PEMBELAJARAN KLASIFIKASI *DECISION TREE* MENGGUNAKAN ALGORITMA C4.5 UNTUK PEREKOMENDASIAN *HANDPHONE*

SKRIPSI

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

oleh :
Dewi Purbasari
0410963011

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

repository.ub.ac
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PEMBELAJARAN KLASIFIKASI *DECISION TREE*
MENGGUNAKAN ALGORITMA C4.5
UNTUK PEREKOMENDASIAN *HANDPHONE*

oleh:
DEWI PURBASARI
0410963011-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 6 Januari 2008
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,	Pembimbing II,
Drs. Muh. Arif Rahman, M.Kom	**Bayu Rahayudi, MT, MM**
NIP. 131 971 481	**NIP. 132 318 424**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, Msc
NIP. 131 126 049

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	**:**	Dewi Purbasari
NIM	**:**	0410963011-96
Jurusan	**:**	Matematika / Ilmu Komputer
Penulis tugas akhir berjudul	**:**	Pembelajaran Klasifikasi *Decision Tree* menggunakan Algoritma C4.5 untuk Perekomendasian *Handphone.*

Dengan ini menyatakan bahwa :

1. **Isi dari tugas akhir 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 Tugas Akhir ini.**
2. **Apabila dikemudian hari ternyata Tugas Akhir 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, 6 Januari 2009
Yang menyatakan,

Dewi Purbasari
NIM. 0410963011

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PEMBELAJARAN KLASIFIKASI *DECISION TREE* MENGGUNAKAN ALGORITMA C4.5 UNTUK PEREKOMENDASIAN *HANDPHONE*

ABSTRAK

Dengan berkembangnya teknologi, *handphone* tidak saja berfungsi sebagai penghubung komunikasi antar manusia melainkan juga dapat mendengarkan musik atau bermain *game*. Selain itu juga *handphone* telah dilengkapi dengan kamera, dan fasilitas lainnya yaitu Internet. Untuk mendapatkan *handphone* yang sesuai dengan keinginan maka dibutuhkan suatu sistem rekomendasi yang tepat. Perekomendasian yang tepat membutuhkan data–data dalam jumlah banyak, maka dibutuhkan suatu teknologi untuk pengolahan data. Salah satu teknologi yang dapat menganalisis data dalam jumlah banyak adalah *data mining*. Salah satu metode yang sering digunakan untuk menganalisis data yaitu klasifikasi, yang merupakan salah satu metode dari teknologi *data mining*.

Pada penelitian ini dibuat suatu aplikasi yang mengimplementasikan salah metode dalam klasifikasi yaitu *decision tree* dengan algoritma C4.5. Algoritma ini digunakan untuk menganalisis data bertipe diskrit dan kontinyu. Data yang digunakan dalam penelitian ini adalah data penjualan *handphone.* Kemudian dari data tersebut dianalisis dan diuji tingkat kebenaran atau akurasi dari hasil rekomendsi, serta *running time* (waktu yang dibutuhkan oleh sistem) dalam memberikan rekomendasi.

Hasil pengujian sistem pertama menunjukkan bahwa *data mining* dengan metode *decision tree* menggunakan algoritma C4.5 dapat diimplementasikan untuk perekomendasian *handphone*. Hasil pengujian sistem kedua menunjukkan bahwa semakin bertambahnya data yang digunakan dalam proses pembelajaran maka kebenaran sistem dalam pemberian rekomendasi akan semakin meningkat. Begitu juga pada *running time* (waktu yang dibutuhkan oleh sistem) semakin meningkat, seiring dengan meingkatnya jumlah data pembelajaran yang digunakan pada saat tes.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEARNING CLASSIFICATION DECISION TREE USING ALGORITHM C4.5 FOR RECOMMENDATION HANDPHONE

ABSTRACT

With the development of technology, mobile phones not only serve as the communication link between people, but also can listen to music or play games. Besides mobile phones also have equipped with a camera, and other facilities, namely the Internet. In order to get a accordance handphone on demand, its needed an appropriate recommendations system. Appropriate recommendations need the right data in the number of lots, so it needed a technology for data processing. One technology that can analyze a large amount of data is data mining. One method often used to analyze the data classification, which is one method of data mining technology.

In this research, which made an application to one of the methods in the classification decision tree with a algorithm C4.5, this algorithm is used to analyze the data with type discrete and continuous. Data used in this research is the mobile phone sales data. Then, from the data analyzed and tested the level of truth or accuracy of the results of the recommendations, and the running time (the time needed by the system) in providing recommendations.

The results of the test first system show that with data mining methods using decision tree algorithm C4.5 can be implemented to recommendations mobile phone. The results of the test second system show that increasing the data used in the process of learning the truth of the recommendations in the system will be increased. So do in the running time (time needed by system) getting increased, due increasing number of learning data being used while on test.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Alhamdullillahi Robbil alamin, puji syukur Penulis panjatkan kepada Allah SWT atas segala limpahan rahmat dan hidayah-Nya, sehingga Penulis dapat menyelesaikan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer

Tugas akhir ini bertujuan untuk menerapkan algoritma c4.5 dalam klasifikasi *decision tree*. Berdasarkan tujuan tersebut Penulis mengambil judul untuk tugas akhir ini "Pembelajaran Klasifikasi *Decision Tree* menggunakan Algoritma C4.5 untuk Perekomendasian *Handphone*".

Pada penyusunan tugas akhir ini, Penulis ingin mengucapkan terima kasih kepada :

1. Drs. M. Arif Rahman, M.Kom., selaku pembimbing utama atas arahan serta bimbingannya dalam penyusunan tugas akhir ini.
2. Bayu Rahayudi, MT., MM., selaku pembimbing pendamping atas arahan serta bimbingannya dalam penyusunan tugas akhir ini.
3. Wayan Firdaus Mahmudi, S.Si, MT, selaku Ketua Program Studi Ilmu Komputer, Jurusan Matematika, FMIPA Universitas Brawijaya.
4. Dr. Agus Suryanto, MSc., selaku Ketua Jurusan Matematika, FMIPA Universitas Brawijaya.
5. Drs. Mardji, MT., selaku penasehat akademik.
6. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
7. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan tugas akhir ini.
8. Orang tua Penulis atas dukungan materi dan doa restunya kepada Penulis.
9. Rekan-rekan di Program Studi Ilmu Komputer FMIPA Universitas Brawijaya yang telah banyak memberikan

bantuannya demi kelancaran pelaksanaan penyusunan tugas akhir ini.

10. Dan semua pihak yang telah membantu dalam penyusunan tugas akhir ini yang tidak dapat Penulis sebutkan satu per satu.

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

Semoga laporan tugas akhir ini dapat bermanfaat. Amin.

Malang, Desember 2009

Penulis

repository.ub.ac
UNIVERSITAS BRAWIJAYA

DAFTAR ISI

Halaman

HALAMAN JUDUL i
HALAMAN PENGESAHAN iii
HALAMAN PERNYATAAN v
ABSTRAK vii
KATA PENGANTAR ix
DAFTAR ISI xiii
DAFTAR GAMBAR xv
DAFTAR TABEL xvii
BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian..... 3
1.4 Batasan Masalah 3
1.5 Manfaat Penelitian 3
1.6 Metodologi Pemecahan Masalah 4
1.7 Sistematika Penulisan 4

BAB II TINJAUAN PUSTAKA
2.1 *Data Mining* 7
2.1.1 Definisi 7
2.1.2 *Knowledge discovery in database* (KDD) 8
2.1.3 Fungsionalitas *data mining* 9
2.2 Klasifikasi *Data Mining* 10
2.2.1 Definisi klasifikasi 10
2.3 *Decision Tree* 11
2.3.1 Pembangunan *decision tree* 13
2.3.2 Pembangunan *decision tree* dalam algoritma C4.5 15
2.3.3 Information Gain..... 16
2.3.4 Penanganan atribut kontinyu 18
2.3.5 Mengubah *decision tree* menjadi *classification rules* 19

2.3.6 *Pruning tree* 19
2.4 Rekomendasi 20
BAB III METODOLOGI DAN PERANCANGAN SISTEM
3.1 Analisis Sistem 22
3.1.1 Deskripsi sistem 22
3.1.2 Batasan sistem 22
3.2 Perancangan Sistem 22
3.2.1 Perancangan basis data 22
3.2.2 Perancangan penelitian 26
3.2.3 Perancangan antarmuka 31
3.3 Perancangan Uji Coba 33
3.3.1 Data yang digunakan dalam penelitian 33
3.3.2 Pengujian sistem 33
3.3.3 Pengujian pada saat rekomendasi sistem 34
3.3.4 Pengujian *running time* rekomendasi sistem 35
3.4 Perhitungan Manual 36

BAB IV IMPLEMENTASI DAN PEMBAHASAN
4.1 Lingkungan Implementasi 43
4.1.1 Lingkungan perangkat keras 43
4.1.2 Lingkungan perangkat lunak 43
4.2 Implementasi Program 44
4.2.1 Struktur data 44
4.2.2 Implementasi proses pembelajaran 49
4.2.3 Implementasi proses perekomendasian 61
4.3 Penerapan Aplikasi 63
4.4 Pengujian sistem rekomendasi 67
4.4.1 Pengujian tingkat kebenaran 67
4.4.2 Pengujian *running time* rekomendasi sistem 68
4.5 Analisis Hasil 70

BAB V KESIMPULAN DAN SARAN
5.1 Kesimpulan..... 71
5.2 Saran 71

DAFTAR PUSTAKA 73

DAFTAR GAMBAR

Halaman

Gambar 2.1 Tahapan Proses KDD 9
Gambar 2.2 Klasifikasi dengan *Input Atribut Set* dan *Output Class Label* 11
Gambar 2.3 Contoh Penerapan Metode *Decision Tree* untuk Klasifikasi *Handphone* Beserta Kriteria Pembeli 12
Gambar 3.1 Tahapan Penelitian 21
Gambar 3.2 Hubungan Antar Tabel 25
Gambar 3.3 Alur Proses 26
Gambar 3.4 Proses Cari Atribut Terbaik 28
Gambar 3.5 Proses Pembentukan *Tree* 30
Gambar 3.6 Proses Perekomendasi HP 31
Gambar 3.7 Diagram Apikasi Sistem 32
Gambar 3.8 *Decision Tree* 41
Gambar 4.1 Struktur Data Cari Atribut Terbaik 45
Gambar 4.2 Struktur Data *Pruning* 47
Gambar 4.3 Pengambilan Data dari *Database* 49
Gambar 4.4 Proses Perhitungan Jumlah Kategori Atribut Diskrit Dan Atribut Kontinyu Pada Data Pembelajaran .. 51
Gambar 4.5 Proses Perhitungan Probabilitas Atribut Bertipe Diskrit 51
Gambar 4.6 Proses Perhitungan Probabilitas Atribut Bertipe Kontinyu 51
Gambar 4.7 Proses Perhitungan Probabilitas Atribut Bertipe Diskrit 52
Gambar 4.8 Proses Perhitungan Probabilitas Atribut Bertipe Kontinyu 53
Gambar 4.9 Proses Perhitungan *Information* Setiap Atribut 53
Gambar 4.10 Proses Perhitungan *Entropy* Atribut Diskrit 54
Gambar 4.11 Proses Perhitungan *Entropy* Atribut Kontinyu 55
Gambar 4.12 Proses Pembentukan *Tree* 58
Gambar 4.13 Proses Mengekstrak Aturan Klasifikasi 59
Gambar 4.14 Proses *Prunning Rule* 60

repository.ub.ac
UNIVERSITAS BRAWIJAYA

Gambar 4.15 Proses Memasukkan Aturan Klasifikasi ke *Database* 60
Gambar 4.16 Proses Perekomendasian 62
Gambar 4.17 Tampilan Menu Transaksi Penjualan 63
Gambar 4.18 Tampilan Menu Data Pembelajaran 64
Gambar 4.19 Tampilan Menu Perekomendasian 65
Gambar 4.20 Tampilan Menu Pengujian Sistem 66
Gambar 4.21 Grafik Jumlah Sampel Benar 67
Gambar 4.22 Grafik Tingkat Kebenaran 68
Gambar 4.23 Grafik Hasil Pengujian *Running Time* Rekomendasi 69

