, SLIQ, SPRINT. Kebanyakan algoritma tersebut memiliki 2 fase yang terpisah, yaitu fase pertumbuhan yang diikuti dengan fase pemangkasan. Membangun pohon keputusan dalam 2 fase yang terpisah menjadi usaha yang sia-sia, ketika keseluruhan *subtree* yang dibangun pada fase pertama kemudian dipangkas pada fase berikutnya. Inilah alasan mengapa dalam klasifikasi data produk rumah digunakan algoritma PUBLIC. Karena PUBLIC lebih memilih mengintegrasikan fase pemangkasan ke dalam fase pertumbuhan daripada melakukan tiap fase secara berurutan. Pembahasan lebih lanjut mengenai algoritma PUBLIC akan dijelaskan pada subbab 2.5 Algoritma PUBLIC.

2.5 Algoritma PUBLIC

Algoritma PUBLIC (*PrUning and BuiLding Integrated in Classification*) adalah sebuah algoritma baru *decision tree* yang menawarkan integrasi fase pemangkasan dalam fase pertumbuhan. Selama fase pertumbuhan, sebelum melakukan *split* (percabangan) pada sebuah *node*, jika dapat diketahui bahwa *node* tersebut akan dipangkas pada fase pemangkasan, maka pembangunan *subtree* yang berakar pada *node* tersebut dapat dihindari. Sehingga jumlah *node* yang diperluas pada fase pertumbuhan menjadi jauh berkurang, dan tentunya ini meningkatkan performansi *decision tree* dan reduksi yang signifikan dalam pembacaan I/O.

Seperti yang dihasilkan dari percobaan yang dilakukan Rastogi dan Shim dengan data dalam bidang kedokteran (1998) pada Tabel 2.2. Rastogi mencoba membandingkan waktu eksekusi yang timbul dalam pengklasifikasian tiap himpunan data nyata yang ditunjukkan pada Tabel 2.1, antara algoritma SPRINT dengan algoritma PUBLIC. SPRINT dianggap sebagai perwakilan klassifier tradisional yang fase pertumbuhan dan pemangkasannya dilakukan terpisah.

Hasilnya, dari perbandingan waktu eksekusi, didapatkan waktu yang ditempuh algoritma SPRINT jauh lebih buruk dari algoritma PUBLIC. Misalnya pada himpunan data 'yeast', SPRINT membutuhkan waktu eksekusi 83% lebih banyak daripada PUBLIC. Ini membuktikan bahwa pemangkasan di awal yang dilakukan PUBLIC dapat mengurangi jumlah pertumbuhan *node* sejenis secara *significant*, sehingga berpengaruh pula pada waktu eksekusi yang diperlukan.

Tabel 2.1 Himpunan Data Nyata (Rastogi dan Shim, 1998)

Data Set	breast cancer	car	letter	Satima Ge	shuttle	VeHicle	yeast
No. of Categorical Attributes	0	6	0	0	0	0	0
No. of Numeric Attributes	9	0	16	36	9	18	8
No. of Class	2	4	26	7	5	4	10
No. of Records (Train)	469	1161	13368	4435	43500	559	1001
No. of Records (Test)	214	567	6632	2000	14500	287	483

Tabel 2.2 Waktu Eksekusi Data Nyata (detik) (Rastogi dan Shim, 1998)

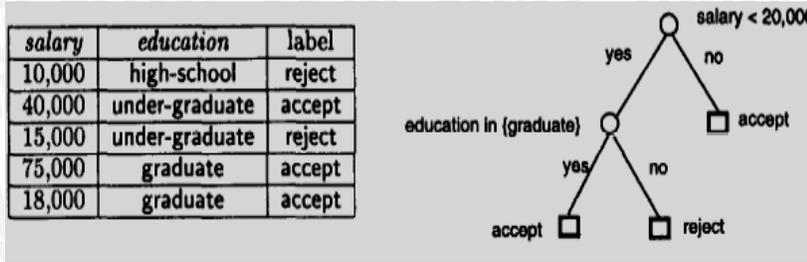
Data Set	breast cancer	car	Letter	satimage	shuttle	vehicle	yeast
SPRINT	21.49	45.85	3283.20	1471.00	457.78	151.90	253.96
PUBLIC	13.90	33.93	2793.32	1042.39	455.20	97.09	139.02
Max Ratio	56%	38%	18%	43%	0.6%	55%	83%

Dengan hanya memangkas *node* yang diketahui akan dipangkas pada fase pemangkasan, dijamin bahwa *tree* yang digenerasi dengan pendekatan integrasi PUBLIC akan sama persis dengan *tree* yang digenerasi dari hasil melakukan 2 fase secara terpisah.

2.5.1. Fase Pertumbuhan Pohon

Pada fase pertumbuhan, *tree* dibangun secara *breadth-first*, yaitu pembangunan per level atau ke samping, dengan membagi data pelatihan berulang kali hingga setiap partisi berisi *record-record* pada kelas yang sama. Kondisi *split* untuk membagi data pelatihan, berbentuk $A < v$ (v adalah nilai pada domain A) bila A adalah atribut numerik, atau berbentuk $A \in V$ (V adalah himpunan nilai dari domain A), bila A adalah atribut kategoris. Dengan demikian tiap percabangan akan dibagi menjadi 2.

Atribut numerik adalah atribut-atribut dengan domain terurut, sedangkan atribut kategoris adalah atribut dengan domain diskrit. Contoh penggunaan atribut numerik dan diskrit dapat dilihat pada Gambar 2.3. Dimana *salary* merupakan contoh dari atribut numerik, dan *education* sebagai contoh atribut kategoris dengan domain {*high-school, undergraduate, graduate*}.



Gambar 2.3 Contoh Aplikasi Persetujuan Pinjaman (Rastogi dan Shim, 1998)

Gambar 2.3(a) menunjukkan sebuah contoh himpunan pelatihan untuk sebuah aplikasi persetujuan pinjaman. Ada sebuah *record* tunggal yang bersesuaian dengan masing-masing permintaan pinjaman, yang masing-masing dicocokkan dengan satu dari dua label – *accept* jika permintaan pinjamannya diterima atau *reject* jika permintaan pinjamannya ditolak. Tiap *record* dikarakteristikan dengan 2 atribut, *salary* dan *education*. Atribut-atribut menyatakan pendapatan dan level pendidikan dari pemohon pinjaman. Tujuan *classifier* adalah untuk melakukan deduksi, dari data training, kondisi-kondisi yang jelas dan berarti yang mengandung *salary* dan *education* dalam sebuah permintaan pinjaman diterima atau ditolak.

Gambar 2.3(b) adalah sebuah pohon keputusan untuk data pelatihan pada Gambar 2.3(a). Tiap *node* internal dari pohon keputusan memiliki sebuah tes yang mengandung sebuah atribut, dan sebuah cabang keluar untuk setiap kemungkinan keluaran. Tiap daun memiliki sebuah kelas yang berhubungan. Untuk mengklasifikasikan *record-record* baru dengan sebuah pohon keputusan, dimulai dari *node* akar, *node-node* yang mengikutinya dikunjungi hingga dicapai sebuah daun. Pada tiap *node* internal, tes untuk *node* tersebut diterapkan pada *record*. Keluaran dari tes pada sebuah *node* internal menentukan cabang yang akan ditelusuri, dan *node* yang dikunjungi berikutnya. Kelas untuk *record* tersebut adalah kelas dari *node* daun terakhir. Maka, pohon keputusan pada Gambar 2.3(b) menerima sebuah permintaan pinjaman hanya jika $salary \geq 20,000$ atau $education \in \{graduate\}$; selain itu, aplikasi pinjaman ditolak

Langkah-langkah proses dalam fase pertumbuhan *tree* adalah sebagai berikut:

✓ **Pembangunan Struktur Data**

Tiap *node* pada pohon keputusan membangun daftar pemisahan untuk tiap atribut. Setiap daftar atribut berisi masukan tunggal untuk tiap *record* pada partisi untuk *node* tersebut. Masukan daftar atribut untuk sebuah *record* terdiri atas 3 *fields* – nilai atribut dalam *record*, label kelas untuk *record* tersebut, dan sebuah *record* identifier. Daftar- atribut untuk *node* akar dibuat dahulu dari data masukan, sedangkan untuk *node* lain diturunkan dari daftar atribut induknya saat *node-node* induk di-*split*. Daftar atribut untuk atribut numerik pada *node* akar yang diurutkan terlebih dahulu, dan urutan tersebut dipertahankan untuk *node-node* lainnya oleh prosedur *splitting*. Juga, pada tiap *node*, dibuat histogram distribusi kelas dari *record-record* pada *node* tersebut.

Sehingga, dapat dikatakan inisialisasi untuk *node* akar pada algoritma pertumbuhan melibatkan:

1. Pembangunan daftar atribut
2. Pengurutan daftar atribut untuk atribut numerik
3. Pembangunan histogram distribusi kelas

✓ **Pemilihan *Cutpoint* pada Atribut**

Untuk sebuah himpunan *record* S , nilai *entropy* atributnya dihitung dengan menggunakan Persamaan 2.1.

$$E(S) = -\sum_{i=1}^m p_i \log_2(p_i) \quad (2.1)$$

di mana p_i adalah frekuensi relatif kelas i dalam S dan $E(S)$ adalah nilai *entropy* pada himpunan *record* S .

Entropy adalah sebuah rumus untuk menghitung tingkat homogenitas data sampel. Makin homogen sebuah himpunan dengan kelas-kelas dalam himpunan tersebut, makin rendah entropinya.

Berbeda dengan tipe atribut kategoris dimana subset dari tiap atribut bernilai diskrit, atribut numerik memiliki nilai dengan range dari null sampai tak terhingga, sedangkan dalam proses klasifikasi semua atribut harus bernilai kategoris. Sehingga atribut dengan nilai kontinyu (*continuous*) harus didiskritkan.

Pemilihan *cutpoint* terbaik pada suatu atribut ditentukan berdasarkan perhitungan *class information entropy*. *Cutpoint* (titik percabangan) yang terbaik adalah yang memberikan *class information entropy* yang paling kecil diantara semua kandidat

repository.ub.ac

cutpoint. *Class information entropy* yang membagi S dengan n *record* menjadi himpunan-himpunan S_1 dengan n_1 *record* dan S_2 dengan n_2 *record* dapat dirumuskan pada Persamaan 2.2.

$$E(S_1, S_2) = \frac{n_1}{n} E(S_1) + \frac{n_2}{n} E(S_2) \quad (2.2)$$

Dimana $E(S_1)$ = *class entropy* dari subset S_1 ,
 S_1 = subset dari S ,
 $E(S_1, S_2)$ = *class information entropy* partisi dengan *cutpoint* di S_1 ,
 n_i = jumlah sampel di S_i .

Untuk menghitung *cutpoint* terbaik untuk atribut numerik, daftar atribut (terurut) ditelusuri dari awal dan pada tiap titik *split*, distribusi kelas dalam 2 partisi ditentukan dengan histogram kelas untuk *node* tersebut. Entropi untuk tiap titik *split* dihitung dari daftar terurut. Untuk atribut kategoris, daftar atribut ditelusuri untuk pertama kali dengan membangun sebuah histogram yang mengandung distribusi kelas untuk tiap nilai atribut. Histogram ini kemudian digunakan untuk menghitung entropi untuk tiap titik *split*.

✓ Daftar Atribut Hasil *Splitting*

Sekali *split* terbaik pada *node* ditemukan, titik *split* tersebut akan digunakan untuk membagi daftar atribut pada atribut *splitting* diantara 2 *node* anak. Tiap *record identifier* beserta informasi tentang *node* anak ditugaskan untuk (ke kiri atau kanan) dan dimasukkan ke dalam *hashtable*. Sisa daftar atribut kemudian akan dipecah dan dimasukkan ke dalam *hashtable* beserta informasinya menggunakan *record identifier* tersebut. Histogram distribusi kelas kedua *node* anak juga akan dihitung selama tahap ini.

2.5.2. Fase Pemangkasan Pohon

Sebagian besar algoritma melaksanakan fase pemangkasan setelah fase pertumbuhan di mana *node-node* dipangkas secara iteratif untuk mencegah *overfitting* dan untuk memperoleh pohon dengan akurasi yang lebih tinggi (Gehrke, 1996). *Overfitting* adalah *error* yang timbul ketika pohon keputusan diterapkan pada data tes, karena ukuran pohon terlalu besar.

Pohon yang sudah dibangun pada fase pertama telah berisi klasifikasi yang lengkap berdasarkan dataset pelatihan, bahkan untuk data *noise*. Dari hasil klasifikasi tersebut terbentuk sebuah *tree* beserta cabang-cabang yang menunjuk pada tiap kelas tertentu. Cabang-cabang tersebut dapat menyebabkan *error* saat klasifikasi data tes. Fase pemangkasan ditujukan untuk menghapus cabang-cabang tersebut dari pohon keputusan dengan menyeleksi *subtree* dengan perkiraan *error rate* terendah.

Berikut ini ditampilkan beberapa pendekatan pemangkasan yang populer:

1. Pemangkasan berdasarkan kompleksitas *error*

Sebuah metode pemangkasan populer yang biasa disebut pemangkasan berdasarkan kompleksitas *error* dinyatakan dalam Persamaan 2.3

$$E_{er} = \frac{\tilde{R} - R}{L} \quad (2.3)$$

Dengan R = *error rate* (probabilitas *error*) dari *tree* yang belum dipangkas,

\tilde{R} = *error rate* setelah sebuah *node* dihapus

L = jumlah *node-node* daun pada *subtree* dari *node* yang sedang dievaluasi dalam pemangkasan.

Error rate dihitung berdasarkan sekumpulan *instance* yang tidak terpengaruh dengan himpunan yang digunakan untuk membangun pohon. E_{er} dihitung untuk tiap *node* non-terminal dan *node* dengan ukuran kompleksitas error terkecil dihapus.

2. Pemangkasan berdasarkan *error* minimum

Pemangkasan berdasarkan *error* minimum dilakukan berdasarkan Persamaan 2.4 untuk menghitung *error rate* yang diharapkan,

$$E_{me} = \frac{N - N_{\omega_k} + C - 1}{N + C} \quad (2.4)$$

Di mana, N = jumlah *record* pada sebuah *node* tertentu

N_{ω_k} = jumlah *record* dari kelas yang dominan.

Error rate yang diharapkan dari pohon yang tidak dipangkas didapat dengan cara menghitung *error rate* perkiraan untuk tiap

cabang yang mengandung bobot berupa jumlah *record* pada cabang tersebut.

Error rate yang diharapkan jika sebuah *node* dipangkas (yaitu, seluruh anaknya dihapus) dihitung berdasarkan *record-record* pada *node* tersebut.

Jika *error rate* yang diharapkan dengan pemangkasan lebih rendah daripada tanpa pemangkasan, anak-anak di bawah *node* tersebut dipangkas dan *node* tersebut dibuat menjadi sebuah *node* terminal.

3. Pemangkasan berdasarkan *Minimum Description Length* (MDL)
Sebuah kelas penting dari algoritma-algoritma pemangkasan adalah algoritma yang berdasarkan prinsip *Minimum Description Length* (MDL) (Mehta, 1995).

Pohon keputusan 'terbaik' dapat ditentukan sebagai sebuah pohon yang dapat mengkodekan kelas-kelas dari *record-record* dengan *cost* minimum. Hal ini dapat dicapai sebagai berikut. Sebuah *subtree* *S* dipangkas, jika *cost* pengkodean langsung terhadap *record-record* dalam *S* tidak melebihi *cost* pengkodean *subtree* ditambah *cost* dari *record-record* pada tiap daun dari *subtree*.

Dalam (Mehta, 1995), ditunjukkan bahwa pruning MDL:

- a) menghasilkan pohon akurat untuk lingkup himpunan data yang luas,
- b) menghasilkan pohon yang secara signifikan lebih ringkas,
- c) secara perhitungan lebih efisien dan tidak menggunakan himpunan data yang terpisah untuk pemangkasan.

Tujuan pruning MDL adalah untuk menemukan *subtree* yang mendeskripsikan training set *S* dengan *cost* minimum. Prinsip MDL dapat digunakan untuk mencegah *overfitting*, dengan cara memangkas *tree* yang dibangun pada fase pertumbuhan.

Terdapat beberapa metode *heuristic* pemangkasan lainnya yang telah diusulkan. Misalnya, antara lain:

- 1) Metode nilai kritis, menggunakan sebuah *threshold* pada nilai-nilai yang diperoleh selama pemecahan *node* hingga penghapusan *subtree-subtree*.
- 2) Metode *error* yang tereduksi adalah berdasarkan pada penggunaan sebuah himpunan data terpisah dan pengamatan

jumlah *error* pada tiap *node* non-terminal saat sebuah *node* dipertahankan dan saat *subtree* anak dihapus. *Subtree* anak bersesuaian dengan *node* yang menghasilkan pengurangan jumlah klasifikasi salah terbesar.

Biaya pemangkasan *tree* tergantung pada *cost* pengkodean *tree*.