DAFTAR TABEL

Halaman

Tabel 3.1 Tabel Konsumen (T_Kons) 23
Tabel 3.2 Tabel Transaksi (T_Transaksi) 23
Tabel 3.3 Tabel Barang (T_HP) 24
Tabel 3.4 Tabel Detail Barang (T_DetBrg) 24
Tabel 3.5 Tabel Detail Fitur (T_DetFitur) 25
Tabel 3.6 Tipe Atribut Dan Kategori Pada kriteria Pembeli 33
Tabel 3.7 Rancangan Tabel Data Pembelajaran 34
Tabel 3.8 Rancangan Tabel Pengujian Rekomendasi Sistem 35
Tabel 3.9 Rancangan Tabel Pengujian Rekomendasi Sistem 35
Tabel 3.10 Contoh Data Sampel 37
Tabel 3.11 Probabilitas Jenis HP 36
Tabel 3.12 Probabilitas Jenis Kelamin..... 38
Tabel 3.13 Probabilitas Jenis Kelamin Terhadap Jenis HP .. 38
Tabel 3.14 Probabilitas Umur 39
Tabel 3.15 Probabilitas Umur Terhadap Jenis HP 39
Tabel 3.16 Contoh Data Yang Akan DiCari Produk HP Yang Sesuai 42
Tabel 4.1 Keterangan Struktur Data Cari Atribut Terbaik.. 45
Tabel 4.2 Keterangan Struktur Data *Prunning* 47
Tabel 4.3 Deskripsi Variabel Pada `procedure buatTree` 48
Tabel 4.4 Hasil Pengujian Tingkat Kebenaran 67
Tabel 4.5 Hasil Pengujian *Running Time* Rekomendasi..... 69

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sebagai makhluk sosial komunikasi merupakan kegiatan yang paling penting, dimana kegiatan ini memiliki tujuan untuk menyampaikan pesan kepada orang lain. Penyampaian pesan ini membutuhkan suatu alat atau fasilitas untuk berkomunikasi, karena tidak semua lawan bicara dekat dengan kita. Fasilitas ini dapat berupa *handphone*.

Dengan berkembangnya teknologi, *handphone* tidak saja berfungsi sebagai penghubung komunikasi antar manusia melainkan juga dapat digunakan untuk menghilangkan kejenuhan dengan mendengarkan musik atau bermain *game*. Selain itu juga *handphone* dapat digunakan untuk mendokumentasikan sebuah gambar nyata. Dari beberapa kegunaan yang disebut tadi, *handphone* juga telah dilengkapi dengan Internet yang digunakan untuk mencari informasi dalam waktu yang singkat. Dari penjelasan sebelumnya *handphone* dapat diartikan sebagai salah satu alat yang digunakan untuk mengirim suara, data, video untuk keperluan komunikasi antar manusia.

Untuk mendapatkan *handphone* yang sesuai dengan keinginan maka dibutuhkan suatu sistem rekomendasi yang tepat. Perekomendasian yang tepat dapat menunjang keberhasilan suatu perusahaan karena konsumen disini berperan sebagai pengambil keputusan. Suatu perusahaan bisa memanfatkan data konsumen untuk target pemasaran suatu produk dimana yang diperoleh dari data penjualan *handphone*. Oleh karena itu suatu perusahaan harus menyediakan infomasi-informasi yang detail dan tepat untuk konsumen, maka dibutuhkan data–data dalam jumlah banyak untuk menghasilkan informasi yang tepat. Data dalam jumlah banyak membutuhkan suatu teknologi untuk pengolahan data dengan mengharapkan hasil yang maksimal dan waktu yang minimum. Ketersediaan data yang berlimpah dihasilkan dari penggunaan teknologi informasi di hampir semua bidang kehidupan, menimbulkan suatu kebutuhan untuk dapat memanfaatkan informasi dan pengetahuan yang terkandung di dalam limpahan data tersebut, yang kemudian ditemukan teknologi pengolahan data yaitu *data mining*.

repository.ub.ac
UNIVERSITAS BRAWIJAYA

Teknologi *data mining* dapat didefinisikan sebagai proses analisis data dalam jumlah banyak dengan menggunakan perangkat lunak untuk menggali nilai tambah berupa informasi atau penemuan pengetahuan (*knowledge discovery in database*) yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Han, 2000). Hasil dari teknologi ini adalah suatu aplikasi untuk menghasilkan informasi baru yang menarik dan berharga bagi perusahaan, yang kemudian digunakan sebagai pengambilan keputusan.

Sekarang ini telah banyak metode-metode yang ditawarkan untuk memudahkan dalam pengambilan keputusan yaitu klasifikasi yang merupakan salah satu metode dari teknologi *data mining*. Klasifikasi memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (data pembelajaran) dan menggunakan model tersebut untuk klasifikasi data tes serta menguji tingkat kebenaran dari model. Model klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision tree*. *Decision tree* (pohon keputusan) adalah salah satu metode yang sangat populer dari klasifikasi. Salah satu algoritma *decision tree* yang sering digunakan C4.5 (Quinlan,1993).

Penelitian tentang pengklasifikasian data menggunakan algoritma C4.5 sebelumnya pernah dilakukan, dengan judul penelitian "Penggunaan Algoritma C4.5 Untuk Pengklasifikasian Dataset Perbankan" (Veronica S. Moertini, 2003). Dalam penelitian ini membahas tentang teknik-teknik klasifikasi data secara umum, metodologi dasar algoritma C4.5, proses dan analisis hasil eksperimen yang menggunakan C4.5 untuk mengklasifikasi data perbankan.

Penggunaan algoritma C4.5 untuk pengklasifikasian bisa dikembangkan menjadi suatu sistem rekomendasi untuk penjualan suatu produk.

Berdasarkan latar belakang yang telah dipaparkan, maka judul yang diambil dalam Tugas Akhir ini adalah "**Pembelajaran Klasifikasi *Decision Tree* menggunakan *Algoritma C4.5* Untuk Perekomendasian *Handphone*".**

Dengan adanya sistem rekomendasi pembelian *handphone* ini diharapkan dapat mempermudah konsumen yang dimana konsumen tidak perlu melakukan browsing ke internet atau membeli koran dalam mencari produk handphone yang diinginkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka dalam pembuatan tugas akhir ini dapat dirumuskan masalah sebagai berikut :

1. Bagaimana menerapkan *decision tree* untuk perekomendasian *handphone*.
2. Menganalisis dan menguji tingkat kebenaran pada metode *decision tree* sebagai salah satu metodologi pada klasifikasi *data mining* dengan menggunakan algoritma C4.5.
3. Menganalisis *running time* (waktu yang dibutuhkan oleh sistem) dalam perekomendasian *handphone.*

1.3 Tujuan

Tujuan yang dicapai dari tugas akhir ini adalah :

1. Mengimplementasikan metode *decision tree* dengan menggunakan algoritma C4.5 untuk perekomendasian *handphone*.
2. Melakukan analisis dan pengujian terhadap *decision tree* tingkat kebenaran atau akurasi, yang nantinya dapat digunakan untuk pengembangan selanjutnya.
3. Melakukan analisis *running time* (waktu yang dibutuhkan oleh sistem) dalam perekomendasian *handphone.*

1.4 Batasan Masalah

Batasan masalah untuk tugas akhir ini adalah :

1. *Dataset* yang digunakan bertipe record.
2. Jika pada *dataset* terdapat atribut yang kontinyu maka jumlah kelas dibatasi paling banyak 2 kelas.
3. Perhitungan data yang dilakukan tanpa *missing value*.

1.5 Manfaat Peneitian

Manfaat dari penulisan tugas akhir ini adalah :

1. Menyediakan sebuah aplikasi untuk perekomendasian *handphone*.
2. Mengetahui tingkat kebenaran atau akurasi, yang nantinya dapat digunakan untuk pengembangan selanjutnya.
3. Mengetahui *running time* (waktu yang dibutuhkan oleh sistem) dalam perekomendasian *handphone.*

repository.ub.ac
UNIVERSITAS BRAWIJAYA

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnnya, maka metodologi yang digunakan dalam penulisan tugas akhir ini adalah :

1. Studi Literatur
   Pencarian referensi-referensi untuk pendukung penulisan tugas akhir ini dan pendalaman materi.
2. Pendefisian dan analisis masalah
   Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.
3. Perancangan dan implementasi sistem
   Membuat perancangan perangkat lunak dengan mengalisis kebutuhan sistem dan mengimplementasikan hasil rancangan.
4. Uji coba dan analisis hasil implementasi
   Menguji perangkat lunak dengan data yang sebenarnya, dan menganalisis hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, , manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini berisi tentang teori-teori yang menjadi acuan untuk pelaksanaan penulisan tugas akhir yang meliputi teori tentang klasifikasi *data mining,* diantaranya: definisi *data mining*, klasifikasi *data mining*, tahap-tahap *Knowledge Discovery in Database* (KDD), fungsionalitas *data mining* karakteristik dan cara pencarian pola menggunakan *decision tree* dan Algoritma C4.5.

BAB III PERANCANGAN DAN METODOLOGI

Bab ini berisi tentang aliran proses atau alur dari sistem yang akan dibuat serta algoritma dan *flowchart* dari program serta pembahasannya.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini membahas tentang analisis dan pembahasan tentang kinerja dari program yang telah dibuat baik dari antar muka maupun proses.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari sistem yang dirancang serta saran dari hasil tugas akhir ini untuk pengembangan selanjutnya.

repository.ub.ac
UNIVERSITAS BRAWIJAYA

BAB II
TINJAUAN PUSTAKA

2.1 Data Mining

2.1.1 Definisi

Ketersediaan data yang berlimpah yang dihasilkan dari penggunaan teknologi informasi di hampir semua bidang kehidupan, menimbulkan kebutuhan untuk dapat memanfaatkan informasi dan pengetahuan yang terkandung di dalam limpahan data tersebut, pengolahan data tersebut tidak dapat menggunakan alat dan metode analisis tradisional untuk mengekstrak data dalam volume yang besar. *Data mining* merupakan teknologi yang menggabungkan metode analisis tradisional dengan algoritma yang canggih untuk memproses data dengan volume besar. *Data mining tools* memperkirakan perilaku dan tren masa depan, memungkinkan bisnis untuk membuat keputusan yang proaktif dan berdasarkan pengetahuan. *Data mining tools* mampu menjawab permasalahan bisnis, yang dimana jika dihitung secara tradisional membutukan waktu yang lama untuk diselesaikan. *Data mining tools* menjelajah database untuk mencari pola tersembunyi, menemukan infomasi yang prediktif.

Ada beberapa pengertian yang berkaitan dengan *data mining* dari beberapa referensi, yaitu sebagai berikut :

1. *Data mining* adalah semua proses pencarian pola-pola (*patterns*) pada data secara otomatis (Moore, 2008).
2. *Data mining* adalah penemuan tren atau pola yang menarik dalam dataset besar untuk memandu pengambilan keptusan mengenai aktivitas yang akan datang (Ramakrishnan, 2003).
3. *Data mining* adalah penggalian dan analisis (dengan alat otomatis atau semi - otomatis) terhadap himpunan data yang besar untuk menemukan pola-pola yang bermakna (Hidayat, 2008).
4. *Data Mining* adalah banyak istilah yang digunakan untuk menunjukkan pengertian *data mining* seperti *knowledge mining from data, knowledge extraction,* data/ *pattern analysis, data archeology,* dan *data dredging*. Banyak orang beranggapan bahwa *data mining* mempunyai persamaan dengan *knowledge discovery in database or* KKD (Han, 2006).