- **Cost Pengkodean Tree**

Cost pengkodean *tree* terdiri atas:

1. *Cost* pengkodean struktur *tree*.

Struktur *tree* dapat dikodekan dengan bit tunggal untuk menentukan *node* dari *tree* adalah *node* internal (1) atau daun (0). Teknik pengkodean ini sudah mendekati optimal.

2. *Cost* pengkodean tiap *split*, atribut dan nilai untuk *split*.

Cost pengkodean *split* merupakan bagian penting dalam pengkodean *tree*. *Cost* pengkodean tiap *split* dinyatakan dengan $C_{split}(N)$. Dimana termasuk di dalamnya penentuan atribut yang digunakan untuk men-*split node* dan nilai atribut.

Atribut *splitting* dapat dikodekan dengan $\log a$ bit (karena ada a atribut), sementara penentuan nilai kriteria tergantung pada apakah atribut tersebut kategoris atau numerik. v adalah jumlah nilai berbeda pada atribut *splitting* pada *record* suatu *node*. Jika atribut *splitting* numerik, maka ada $v-1$ titik berbeda pada *node* yang dapat dibagi, dan memerlukan $\log(v-1)$ untuk mengkodekan titik *splitnya*.

Di sisi lain, untuk atribut kategoris, terdapat 2^v subset nilai berbeda. Dimana himpunan kosong dan himpunan yang mengandung seluruh nilai bukan kandidat *splitting*. Dan *cost splitnya* adalah $\log(2^v-2)$.

Sehingga, *cost* pengkodean *split* ($C_{split}(N)$):

a) Untuk atribut kategoris, *cost* $C_{split}(N)$ tampak pada Persamaan 2.5.

$$cost = \log a + \log(2^v - 2) \quad (2.5)$$

b) Untuk atribut numerik, *cost* $C_{split}(N)$ tampak pada Persamaan 2.6

$$cost = \log a + \log(v - 1) \quad (2.6)$$

3. *Cost* pengkodean kelas-kelas *record* data pada tiap daun dari *tree*.

Sebuah himpunan S yang terdiri dari n record, dimana masing-masing merupakan bagian dari kelas k . Biaya pengkodean kelas untuk sejumlah n record dinyatakan pada Persamaan 2.7

$$C(S) = \log_2 \binom{n+k-1}{k-1} + \log_2 \frac{n!}{n_1! \dots n_k!} \quad (2.7)$$

Dimana, $C(S)$ = biaya pengkodean kelas untuk record pada himpunan S

n_i = jumlah record pada kelas i untuk $1 \leq i < k$

k = jumlah kelas

2.5.3. Fase Pertumbuhan dan Pemangkasan dalam PUBLIC

Algoritma PUBLIC menggunakan prosedur pertumbuhan yang biasa digunakan dalam algoritma klasifikasi lain. Perbedaannya adalah bahwa secara periodik atau setelah sejumlah tertentu *node split, tree* yang sudah dibangun parsial tersebut dipangkas. Algoritma pemangkasan yang biasa tidak dapat digunakan untuk memangkas *tree* parsial tersebut.

Masalah yang dihadapi dengan prosedur pemangkasan sebelum *tree* lengkap adalah bahwa *cost* termurah *subtree* yang berakar pada sebuah daun N diasumsikan sebagai $C(S)+1$. Ini tidak benar untuk *tree* yang baru dibangun sebagian karena sebuah daun pada *tree* parsial mungkin akan di-*split* belakangan. $C(S)+1$ menyebabkan *over-estimate cost subtree* termurah pada N dan hal ini dapat menyebabkan *over-pruning*, yaitu, *node-node* dipangkas pada fase pembangunan padahal tidak akan dipangkas selama fase pemangkasan. Hal tersebut tidak diinginkan karena *decision tree* PUBLIC diharapkan identik dengan pohon keputusan yang dihasilkan oleh *classifier* tradisional.

Pada PUBLIC, sekali fase pembangunan berakhir, tidak ada lagi *node-node* daun yang termasuk dalam kategori ‘akan diperluas’. Penerapan algoritma pemangkasan PUBLIC menghasilkan *tree* terpangkas sama dengan yang dihasilkan dengan algoritma pemangkasan terpisah.

Algoritma pemangkasan pada PUBLIC dapat dilihat pada Gambar 2.4

```

procedure computeCost&PrunePublic(Node N) :
/* S is the set of data records for N */
1.if N is a "yet to be expanded" leaf
    Return lower bound on subtree cost at N
2.if N is a "pruned" or "not expandable" leaf
    Return (C(S)+1)
    /* N1 and N2 are N' children */
3.minCost1 := computeCost&PrunePublic(N1);
4.minCost2 := computeCost&PrunePublic(N2);
5.minCostN:=min{C(S)+1,Csplit(N)+1+minCost1+minCost2};
6.if minCostN = C(S)+1 {
7.    prune child nodes N1 and N2 from tree
8.    delete nodes N1 and N2 and all their
        descendants from Q
9.    mark node N as pruned
10. }
11.return minCostN

```

Gambar 2.4 Algoritma pemangkasan PUBLIC (Rastogi dan Shim, 1998)

- **Komputasi Lowerbound pada cost subtree**

Umumnya *subtree* yang berakar pada *node N* mempunyai biaya sedikitnya 1. Satu merupakan estimasi biaya *subtree* termurah pada daun yang diperluas. Algoritma PUBLIC mengurangi jumlah *node* yang dihasilkan dan memangkaskan lebih banyak *node*.

Pertimbangan dalam menghitung *cost subtree minimum* pada *node* daun antara lain:

- 1) Biaya *split*
- 2) Penggabungan biaya penguraian nilai tiap *split*

Algoritma untuk menghitung *lowerbound* pada *subtree* ditunjukkan pada Gambar 2.5

```

procedure computeMinCostS(Node N) :
/* n1, ..., nk are sorted in decreasing order */
1. if k = 1 return (C(S)+1)
2. s:=1
3. tmpCost := 2*s+1+s*log a +  $\sum_{i=s+2}^k n_i$ 
4. while s+1 < k and ns+2 > 2+log a do {
5.    tmpCost := tmpCost + 2 + log a - ns+2
6.    s++
7. }
8. return min{C(S)+1, tmpCost}

```

Gambar 2.5 Algoritma perhitungan *lowerbound* (Rastogi dan Shim, 1998)

Dari Gambar 2.5, bila *node* N bukan *split* yaitu $k=0$, maka biaya minimum *subtree* pada *node* N adalah $C(S) + 1$.

Untuk $s > 0$, *lowerbound* pada biaya pengkodean *subtree* dengan s *split* dan berakar pada *node* N ditunjukkan pada Persamaan 2.8

$$tmpCost = 2 * s + 1 + s * \log a + \sum_{i=s+2}^k n_i \quad (2.8)$$

Dimana: s = jumlah *split*

1 bit pada $(2*s+1)$ berguna untuk menetapkan tipe tiap *node*

a = jumlah atribut

k = jumlah kelas untuk *record* dalam S

n_i = jumlah *record* pada kelas i dalam S

dan $n_i > n_{i+1}$ untuk $1 \leq i < k$ (n_1, \dots, n_k diurutkan menurun berdasarkan nilainya).



BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab metodologi dan perancangan ini akan dibahas metode perancangan yang digunakan dan langkah-langkah yang dilakukan dalam penelitian Penerapan *Decision Tree* Menggunakan Algoritma PUBLIC dalam Sistem Rekomendasi Pembelian Rumah.

Penelitian dilakukan dengan tahapan-tahapan sebagai berikut:

1. Mempelajari berbagai literatur yang terkait dengan masalah klasifikasi data menggunakan algoritma PUBLIC, baik melalui buku ataupun source dari internet.
2. Mengumpulkan data hasil produk para pengembang perumahan di kota/kabupaten Malang, dengan cara mengumpulkan brosur-brosur hasil produk rumah para pengembang.
3. Menganalisis dan melakukan perancangan sistem rekomendasi pembelian rumah menggunakan algoritma PUBLIC.
4. Mengimplementasikan hasil analisis dan rancangan yang dilakukan pada tahap sebelumnya menjadi perangkat lunak sistem rekomendasi pembelian rumah.
5. Melakukan uji coba sistem dengan memasukkan data kriteria pembeli ke dalam sistem. Hasil yang diperoleh adalah rekomendasi tipe rumah.
6. Mengevaluasi hasil rekomendasi sistem dengan menggunakan data uji.

Langkah-langkah yang dilakukan dapat dijelaskan kembali dalam bentuk alur yang ditunjukkan dalam Gambar 3.1

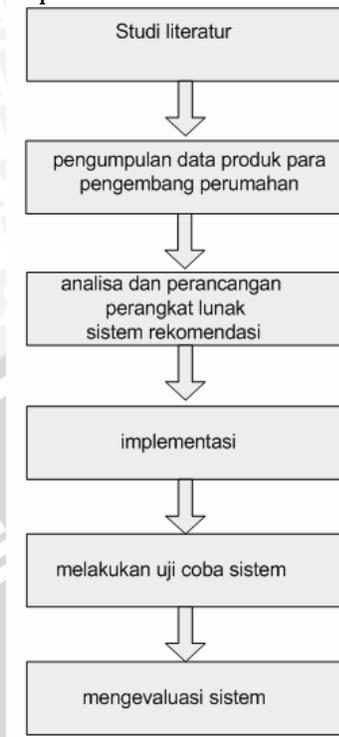
3.1 Analisis Sistem

Pada subbab analisis sistem ini akan dijelaskan mengenai deksripsi sistem dan batasan sistem.

3.1.1 Deskripsi sistem

Sistem yang akan dibangun adalah sistem rekomendasi pembelian rumah. Sistem ini bertujuan untuk memberikan rekomendasi kepada pembeli, berupa produk rumah dari pengembang tertentu berdasarkan data kriteria yang diinputkan oleh pembeli. Sistem yang dibangun lebih ditujukan bagi pihak asosiasi pengembang perumahan dengan tujuan untuk meningkatkan pelayanan kepada

pembeli sekaligus memberikan rekomendasi produk rumah yang sesuai dengan kriteria pembeli.



Gambar 3.1 Langkah-langkah penelitian

Dalam penelitian ini, produk rumah yang akan dijadikan sampel adalah produk rumah dari para pengembang perumahan di daerah kota/kabupaten Malang. Data produk perumahan dari para pengembang perumahan ini dikumpulkan dan diklasifikasikan berdasarkan tipe rumahnya, untuk kemudian dihasilkan *output* produk rumah yang paling mendekati data kriteria pembeli.

Secara umum tipe rumah masing-masing pengembang dibedakan menjadi tipe rumah 30, tipe 40, tipe 50, tipe 60, tipe 70, dan tipe 80. Rumah tipe 30 adalah kumpulan rumah yang luas bangunannya antara 30 m² sampai dengan kurang dari 40 m², tipe 40 adalah kumpulan rumah yang luas bangunannya antara 40 m² sampai dengan kurang dari 50 m², begitu juga dengan tipe-tipe yang lain. Sedangkan untuk tipe 80 adalah kumpulan rumah yang luas

bangunannya sama dengan atau lebih dari 80 m². Semua data produk rumah dari tiap pengembang nantinya akan diklasifikasikan berdasarkan tipe-tipe tersebut.

Proses klasifikasi tipe rumah dilakukan dengan tujuan untuk mengarahkan proses pencarian produk pada saat rekomendasi, sehingga sistem akan mencari data produk yang paling sesuai dengan data inputan kriteria pembeli.

Data kriteria pembeli yang akan diperhitungkan antara lain harga rumah (harga tunai, uang muka, harga angsuran) yang diinginkan, jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja pembeli. Dari data kriteria tersebut, sistem akan menentukan tipe rumah yang sesuai berdasarkan aturan klasifikasi yang telah terbentuk. Penjelasan dari tiap-tiap prosesnya akan dibahas lebih lanjut pada subbab 3.3.1 Perancangan proses.

3.1.2 Batasan sistem

Batasan sistem yang akan dikembangkan adalah:

1. Data *sampel* yang digunakan dalam sistem adalah data produk pada periode bulan Maret 2008 di wilayah kota/kabupaten Malang.
2. Sebelum sistem rekomendasi dijalankan (dieksekusi) proses klasifikasi tipe rumah dari tiap pengembang harus dilakukan terlebih dahulu.
3. Sistem hanya memberikan rekomendasi berupa nama produk tanpa memperhatikan kondisi produk, spesifikasi produk, (ataupun kualitas pengembang).

3.2 Perancangan Basis Data

Dari analisis sistem pada Subbab 3.1, maka diperlukan adanya tabel-tabel dalam basis data yang digunakan untuk menyimpan data. Tabel-tabel yang digunakan pada penelitian ini hanya terbatas pada atribut-atribut yang terlibat dalam proses perhitungan, serta beberapa tambahan atribut sebagai keterangan. Selain itu relasi antar tabelnya pun dibuat cukup sederhana untuk menyederhanakan *query* yang digunakan saat penelitian dilakukan.

Adapun tabel-tabel tersebut adalah:

1. Tabel kriteria (t_Kriteria)
2. Tabel rumah (t_Rumah)

3. Tabel pengembang (t_Pengembang)
4. Tabel lokasi (t_Lokasi)
5. Tabel klasifikasi (t_Klasifikasi)

Tabel rumah, pengembang, dan lokasi adalah tabel-tabel yang digunakan untuk menyimpan data *training* pada tahap pembelajaran. Tabel klasifikasi digunakan oleh dua proses yaitu penyimpanan data *training* dan proses pengklasifikasian data produk. Sedangkan Tabel kriteria digunakan untuk menyimpan data tes pada tahap pengujian.

Penjelasan lebih lanjut mengenai tabel-tabel tersebut adalah sebagai berikut:

1. Tabel kriteria

Tabel kriteria adalah tabel yang digunakan untuk menyimpan data kriteria rumah yang diinginkan pembeli, seperti No.KTP, tunai_user, jenis_bayar, muka_user, angsuran_user, waktu_user, bangunan_user, tanah_user, dan lokasi_user.

Tabel 3.1 Tabel Kriteria (t_Kriteria)

Field	Type	Allow Null
No_KTP	nVarchar(16)	No
tunai_user	Int	No
jenis_bayar	nVarchar(16)	No
muka_user	Int	No
angsuran_user	Int	No
waktu_user	Int	No
bangunan_user	Int	No
tanah_user	Int	No
lokasi_user	nVarchar(26)	Yes
jml_rekomen	Int	No

Keterangan:

- No_KTP : No KTP pembeli
- tunai_user : Besar *budget* yang dimiliki untuk membeli rumah
- jenis_bayar : Jenis pembayaran yang diinginkan (tunai/ kredit)
- muka_user : Batas maksimal uang muka pembelian rumah yang diinginkan pembeli
- angsuran_user : Batas maksimal harga angsuran pembelian rumah yang diinginkan pembeli

waktu_ : Jangka waktu mengangsur yang diinginkan
 angsuran pembeli
 bangunan_ : Luas bangunan yang diinginkan pembeli
 user
 tanah_user : Luas tanah yang diinginkan pembeli
 lokasi_user : Lokasi tempat bekerja pembeli
 jml_ : Jumlah hasil rekomendasi yang diinginkan
 rekomen pembeli

2. Tabel rumah

Tabel rumah adalah tabel yang digunakan untuk menyimpan data hasil produk para pengembang perumahan, seperti id_rumah, nama_rumah, harga tunai, uang_muka, angsuran_5th, angsuran_10th, angsuran_15th, luas_bangunan, luas_tanah, tipe_rumah, dan id_pengembang. Untuk lebih jelasnya dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel rumah (t_Rumah)

Field	Tipe	Allow Null
id_rumah	nChar(3)	No
nama_rumah	nVarchar(20)	No
harga_tunai	Int	No
uang_muka	Int	No
angsuran_5th	Int	No
angsuran_10th	Int	No
angsuran_15th	Int	No
luas_bangunan	Int	No
luas_tanah	Int	No
tipe_rumah	Int	No
id_pengembang	nVarchar(10)	No

Keterangan:

id_rumah : Kode rumah
 nama_rumah : Nama produk rumah
 harga_tunai : Harga tunai produk rumah
 uang_muka : Harga uang muka pembayaran rumah secara kredit
 angsuran_5th : Harga angsuran pembayaran rumah secara kredit dalam jangka waktu 5 tahun

angsuran_10th	:	Harga angsuran pembayaran rumah secara kredit dalam jangka waktu 10 tahun
angsuran_15th	:	Harga angsuran pembayaran rumah secara kredit dalam jangka waktu 15 tahun
luas_bangunan	:	Luas bangunan produk rumah
luas_tanah	:	Luas tanah produk rumah
tipe_rumah	:	Tipe rumah
id_pengembang	:	Kode pengembang perumahan yang memproduksi perumahan tersebut

3. Tabel pengembang

Tabel pengembang menyimpan data mengenai pengembang perumahan seperti `id_pengembang`, `nama_pengembang` dan `lokasi_rumah`. Untuk lebih jelasnya dapat dilihat pada Tabel 3.3

Tabel 3.3 Tabel pengembang (t_Pengembang)

Field	Tipe	Allow Null
<code>id_pengembang</code>	nVarchar(10)	No
<code>nama_pengembang</code>	nVarchar(30)	No
<code>lokasi_rumah</code>	nVarchar(20)	No

Keterangan:

<code>id_pengembang</code>	:	Kode pengembang
<code>nama_pengembang</code>	:	Nama pengembang perumahan
<code>lokasi_rumah</code>	:	Lokasi suatu perumahan berada

4. Tabel lokasi

Tabel lokasi menyimpan data mengenai jarak antara lokasi produk perumahan dengan kantor pembeli, seperti `lokasi_rumah`, `lokasi_kerja`, `jarak`, dan `id_pengembang`. Untuk lebih jelasnya dapat dilihat pada Tabel 3.4

Tabel 3.4 Tabel lokasi (t_lokasi)

Field	Tipe	Allow Null
<code>lokasi_rumah</code>	nVarchar(10)	No
<code>lokasi_kerja</code>	nVarchar(10)	No
Jarak	Double	No
<code>id_pengembang</code>	nVarchar(10)	No

Keterangan:

- lokasi_rumah : Lokasi suatu perumahan berada
- lokasi_kerja : Lokasi tempat kerja pembeli
- Jarak : Jarak antara lokasi perumahan dengan lokasi kerja pembeli
- id_pengembang : Kode pengembang

5. Tabel klasifikasi

Tabel klasifikasi menyimpan data mengenai proses klasifikasi dan perekomendasi, seperti No.KTP, id_klasifikasi, rekomendasi dan waktu. Untuk lebih jelasnya dapat dilihat pada Tabel 3.5

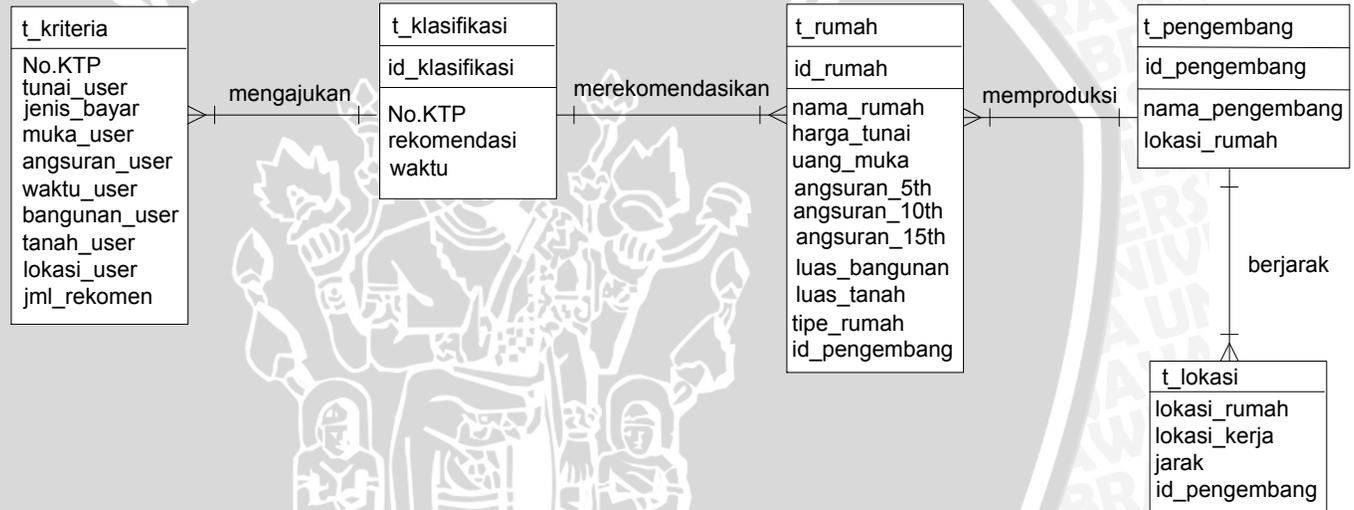
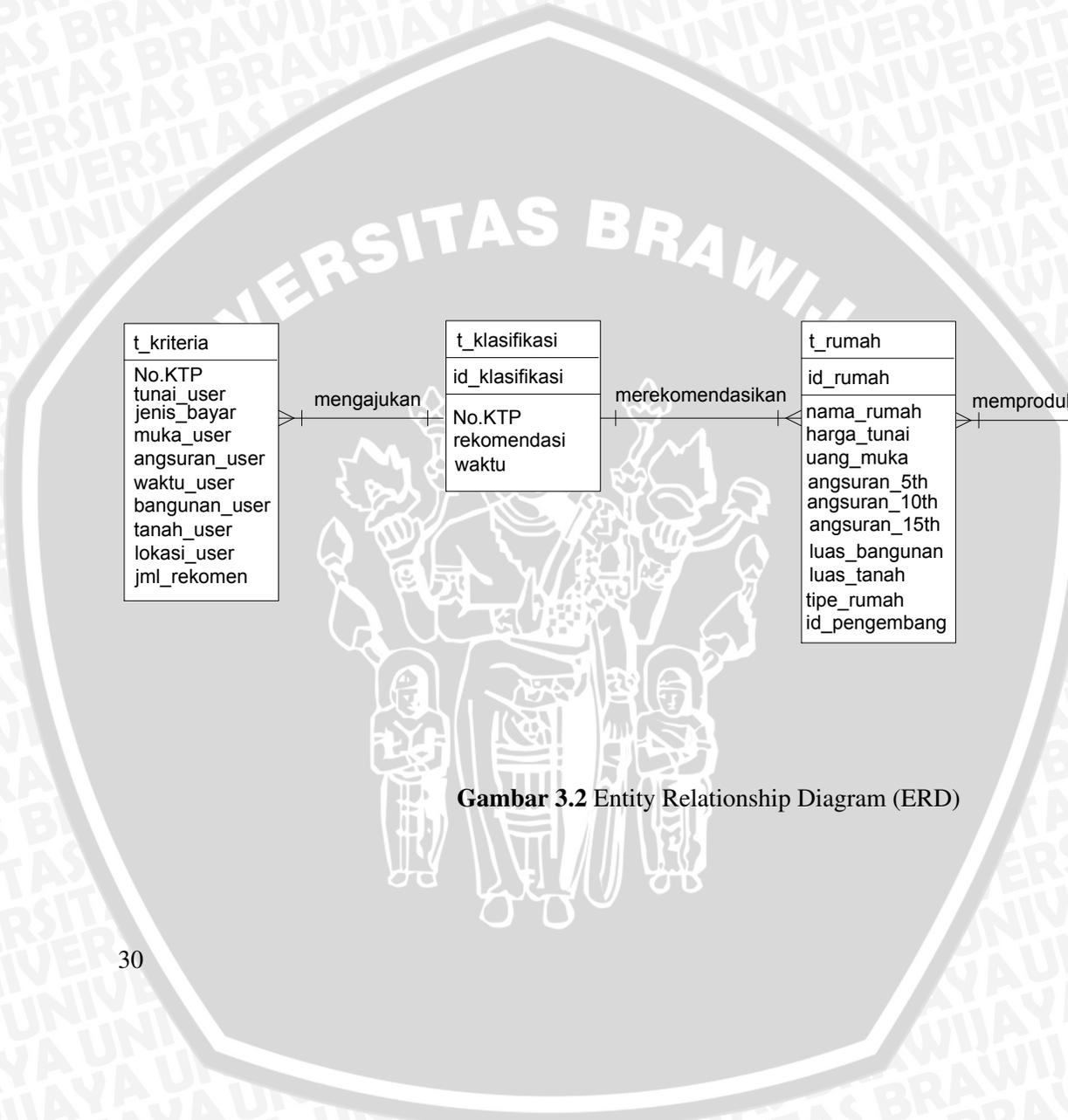
Tabel 3.5 Tabel klasifikasi (t_ Klasifikasi)

Field	Tipe	Allow Null
No.KTP	nVarchar(10)	No
id_klasifikasi	nVarchar(15)	No
Rekomendasi	nVarchar(60)	No
Waktu	Datetime	No

Keterangan:

- No.KTP : No.KTP pembeli
- id_klasifikasi : Kode klasifikasi
- Rekomendasi : Kode rumah hasil rekomendasi
- Waktu : Waktu yang diperlukan sistem dalam proses rekomendasi

Struktur dan relasi antar Tabel digambarkan pada Gambar 3.2



Gambar 3.2 Entity Relationship Diagram (ERD)

3.3 Perancangan Sistem

Pada subbab ini akan dijelaskan mengenai berbagai proses yang terjadi dalam membangun sistem rekomendasi pembelian rumah, antara lain perancangan proses dan perancangan antarmukanya.

3.3.1 Perancangan Proses

Dalam memberikan rekomendasi, sistem akan melakukan dua tahapan proses, yaitu proses klasifikasi tipe rumah, dan proses pencarian produk yang sesuai dengan data kriteria yang diinputkan pembeli. Pada proses klasifikasi tipe rumah, sistem akan membangun pohon keputusan dengan mengklasifikasikan data hasil produk tiap pengembang berdasarkan tipe rumahnya. Sebagai contoh seseorang yang ingin membeli rumah dengan kriteria harga tunai di bawah 120 juta, maka akan diarahkan pada tipe rumah tertentu dari produk-produk tiap pengembang yang sesuai dengan kriteria harga tunai tersebut.

Jadi proses klasifikasi pada data *trainingnya* akan mengacu pada tipe rumahnya, atau dengan kata lain tipe rumah merupakan kelas dari proses klasifikasinya.

Metode yang akan digunakan untuk membangun pohon keputusan tersebut adalah *decision tree* menggunakan algoritma PUBLIC. Hasil dari proses tersebut adalah aturan klasifikasi tipe rumah. Aturan klasifikasi tersebut akan digunakan pada saat proses rekomendasi dijalankan, yang berfungsi untuk mengarahkan proses pencarian produk pada tipe rumah yang ditentukan oleh sistem berdasarkan data inputan.

Dalam proses rekomendasi, sistem akan menerima inputan data berupa data kriteria rumah yang diinginkan pembeli (seperti harga tunai, uang muka, harga angsuran, jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja pembeli). Dari data tersebut, sistem akan menentukan produk rumah yang sesuai dengan membaca aturan klasifikasi yang telah terbentuk. Produk rumah yang paling mendekati kriteria pembelilah yang direkomendasikan.

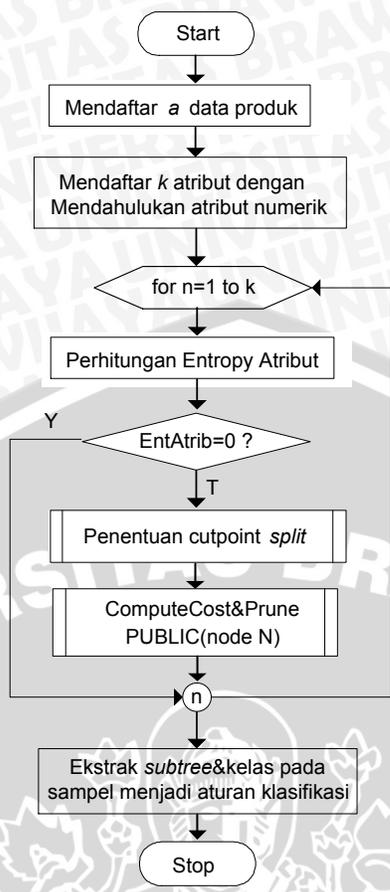
Output dari proses rekomendasi tersebut adalah produk rumah yang direkomendasikan dari data produk para pengembang perumahan. Proses perhitungan dari masing-masing tahapan akan dijelaskan pada subbab 3.5 Contoh perhitungan manual.

3.3.1.1 Proses klasifikasi tipe rumah pada data *training* menggunakan *decision tree* dengan algoritma PUBLIC

Pada proses ini sistem akan melakukan klasifikasi pada data *training* berdasarkan tipe rumah. Metode yang digunakan pada proses klasifikasinya adalah *decision tree* menggunakan algoritma PUBLIC. Tahapan prosesnya adalah sebagai berikut:

1. Tahap pembentukan aturan klasifikasi tipe rumah pada data *training*.
 - Data *training* yang sebelumnya telah disimpan dalam *database*, akan diklasifikasikan dengan membentuk pohon keputusan (*decision tree*) berdasarkan atribut-atribut (seperti uang muka, jangka waktu angsuran, besar angsuran, luas bangunan, luas tanah, dan harga tunai) produk rumah, dengan tipe rumah sebagai kelasnya.
 - Pembentukan *decision tree* dimulai dengan mengutamakan atribut numerik.
 - Dilakukan perhitungan nilai *entropy* atribut dengan Persamaan 2.1.
Bila nilai *entropy* atribut sama dengan null, maka artinya atribut tersebut tidak dapat diperluas lagi dan dilanjutkan pada atribut selanjutnya.
Namun bila nilai *entropy* atribut lebih lebih besar dari null, maka dilakukan penentuan *cutpoint split* pada atribut terpilih, yang akan dijelaskan lebih detail pada tahap 2.
 - Kemudian dilakukan perhitungan *cost* dan pemangkasan *node*, yang akan dijelaskan lebih detail pada tahap 3.
Sistem akan berjalan secara rekursif sampai seluruh atribut telah diseleksi.
 - Setelah proses klasifikasi selesai, sistem akan mengekstraknya menjadi suatu aturan klasifikasi dan menyimpannya dalam *database*.

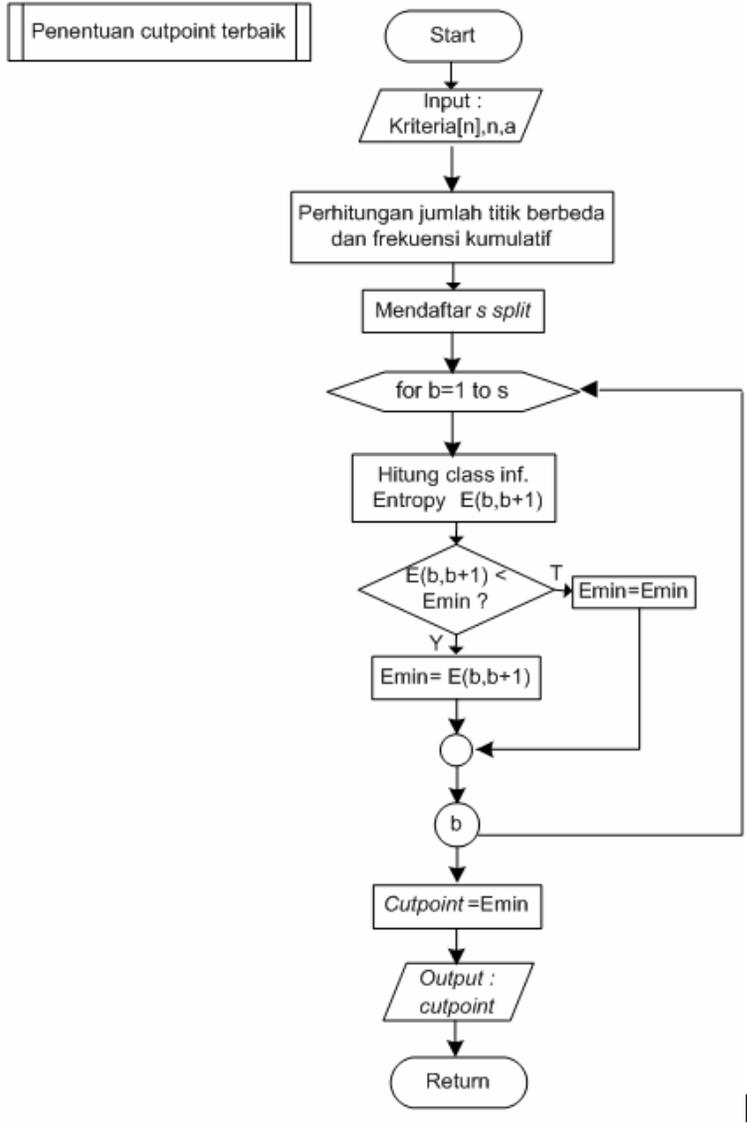
Untuk lebih jelasnya perhatikan Gambar 3.3 *flowchart* pembentukan aturan klasifikasi tipe rumah pada data *training*.



Gambar 3.3 Flowchart pembentukan aturan klasifikasi tipe rumah

2. Tahap penentuan *cutpoint split*
 - Pada atribut terpilih, dilakukan perhitungan jumlah *s* titik berbeda dan jumlah frekuensi kumulatif pada tiap titiknya.
 - Didapatkan *s* titik berbeda.
 - Untuk setiap *s* titik, dilakukan perhitungan *class information entropy* dengan Persamaan 2.2.
 - Dari semua *class information entropy* *s* titik yang ada, nilai *class information entropy* terkecil yang akan dipilih sebagai *cutpoint splitting*.

- *Cutpoint* terpilih akan membagi *node* N menjadi 2 *node* anak, *node* N_1 dan N_2 .
- Untuk lebih jelasnya perhatikan Gambar 3.4 *flowchart* prosedur penentuan *cutpoint splitting*.