5. *Data mining* atau *Knowledge Discovery in Databases* (KDD) adalah pengambilan informasi yang tersembunyi, dimana informasi tersebut sebelumnya tidak dikenal dan berpotensi bermanfaat (Han dan Khamber, 2000).

Secara sederhana *data mining* adalah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di basis data yang besar. Dalam jurnal ilmiah, *data mining* juga dikenal dengan nama KDD.

2.1.2 *Knowledge discovery in database* (KDD)

KDD adalah keseluruhan proses non - trivial untuk mencari dan mengidentifikasi pola (*pattern*) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti (Fayyad, 1996).

Proses KDD secara garis besar dapat dijelaskan sebagai berikut :

1. Data *Selection*

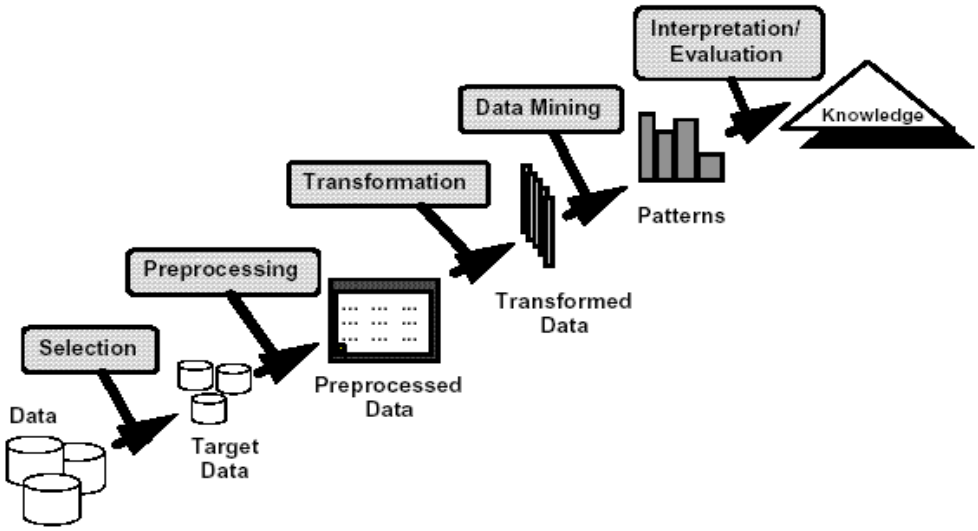
   Sebelum tahap penggalian informasi dalam KDD dimulai, perlu dilakukan proses seleksi (*selection*) data yang nantinya digunakan untuk proses *data mining*.
2. *Pre-processing/ Cleaning*

   Proses *cleaning* data menjadi fokus pada KDD. Proses *cleaning* mencakup beberapa proses yaitu :
   - membuang duplikasi data
   - membuang data yang tidak konsisten
3. *Transformation*

   Proses transformasi data adalah melakukan diskretisasi terhadap atribut bertipe kontinyu. Proses pembelajaran ukuran data yang diproses harus lebih kecil.
4. *Data mining*

   Pada tahap ini merupakan bagian dari teknik *data mining*.
5. *Interpretation/ Evaluation*

   Pada tahap ini pencarian informasi yang dihasilkan dari proses *data mining* ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Adapun proses KDD dapat dilihat pada gambar 2.1.



Gambar 2.1 Tahapan Proses KDD

Pada umumnya proses KDD terdiri dari 5 tahap, tetapi sebenarnya pada proses tertentu terdapat suatu pengulangan. Seorang analis bisa kembali ke tahap semula, karena analis menyadari bahwa proses *cleaning* belum dilakukan dengan sempurna, atau karena analis menemukan data atau informasi baru untuk "memperkaya" data yang sudah ada. Di dalam proses KDD terdapat *data mining* yang nantinya akan dibahas pada penulisan tugas akhir ini.

2.1.3 Fungsionalitas *data mining*

Menurut Hand and Kamber,2001, ada beberapa *fungsionalitas* dari data mining dan pola pencarian yang dapat ditemukan dalam data mining dapat dijelaskan seperti dibawah ini:

1. Deskripsi kelas
   Data dapat diasosiasikan dengan kelas. yang berguna untuk menggambarkan kelas secara individual dan konsep secara tepat.
2. Analisis *association*
   Analisis *association* adalah penemuan *association rules* yang menunjukkan nilai kondisi dari atribut yang terjadi secara bersama-sama dan terus-menerus dalam membentuk sekumpulan data.
3. Klasifikasi dan Prediksi
   Klasifikasi adalah proses untuk menemukan sekumpulan model yang menggambarkan dan membedakan kelas data, yang bertujuan agar model tersebut dapat digunakan untuk

memprediksi kelas suatu obyek yang belum diketahui kelasnya. *Classification* dapat digunakan untuk memprediksi data nominal ataupun data diskrit.

4. Analisis *Cluster*

   Analisis *Cluster* berbeda dengan *classification. Clustering* menganalisis obyek data dengan kelas yang belum diketahui, sedangkan *classification* menganalisis obyek data dari kelas yang telah diketahui.

5. Analisis *Outlier*

   Outlier dapat dideteksi menggunakan tes yang bersifat statistik yang mengambil sebuah distribusi atau probabilitas model untuk data atau menggunakan ukuran jarak, dimana obyek yang jaraknya jauh dari *cluster* yang lain dianggap sebagai *outlier.*

6. Analisis *Evolution*

   Data analisis *evolution* ini mencari model atau obyek yang memiliki kebiasaan berubah setiap waktu. Analisis ini berkaitan dengan *time-series*. *Evolution analysis* bisa berupa *characterizaiton, discrimination, association, classification*, *clustering*.

Fungsionalitas (kegunaan) *data mining* adalah untuk menspesifikasikan pola yang harus ditemukan dalam tugas *data mining*. Secara umum tugas *data mining* dapat diklasifikasikan dalam dua kategori yaitu deskriptif dan prediktif. Tugas *data mining* secara deskriptif adalah untuk mengklasifikasikan sifat umum suatu data didalam *database*. Sedangkan tugas *data mining* secara prediktif adalah untuk mengambil kesimpulan terhadap data dalam membuat prediksi (Hand and Kamber,2001).

Dalam penulisan tugas akhir ini menggunakan metode prediktif, karena metode klasifikasi yang digunakan pada tugas akhir ini termasuk dalam metode prediktif.

2.2 Klasifikasi *Data Mining*

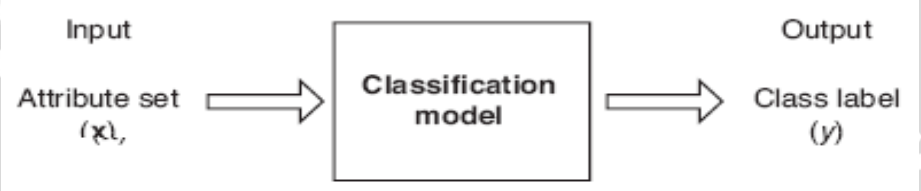
2.2.1 Definisi klasifikasi

Klasifikasi adalah sebuah proses untuk menemukan model yang menjelaskan atau membedakan konsep atau kelas data, dengan

tujuan untuk dapat memperkirakan kelas dari suatu obyek yang kelasnya tidak diketahui (Tan, 2006).

Untuk mendapatkan model, harus dilakukan analisis terhadap data pembelajaran (*training set*). Sedangkan data uji (*test set*) digunakan untuk mengetahui tingkat kebenaran dari model yang telah dihasilkan.

Input data untuk klasifikasi adalah kumpulan *record*. Setiap *record* dikenal sebagai *instance* atau contoh yang ditandai oleh *tuple* (x,y) dimana x adalah atribut dan y adalah atribut khusus yang menunjukkan label kelas (disebut juga kategori atau atribut target), yang dapat dilihat pada gambar 2.2.



Gambar 2.2 Klasifikasi dengan *Input Attribute Set* dan *Output Class Label*

Sasaran dari permasalahan klasifikasi ialah untuk mengidentifikasi karakteristik yang menunjukkan kelompok dimana setiap kasus harus ditempatkan. Hal ini dapat digunakan untuk memahami data yang sudah ada, atau juga untuk memprediksi data yang baru.

2.3 *Decision Tree*

Decision tree merupakan salah satu metode pada klasifikasi *data mining* yang mudah diinterpretasikan. Metode *Decision tree* sangat terkenal daripada metode klasifikasi yang lainnya, karena metode ini tidak membutuhkan pengetahuan yang lebih atau pengaturan parameter (Han dan Khamber, 2000). Ada dua tipe *decision tree* (www.dtreg.com,2008), yaitu :

- *Classification tree*

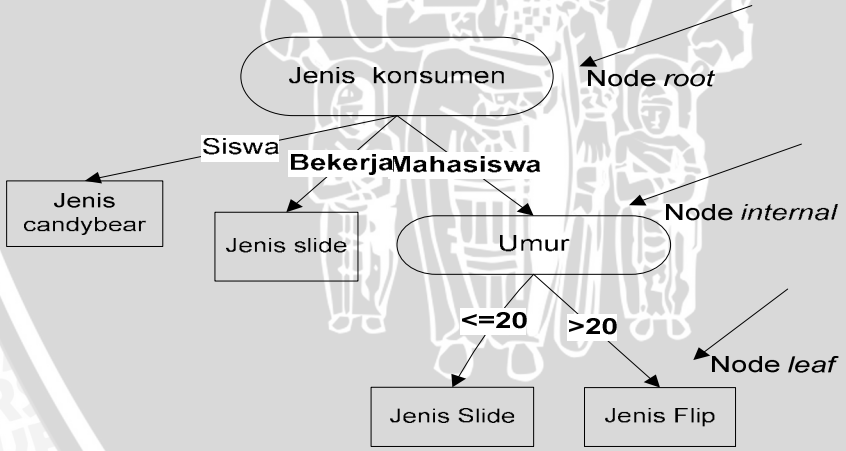
  Memasukkan *record* ke kelas data yang telah diklasifikasikan untuk menghasilkan sebuah node.
- *Regression tree*

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

Dalam penelitian tugas akhir ini menggunakan *classification tree*, karena penelitian ini hanya mengklasifikasikan data-data ke dalam kelas-kelas yang telah tersedia.

Decision tree merupakan *supervised learning* dimana kelas dari sekumpulan data telah ditentukan sebelumnya. Pada pembentukan *decision tree* membutuhkan data pembelajaran dalam jumlah yang data yang banyak agar terbentuknya variasi yang berbeda-beda pada tiap-tiap kelas yang telah didefinisikan. Hal ini disebabkan *decision tree* menganalisis dari data yang telah ada, apabila data tersebut secara sistematis kekurangan data pembelajarannya maka hasil klasifikasi akan menjadi kurang optimal.

Pada *decision tree* terdapat beberapa algoritma, seperti CLS, ID3, C4.5, CART, SLIQ, SPRINT. *Decision tree* sangat mudah untuk dipahami dan diinterpretasikan. Akan tetapi, *tree* bisa menjadi sangat kompleks, yang dimana suatu *tree* harus dimasukkan ke dalam database dengan ratusan atribut. Algoritma yang digunakan dalam penulisan tugas akhir ini adalah menggunakan algoritma C4.5, yang dimana setiap node haru memiliki dua nilai atau lebih. Contoh struktur sebuah *decision tree* (Larose, 2005), dapat dilihat pada gambar 2.3.



Gambar 2.3 Contoh Penerapan Metode *Decision Tree* Untuk Klasifikasi *Handphone* Beserta Kriteria Pembeli

Keterangan :

node *root* : menunjukkan pengujian terhadap sebuah nilai atribut.

node *internal* : merepresentasikan hasil dari pengujian tersebut.

node *leaf* : menunjukkan distribusi kelas.

Komponen pertama adalah node *root*, yang menentukan test yang akan dijalankan. Node *root* dalam contoh ini adalah "jenis konsumen". Hasil dari tes ini menyebabkan *tree* terpecah menjadi tiga cabang, dengan tiap cabang merepresentasikan satu dari jawaban yang mungkin. Proses pembentukan *tree* ini akan terus menggunakan proses yang sama atau bersifat rekursif untuk dapat membentuk sebuah *decision tree*. Jika sebuah atribut telah dipilih menjadi atribut tes, maka atribut tersebut tidak diikutkan lagi dalam penghitungan nilai information gain.

2.3.1 Pembangunan *decision tree*

Decision tree (pohon keputusan) adalah suatu proses yang membagi kumpulan data yang berukuran besar menjadi kumpulan *record* yang berukuran lebih kecil. Sehingga setiap *record* memiliki data yang mirip satu dengan yang lain (Berry, 2004).

Proses pembangunan *tree* ini dilakukan dengan cara membagi data secara rekursif atau terus menggunakan proses yang sama, hingga tiap bagian terdiri dari beberapa sampel yang berasal dari kelas yang sama. Ketika sampel tidak dalam kelas yang sama maka dilakukan perhitungan *information gain.* Suatu atribut yang telah dipilih menjadi tes atribut maka atribut tersebut tidak diikutkan lagi dalam perhitungan *information gain* (Tan., dkk, 2004). Pembangunan *tree* dilakukan dengan sebuah algoritma induksi *decision tree*. Rangka algoritma induksi *decision tree* disebut *TreeGrowth* yang terlihat pada algoritma di bawah ini. *Input* terdiri dari record pembelajaran (*training record*) E dan atribut F. Cara kerja algoritma yaitu dengan memilih atribut yang terbaik untuk memisahkan data secara rekursif dan mengembangkan node leaf pada *tree* sampai ditemui kriteria untuk berhenti (Kohavi dan Quinlan, 1999). Berikut ini merupakan algoritma induksi *decision tree* :

TreeGrowth(*E,F*)

```
if hentikan_kond(E,F) = true then
    leaf = BuatNode()
    leaf.label = Klasifikasi(E)
    return leaf.
else
    root = BuatNode()
    root.tes_kond = find_best_split(E,F)
    let V = {v| v mungkin melakukan root.tes_kond}
  for each v ∈ V do
        E_v = {e| root. tes_kond(e) and e ∈ E }
        child = TreeGrowth(E_v,F)
        tambah child keturunan dari root dan label
    (root→ child) as v
    end for
end if
  return root
```

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

1. Fungsi `BuatNode()` memperluas pohon keputusan (*decision tree*) dengan membuat node-node baru. Node yang terdapat pada *decision tree* mempunyai sebuah tes kondisi (atribut tes) yang ditandai dengan **node.tes_kond**, atau sebuah label kelas yang ditandai dengan **node.label**.
2. Fungsi `find_best_split()` menentukan atribut mana yang harus dipilih sebagai suatu tes kondisi untuk membagi data pembelajaran (*record* pembelajaran). Pemilihan tes kondisi bergantung pada perhitungan kemurnian (*impurity*) yang digunakan untuk menentukan split terbaik (ukuran seleksi atribut), misalkan : *entropy*, dan *infomation Gain*.
3. Fungsi `Klasifikasi()` menentukan label kelas untuk ditempatkan pada node *leaf*. Untuk setiap node *leaf* t, umpamakan $p(i \mid t)$ adalah fraksi dari tiap *record* pembelajaran dari kelas i pada node t. Pada umumnya, node *leaf* menempatkan kelas yang mempunyai jumlah mayoritas pada record pembelajaran, leaf.label **=** $\arg\max_i p(i \mid t)$, dimana operator argmax mengembalikan nilai maksimum dari $p(i \mid t)$. Selain menyediakan informasi yang dibutuhkan untuk label kelas dari

suatu node *leaf*, fraksi $p(i \mid t)$ dapat juga digunakan untuk memperkirakan kemungkinan (probabilitas) suatu record ditempatkan dalam node *leaf* t pada kelas i.

4. Fungsi `hentikan_kond()` digunakan untuk menghentikan proses pembentukan pohon dengan menguji apakah semua *record* mempunyai label kelas yang sama atau nilai atribut yang sama. Cara lain untuk menghentikan fungsi rekursif pembentukan pohon adalah dengan menguji apakah jumlah atributnya telah habis, jika demikian maka label kelas akan ditentukan dengan perhitungan kelas terbanyak (*majority class*).

2.3.2 Pembengunan *decision tree* dalam algoritma C4.5

Secara umum pembangunan *decision tree* menggunakan algoritma C4.5 adalah sebagai berikut:

1. Pilih atribut sebagai node *root*.
2. Buat cabang dari *root* untuk setiap nilai dari atribut *root*.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Algoritma ini secara rekursif membuat sebuah *decision tree* berdasarkan data pembelajaran yang telah disiapkan. Algoritma ini mempunyai inputan berupa data pembelajaran dan data. Data pembelajaran berupa data yang akan digunakan untuk membangun sebuah *tree*. Sedangkan data merupakan *field-field* yang nantinya akan di gunakan sebagai parameter dalam melakukan pengelompokan data. Di bawah ini merupakan gambaran dari permasalahan di atas yang diterjemahkan ke dalam bahasa pemrograman.

❖ **Algoritma ID3** (Han dan Khamber, 2000)

Algoritma: Membangun *decision tree*.

Deskripsi : Membangun sebuah *decision tree* dari data pembelajaran.

Input : Data Pembelajaran, Data.

Output: *decision tree*.

Method :

```
1. Buat node N;
2. if  semua data pada same class, C then
3.     Return N beri label node leaf pada class C;
4. if atribute-list habis then
5.     Return N beri label node leaf dengan pemilihan
       class terbanyak pada data; // majority voting
6. seleksi atribut tes, memilih atribut dengan
   information gain tertinggi pada atribut-list;
7. label node N dengan atribut tes;
8. for setiap value ai yang diketahui pada atribut tes
   // batas sampel
9.     terbentuk sebuah cabang dari node N untuk kondisi
       atribut tes = ai;
10.    abaikan si pada sekumpulan data yang datanya dari
       atribut tes = ai;  // a partition
11.    if si kosong then
12.        Beri sebuah label leaf , yang class terbanyak
           pada data;
13.    Else beri node returned pada
       Hasil_decision_tree(si, attriute-list-atribut-tes);
```

❖ **Algoritma C4.5** (Solorio)

1. Dibangun *decision tree* dari data pembelajaran (conventional ID3).
2. Ekstrak hasil *tree* dalam bentuk *rule*. Nomor *rule* menunjukkan nomor alur dari *root* untuk sebuah node *leaf*.
3. *Prune* (*generalize*) setiap *rule* oleh pemindahan prasyarat peningkatan klasifikasi yang akurat.
4. Pegurutan *pruned rule* yang akurat, dan digunakan ketika tes klasifikasi.

2.3.3 *Information gain*

Pemilihan atribut pada algoritma induksi *decision tree* menggunakan ukuran berdasarkan *entropy* yang dikenal dengan *information gain* sebagai sebuah heuristik untuk memilih atribut terbaik dari data ke dalam kelas. Semua atribut adalah bersifat kategori yang bernilai diskrit (berhingga). Atribut dengan nilai kontinyu harus didiskritkan.

Information gain adalah salah satu atribut *selection measure* yang digunakan untuk memilih atribut tes tiap node pada *tree*. Atribut dengan *information gain* tertinggi dipilih sebagai atribut tes dari suatu node (Han dan Khamber, 2000). Ada 2 kasus berbeda ada

saat penghitungan *information gain*, pertama untuk kasus penghitungan atribut tanpa *missing value* dan kedua, penghitungan atribut dengan *missing value.*

- **Penghitungan *information gain* tanpa *missing value***

Misalkan S berisi *s* data *sample*. Anggap atribut untuk *class* memiliki *m* nilai yang berbeda, C_i (untuk i = 1, …,m). Anggap s_i menjadi jumlah sampel S pada *class* C_i. Maka besar *information*-nya dapat dihitung dengan :

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i * \log_2(p_i) \qquad (2.1)$$

Dimana $p_i = \frac{s_i}{s}$ adalah probabilitas dari sampel yang mempunyai *class* C_i. Misalkan atribut A mempunyai *v* nilai yang berbeda, $\{a_1, a_2, \ldots, a_v\}$. Atribut A dapat digunakan untuk mempartisi S menjadi *v* subset $\{S_1, S_2, \ldots, S_v\}$, dimana S_j berisi sampel pada S yang mempunyai nilai a_j dari A. Jika A terpilih menjadi atribut tes (yaitu, atribut terbaik untuk *splitting*), maka subset-subset akan berhubungan dengan pertumbuhan node-node cabang yang berisi S. Anggap s_{ij} sebagai jumlah *class* sampel C*i* pada subset S_j. *Entropy*, atau nilai *information* dari subset A adalah :

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj}) \qquad (2.2)$$

$\frac{s_{1j} + \ldots + s_{mj}}{s}$ adalah bobot dari subset ke *j* dan jumlah sampel pada subset (yang mempunyai nilai a_j dari A) dibagi dengan jumlah total sampel pada S. Untuk subset S_j,

$$I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} * \log_2(p_{ij}) \qquad (2.3)$$

Dimana $p_{ij} = \frac{s_{ij}}{|s_j|}$ adalah probabilitas sampel S_j yang mempunyai *class* C*i*. Maka nilai *information gain* atribut A pada subset S adalah

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \quad (2.4)$$

2.3.4 Penanganan atribut kontinyu

Algoritma C4.5 juga menangani masalah atribut kontinyu. Perhitungan atribut bertipe kontinyu berbeda dengan atribut bertipe diskrit, dimana atribut kontinyu memiliki nilai dengan *range* dari null sampai tak hingga. Pada proses klasifikasi semua atribut harus bernilai kategori, maka atribut bertipe kontinyu harus didiskritkan. Dalam permasalahan ini dibutuhkan suatu metode untuk mendiskritkan atribut kontinyu. Salah satu cara adalah dengan *entropy - based discretization* yang melibatkan penghitungan *class entropy*.

Misalkan T membagi S contohnya menjadi subset S1 dan S2. Misal ada *k class* C1, C2, …,Ck dan P (C*i*, S*j*) menjadi perbandingan dari contoh pada S*j* yang mempunyai *class i*. Maka *class entropy* dari subset S*j* didefinisikan dengan :

$$Ent(S) = -\sum_{i=1}^{k} P(Ci, S) \log_2 (P(Ci, S)) \quad (2.5)$$

Dan *class information entropy* E (A, T;S)

$$E(A, T;S) = \frac{|S1|}{|S|} Ent(S1) + \frac{|S2|}{|S|} Ent(S2) \quad (2.6)$$

Dimana; Ent(S*j*) = *class entropy* dari subset S*j*

S*j* = subset dari S

C*i* = *class i*

P(C*i*, S*j*) = perbandingan *instance* dari S*j* yang berada pada *class* C*i*

$E(A, T_A;S)$ = *class information entropy* partisi dengan *cut point* T_A di A

A = atribut

|S*k*| = jumlah *instance* di S*k*

Cut point yang terbaik adalah yang memberikan *class information entropy* yang paling kecil diantara semua kandidat *cut point*.

2.3.5 Mengubah *decision tree* menjadi *classification rules*

Hasil dari struktur *desicion tree* yang telah terbentuk diterjemahkan ke dalam bentuk *rule IF - THEN*. Penerjemahan satu *rule* berasal dari *root* sampai pembentukan node *leaf*. Masing-masing attribut memiliki nilai untuk menghubungkan *rule* ("*IF*" *part*). Pada suatu node *leaf* diberikan *class prediction*, untuk membentuk *rule* ("*THEN*" *part*). *IF - THEN rule* lebih mudah dipahami oleh manusia, terutama jika diberi *tree* yang sangat besar.