Gambar 3.4 *Flowchart* prosedur penentuan *cutpoint* terbaik

3. Tahap perhitungan *cost* dan pemangkasan *node*

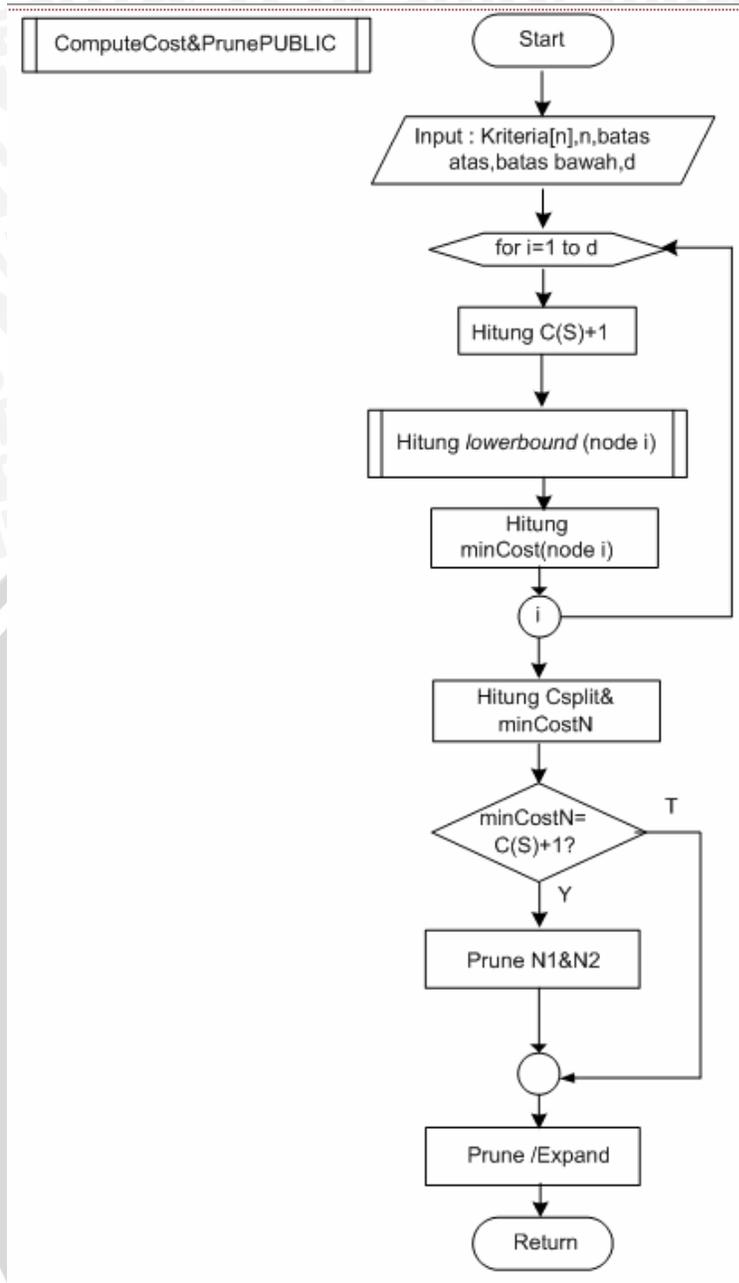
- Dilakukan perhitungan nilai *cost* $C(S)+1$ pada *node* N dengan Persamaan 2.7.
- Dilakukan perhitungan *lowerbound*, yang akan dijelaskan lebih detail pada Tahap 4.
- Secara rekursif, untuk setiap *node* anak N_1 dan *node* anak N_2 , dilakukan perhitungan *cost* $C(S)+1$ dan perhitungan *lowerbound*.
- Dilakukan perhitungan *cost* pengkodean *split* ($C_{split}(N)$)
 1. Untuk atribut kategoris, perhitungannya seperti pada Persamaan 2.5
 2. Untuk atribut numerik, perhitungannya seperti pada Persamaan 2.6
- Dilakukan perhitungan *cost minimum* ($MinCostN$) dengan cara membandingkan *cost* pengkodean langsung dengan *cost* pengkodean *subtree* ditambah *cost* pengkodean *node* anak seperti tampak pada Gambar 2.4
- Bila *cost minimum* yang didapat nilainya sama dengan *cost* pengkodean langsung $C(S)+1$ *node* N , maka *node* anak N_1 dan N_2 dipangkas dari *tree*.
Sistem akan menghapus *node* anak N_1 dan N_2 dari *queue* Q .
- Bila nilainya tidak sama dengan *cost* pengkodean langsung $C(S)+1$ *node* N , maka *node* N , *node* anak N_1 dan N_2 akan diperluas.

Untuk lebih jelasnya perhatikan Gambar 3.5 *flowchart* prosedur *ComputeCost&PrunePUBLIC*.

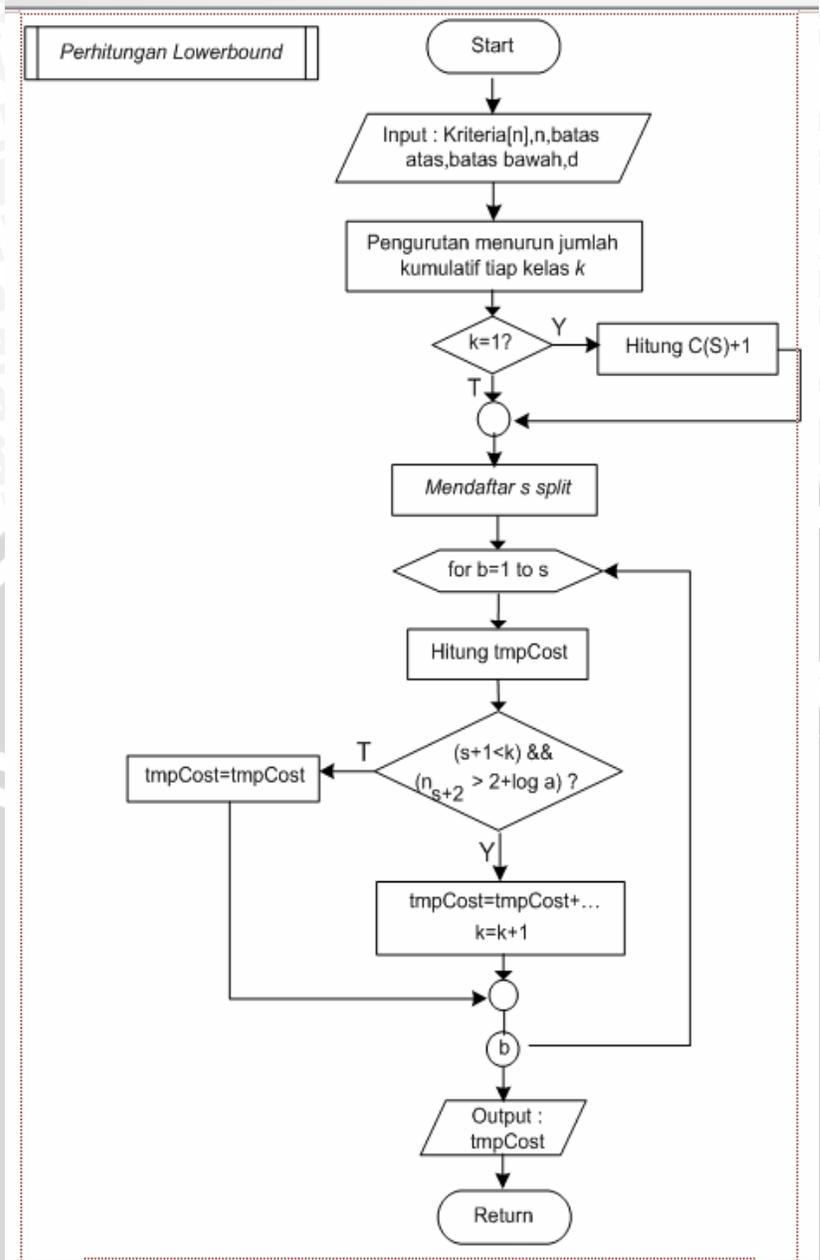
4. Tahap perhitungan *cost lowerbound*

- Sistem akan mengurutkan secara menurun frekuensi relatif dari *record* pada k kelas yang berbeda.
- Bila $k=1$, dilakukan perhitungan *cost*.
- Untuk s *split*, sistem akan menghitung *tmpCost* dengan Persamaan 2.8 Sistem akan mengembalikan nilai minimum dari perbandingan antara *cost* pengkodean data *record*+1 dengan *tmpCost*.

Untuk lebih jelasnya perhatikan Gambar 3.6 *flowchart* prosedur perhitungan *cost lowerbound*.



Gambar 3.5 Flowchart prosedur ComputeCost&PrunePUBLIC



Gambar 3.6 Flowchart prosedur perhitungan Lowerbound

3.3.1.2 Proses Rekomendasi Produk

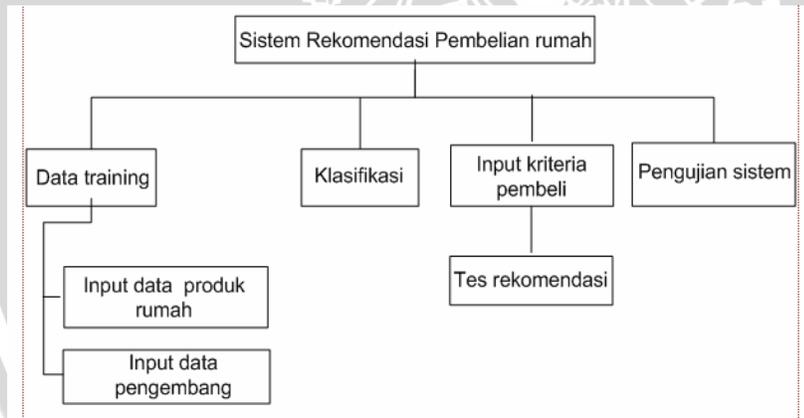
Proses rekomendasi produk dilakukan dengan mencari tipe rumah yang paling mendekati semua kriteria yang diinputkan pembeli. Tahapan-tahapan prosesnya adalah sebagai berikut:

1. Pembeli menginputkan data kriteria berupa No.KTP, *budget* yang dimiliki, jenis pembayaran (tunai/kredit), harga rumah yang diinginkan (harga tunai, uang muka, harga angsuran), jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja pembeli serta jumlah output rekomendasi yang diinginkan.
2. Sistem akan melakukan proses pencarian produk rekomendasi berdasarkan kriteria yang diinputkan dengan membaca aturan klasifikasi yang telah terbentuk dalam *database*.
3. Setelah didapatkan n item yang mendekati kriteria pembeli, produk rumah hasil rekomendasi diurutkan berdasar jarak terpendek dari lokasi kerja dengan lokasi perumahan berada.
4. Hasil akhir berupa c item produk rekomendasi dengan jarak terkecil.

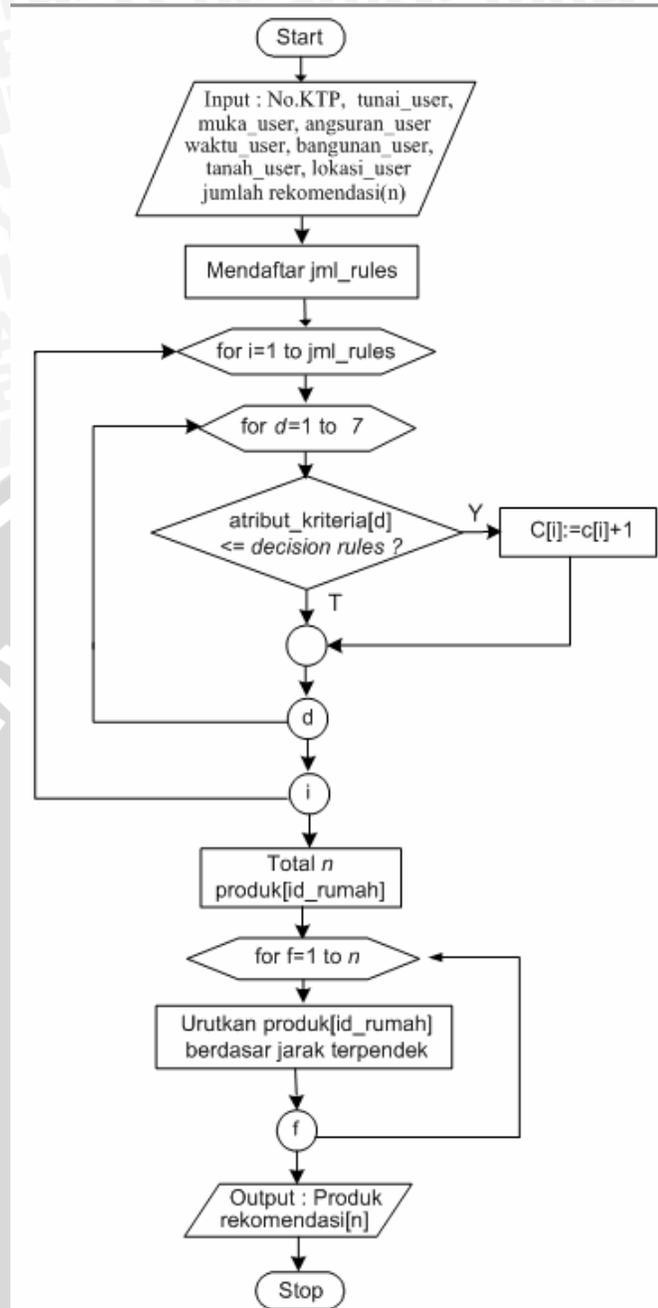
Untuk lebih jelasnya perhatikan Gambar 3.7 *flowchart* prosedur proses rekomendasi produk.

3.3.2 Perancangan antarmuka

Berdasarkan analisis yang telah dilakukan, diagram aplikasi sistem yang akan dibangun ditunjukkan pada Gambar 3.8



Gambar 3.8 Diagram aplikasi sistem



Gambar 3.7 Flowchart prosedur proses rekomendasi produk

Aplikasi sistem akan terdiri dari empat menu utama, diantaranya menu data *training*, data kriteria, tes rekomendasi, dan menu pengujian sistem. Berikut merupakan penjelasan dari masing-masing menu utama tersebut:

- **Data *Training***
Menu ini dirancang untuk memproses seluruh input data baru baik data pengembang maupun data produk rumah, serta data produk yang telah diupdate.
- **Klasifikasi**
Menu ini dirancang untuk melakukan proses klasifikasi data.
- **Tes Rekomendasi**
Menu ini dirancang untuk memberikan rekomendasi produk rumah berdasarkan data kriteria yang diinputkan oleh pembeli. Data inputan tersebut merupakan data kriteria seperti No.KTP, harga rumah yang diinginkan (harga tunai, uang muka, harga angsuran), jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja.
- **Pengujian Sistem**
Menu ini dirancang mengukur tingkat kesalahan atau prosentase *error* dan mengukur waktu yang diperlukan sistem dalam memberikan rekomendasi pada sejumlah data *test* dengan jumlah data *training* yang berbeda-beda.

3.4 Perancangan Uji Coba

Pada subbab ini akan dijelaskan mengenai data yang digunakan dalam penelitian, dan skenario pengujian datanya.

3.4.1 Data yang digunakan dalam penelitian

Data yang digunakan dalam penelitian ini adalah data produk rumah dan data kriteria pembeli. Data produk rumah digunakan sebagai kelas dalam proses klasifikasi, sedangkan data kriteria pembeli digunakan sebagai acuan dalam proses rekomendasinya. Proses pengolahan datanya telah dijelaskan pada subbab-subbab sebelumnya.

3.4.1.1 Data produk rumah

Data produk rumah yang akan dijadikan sampel dalam penelitian ini adalah data produk yang dihasilkan para pengembang perumahan

wilayah kota/kabupaten Malang periode bulan Maret 2008. Karena jumlah produk yang dihasilkan tiap pengembang berbeda-beda, maka jumlah *record* yang dijadikan data sampel dihitung dari jumlah total *record* yang tersimpan dalam *database*, yaitu 300 data *record*.

3.4.1.2 Data kriteria pembeli

Dalam penelitian ini data kriteria pembeli yang akan digunakan diantaranya No.KTP, harga rumah yang diinginkan (harga tunai, uang muka, harga angsuran), jenis pembayaran, jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja. Tabel 3.6 merupakan daftar tipe atribut dan kategori yang digunakan dalam penelitian pada masing-masing kriteria pembeli tersebut.

Tabel 3.6 Tipe atribut dan kategori pada kriteria pembeli

No	Daftar Atribut	Tipe Atribut	Kategori
1	No.KTP	Kontinyu	Tak hingga
2	Harga tunai	Kontinyu	Tak hingga
3	uang muka	Kontinyu	Tak hingga
4	Jenis pembayaran	Diskrit	Tunai Kredit
5	Harga angsuran	Kontinyu	Tak hingga
6	Jangka waktu angsuran	Diskrit	5 tahun 10 tahun 15 tahun
7	luas bangunan	Kontinyu	Tak hingga
8	luas tanah	Kontinyu	Tak hingga
9	lokasi_kerja	Kontinyu	Tak hingga

3.4.2 Pengujian sistem rekomendasi

Proses pengujian ini akan terdiri dari dua pengujian, yaitu pengujian tingkat kesalahan dan pengujian waktu rekomendasi sistem.