2.3.6 *Pruning tree*

Pruning tree adalah melakukan suatu kegiatan untuk mengganti suatu *subtree* dengan suatu node. Penggantian dilakukan jika *error rate* pada *subtree* lebih besar jika dibandingkan dengan node *single*.

Menurut Frank, 2005, pada algoritma C4.5 perkiraan *error* untuk satu node dapat dihitung dengan menggunakan rumus sebagai berikut:

$$e = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Bigg/ \left(1 + \frac{z^2}{N} \right) \qquad (2.7)$$

Jika c = 25% (asumsi untuk C4.5) maka z = 0,69 (dari distribusi binomial).

f : *error* pada data pembelajaran

N : jumlah *instance* pada satu *node*

Metode *pruning* yang digunakan pada algoritma *C4.5* adalah *rule post-pruning*. Pada metode ini proses *pruning* dilakukan setelah aturan klasifikasi dari pohon keputusan (*decision tree*) terbentuk. Berikut ini adalah tahapan dari *rule post-pruning* (Baltes,__) :

1. Bentuk pohon keputusan (*decision tree*) dari data *training* yang telah ditetapkan.
2. Ekstrak tiap-tiap *subtree* pada *decision tree* menjadi suatu aturan klasifikasi.

3. Hitung tingkat *error rate* dari tiap-tiap aturan klasifikasi yang terbentuk. Hapus duplikasi *rule* yang terbentuk setelah proses *pruning* berakhir.
4. Urutkan tiap-tiap aturan yang telah di-*pruning* dan sesuaikan dengan tingkat akurasinya. Kemudian gunakan aturan-aturan tersebut dalam proses klasifikasi selanjutnya.

2.4 Rekomendasi

Sistem rekomendasi merupakan suatu teknik pengumpulan informasi yang dimana sebagai suatu usaha untuk memberikan informasi yang tepat kepada *user*. Sistem rekomendasi dapat juga diartikan sebagai teknologi pemilihan informasi personal yang digunakan untuk memprediksi apakah seorang *user* tertentu akan tertarik dengan sesuatu (Han.dkk, 2006). Sebuah sistem rekomendasi dirancang untuk dapat memberikan sebuah masukan pada *user* berupa saran atau rekomendasi terhadap sesuatu. Dalam membangun sebuah sistem rekomendasi, terdapat dua metode pengumpulan informasi (*information filtering*) sebagai bahan rekomendasi, yaitu :

1) Dengan menganalisis 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* (Haruechaiyasak.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).

BAB III

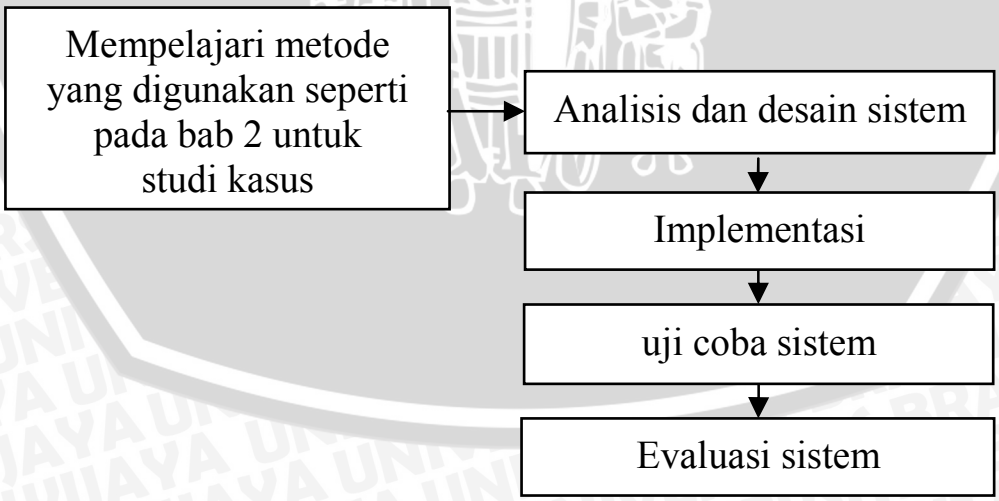
METODOLOGI DAN PERANCANGAN SISTEM

Pada bab metodologi dan perancangan sistem, akan dibahas langkah–langkah yang dilakukan dalam penelitian tentang perekomendasian HP (*handphone*) menggunakan algoritma C4.5.

Penelitian dilakukan dengan tahapan–tahapan berikut ini :

1. Mempelajari metode yang nantinya digunakan sebagai acuan untuk mendapatkan solusi dari studi kasus yang digunakan dalam penelitian ini.
2. Melakukan analisis dan desain sistem perekomendasian HP (*Handphone*) menggunakan algoritma C4.5.
3. Mengimplementasikan hasil analisis dan desain yang dilakukan pada tahap sebelumnya menjadi perangkat lunak perekomenendasian HP (*handphone*).
4. Langkah selanjutnya melakukan uji coba sistem dengan memasukkan data konsumen ke dalam sistem. Hasil yang diperoleh adalah perekomendasian HP (*handphone*).
5. Mengevaluasi hasil rekomendasi sistem dengan menggunakan data uji.

Langkah-langkah di atas dijelaskan kembali dalam bentuk alur, yang dapat dilihat pada gambar 3.1.



Gambar 3.1. Tahapan Penelitian

3.1 Analisis Sistem

Pada tahap ini akan dijelaskan tentang deskripsi umum sistem dan batasan sistem.

3.1.1 Deskripsi sistem

Dalam memberikan rekomendasi, sistem akan melakukan klasifikasi jenis HP. Pada penelitian ini, data transaksi penjualan *handphone* yang digunakan merupakan data fiktif yang di *generate*. Contoh klasifikasi dalam sistem ini misalnya HP sony ericsson S700i dan nokia 6111 termasuk dalam jenis HP *slider*, walaupun kedua HP tersebut memiliki jenis hp yang sama tetapi pada saat proses pembelajaran tidak dapat dilakukan secara bersama-sama karena proses pembelajaran dilakukan berdasarkan merk HP. Pada proses pembelajaran atau klasifikasi jenis HP, sistem akan membangun *decision tree* (pohon keputusan) dengan mengkalsifikasikan data transaksi penjualan. Jadi klasifikasi pada data pembelajaran berdasarkan jenis HP. Dalam proses pembelajaran akan menghasilkan rekomendasi berdasarkan jenis HP, yang dimana diperoleh dari hasil transaksi penjualan HP.

3.1.2 Batasan sistem

Batasan sistem yang akan dikembangkan, yaitu:

1. Sebelum dilakukan proses rekomendasi HP, harus dilakukan proses pembelajaran dengan membentuk aturan klasifikasi terhadap jenis HP terlebih dahulu.
2. Sistem hanya memberikan rekomendasi produk tanpa memperhatikan kondisi produk.

3.2 Perancangan Sistem

Pada tahap ini terdiri dari perancangan basis data, perancangan penelitian dan perancangan antarmuka.

3.2.1 Perancangan basis data

Pada penelitian ini untuk menghasilkan suatu rekomendasi HP, maka dibutuhkan data konsumen yaitu kd_konsumen, nama, umur, jenis kelamin, budget, jenis konsumen dan kebutuhan konsumen. Sedangkan data barang yang digunakan ialah kode barang, tipe HP

repository.ub.ac
UNIVERSITAS BRAWIJAYA

jenis HP, merk HP, harga dan keterangan. Dibutuhkan juga data transaksi penjualan, yang dimana diperoleh dari konsumen yang membeli suatu produk. Data ini yang nantinya digunakan sebagai data pembelajaran bagi sistem dalam memberikan rekomendasi pada *user*.

Tabel-tabel yang digunakan dalam penelitian ini ialah tabel konsumen (T_kons), tabel transaksi pembelian (T_TransPemb), tabel barang (T_HP), tabel detail *handphone* (T_detHP) dan tabel detail fitur (T_detFitur).

Tabel konsumen ini digunakan untuk menyimpan data konsumen, yang dapat dilihat pada tabel 3.1.

Tabel 3.1. Tabel Konsumen (T_kons)

Field Name	Data Type	Field Size
kd_kons	Char	7
Umur	Integer	4
JK	Char	2
Budget	VarChar	20
Jns_kons	VarChar	20
kebutuhan	VarChar	20

Keterangan :

kd_kons	:	kode konsumen
Umur	:	umur konsumen
JK	:	Jenis kelamin konsumen
Budget	:	anggaran konsumen
jns_kons	:	jenis konsumen
kebutuhan	:	kebutuhan konsumen

Tabel transaksi penjualan digunakan untuk menyimpan transaksi pembelian suatu produk (HP), yang dapat dilihat pada tabel 3.2.

Tabel 3.2. Tabel Transaksi (T_Transaksi)

Field Name	Data Type	Field Size
kd_trans	Char	5
Tanggal	Date/Time	9
kd_brg	Char	5
Kd_kons	Char	7

Keterangan :

kd_trans	:	kode transaksi pembelian
Tanggal	:	tanggal transaksi pembelian
kd_brg	:	kode barang (HP)
kd_kons	:	kode konsumen

Tabel barang digunakan untuk menyimpan data HP yang dapat dilihat pada tabel 3.3.

Tabel 3.3. Tabel Barang (T_HP)

Field Name	Data Type	Field Size
kd_brg	Char	7
tipe_HP	VarChar	20
Merk_HP	VarChar	15
Jenis_HP	VarChar	15
Harga	Money	8
Keterangan	VarChar	50

Keterangan :

kd_brg	:	Kode barang
tipe_HP	:	tipe *handphone*
Merk_HP	:	Merk *handphone*
Jenis_hp	:	Jenis *handphone*
Harga	:	Harga *handphone*
Keterangan	:	Keterangan

Tabel detail *handphone* digunakan untuk menyimpan data detail HP , yang yang dapat dilihat pada tabel 3.4.

Tabel 3.4. Tabel Detail Barang (T_DetBrg)

Field Name	Data Type	Field Size
kd_brg	Char	7
Kd_fitur	VarChar	20
Keterangan	VarChar	50
Picture	VarChar	50

Keterangan :

kd_brg	:	kode barang
kd_fitur	:	kode fitur HP
Keterangan	:	Keterangan fitur HP
Picture	:	Picture HP

Tabel detail fitur *handphone* digunakan untuk menyimpan data detail fitur HP , yang yang dapat dilihat pada tabel 3.5.

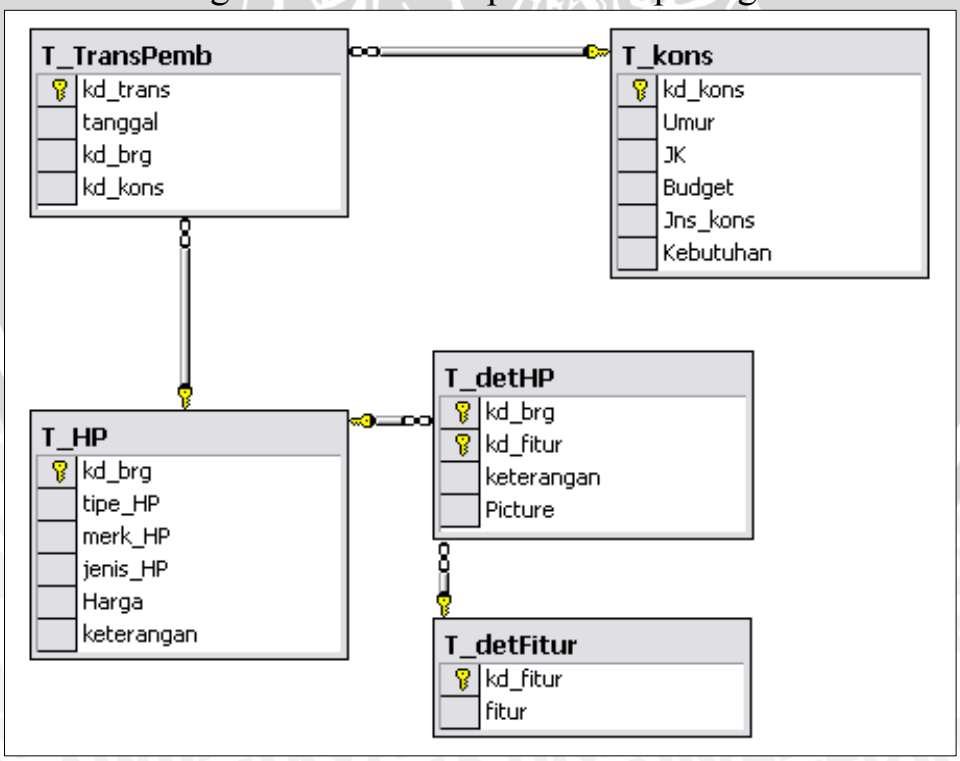
Tabel 3.5. Tabel Detail Fitur (T_detfitur)

Field Name	Data Type	Field Size
Kd_fitur	VarChar	20
Fitur	VarChar	20

Keterangan :

kd_fitur	:	kode fitur HP
Fitur	:	fitur HP

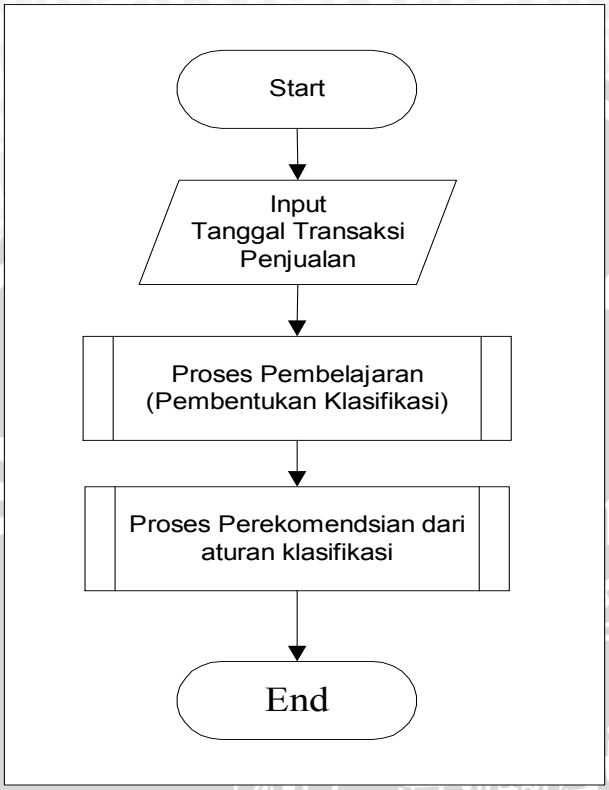
Model hubungan antar tabel dapat dilihat pada gambar 3.2.



Gambar 3.2. Hubungan Antar Tabel

3.2.2 Perancangan penelitian

Perancangan penelitian ini meliputi *input* data transaksi penjualan, proses pembelajaran dari data transaksi penjualan dan pembuatan sistem rekomendasi. Alur proses dapat dilihat pada gambar 3.3.



Gambar 3.3. Alur Proses

a. *Input* data transaksi penjualan

Input data transaksi penjualan telah dijelaskan di atas, data yang digunakan ialah data konsumen, data barang dan data transaksi penjualan. Dalam prosedur pengumpulan data telah diperoleh data konsumen dan data barang, kemudian data transaksi penjualan diperoleh dari data konsumen yang melakukan transaksi.

Proses input data penjualan ini adalah tahap awal *user* akan memasukan data konsumen kemudian produk yang dipilih atau dibeli oleh konsumen. Data transaksi penjualan ini digunakan untuk proses pembelajaran.

b. Proses pembelajaran data transaksi penjualan

Pada proses ini sistem membutuhkan masukkan berupa tanggal transaksi penjualan dan merk HP. Proses pembelajaran digunakan untuk perekomendasian HP.

c. Pembuatan sistem rekomendasi

Dalam proses penelitian ini, metode klasifikasi data yang digunakan ialah *decision tree*. Metode ini diterapkan untuk membuat pohon keputusan. *Output* yang dibutuhkan adalah hasil dari aturan klasifikasi jenis HP. Aturan klasifikasi ini yang nantinya digunakan untuk proses rekomendasi, yang berfungsi untuk pencarian jenis HP.

Tahap pertama yang dilakukan adalah memasukkan (*input*) berupa umur, jenis kelamin, jenis konsumen, budget dan kebutuhan. Setelah *user* menginputkan maka dilakukan proses pemilihan atribut yang sesuai dengan keperluan konsumen, dari inputan tersebut sistem akan menentukan jenis HP yang sesuai dari aturan klasifikasi tersebut.

Tahap kedua membuat daftar (*output*) yaitu berupa tipe HP dan jenis HP beserta detail HP yang sesuai dengan inputan *user*. Untuk lebih jelasnya dapat dilihat pada perhitungan manual, pada sub bab 3.4.

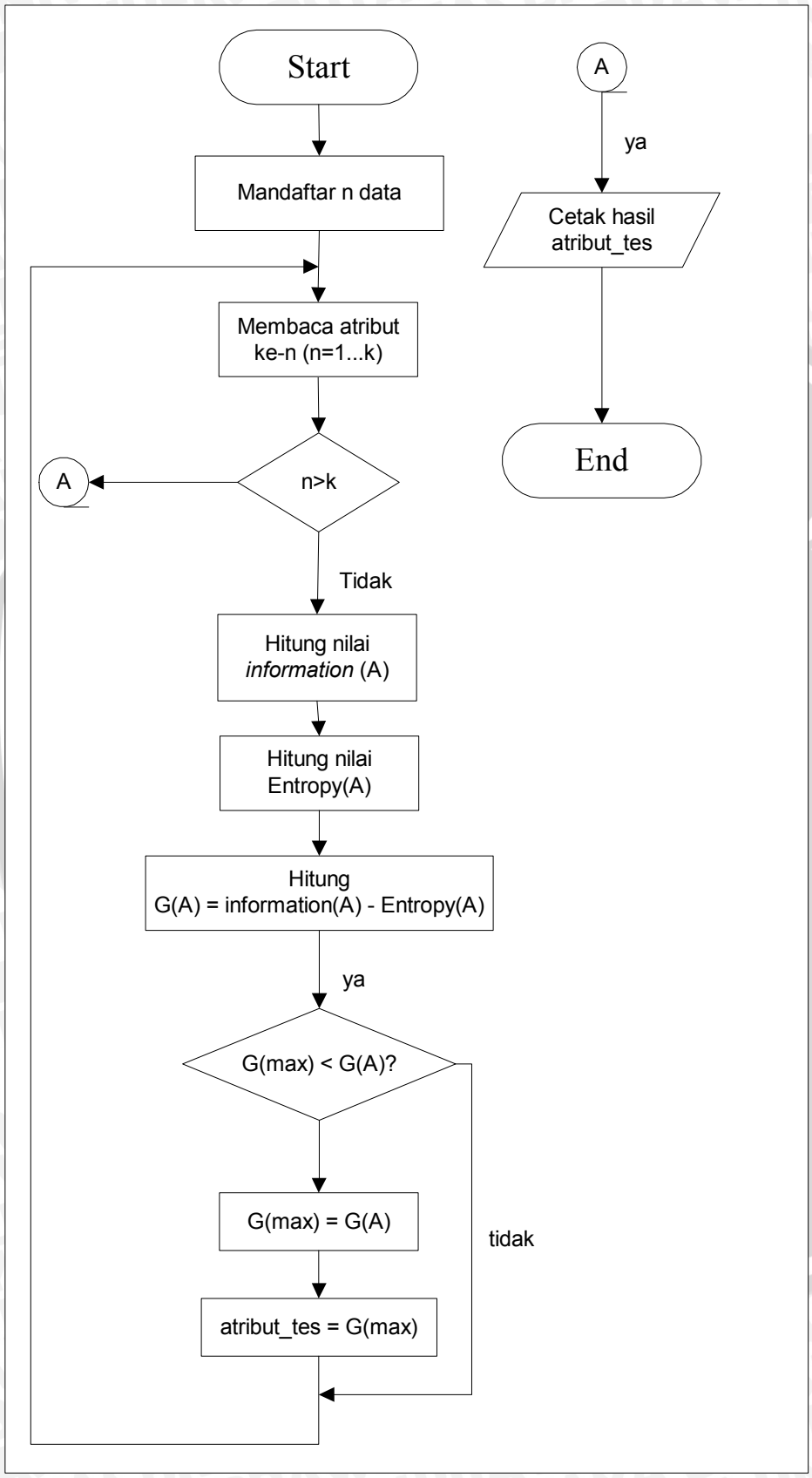
d. Proses Tes Data

Setelah pemberian rekomendasi maka dilakukan tes data yang bertujuan untuk mengetahui tingkat kebenaran dari aturan klasifikasi yang telah dihasilkan.

Dari proses rekomendasi di atas akan dijelaskan tahap perhitungan atribut terbaik dan aturan klasifikasi jenis HP dengan aliran data atau *flowchart*.

- Proses pemilihan atribut terbaik untuk pembentukan *tree,* yaitu :
  - Sistem akan mendaftar k atribut.
  - Menghitung probabilitas data pembelajaran.
  - Sistem akan menghitung informasi data pembelajaran.
  - Hitung *entropy* data pembelajaran tersebut.
  - Pilih *gain* terbesar dengan cara melakukan perbandingan antar *gain*, yang nantinya digunakan untuk atribut tes berikutnya.
  - Sistem akan menghitung terus sampai atribut habis.

Proses pemilihan atribut diatas akan dijelaskan dalam bentuk *flowchart,* yang dapat dilihat pada gambar 3.4.