3.4.2.1 Pengujian tingkat kesalahan

Pengukuran tingkat kesalahan dilakukan untuk memperoleh data mengenai ukuran ketepatan sistem rekomendasi dalam memberikan rekomendasi produk rumah kepada pembeli.

Sebuah rekomendasi dianggap gagal atau salah, jika produk rumah yang direkomendasikan tidak sesuai dengan kriteria yang diinputkan.

Tingkat kesalahan yang diukur dinyatakan dalam persen (%). Perhitungannya dapat dinyatakan dengan Rumus 3.1

$$error = \frac{\text{jumlah_kesalahan}}{\text{jumlah_data_tes}} \times 100\% \quad (3.1)$$

Pengukuran kesalahan ini menggunakan data sampel produk rumah yang telah ditetapkan sebelumnya. Proses pengujian ini akan dilakukan dengan asumsi sebanyak n kali dengan jumlah data *training* yang telah ditentukan. Hasil perhitungan jumlah data *training* dapat dilihat pada Tabel 3.7

Tabel 3.7 Tabel Jumlah Data *Training*

No.	Data Training ke-	Jumlah Transaksi
1	datatraining[1]	50
2	datatraining[2]	100
3	datatraining[3]	150
4	datatraining[4]	200
5	datatraining[5]	250
6	datatraining[6]	300

Tabel 3.8 adalah rancangan tabel yang akan digunakan untuk mencatat hasil dalam proses pengujian tingkat kesalahan:

Tabel 3.8 Rancangan Tabel Catatan Tingkat Kesalahan

No.	Banyak Data <i>Training</i>	Banyak Data Tes	Jumlah Benar	Jumlah Salah	Error (%)

Keterangan:

Banyak Data *training* : Jumlah data produk dikali jumlah kriteria rumah yang diseleksi.

Banyak Data	: Jumlah n produk rumah keinginan pembeli
Tes	dikali jumlah kriteria rumah yang diseleksi.
Jumlah Benar	: Jumlah sampel yang sesuai dengan rekomendasi yang diberikan
Jumlah Salah	: Jumlah sampel yang tidak sesuai dengan rekomendasi.

3.4.2.2 Pengujian waktu rekomendasi sistem

Pengujian waktu dilakukan untuk mendapatkan catatan waktu yang dibutuhkan untuk menghasilkan sebuah rekomendasi produk rumah. Pengujian dilakukan beberapa kali pada sampel yang berbeda untuk memperoleh perbandingan antara waktu yang dibutuhkan dengan jumlah produk rumah yang dijadikan data *training*.

Dan Tabel 3.9 adalah tabel yang digunakan untuk pengukuran waktu yang diperlukan sistem dalam memberikan rekomendasi pada sejumlah data *test* dengan jumlah data *training* yang berbeda-beda.

Tabel 3.9 Pengukuran waktu rekomendasi sistem

Banyak Data <i>Training</i>	Banyak Data Tes	Waktu

Keterangan:

Banyak Data <i>Training</i>	: Jumlah produk rumah yang dijadikan data pembelajaran.
Banyak Data Tes	: Jumlah sampel yang diujikan
Waktu	: Waktu yang diperlukan sistem dalam proses rekomendasi

3.5 Contoh Perhitungan Manual

Tabel 3.10 merupakan contoh tabel data sampel dan langkah-langkah perhitungan yang akan diproses dalam sistem.

Dari tabel 3.10 langkah awalnya adalah membuat *decision tree* dengan algoritma PUBLIC untuk mengklasifikasikan data produk rumah berdasarkan tipe rumahnya. Kemudian setelah aturan klasifikasi tipe rumah terbentuk, maka langkah selanjutnya adalah merekomendasikan produk menggunakan aturan klasifikasi yang terbentuk.

Tabel 3.10 Contoh data sampel

no	id_rumah	u.muka	Angsuran			luas bangunan	luas tanah	h.tunai	Lokasi	type Rumah
			5th	10th	15th					
1	VBT01	8	1.996	1.211	1.002	36	90	103.8	V.bukittidar	30
2	VBT02	9.8	2.274	1.445	1.195	40	104	123.8	V.bukittidar	40
3	VBT04	13.6	3.329	2.058	1.702	50	126	176.3	V.bukittidar	50
4	VBT05	19.6	4.57	2.856	2.331	70	162	255.3	V.bukittidar	70
5	GBI01	43	3.472	2.123	1.703	55	105	203	ikan belida	50
6	GBI02	36	3.167	1.979	1.615	47	98	177.35	ikan belida	40
7	GBI03	30	2.594	1.621	1.323	36	84	145.8	ikan belida	30
8	PCP01	11.3	2.211	1.362	1.099	40	70	131.6	may.sungkono	40
9	PCP04	29.9	3.766	2.326	1.877	60	112	224.8	may.sungkono	60
10	PCP05	34.9	4.196	2.585	2.086	70	128	249.8	may.sungkono	70
11	PCP06	40.9	4.7	2.896	2.337	70	133	279.8	may.sungkono	70
12	MEA01	34.9	2.9	1.816	1.468	40	79	183.7	me-bandulan	40
13	MEA05	59.3	5.001	3.088	2.497	65	143	312.4	me-bandulan	60
14	MEB01	45.7	3.848	2.376	1.922	50	102	240.4	me-bandulan	50
15	MEB02	45.6	3.839	2.371	1.917	50	103	239.8	me-bandulan	50
16	MEB03	47.2	3.89	2.458	1.987	50	109	248.6	me-bandulan	50
17	MEB05	48.1	4.05	2.501	2.023	50	113	253	me-bandulan	50
18	MEB08	48.6	4.094	2.528	2.045	50	116	255.75	me-bandulan	50
19	MEC03	58.1	4.895	3.023	2.444	65	134	305.8	me-bandulan	60
20	MEC02	59.9	5.054	3.121	2.524	65	142	315.7	me-bandulan	60
21	MED02	36.9	3.109	1.92	1.553	40	90	194.3	me-bandulan	40
22	MED03	57.3	4.834	2.985	2.414	65	134	301.9	me-bandulan	60
23	MEE07	61.1	5.151	3.181	2.572	65	159	321.8	me-bandulan	60
24	MEE09	36.4	3.073	1.897	1.534	40	90	191.95	me-bandulan	40
25	MEF05	52.7	4.448	2.747	2.221	50	151	277.86	me-bandulan	50
26	MEG09	53.8	4.536	2.801	2.265	65	118	283.3	me-bandulan	60

Dijelaskan dalam langkah-langkah berikut ini:

Proses pembentukan *decision tree* dengan algoritma *PUBLIC*

Langkah 1. Menghitung nilai *entropy* dari atribut yang dipilih menjadi *node* akar.

Berdasarkan Tabel 3.10, atribut yang dipilih menjadi *node* akar ke-1 adalah atribut uang muka. maka nilai *entropy* uang muka dapat dihitung dengan menggunakan Persamaan 2.1

$$\begin{aligned}
 E(S) &= E(\text{uang muka}) \\
 &= -\frac{2}{26} \log_2 \frac{2}{26} - \frac{6}{26} \log_2 \frac{6}{26} - \frac{8}{26} \log_2 \frac{6}{26} - \frac{7}{26} \log_2 \frac{7}{26} - \frac{3}{26} \log_2 \frac{3}{26} \\
 &= 2,1652
 \end{aligned}$$

$E(\text{harga_tunai}) = 2,1652 \rightarrow$ menunjukkan bahwa *node* tidak homogen

Langkah 2. Membuat Tabel distribusi kelas dengan mengurutkan nilai atribut yang diuraikan pada Tabel 3.11

Tabel 3.11 Tabel Distribusi Kelas

u.muka	Type rumah
8	30
9.8	40
11.3	40
13.6	50
19.6	70
29.9	60
30	30
34.9	70
34.9	40
36	40
36.4	40
36.9	40
40.9	70
43	50
45.6	50
45.7	50
47.2	50
48.1	50
48.6	50
52.7	50
53.8	60
57.3	60
58.1	60
59.3	60
59.9	60
61.1	60



Langkah 3. Menghitung *class information entropy*

Perhitungan *class information entropy* digunakan untuk mencari nilai *cutpoint* terbaik. Rumus perhitungan yang digunakan adalah Persamaan 2.2.

Dengan menggunakan rumus tersebut dapat diketahui nilai *class information entropy* dari masing-masing *split point*nya. Berikut ini merupakan contoh perhitungan titik *split* pada atribut uang muka

$$\begin{aligned}
 E(8, 9,8) &= \frac{1}{26}I(1,0,0,0,0) + \frac{25}{26}I(1,6,8,7,3) \\
 &= 0 + \frac{25}{26} \left(-\frac{1}{25} \log\left(\frac{1}{25}\right) - \frac{6}{25} \log\left(\frac{6}{25}\right) - \frac{8}{25} \log\left(\frac{8}{25}\right) - \frac{7}{25} \log\left(\frac{7}{25}\right) - \frac{3}{25} \log\left(\frac{3}{25}\right) \right) \\
 &= 2,00693
 \end{aligned}$$

Dengan menggunakan cara yang sama, maka tiap-tiap *split point* dapat dihitung nilai *class information entropy*nya.

Histogram distribusi kelas kemudian disusun seperti Tabel 3.12 untuk mempermudah perhitungan *class information entropy*nya.

Tabel 3.12 Perhitungan *class information entropy* uang muka

Cutpoint	Tipe rumah								<=	>	Esplit
	30	40	50	60	70						
8	<= 1	0	0	0	0	1					
	> 1	6	8	7	3	25	26	0	2.006933	2.00693	
9.8	<= 1	1	0	0	0	2					
	> 1	5	8	7	3	24	26	0.07692	1.9239649	2.00089	
11.3	<= 1	2	0	0	0	3					
	> 1	4	8	7	3	23	26	0.10596	1.8321367	1.93809	
13.6	<= 1	2	1	0	0	4					
	> 1	4	7	7	3	22	26	0.23077	1.77114	2.00191	
19.6	<= 1	2	1	0	1	5					
	> 1	4	7	7	2	21	26	0.3696	1.651373	2.02097	
29.9	<= 1	2	1	1	1	6					
	> 1	4	7	6	2	20	26	0.51961	1.5875886	2.1072	
30	<= 2	2	1	1	1	7					
	> 0	4	7	6	2	19	26	0.60198	1.3672832	1.96926	
34.9	<= 2	3	1	1	2	9					
	> 0	3	7	6	1	17	26	0.76056	1.1373354	1.89789	
36	<= 2	4	1	1	2	10					
	> 0	2	7	6	1	16	26	0.81613	1.0322593	1.84839	
36.4	<= 2	5	1	1	2	11					
	> 0	1	7	6	1	15	26	0.86323	0.9016192	1.76485	

36.9	<=	2	6	1	1	2	12					
	>	0	0	7	6	1	14	26	0.90422	0.6977581	1.60198	
40.9	<=	2	6	1	1	3	13					
	>	0	0	7	6	0	13	26	0.99389	0.4978637	1.49175	
43	<=	2	6	2	1	3	14					
	>	0	0	6	6	0	12	26	1.11686	0.4615385	1.5784	
45.6	<=	2	6	3	1	3	15					
	>	0	0	5	6	0	11	26	1.21476	0.4205512	1.63531	
45.7	<=	2	6	4	1	3	16					
	>	0	0	4	6	0	10	26	1.29751	0.3734425	1.67096	
47.2	<=	2	6	5	1	3	17					
	>	0	0	3	6	0	9	26	1.36971	0.3178716	1.68759	
48.1	<=	2	6	6	1	3	18					
	>	0	0	2	6	0	8	26	1.43401	0.249624	1.68363	
48.6	<=	2	6	7	1	3	19					
	>	0	0	1	6	0	7	26	1.4921	0.1592965	1.65139	
52.7	<=	2	6	8	1	3	20					
	>	0	0	0	6	0	6	26	1.54515	0	1.54515	
53.8	<=	2	6	8	2	3	21					
	>	0	0	0	5	0	5	26	1.69131	0	1.69131	
57.3	<=	2	6	8	3	3	22					
	>	0	0	0	4	0	4	26	1.81107	0	1.81107	
58.1	<=	2	6	8	4	3	23					
	>	0	0	0	3	0	3	26	1.91451	0	1.91451	
59.3	<=	2	6	8	5	3	24					
	>	0	0	0	2	0	2	26	2.00634	0	2.00634	
59.9	<=	2	6	8	6	3	25					
	>	0	0	0	1	0	1	26	2.08931	0	2.08931	
61.1	<=	2	6	8	7	3	26					
	>	0	0	0	0	0	0	26	2.1652	0	2.1652	

Cutpoint (titik percabangan) yang terbaik adalah yang memberikan *class information entropy* yang paling kecil diantara semua kandidat *cutpoint*. Diantara semua kandidat *split point* diatas, nilai *class information entropy* yang paling kecil adalah *split point* 40,9 dengan nilai $Ent(40,9, 43) = 1,492$. Dengan demikian *cutpoint* pada atribut uang muka terletak di 40,9.

Langkah 4. Menghitung Biaya dan Pemangkasan *Node* (*ComputeCost&PrunePUBLIC*)

Setelah diketahui nilai titik percabangan, maka *node* N beserta anak-anak *node* N_1 dan N_2 disimpan dalam *queue* Q.

Kemudian *node* N diseleksi apakah perlu dipangkas atau diperluas, dengan cara:

- a. Menghitung biaya $C(S)$ dari *node* N.
Rumus perhitungannya dinyatakan dengan Persamaan 2.7.
Berikut ini merupakan contoh perhitungan *cost node* N pada atribut uang muka:

$$\begin{aligned}
 C(S) &= \log_2 \binom{26+5-1}{5-1} + \log_2 \frac{26!}{8!7!6!3!2!} \\
 &= 14,744 + 47,711 = 62,455
 \end{aligned}$$

- b. Menghitung *lowerbound node* daun
 - Pertama-tama yang harus dilakukan, adalah mengurutkan jumlah frekuensi relatif dari tiap kelas
Pada atribut uang muka, pengurutan jumlah *record* untuk tiap kelas adalah sebagai berikut: 5, 4, 3, 2, 2
 - Untuk $s = 1$, dihitung *tmpCost*nya dengan persamaan 2.8
tmpCost untuk *node* uang muka adalah:

$$\begin{aligned}
 tmpCost &= 2*1+1+1*\log 7 + \sum_3^5 n_i \\
 &= 2+1+2,322+11 = 16,322
 \end{aligned}$$

```

while s+1 < k and ns+2 > 2+log a do
{
    tmpCost = tmpCost + 2 + log a - ns+2;
    s++;
}
    
```

- Untuk $s=2$
while $2 < 5$ and $7 > 4,322$ do
{ $tmpCost = 16,322+2+2,322-7=13,644$ }
- Pada prosedur *lower bound* ini dicari nilai minimum dari $\{C(S)+1, tmpCost\}$, dan pada *node* uang muka didapatkan nilai minimum $tmpCost = 13,644$

- c. Secara rekursif, sistem menghitung *ComputeCost&Prune* PUBLIC dari *node* anak N_1 dan N_2 .

- *Cost* $C(S)$ dari *node* N_1

$$\begin{aligned}
 C(S) &= \log_2 \binom{13+5-1}{5-1} + \log_2 \frac{13!}{6!3!2!1!1!} \\
 &= \log_2 \binom{17}{4} + 19.461 = 30,679
 \end{aligned}$$

- *Cost lowerbound* untuk *node* daun
Urutan frekuensi relatif *node* N_1 : 5, 2, 1

$$s = 1$$

$$\begin{aligned} \text{tmpCost} &= 2 * 1 + 1 + 1 * \log 5 + \sum_3^5 n_i \\ &= 2 + 1 + 2,322 + 4 = 9,322 \end{aligned}$$

$$\begin{aligned} \text{nilai minimum dari } \{C(S)+1, \text{tmpCost}\} &= \min(31,679, 9,322) \\ &= 9,322 \end{aligned}$$

- Cost C(S) dari node N₂

$$\begin{aligned} C(S) &= \log_2 \left(\frac{13+2-1}{2-1} \right) + \log_2 \frac{13!}{7!6!} \\ &= \log_2 \left(\frac{14}{1} \right) + \log_2 1716 \\ &= 14,554 \end{aligned}$$

- Cost lower bound untuk node daun
Urutan frekuensi relatif node N₂: 3, 3, 2
s = 1

$$\begin{aligned} \text{tmpCost} &= 2 * 1 + 1 + 1 * \log 5 + \sum_3^2 n_i \\ &= 2 + 1 + 2,322 = 5,322 \end{aligned}$$

$$\begin{aligned} \text{nilai minimum dari } \{C(S)+1, \text{tmpCost}\} &= \{15,554, 5,322\} \\ &= 5,322 \end{aligned}$$

d. Menghitung $C_{split}(N)$

1. Untuk atribut kategoris dengan Persamaan 2.5
2. Untuk atribut numerik dengan Persamaan 2.6

Pada atribut uang muka yang bertipe numerik,

$$\begin{aligned} \text{Nilai } C_{split}(N) &= \log 5 + \log 25 \\ &= 2,322 + 4,644 = 6,966 \end{aligned}$$

e. Menghitung $\min Cost_N$

$$\begin{aligned} \min Cost_N &= \{C(S)+1, C_{split}(N) + 1 + \min Cost_1 + \min Cost_2\} \\ &= \{63,455, 6,966+1+5,322+9,322\} \\ &= 22,61 \end{aligned}$$

- f. Bila nilai $\min Cost_N = C(S)+1$, maka node N, node anak N₁ dan N₂ dipangkas
- g. Untuk atribut uang muka, nilai $\min Cost_N$ tidak sama dengan C(S)+1, sehingga anak N₁ dan N₂ diperlukan

Proses perhitungan dilanjutkan pada atribut selanjutnya. Langkah-langkah perhitungannya sama, dan hal tersebut dilakukan sampai semua atribut selesai diproses.

Untuk atribut angsuran, sebelum diproses root dipecah menjadi 3 bagian, jangka waktu 5 tahun, 10 tahun, dan 15 tahun. Gambar 3.9 adalah hasil akhir *decision tree* klasifikasi tipe rumah dari 26 data *record*.

Proses ekstraksi *decision tree* ke dalam *rule*

Setelah pohon keputusan terbentuk maka langkah selanjutnya adalah mengekstrak pohon ke dalam aturan klasifikasi. Maka berdasarkan *decision tree* tersebut aturan klasifikasi yang terbentuk adalah sebagai berikut:

1. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran ≤ 3.167 **AND** luas bangunan ≤ 36 **THEN** produk rumah : VBT01,GBI03
2. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran ≤ 3.167 **AND** luas bangunan > 36 **THEN** produk rumah : PCP01, VBT02, MEA01, MEE09, MED02, GBI02.
3. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran > 3.167 **AND** luas bangunan ≤ 60 **THEN** produk rumah : VBT04, PCP04.
4. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran > 3.167 **AND** luas bangunan > 60 **THEN** produk rumah : PCP05, VBT05, PCP06.
5. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran ≤ 1.98 **AND** luas bangunan ≤ 36 **THEN** produk rumah : VBT01,GBI03.
6. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran ≤ 1.98 **AND** luas bangunan > 36 **THEN** produk rumah : PCP01, VBT02, MEA01, MEE09, MED02, GBI02.
7. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran > 1.98 **AND** luas bangunan ≤ 60 **THEN** produk rumah : VBT04, PCP04.
8. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran > 1.98 **AND** luas bangunan > 60 **THEN** produk rumah : PCP05, VBT05, PCP06.

9. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran ≤ 1.615 **AND** luas bangunan ≤ 36 **THEN** produk rumah : VBT01,GBI03.
10. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran ≤ 1.615 **AND** luas bangunan > 36 **THEN** produk rumah : PCP01, VBT02, MEA01, MEE09, MED02, GBI02.
11. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran > 1.615 **AND** luas bangunan ≤ 60 **THEN** produk rumah : VBT04, PCP04.
12. **IF** uang muka $\leq 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran > 1.615 **AND** luas bangunan > 60 **THEN** produk rumah : PCP05, VBT05, PCP06.
13. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran ≤ 4.448 **THEN** produk rumah : GBI01, MEB02, MEB01, MEB03, MEB05, MEB08, MEF05.
14. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=5 tahun **AND** harga angsuran > 4.448 **THEN** produk rumah : MEG09, MED03, MEC03, MEA05, MEC02, MEE07.
15. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran ≤ 2.747 **THEN** produk rumah : GBI01, MEB02, MEB01, MEB03, MEB05, MEB08, MEF05.
16. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=10 tahun **AND** harga angsuran > 2.747 **THEN** produk rumah : MEG09, MED03, MEC03, MEA05, MEC02, MEE07.
17. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran ≤ 2.22 **THEN** produk rumah : GBI01, MEB02, MEB01, MEB03, MEB05, MEB08, MEF05.
18. **IF** uang muka $> 40,9$ juta **AND** waktu angsuran=15 tahun **AND** harga angsuran > 2.22 **THEN** produk rumah : MEG09, MED03, MEC03, MEA05, MEC02, MEE07.

Aturan klasifikasi ini yang akan digunakan untuk merekomendasikan produk rumah tertentu berdasarkan kriteria yang diajukan pembeli.

Proses rekomendasi

Berikut ini merupakan contoh data baru kriteria rumah yang diinginkan pembeli. Berdasarkan data di bawah ini produk apa yang paling sesuai untuk direkomendasikan?

Tabel 3.13 Contoh data yang akan dicari produk rumah yang sesuai

Field	Value
No_KTP	0410960023
tunai_user	200 juta
jenis_bayar	Kredit
muka_user	38 juta
angsuran_user	3.2 juta
waktu_user	5
bangunan_user	50
tanah_user	120
lokasi_user	Jl.Sumbersari
jml_rekomen	5

Penyelesaian :

Untuk mengetahui produk rumah yang sesuai dengan data diatas adalah dengan membaca aturan klasifikasi yang telah terbentuk dari proses klasifikasi tipe rumah.

Data mulai diseleksi dari aturan klasifikasi no 1,

- aturan klasifikasi no 1
apakah uang muka ≤ 40.9 juta?
Karena pada data kriteria, uang muka yang diinginkan < 38 juta, maka rule 1 masih memenuhi,
Apakah lama angsuran yang diinginkan=5 tahun?
rule 1 masih memenuhi,
Apakah harga angsuran yang diinginkan ≤ 3.167 juta?
Karena pada data kriteria, harga angsuran yang diinginkan < 3.2 juta, maka tidak memenuhi aturan 1,
- Pembacaan aturan klasifikasi dilanjutkan pada aturan kedua.
apakah uang muka ≤ 40.9 juta? ya, memenuhi
Apakah lama angsuran yang diinginkan=5 tahun? ya, memenuhi
Apakah harga angsuran yang diinginkan ≤ 3.167 juta? Tidak memenuhi.
Maka pembacaan aturan klasifikasi dilanjutkan pada aturan no.3, begitu seterusnya hingga didapat aturan klasifikasi yang paling sesuai dengan kriteria pembeli.

Dari pembacaan aturan klasifikasi, akhirnya didapatkan aturan yang paling sesuai kriteria pembeli adalah aturan no.3. Produk rumah yang memenuhi kriteria sementara ditunjukkan pada Tabel 3.15

Tabel 3.14 Data Produk hasil rekomendasi sementara

no	id_rumah	u.muka	Angsuran			Luas bangun	luas tanah	h.tunai	lokasi	tipe rumah
			5 th	10 th	15 th					
1	PCP01	11.3	2.211	1.362	1.099	40	70	131.6	M.sungkono	40
2	VBT02	9.8	2.274	1.445	1.195	40	104	123.8	V.bukittidar	40
3	MEA01	34.9	2.9	1.816	1.468	40	79	183.7	bandulan	40
4	MEE09	36.4	3.073	1.897	1.534	40	90	191.95	bandulan	40
5	MED02	36.9	3.109	1.92	1.553	40	90	194.3	bandulan	40
6	GBI02	36	3.167	1.979	1.615	47	98	177.35	ikan belida	40

Kemudian pada data hasil sementara dilakukan proses perhitungan jarak antara lokasi pembeli bekerja dengan lokasi perumahan berada. Proses ini melibatkan *database* lokasi. Contoh isi *database* lokasi ditunjukkan pada Tabel 3.16

Tabel 3.15 Database Lokasi

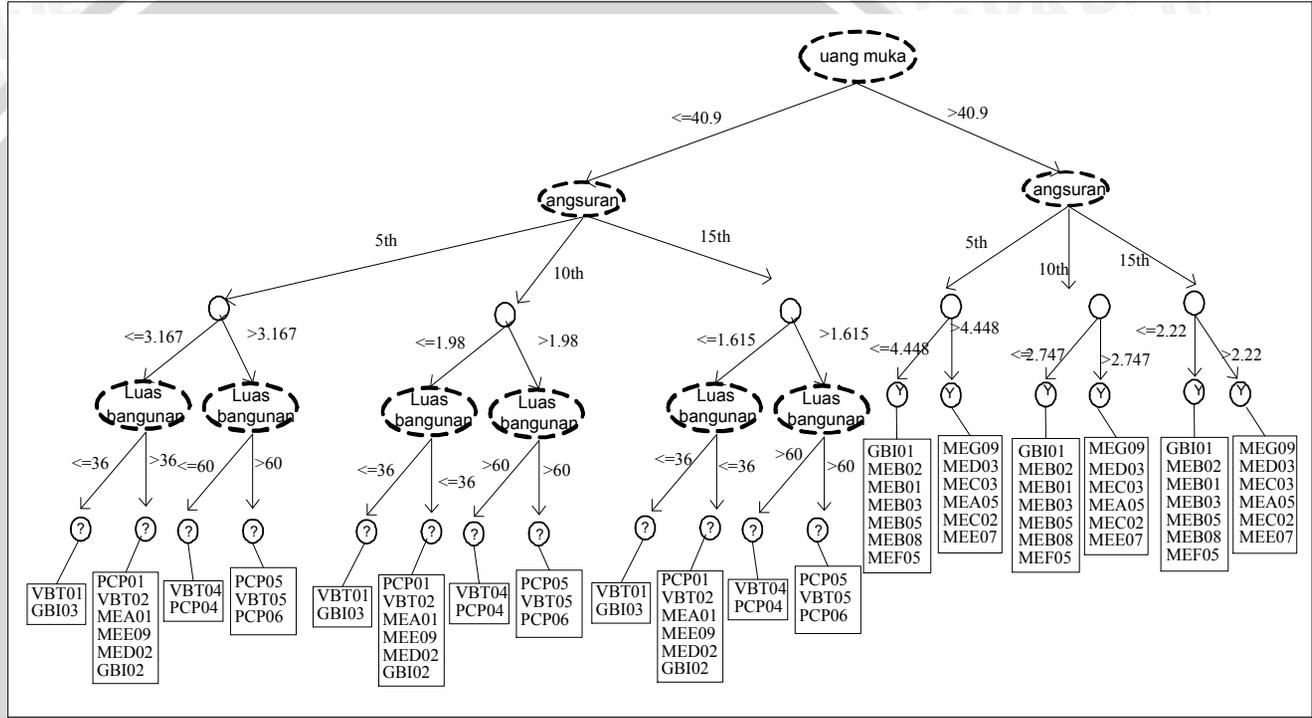
lokasi_kerja	lokasi_rumah	Jarak(km)	Id_pengembang
Sumbersari	V.bukittidar	22	VBT
Sumbersari	ikan belida	20	GBI
Sumbersari	me-bandulan	16	MEA
Sumbersari	me-bandulan	19	MEE
Sumbersari	me-bandulan	15	MED
Sumbersari	may.sungkono	22	PCP

Hasil perhitungan kemudian diurutkan mulai dari jarak terdekat sampai yang terjauh. Perhitungan ini dilakukan untuk mendapatkan hasil rekomendasi yang paling mendekati data kriteria yang diinginkan.

Dari perhitungan diatas didapatkan hasil rekomendasi produk rumah, diurutkan dari produk yang paling mendekati kriteria pembeli seperti tampak pada Tabel 3.17

Tabel 3.16 Hasil Rekomendasi

No	id_rumah	u.muka	Angsuran			luas bangun	luas tanah	h.tunai	lokasi	type rumah
			5 th	10 th	15 th					
1	MED02	36.9	3.109	1.92	1.553	40	90	194.3	me-bandulan	40
2	MEA01	34.9	2.9	1.816	1.468	40	79	183.7	me-bandulan	40
3	MEE09	36.4	3.073	1.897	1.534	40	90	191.95	me-bandulan	40
4	GBI02	36	3.167	1.979	1.615	47	98	177.35	ikan belida	40
5	PCP01	11.3	2.211	1.362	1.099	40	70	131.6	may.sungkono	40
6	VBT02	9.8	2.274	1.445	1.195	40	104	123.8	V.bukittidar	40



Gambar 3.9 Decision tree untuk klasifikasi type rumah

BAB IV HASIL DAN PEMBAHASAN

4.1 Perangkat Sistem

Perangkat sistem terdiri dari perangkat lunak dan perangkat keras yang digunakan dalam penelitian.

4.1.1 Perangkat Lunak

Perangkat Lunak yang digunakan dalam penelitian ini adalah :

1. Sistem operasi Microsoft Windows XP Professional Edition Service Pack 2 sebagai server *database* sekaligus sebagai *client* tempat aplikasi rekomendasi rumah dijalankan.
2. Borland Delphi 7.0 sebagai *software development* dalam pembuatan sistem rekomendasi rumah ini, baik untuk proses input data transaksinya, proses perhitungan rekomendasinya, maupun proses pengujian sistemnya.
3. Microsoft Access 2003 sebagai DBMS (*Database Management System*).

4.1.2 Perangkat Keras

Perangkat keras yang digunakan dalam pengamatan ini adalah sebuah PC (*Personal Computer*) dengan spesifikasi sebagai berikut :

1. Processor Intel Celeron 2,4 GHz
2. Memory 256 MB RAM
3. VGA 64 MB
4. Space Hardisk 40 GB
5. Keyboard
6. Mouse

4.2 Implementasi

4.2.1 Implementasi Basis Data

Pada bab sebelumnya telah dijelaskan rancangan tabel dan relasi basis data yang terdiri dari 5 buah tabel. Tabel-tabel tersebut antara lain *t_rumah*, *t_pengembang*, dan *t_lokasi* digunakan untuk menyimpan data produk, *t_klasifikasi* untuk menyimpan data hasil rekomendasi, dan *t_kriteria* untuk menyimpan data kriteria pembeli.

Sedangkan jenis kolom dan fungsinya sudah dijelaskan lebih detail pada bab III tersebut.

Tabel-tabel tersebut beserta relasinya diimplementasikan dengan menggunakan Microsoft Access 2003.

4.2.2 Struktur Data

Struktur data yang digunakan untuk melakukan proses klasifikasi data pada setiap elemen kriteria rumah adalah *record*. Struktur data tersebut dapat dilihat pada Gambar 4.1.

```
TKriteria = record
    CostNode: array [1..4] of double;
    Noun: array [1..500] of integer;
    Tipe: array [1..500] of integer;
    Kode: array [1..500] of string
end;
const
    Atribut : array [1..7] of string =
('uang_muka', 'angsuran_5th', 'angsuran_10th', 'angsuran_15th',
'luas_bangunan', 'luas_tanah', 'harga_tunai');
var
    Kriteria : array [1..7] of TKriteria;
```

Gambar 4.1 Struktur data *record* tiap kriteria

Dari struktur data pada Gambar 4.1 dapat dijelaskan bahwa sebuah elemen kriteria rumah terdiri dari *CostNode*, *Noun*, *Tipe*, *Kode*. *CostNode* digunakan untuk menyimpan nilai hasil proses perhitungan *cost* pada tiap *node* pada tiap kriteria. *CostNode* adalah *array* dengan elemen bertipe data *double*. *Noun* yang merupakan *array* dengan elemen bertipe data *integer*, digunakan untuk menyimpan nilai dari tiap id rumah yang bersangkutan pada tiap kriteria. Sedangkan *Tipe* digunakan untuk menyimpan tipe rumah dari tiap id rumah yang bersangkutan pada tiap kriteria. *Tipe* juga merupakan *array* dengan elemen bertipe data *integer*. Dan *Kode* yang merupakan *array* dengan elemen bertipe data *string*, digunakan untuk menyimpan nama id rumah pada tiap kriteria.

Urutan dari kriteria yang akan diproses oleh sistem disesuaikan dengan nilai dari *Atribut*, yang merupakan *array* dengan elemen bertipe data *string* sebanyak 7 indeks.

4.2.3 Pembentukan *decision tree*

Sistem rekomendasi dapat dijalankan setelah proses klasifikasi rumah dari semua pengembang dilakukan. Langkah awal untuk proses pembentukan klasifikasi rumah adalah dengan menetapkan data *training* dan tipe rumahnya. Data *training* didapatkan dari data produk rumah para pengembang perumahan, yang kemudian dikelompokkan ke dalam tipe rumah tertentu berdasarkan besarnya luas bangunan. Data produk yang telah memiliki tipe rumah selanjutnya diproses menggunakan metode PUBLIC. Gambar 4.2 adalah contoh penggunaan *query* pada pemanggilan data produk dari *database* sebagai proses awal klasifikasi.

```

.....
.....
for i:=1 to a do
begin
  Adoquery1.SQL.Clear;
  Adoquery1.SQL.Add('SELECT '+Atribut[b]+' ,id_rumah,
    tipe_rumah FROM t_Rumah WHERE t_Rumah.no='+
    inttostr(i)+ ' order by '+Atribut[b]);
  Adoquery1.Active:=true;
  with ADOQUERY1 do
  begin
    Kriteria[b].noun[i]:= strtoint(fieldbyname
      (Atribut[b]).AsString);
    Kriteria[b].kode[i]:= fieldbyname('id_rumah').AsString;
    Kriteria[b].tipe[i]:= strtoint(fieldbyname
      ('tipe_rumah').AsString);
  end;
end;
end;

```

Gambar 4.2 Penggunaan *query* pada pemanggilan data produk dari *database*

Setelah nilai tiap atribut disimpan ke dalam *array*, data diurutkan *ascending*, dari kecil ke besar. Pengurutan ini berguna untuk mencari nilai *cutpoint* pada tiap atribut. Tetapi sebelum *cutpoint* atribut tertentu dihitung, dilakukan proses perhitungan *Entropy* Atribut.