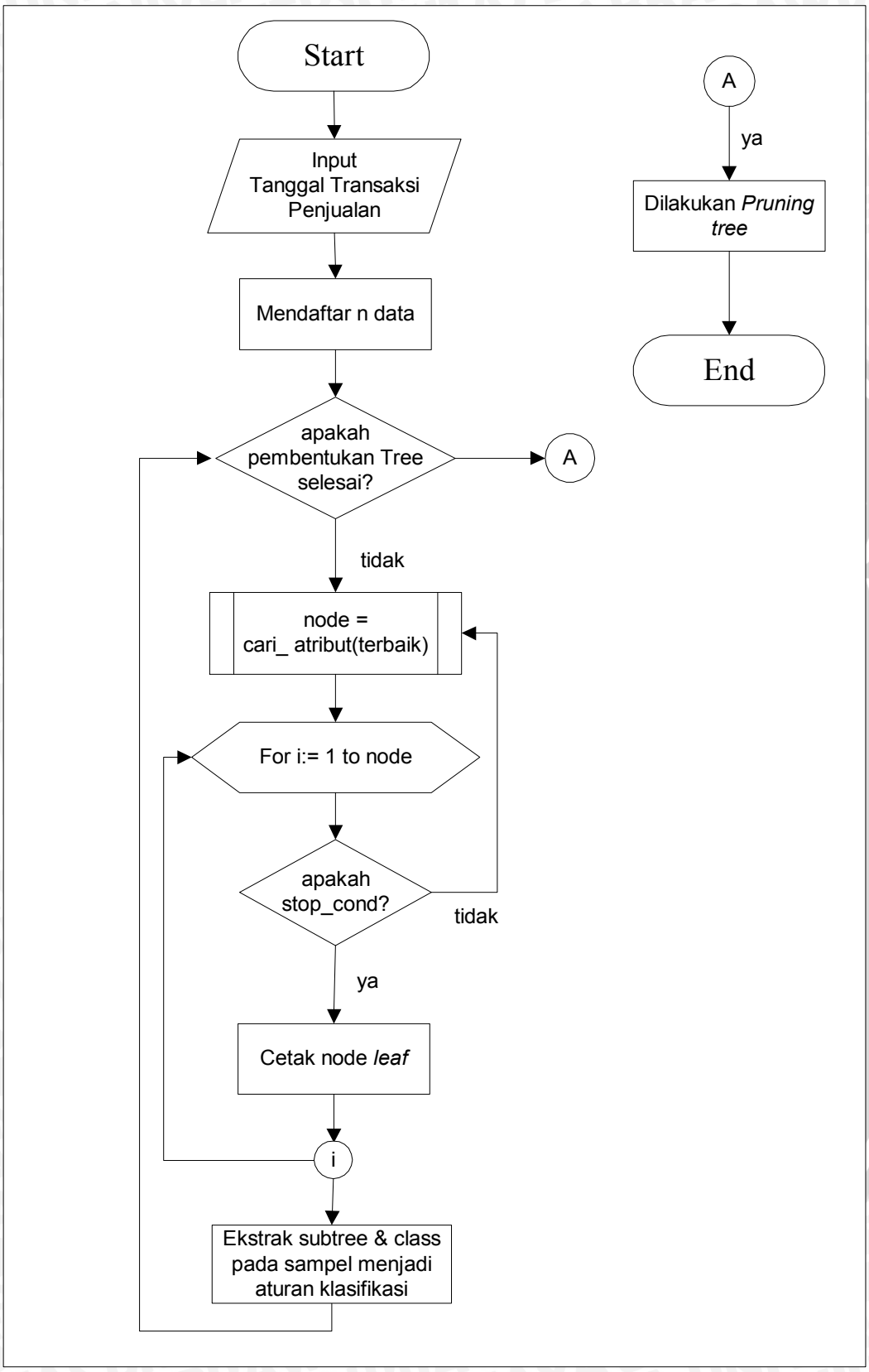
Gambar 3.4 Proses Cari Atribut Terbaik

- Proses pembentukan *tree* (pembelajaran), yaitu meliputi:
  - *User* meng*input*kan data konsumen dan produk yang dibeli, yang nantinya akan digunakan dalam proses pembelajaran.
  - Sistem akan mendaftar n data transaksi penjualan kemudian dilakukan pembelajaran.
  - Pembentukan *decision tree* dengan atribut–atribut yang di *input*kan dimana jenis HP sebagai kelasnya.
  - Proses mencari node dengan atribut terbaik berdasarkan nilai gain terbesar.
  - Menghitung jumlah data sampel dari tiap node yang terpilih, jika data sampel pada setiap node mempunyai kelas yang sama maka sistem langsung membentuk suatu node. Apabila tidak, sistem akan memanggil untuk pembentukan *decision tree.*
  - Sistem akan terus memanggil fungsi rekursif pembentukan pohon sampai seluruh atribut telah selesai diseleksi.
  - Dari pembentukan node tersebut, sistem akan mengekstrak menjadi suatu atruran klasifikasi, yang kemudian disimpan ke dalam database.
  - Setelah pembentukan tree maka dilakukan proses *pruning* untuk mengurangi *error rate* pada *subtree* yang terbentuk berdasarkan hasil klasifikasi.

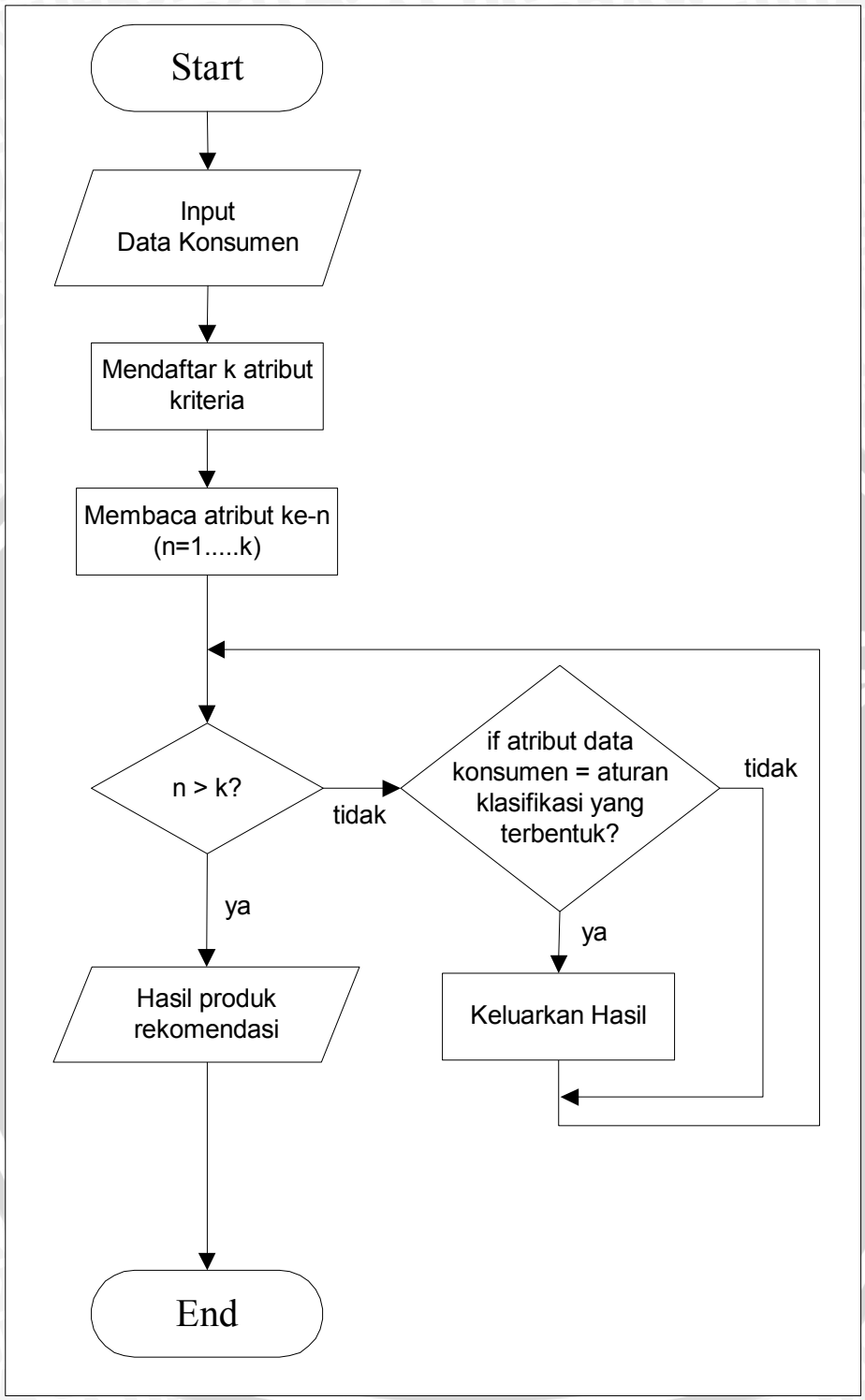
Proses pembentukan klasifikasi diatas akan dijelaskan dalam bentuk *flowchart,* yang dapat dilihat pada gambar 3.5.

- Proses rekomendasi produk

  Dalam proses ini yang dilakukan adalah merekomendasikan jenis HP dan tipe HP beserta informasi harga HP yang diiginkan konsumen dari kriteria yang diinputkan *user*. Tahapan-tahapan prosesnya adalah sebagai berikut :
  - *User* menginputkan data konsumen.
  - Sistem akan melakukan proses pencarian jenis HP, rekomendasi berdasarkan kriteria yang diinputkan.
  - Sistem akan merekomendasikan jenis HP dengan membaca aturan klasifikasi yang telah terbentuk.
  - Proses akan dilakukan sampai didapatkan beberapa produk sesuai input *user* .



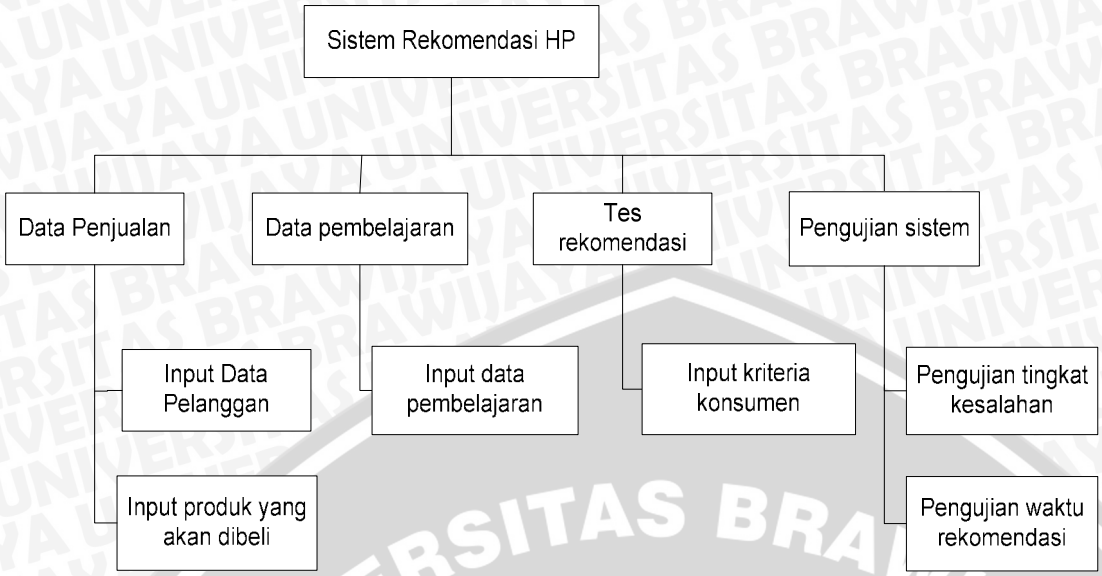
Gambar 3.5 Proses pembentukan *Tree* (Pembelajaran)



Gambar 3.6. Proses Perekomendasian HP

3.2.3 Perancangan antarmuka

Berdasarkan analisis yang telah dilakukan, berikut ini merupakan diagram aplikasi sistem yang akan dibangun, yang terlihat pada gambar 3.7.



Gambar 3.7. Diagram Aplikasi Sistem

Aplikasi sistem akan terdiri dari empat menu utama, diantaranya menu data transaksi penjualan, data pembelajaran, tes rekomendasi, menu pengujian sistem. Berikut merupakan penjelasan dari masing-masing menu utama tersebut :

- Data transaksi penjualan
  Menu ini dirancang untuk memproses seluruh input data penjualan maupun data barang yang telah diupdate untuk digunakan sebagai data pembelajaran.
- Data pembelajaran
  Menu ini dirancang untuk memproses seluruh input data transaksi penjualan HP, yang nantinya digunakan untuk proses pembelajaran (pembentukan tree).
- Tes rekomendasi
  Menu ini dirancang untuk memberikan rekomendasi suatu produk berdasarkan data yang di*input*kan oleh *user*. Data *input*an tersebut berupa data konsumen.
- Pengujian Sistem
  Pada menu ini mencakup dua pengujian yaitu pengujian tingkat kebenaran sistem pada saat pemberian rekomendasi dan pegujian *running time* rekomendasi. Pengujian tingkat kebenaran dilakukan untuk mengukur tingkat kebenaran/ akurasi. Sedangkan pengujian *running time* rekomendasi dilakukan untuk mengukur waktu yang diperlukan sistem dalam memberikan

rekomendasi pada sejumlah data tes dengan jumlah data pembelajaran yang berbeda-beda.

3.3 Perancangan Uji Coba

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

3.3.1 Data yang digunakan dalam penelitian

Data yang digunakan dalam penelitian ini adalah data barang (HP) dan data konsumen. Data barang digunakan untuk proses klasifikasi, sedangkan data konsumen digunakan sebagai acuan dalam proses rekomendasinya. Proses pengolahan datanya telah dijelaskan pada subbab-subbab sebelumnya. Data barang dan data konsumen merupakan hasil dari proses transaksi penjualan, yang dimana konsumen membeli suatu produk.

3.3.2 Pengujian sistem

Proses pengujian ini akan terdiri dari dua bagian, yaitu pengukuran tingkat kebenaran sistem dan pengujian *running time* rekomendasi sistem.

Tabel 3.6 Tipe Atribut Dan Kategori Pada Kriteria Pembeli HP

No	Daftar Atribut	Tipe Atribut	Kategori
1	Jenis Kelamin	Diskrit	L
			P
2	Jenis Konsumen	Diskrit	Siswa
			Mahasiswa
			Pegawai(Swasta)
			Pegawai(Wiraswasta)
			Pegawai(PNS)
3	Umur	Kontinyu	Tak hingga
4	Budget	Kontinyu	Tak hingga
5	Kebutuhan	Diskrit	Game
			Internet
			Kamera
			Musik
			Sms&Telephone

3.3.3 Pengujian pada saat rekomendasi sistem

Pengujian tingkat kebenaran dilakukan untuk memperoleh data mengenai ukuran ketepatan sistem rekomendasi dalam memberikan rekomendasi HP kepada *user*. Sebuah rekomendasi dianggap gagal atau salah, jika produk yang direkomendasikan tidak sesuai dengan kriteria yang diinputkan.

Tingkat kebenaran yang diukur dinyatakan dalam persen (%). Perhitungannya dapat dinyatakan sebagai berikut:

$$akurasi = \frac{jumlah_benar}{jumlah_datates} \times 100\%$$

Pengujian tingkat kebenaran ini menggunakan data penjualan HP yang telah ditetapkan sebelumnya.

Proses pembelajaran akan dilakukan sebanyak 4 kali dengan jumlah data pembelajaran yang telah ditentukan, dapat dilihat pada tabel 3.7. Data pembelajaran tersebut yang nantinya digunakan untuk untuk mencatat hasil dalam proses pengujian tingkat kebenaran, rancangan tabel ini dapat dilihat pada tabel 3.8.

Tabel 3.7 Rancangan Tabel Data Pembelajaran

Tanggal Transaksi	Data Pembelajaran ke-	Jumlah Data Pembelajaran
1/1/2005-7/1/2005	Data [1]	520
1/1/2005-1/1/2006	Data [2]	1012
1/1/2005-8/1/2006	Data [3]	1512
1/1/2005-12/28/2006	Data [4]	2000

Keterangan:

Tanggal Transaksi Penjualan	:	Rentang tanggal transaksi penjualan untuk data pembelajaran.
Data Pembelajaran ke-	:	Data yang digunakan untuk pembelajaran.
Jumlah Data Pembelajaran	:	Jumlah data transaksi penjualan.

Pada tabel 3.8 merupakan rancangan tabel yang digunakan untuk pengujian rekomendasi sistem pada sejumlah data tes dengan jumlah data pembelajaran yang berbeda-beda.

Tabel 3.8 Rancangan Tabel Pengujian Rekomendasi Sistem

Data Pembelajaran ke-	Jumlah Aturan (Rule)	Jumlah Data Tes	Jumlah benar	Akurasi (%)

Keterangan:

Data Pembelajaran ke-	:	Data yang digunakan untuk pembelajaran.
Jumlah Aturan (Rule)	:	Jumlah aturan yang dihasilkan dari pembentukan *tree*
Jumlah Data Tes	:	Jumlah data yang diujikan
Jumlah Benar	:	Jumlah data benar dalam proses pengujian
Akurasi(%)	:	$akurasi = \frac{jumlah_benar}{jumlah_datates} \times 100\%$

3.3.4 Pengujian *running time* rekomendasi sistem

Pada tabel 3.9 merupakan rancangan tabel yang digunakan untuk pengujian waktu. Pengujian *running time* ini dilakukan untuk mendapatkan catatan waktu yang dibutuhkan dalam menghasilkan sebuah rekomendasi HP.

Tabel 3.9 Rancangan Tabel Pengujian Rekomendasi Sistem

Data Pembelajaran ke-	Jumlah Data Tes	*Running time*

Keterangan:

Data Pembelajaran ke-	:	Data yang digunakan untuk pembelajaran.

Jumlah Data Tes	:	Jumlah data yang diujikan
running time	:	Waktu yang diperlukan oleh tiap-tiap metode dalam memberikan rekomendasi dari sejumlah data tes.

3.4 Perhitungan Manual

Pada penelitian ini studi kasus yang digunakan adalah perekomendasian *hanphone.* Contoh data yang digunakan dapat dilihat pada Tabel 3.10.

Untuk lebih jelasnya perhatikan langkah-langkah berikut ini :

Proses pembentukan *decision tree* dengan algoritma C4.5

Sebelum dilakukan proses perhitungan probabilitas *atribut_list* (kategori) terhadap target *class,* terlebih dahulu dilakukan perhitungan *information* target *class,* yang dapat dilihat pada tabel 3.11.

Tabel 3.11. Probabilitas Jenis HP

Jenis HP	Probabilitas
Candybar	13/20
Slider	4/20
Flip	2/20
Komunikator	1/20

$$I(S_1, S_2, \ldots, S_m) = -\sum_{i=1}^{m} p_i * \log_2(p_i)$$

$$I(\text{Jenis HP}) = -\frac{13}{20}\log_2\frac{13}{20} - \frac{4}{20}\log_2\frac{4}{20} - \frac{2}{20}\log_2\frac{2}{20} - \frac{1}{20}\log_2\frac{1}{20}$$

$$= 1.417$$

UNIVERSITAS BRAWIJAYA repository.ub.ac

Tabel 3.10. Contoh Data Sampel

No	Nama	JK	Umur	Jns_kons	Budget	Kebutuhan	Tipe HP	Jenis HP
1	Abi	L	23	Mahasiswa	1.775.000	Musik	K630i	Candybar
2	Setia	P	25	Pekerja(Swasta)	1.900.000	Musik	W880i	Candybar
3	Gimin	L	30	Pekerja(PNS)	3.000.000	Internet	i450i	Slider
4	Dedy	L	27	Pekerja(PNS)	2.900.000	Internet	KF510	Slider
5	Vira	P	16	Siswa	2.000.000	Kamera	N70	Candybar
6	Lovy	P	20	Mahasiswa	560000	Sms&telephone	1650	Candybar
7	Dira	L	17	Siswa	1.825.000	Kamera	6300	Candybar
8	Bita	P	17	Siswa	1.000.000	Kamera	Z530i	Flip
9	Dewi	P	20	Mahasiswa	300000	Sms&telephone	J220i	Candybar
10	Bowo	L	22	Mahasiswa	350000	Sms&telephone	C140	Candybar
11	Lanang	L	18	Siswa	200000	Sms&telephone	3310	Candybar
12	Awan	L	21	Mahasiswa	2.725.000	Musik	N73	Candybar
13	Dito	L	30	Pekerja(Wiraswasta)	7.700.000	Internet	E90	Komunikator
14	Bima	L	22	Mahasiswa	1.200.000	Kamera	KE770	Candybar
15	Dina	P	16	Siswa	2.300.000	Kamera	U600	Slider
16	Ery	P	32	Pekerja(Swasta)	2.000.000	Kamera	N70	Candybar
17	Agusta	P	18	Siswa	1.000.000	Kamera	Z530i	Flip
18	Fira	P	18	Siswa	350000	Sms&telephone	C140	Candybar
19	Lina	P	18	Siswa	200000	Sms&telephone	3310	Candybar
20	Tiwi	P	22	Mahasiswa	2.900.000	Internet	KF510	Slider

- Perhitungan *information* dengan tipe atribut diskrit

Tahap 1. Sebelum menghitung nilai *entropy* jenis HP, terlebih dahulu dilakakukan perhitungan probabilitas setiap kategori dari suatu atribut. Perhitungan ini dijelaskan pada tabel 3.12 kemudian perhitungan antara kategori terhadap target *class*, dapat dilihat pada tabel3.13.

Tabel 3.12. Probabilitas Jenis Kelamin

JK	Probabilitas
L	9/20
P	11/20

Tabel 3.13. Probabilitas Jenis Kelamin Terhadap Jenis Hp

JK	Candybar	Slider	Flip	Komunikator
L	6/9	2/9	0/9	1/9
P	7/11	2/11	2/11	0/11

Entropy untuk atribut Jenis Kelamin adalah;

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj})$$

$$E(JK) = \frac{9}{20}\left(-\frac{6}{9}\log_2(\frac{6}{9}) - \frac{2}{9}\log_2(\frac{2}{9}) - \frac{0}{9}\log_2(\frac{0}{9}) - \frac{1}{9}\log_2(\frac{1}{9})\right) +$$

$$\frac{11}{20}\left(-\frac{7}{11}\log_2(\frac{7}{11}) - \frac{2}{11}\log_2(\frac{2}{11}) - \frac{2}{11}\log_2(\frac{2}{11}) - \frac{0}{11}\log_2(\frac{0}{11})\right)$$

$$= 1.271091$$

Tahap 2. Menghitung nilai *gain*, yang natinya akan digunakan untuk pemilihan atibut tebaik. Rumus perhitungan yang digunakan adalah sebagai berikut :

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

Dimana;

$I(s_1, s_2, \ldots, s_m)$ = *information* data sampel pada *class* C*i*

E(A) = *entropy* dari subset S*j*

$$G(JK) = I(S_1, S_2, S_3, S_4) - E(JK)$$
$$= 1.417 - 1.271091$$
$$= 0.145552$$

- Perhitungan *information* dengan tipe atribut kontinyu.

Metode yang digunakan dalam perhitungan ini adalah *Entropy-Based Discretization*. Perhitungan yang digunakan adalah sebagai berikut :

$$E(A, T;S) = \frac{|S1|}{|S|} Ent(S1) + \frac{|S2|}{|S|} Ent(S2)$$

Tahap 1. Dari rumus di atas maka dapat diketahui nilai *class information entropy* dari masing-masing *split point*nya. Contoh perhitungan dijelaskan pada tabel 3.14 dan tabel 3.15.

Tabel 3.14. Probabilitas Umur

Umur		Probabilitas
<	20	8/20
>=		12/20
<	25	15/20
>=		5/20
<	30	17/20
>=		3/20

Tabel 3.15. Probabilitas Umur Terhadap Jenis Hp

Umur		Candybar	Slider	Flip	Komunikator	Entropy
<	20	5/8	1/8	2/8	0/8	1.232751
>=		8/12	3/12	0/12	1/12	
<	25	11/15	2/15	2/15	0/15	1.207963
>=		2/5	2/5	0/5	1/5	
<	30	12/17	3/17	2/17	0/17	1.223366
>=		1/3	1/3	0/3	1/3	

E(<25, >=25) =

$$\frac{15}{20}\left(-\frac{11}{15}\log_2(\frac{11}{15})-\frac{2}{15}\log_2(\frac{2}{15})-\frac{2}{15}\log_2(\frac{2}{15})-\frac{0}{15}\log_2(\frac{0}{15})\right)+$$

$$\frac{5}{20}\left(-\frac{2}{5}\log_2(\frac{2}{5})-\frac{2}{5}\log_2(\frac{2}{5})-\frac{0}{5}\log_2(\frac{0}{5})-\frac{1}{5}\log_2(\frac{1}{5})\right)$$

= 1.207963

Dengan menggunakan cara yang sama, maka tiap-tiap *split point* dapat dihitung nilai *class information entropy*nya. *