Langkah perhitungan *Entropy* Atribut dilakukan dengan memanggil fungsi `EntAtribut()`. Perhitungan berguna untuk mengetahui apakah data yang akan diproses sejenis atau tidak. Bila data diketahui sejenis atau (`EntAtribut()=0`), maka proses perluasan atribut tidak dilanjutkan. Gambar 4.3 adalah gambaran umum fungsi `EntAtribut()` dalam proses klasifikasi rumah.

```

Function EntAtribut(Kriterias:TKriteria;b:integer; c:integer;
a:integer; d:integer): double;
begin
  //perhitungan jumlah frekuensi kumulatif tiap tipe rumah
  .....
  j:=1;
  while (j<=6) do
  begin
    if (jml_tipe[j]=0) then j:=j+1
    else
    begin
      EntA:=EntA+(-jml_tipe[j]/pjg)*log2(jml_tipe[j]/pjg);
      j:=j+1;
    end;
  end;
  Result := (pjg/d) * entA;
end;

```

Gambar 4.3 Fungsi EntAtribut ()

Bila diketahui nilai $EntAtribut > 0$, maka dilanjutkan pada proses perhitungan *cutpoint*. Proses perhitungan *cutpoint* dimulai dengan pembangunan histogram distribusi kelas. Pembangunan histogram distribusi kelas dilakukan untuk mengetahui jumlah titik potong berbeda pada suatu atribut. Bila jumlah titik potong telah diketahui, proses dilanjutkan pada perhitungan *class information entropy*. Nilai *class information entropy* terkecil yang akan dipilih sebagai *cutpoint splitting*. Gambar 4.4 adalah potongan fungsi EntCutpoint().

```

Function EntCutpoint( Kriterias : TKriteria; b:integer;
a:integer; d:integer):integer;
begin
  //perhitungan jumlah titik berbeda
  .....
  for i:=1 to c do
  begin
    jml_bwh:=0;
    for j:=1 to i do
    begin jml_bwh := jml_bwh+ jml_point[j]; end;
    hsl_bwh[i] :=EntAtribut (Kriteria[b],b,1, jml_bwh, a);
    hsl_atas[i]:=EntAtribut (Kriteria[b],b, jml_bwh+1, a, a);
    total[i] := hsl_bwh[i]+hsl_atas[i];
    if(total[i]< min_Ent) then
    begin
      min_Ent := total[i]; //class information entropy min
      cutpoint:= i;
    end;
  end;
  if (d=4) then Result := c
  else Result := cutpoint;
end;

```

Gambar 4.4 Fungsi EntCutpoint ()

Cutpoint terpilih akan membagi *node* N menjadi 2 *node* anak, *node* N_1 dan *node* N_2 . Namun untuk menentukan *node* N diperluas atau dipangkas, dilakukan perhitungan *cost* terlebih dahulu. Perhitungan *cost* pada *node* N, *node* N_1 dan *node* N_2 dilakukan dengan memanggil fungsi `Cost_Node()`. Pada masing-masing *node* akan dihitung $Cost\ C(S)+1$, *lower bound*, dan $cost\ C_{split}(N)$, yang kemudian dicari nilai minimumnya sebagai *cost* akhir. Bila *cost minimum* yang didapat nilainya sama dengan *cost* $C(S)$ *node* N, maka *node* anak N_1 dan N_2 dipangkas dari *tree*. Gambar 4.5 adalah potongan pemanggilan fungsi `Cost_Node()` dalam prosedur database, sedangkan Gambar 4.6 adalah bentuk umum dari fungsi `Cost_Node()`.

```

procedure TForm_utama.Database;
begin
.....
.....
for d:=1 to 4 do
begin
if (d=1) then Kriteria[b].CostNode[d]:=
Cost_Node(Kriteria[b],b,1,a,d) //cost node N
else if (d=2) then Kriteria[b].CostNode[d]:=
Cost_Node(Kriteria[b],b,1,cutpoint,d) //cost anak N1
else if (d=3) then Kriteria[b].CostNode[d]:=
Cost_Node(Kriteria[b],b,cutpoint+1,a,d) //cost anak N2
else Kriteria[b].CostNode[d]:=log2(7)+
log2(EntCutpoint(Kriteria[b],b,a,d)-1); //cost CSplit(N)
end;
minCostN:=min(Kriteria[b].CostNode[1]+1, Kriteria[b].
CostNode[4]+1+Kriteria[b].CostNode[2]+Kriteria[b].CostNode[3]);
.....
.....
End;

```

Gambar 4.5 Potongan pemanggilan fungsi `Cost_Node()`

```

function Cost_Node(Kriterias:TKriteria; b:integer; x:integer;
y:integer; d:integer): double;
begin
//perhitungan jumlah frekuensi kumulatif tiap tipe rumah
.....
//sorting jml_tipe secara descending
.....
// perhitungan CostS
j:=1; k:=0; y:=0;
while (j<=6) do
begin
if (jml_tipe[j]=0) then k:=k+1
else
begin

```

```

        y:=y+jml_tipe[j];
        jml_fak:=jml_fak*factorial(jml_tipe[j]) ;
    end;
    j:=j+1;
end;
klas:=6-k;        kom2:=1;
for i:=y+1 to (y+klas-1) do
begin
    kom2:= kom2 * i;
end;
kombinasi:= kom2 / factorial(klas-1);
Cost:=log2(kombinasi)+log2(factorial(y)/jml_fak);

// perhitungan Lowerbound
s:=1;        ni:=0;
for i:=s+2 to klas do
begin
    ni := ni+i;
end;
tmpCost := 2*s+1+s*log2(7)+ ni;
while (s+1 < klas) and (jml_tipe[s+2] > 2+log2(7)) do
begin
    tmpCost := tmpCost+2+log2(7)- jml_tipe[s+2];
    s:=s+1;
end;
if (d<>1) then Result := Min(Cost+1,tmpCost)
else Result :=Cost;
end;

```

Gambar 4.6 Fungsi Cost_Node ()

Bila seluruh atribut telah diseleksi atau *node* sudah tidak bisa diperluas lagi, maka langkah selanjutnya adalah mengekstraknya menjadi aturan klasifikasi dan memasukannya ke dalam *database*. Gambar 4.7 adalah contoh penyimpanan *rule* ke dalam *database* pada prosedur Database () .

```

procedure TForm_utama.Database;
begin
while (b<=7) do
begin
if (EntAtribut(Kriteria[b],b,1,a,a)=0) then    p:=7
else
begin
if (minCostN = Kriteria[b].CostNode[1]+1) then .....
else //node diperluas
begin
for cab:=1 to 2 do
begin
.....
.....
harga[cab]:=RpI[h]+(cab*power(10,7-b));
d:=0;

```

```

for j:=1 to jml_final do
begin
.....
.....
if ((harga[cab]=idFinal[j]) or (harga[cab]=0)) then
d:=d+1;
end;
if ((d=0) and (b=7)) then
begin
noFinal:=noFinal+1;
Form_utama.ADOConnection1.Execute('insert into
t_FinalRules values (''+ inttostr(noFinal)+'''+', '''+'''+
floattostr(harga[cab])+'''+', '''+'''+ total_tanda+'''+', '''+
'''+total_cut+'''+', '''+'''+ tview+'''+', '''+'''+ isi+''''+
', '''+inttostr(b)+''''+')');
end; end;
end; end;
if ((b=7) or (p=7)) then
begin
d:=0;
if (jml_final>0)then
begin
for j:=1 to jml_final do
begin
.....
.....
if ((harga[cab]=idFinal[j]) or (harga[cab]=0)) then d:=d+1;
end;
end;
if ((d=0) or (jml_final=0)) then
begin
noFinal:=noFinal+1;
Form_utama.ADOConnection1.Execute('insert into
t_FinalRules values (''+ inttostr(noFinal)+'''+', '''+'''+
floattostr(harga[cab])+'''+', '''+'''+total_tanda+'''+', '''+
'''+total_cut+'''+', '''+'''+tview+'''+', '''+'''+isiI[h]+''''+
', '''+inttostr(b)+''''+')');
end; end;
.....
.....
end; end;

```

Gambar 4.7 Contoh penyimpanan *rule* ke dalam *database* pada prosedur Database ()

4.2.4 Proses rekomendasi

Dalam proses rekomendasi, sistem akan menerima inputan data kriteria pembeli seperti No.KTP, harga rumah yang diinginkan (harga tunai, uang muka, harga angsuran), jangka waktu angsuran, luas bangunan, luas tanah, dan lokasi kerja pembeli. Seperti yang telah dijelaskan pada bab III, dari data kriteria pembeli akan

ditentukan produk rumah yang sesuai berdasarkan aturan klasifikasi yang telah terbentuk oleh *decision tree*. Setelah didapatkan n item yang mendekati kriteria pembeli, produk diurutkan berdasar jarak terpendek dari lokasi kerja dengan lokasi perumahan berada. Gambar 4.8 adalah potongan prosedur yang digunakan dalam proses rekomendasinya.

```
procedure TForm_utama.ButtonRekomendasiClick(Sender:TObject);
begin
    .....
    .....
    // check apakah syarat" pembeli ada di t_Klasifikasi
    if (rekomen='') then //bila tidak ada di t_Klasifikasi
    begin
        // memecah syarat"pembeli per kriteria
        .....
        .....
        //memecah aturan klasifikasi 't_FinalRules' k dalam array
        for j:=1 to jml_final do
        begin
            ADOQuery_pembeli.SQL.Add('SELECT * FROM t_FinalRules
            where t_FinalRules.no='+ inttostr(j));
            with ADOQuery_pembeli do
            begin
                tot_cut[j] :=fieldbyname('total_cut').AsString;
                tot_tanda[j]:=fieldbyname('total_tanda').AsString;
                isiFinal[j] :=fieldbyname('anggota').AsString;
            end;
            //pemecahan cutpoint
            //pemecahan tanda
            .....
            //mengecheck syarat"permbeli pada rules di database
            b:=1; c:=0;
            while (b<=7) do
            begin
                if (part_tanda[b]='<=') then
                begin
                    if (strtoint(nilai[b])<=strtoint(part_cut[b])) then
                    begin c:=c+1; b:=b+1; end
                    else b:=8;
                end
                else if (part_tanda[b]='>') then
                begin
                    if (strtoint(nilai[b])>strtoint(part_cut[b])) then
                    begin c:=c+1; b:=b+1; end
                    else b:=8;
                end
                else if (part_tanda[b]='=') then
                begin
                    if (strtoint(nilai[b])=strtoint(part_cut[b])) then
                    begin c:=c+1; b:=b+1; end
                    else b:=b+1;
                end
            end
        end
    end
end
```

```

end
if (max_c<c) then
begin
max_c:=c;      rekomendasi:=isiFinal[j];
i:=j;
end; end;
end;
//memecah hasil rekomendasi per id_rumah
//sorting brdasarkan selisih terkecil atribut dg kriteria
if (jml_krit>g) then jml_krit:=g;
if (Lokasi<>'') then
begin
// mencari jarak tiap id rumah hasil rekomendasi
// sorting hasil akhir brdasarkan jarak
end;
//hasil rekomendasi terakhir
idresult:=id[1]+'.';
for j:=2 to g do
begin idresult:=idresult+id[j]+'.'; end;
//perhitungan waktu rekomendasi
no:=jml_klasif+1;
Form_utama.ADOConnection1.Execute('insert into
t_klasifikasi values (''+inttostr(no)+'''+', '''+KTP+
'''+'''+syarat+'''+', '''+idresult+'''+', '''+sla
ma+''''+')');
end;
end;

```

Gambar 4.8 Potongan prosedur rekomendasi

4.3 Penerapan Aplikasi

Aplikasi yang telah dibangun digunakan untuk melakukan berbagai proses dalam rekomendasi sistem. Selain itu, aplikasi ini juga digunakan untuk proses pengujian sistem. Berikut ini adalah beberapa tampilan aplikasi pada tiap-tiap menu berdasarkan penjelasan dari subbab 3.3.2 Perancangan antarmuka.

Menu pada Gambar 4.9 digunakan untuk memasukan data baru, baik itu data produk rumah maupun data pengembang ke dalam *database* perumahan. Data produk rumah yang diinputkan berupa id rumah, nama rumah, harga tunai, uang muka, angsuran 5th, angsuran 10th, angsuran 15th, luas bangunan, luas tanah, tipe rumah, lokasi dan id pengembang. Setelah proses input data selesai dilakukan maka sistem akan mengklasifikasikan data *trainingnya* dengan membentuk pohon keputusan (*decision tree*) pada menu klasifikasi.

Input Data Kriteria Pembeli

Nama pembeli :

No KTP :

budget yg dimiliki :

Ulang muka :

Jenis pembayaran :

Besar angsuran per bulan :

Jangka Waktu Angsuran :

Luas bangunan :

Luas tanah :

Lokasi kerja pembeli :

Jumlah rekomendasi :

*) harus diisi

Gambar 4.9 Tampilan menu input produk rumah

Gambar 4.10 merupakan tampilan dari menu klasifikasi data. Pada proses klasifikasinya menu ini juga akan menampilkan *decision tree* dan aturan klasifikasi yang telah terbentuk dalam *database* pada *user*.

Klasifikasi Data

Classification Rule

jumlah record : 100 record

-----> Decision Rules <-----

1. if uang_muka<=43900000 and angsuran_5th>10100000 then PDE.NN4/28.PDE.NN4/24.GHR37.GHR31.PDE.LL2/19.GHR40.PDE.MM1/14.DDR01.INC
2. if uang_muka<=43900000 and angsuran_15th>5100000 then PDE.NN4/28.PDE.NN4/24.DDR01.GHR37.GHR31.PDE.LL2/19.GHR40.PDE.MM1/14.INC
3. if uang_muka<=43900000 and angsuran_5th<=3600000 and luas_bangunan<=37 then GDE02.GDE03.PRe03.SR04.PRe05.PR-T36-09.GSR21.PR-T36-
4. if uang_muka<=43900000 and angsuran_10th<=2300000 and luas_bangunan<=37 then GDE02.GDE03.SR04.PRe03.PRe05.GSR21.GSR27.GSR25.PF
5. if uang_muka<=43900000 and angsuran_10th<=4900000 and luas_bangunan<=60 then GHR19.GHR07.GHR08.GHR13.
6. if uang_muka<=43900000 and angsuran_10th<=4900000 and luas_bangunan<=60 then GHR37.GHR40.GHR31.PDE.NN4/28.PDE.NN4/24.PDE.MM1/14.I
7. if uang_muka<=43900000 and angsuran_15th<=2300000 and luas_bangunan<=37 then GDE02.GDE03.SR04.PRe03.PRe05.GSR21.GSR27.GSR25.GS
8. if uang_muka<=43900000 and angsuran_5th<=3600000 and luas_bangunan<=37 and luas_tanah<=112 then GAMR01.PCP02.MR18.OCM01.MoRe02.Gi
9. if uang_muka<=43900000 and angsuran_5th<=3600000 and luas_bangunan<=37 and luas_tanah<=112 then GriPeln04.OCM02.GARe02.RiverVillage03.Ri
10. if uang_muka<=43900000 and angsuran_5th>3600000 and luas_bangunan<=60 then GHR10.GHR22.GHR08.GHR18.GHR12.GHR21.GHR17.GHR01.

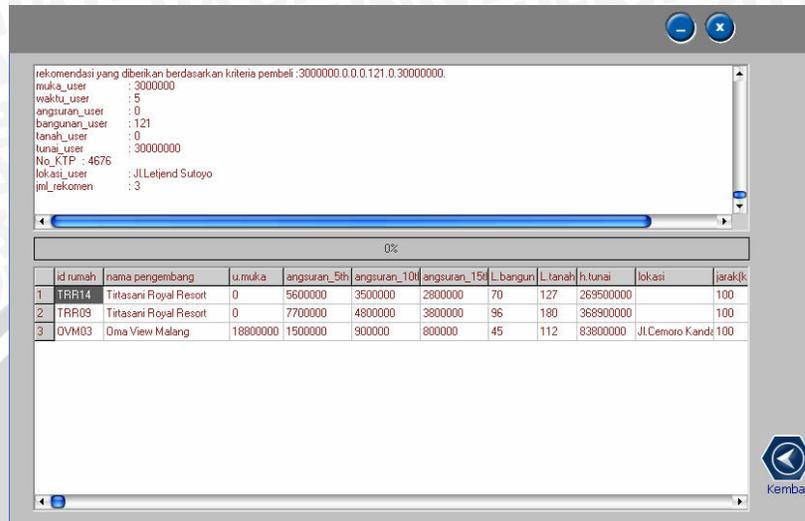
100%

Decision Tree

- uang_muka<=43900000
 - TRR09.TRR14.VBT02.SR04.PCP02.OVM03.OCM01.VBT05.RiveVillage03.PRe03.GDE02.PRe05.RiveVillage04.GriPeln03.PRe04.OCM02.PR
 - angsuran_5th<=3600000
 - GDE02.OVM03.OCM01.GDE03.PRe03.PRe04.SR04.VBT02.PRe05.GriPeln03.PRe06.GAMR01.PCP02.RiverVillage03.OCM02.GriPeln0
 - luas_bangunan<=37
 - GDE02.GDE03.PRe03.SR04.PRe05.PR-T36-09.GSR21.PR-T36-08.PR-T36-05.GSR27.GSR25.GSR17.GSR15.GSR13.GSR11.
 - luas_bangunan<=37
 - angsuran_5th>3600000
 - angsuran_10th<=2300000
 - angsuran_10th>2300000
 - angsuran_15th<=2300000

Gambar 4.10 Tampilan menu klasifikasi data

Untuk menu rekomendasi, akan ditampilkan pada Gambar 4.11. Menu rekomendasi digunakan untuk memberikan rekomendasi produk-produk rumah berdasarkan data kriteria pembeli yang diinputkan *user* ke dalam tabel *t_kriteria*. Untuk lebih memudahkan *user* menu ini juga akan menampilkan informasi berupa id rumah beserta harga untuk masing-masing kriteria.



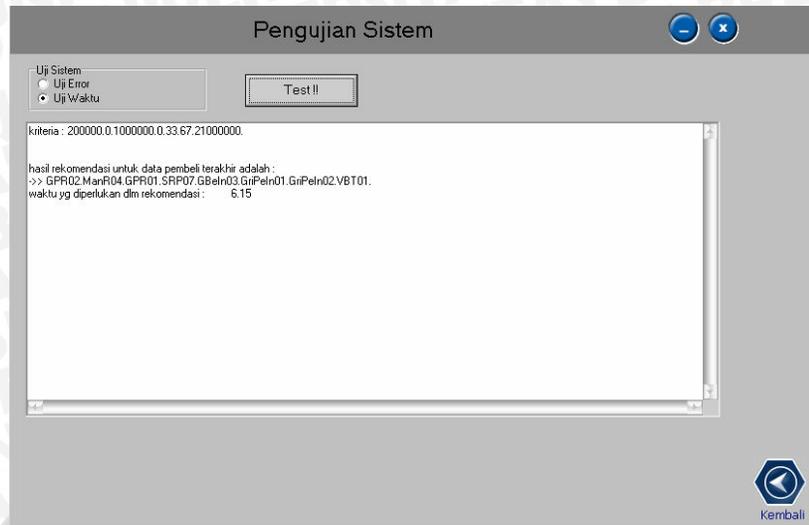
Gambar 4.11 Tampilan menu rekomendasi

Sedangkan Gambar 4.12 merupakan tampilan dari menu pengujian sistem. Untuk melakukan uji coba ini, *user* diharuskan untuk menetapkan kriteria rumah yang diinginkan. Proses ini akan menampilkan informasi jumlah benar, jumlah salah, waktu yang diperlukan sistem dalam proses rekomendasinya dan akan ditampilkan pula prosentase *error* atau tingkat kesalahan yang dinyatakan dalam persen.

4.4 Pengujian sistem rekomendasi

Aplikasi yang telah dibangun juga digunakan untuk menguji tingkat kesalahan sistem dalam memberikan rekomendasi dan mengukur waktu yang diperlukan sistem dalam membangun *tree* berdasarkan sejumlah data *test* dengan jumlah data *training* yang

berbeda-beda. Berikut ini adalah hasil yang diperoleh dari proses pengujian tersebut.



Gambar 4.12 Tampilan menu pengujian sistem

4.4.1 Pengukuran tingkat kesalahan sistem

Tingkat kesalahan diukur dengan cara menghitung jumlah salah pada proses rekomendasi data *test*nya. Jumlah data *test* dihitung dengan cara mengalikan jumlah n hasil kriteria yang diinginkan dengan 7 (sejumlah atribut klasifikasi). Sedangkan kesalahan terjadi apabila hasil rekomendasi dari sistem tidak sesuai dengan nilai kriteria yang diinginkan pembeli, dengan kata lain hasil rekomendasi melampaui batas maksimum dari nilai kriteria yang diinginkan. Misalnya dalam suatu id rumah hasil rekomendasi, nilai harga tunai dan besar angsuran melebihi kriteria pembeli, maka sistem akan menghitungnya sebanyak 2 nilai salah. Jumlah kesalahan yang terjadi dibagi dengan jumlah data sampel untuk mendapatkan *error rate* yang dinyatakan dalam persen.

Pengukuran tingkat kesalahan ini dilakukan sebanyak n kali dengan menggunakan kriteria pengujian yang sama, dan data *training* yang berbeda. Tabel 4.1 merupakan tabel tingkat kesalahan berdasarkan pengujian dengan sejumlah data *training* yang berbeda-beda.

Tabel 4.1 Pengukuran kesalahan rekomendasi sistem

Banyak Data <i>Training</i>	Banyak Data Tes	Jumlah Benar	Jumlah Salah	Error(%)
50	140	10	130	92.86
100	140	76	64	45.71
150	140	101	39	27.86
200	140	101	33	23.57
250	140	101	33	23.57
300	140	101	33	23.57

Keterangan pengujian pada Tabel 4.1:

Banyak Data *training* : Jumlah data produk rumah yang dijadikan data *training* dikali dengan jumlah kriteria rumah yang diseleksi. Misalnya pada pengujian 1, jumlah data produk rumah adalah 50 *record* dan jumlah kriteria adalah 7, maka jumlah data *training* adalah 350.

Banyak Data Tes : Jumlah *n* produk rumah yang diinginkan pembeli dikali jumlah kriteria rumah yang diseleksi. Misalnya pada Tabel 4.1, jumlah *n* produk yang diinginkan pembeli adalah 20 *record* dan jumlah kriteria adalah 7, maka jumlah data tes adalah 140.

Jumlah Benar : Jumlah setiap data tes yang nilainya kurang dari atau sama dengan nilai suatu kriteria yang diinginkan pembeli

Jumlah Salah : Jumlah setiap data tes yang nilainya lebih dari nilai suatu kriteria yang diinginkan pembeli

Error (%) :
$$\frac{\text{jumlah_salah}}{\text{jumlah_data_tes}} \times 100\%$$

4.4.2 Pengukuran waktu rekomendasi sistem

Proses pengukuran waktu rekomendasi dititikberatkan pada saat pengujian data. Pengujian ini akan mengukur waktu yang dibutuhkan dalam memberikan rekomendasi terhadap sejumlah data *test* yang sama dengan jumlah data *training* yang berbeda.

Pengukuran waktu sistem dalam memberikan rekomendasi ini merupakan perbandingan kecepatan proses yang akan diuji terhadap penambahan jumlah aturan klasifikasinya. Pengukuran waktu rekomendasi sistem ini akan ditampilkan saat pengujian tingkat kesalahan selesai dilakukan.

Sama halnya dengan pengujian tingkat kesalahan, proses pengujian waktu rekomendasi ini dilakukan sebanyak n kali dengan menggunakan jumlah data tes yang sama dan jumlah data *training* yang berbeda-beda.

Hasil dari pengukuran waktu yang diperlukan sistem dalam memberikan rekomendasi produk rumah kepada pembeli dapat dilihat pada Tabel 4.2.

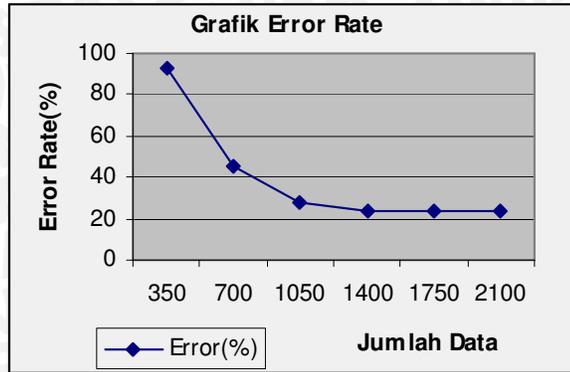
Tabel 4.2 Pengukuran waktu rekomendasi sistem

Banyak Data <i>Training</i>	Banyak Data <i>Test</i>	Waktu (detik)
50	140	1.5
100	140	3.69
150	140	5.86
200	140	4.23
250	140	5.64
300	140	5.55

4.5 Analisa Hasil

Nilai *error rate* menunjukkan tingkat kesalahan pada hasil rekomendasi yang diberikan. Semakin tinggi nilai *error rate* yang diperoleh semakin besar pula tingkat kesalahan hasil rekomendasi yang diberikan oleh sistem. Begitu juga sebaliknya, semakin rendah nilai *error rate* yang dihasilkan semakin kecil tingkat kesalahan pada hasil rekomendasi yang diberikan oleh sistem.

Nilai pengukuran kesalahan rekomendasi sistem yang telah ditunjukkan pada subbab 4.4.1 dapat dinyatakan dengan grafik yang ditunjukkan pada Gambar 4.13

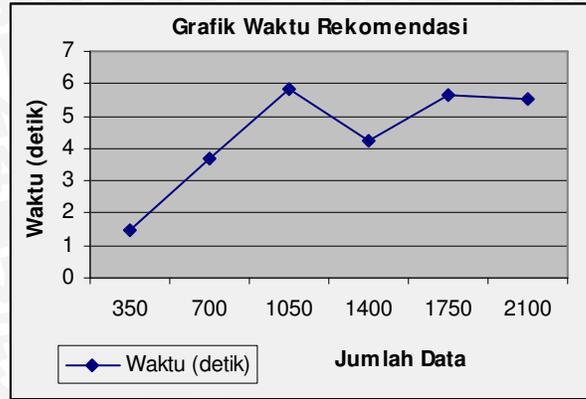


Gambar 4.13 Grafik pengukuran kesalahan rekomendasi

Hasil evaluasi tingkat kesalahan yang digambarkan pada Gambar 4.14 menunjukkan bahwa nilai *error rate* yang diperoleh dengan data sampel sebanyak 1050 data *training* adalah 27.86%. Nilai ini menunjukkan bahwa tingkat kesalahan sistem dalam memberikan rekomendasi produk rumah adalah 27.86%. Dengan kata lain bahwa sistem sudah dapat memberikan rekomendasi produk rumah dengan tingkat kebenaran sebesar 72.14%.

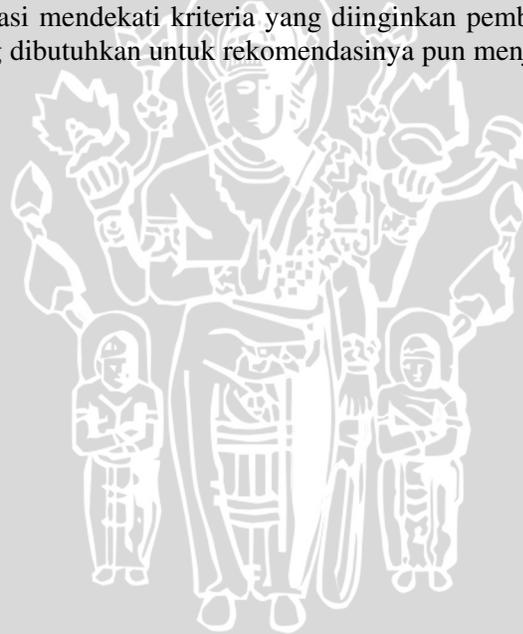
Untuk mengetahui pengaruh antara jumlah data dan tingkat kesalahan yang terjadi maka dilakukan uji coba dengan mengurangi jumlah data *training* menjadi 700 dan 350, serta menambah jumlah data menjadi 1400, 1750, dan 2100. Gambar 4.13 ditunjukkan bahwa nilai *error rate* yang dihasilkan berbeda-beda berdasarkan jumlah data *training* yang digunakan. Nilai *error rate* yang terjadi menurun sebanding dengan meningkatnya jumlah data. Dengan kata lain semakin besar jumlah data yang digunakan semakin kecil nilai *error rate* yang dihasilkan.

Sedangkan untuk pengukuran waktu rekomendasi sistem dapat dinyatakan dengan grafik yang ditunjukkan pada Gambar 4.14



Gambar 4.14 Grafik pengukuran waktu rekomendasi

Gambar 4.14 menunjukkan bahwa waktu yang diperlukan sistem dalam memberikan rekomendasi produk rumah dipengaruhi oleh jumlah aturan klasifikasi yang harus diseleksi. Semakin banyak aturan klasifikasi yang diseleksi, semakin besar pula waktu yang diperlukan sistem dalam memberikan rekomendasi produk rumah kepada pembeli. Hal ini disebabkan dengan meningkatnya jumlah aturan klasifikasi, maka sistem akan lebih banyak melakukan seleksi agar hasil rekomendasi mendekati kriteria yang diinginkan pembeli, sehingga waktu yang dibutuhkan untuk rekomendasinya pun menjadi lebih lama.



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari hasil penelitian ini adalah:

1. Algoritma PUBLIC dapat diimplementasikan dalam pembangunan *decision tree* pada proses klasifikasi data produk rumah.
2. Secara garis besar tingkat kesalahan rekomendasi akan semakin menurun seiring dengan meningkatnya jumlah data *training*, Hal ini ditunjukkan dari hasil penelitian bahwa dengan jumlah 150 *record* data training, *error rate* yang dihasilkan sebesar 27,86%, dan untuk 200 *record* data training *error rate* yang dihasilkan sebesar 23,57%. Sedangkan dari Grafik 4.14 menunjukkan bahwa waktu yang diperlukan dalam memberikan rekomendasi berubah-ubah. Waktu rekomendasi tidak tergantung pada jumlah data training, melainkan tergantung pada jumlah aturan klasifikasi yang terbentuk dalam *database*. Jika jumlah aturan klasifikasi yang terbentuk besar, maka waktu yang diperlukan dalam memberikan rekomendasi akan meningkat pula.

Dan berdasarkan tingkat akurasi yang dihasilkan dan waktu yang diperlukan dalam memberikan rekomendasi, metode PUBLIC ini cukup baik untuk diterapkan dalam membangun sistem rekomendasi pembelian rumah.

5.2 Saran

Saran yang dapat diberikan penulis untuk pengembangan lebih lanjut adalah sebaiknya atribut yang diseleksi lebih kompleks dan jumlahnya lebih banyak, sehingga tingkat keakuratan algoritma PUBLIC dapat ditingkatkan.

DAFTAR PUSTAKA

- Gehrke, J. *Advances in Decision Tree Construction* <http://www.cs.cornell.edu/johannes/papers/2001/kdd2001-tutorial-final.pdf>. 2001. Tanggal akses : 27 Maret 2008.
- Gunduz, S dan Ozsu, T. Recommendation Model for *User Accesses to Web Pages*. <http://www.itu.edu.tr/~sgunduz/papers/icann.pdf>., 2004. Tanggal akses : 27 Maret 2008.
- Guidici, P. 2003. *Applied Data Mining Statistical Methods for Business and Industry*. . John Wiley & Sons. Chicester. Hal 130-132.
- Han, E. dan G. Karypis. Feature Based Recommendation System. <http://glaros.dtc.umn.edu/gkhome/fetch/papers/fbrsCIKM05.pdf>., 2005. Tanggal akses : 22 Maret 2008
- Kantardzic, M. *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons. New Jersey. Subbab Data-Mining Process. 2003.
- Mehta, M. Jorma dan Rakesh. 1995. *MDL-based Decision Tree Pruning*. IBM Almaden Research Center - San Jose, CA.
- Moertini, V. "Data Mining Sebagai Solusi Bisnis". http://home.unpar.ac.id/~integral/Volume_207/Integral_207_20No_201/idadamining.pdf. 2002. Tanggal akses : 22 Maret 2008
- Pramudiono, I. "Pengantar Data Mining : Menambang Permata Pengetahuan di Gunung Data". <http://www.ilmukomputer.com>, 2003. Tanggal akses : 27 Februari 2008.
- Prima, A. 2005. "Analisis dan Implementasi Algoritma PUBLIC sebagai sebuah Classifier Pohon Keputusan yang Scalable dalam Data Mining". <http://www.stttelkom.ac.id>, 2005. Tanggal akses : 27 Februari 2008.

Rastogi, R dan Kyuseok, S. 1998. *PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning*. Bell Laboratories, Murray Hill, NJ.

Zalilia, L. "Penerapan Data Mining untuk IDS". Tugas Akhir Mata Kuliah EC7010. Departemen Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Bandung, 2007.

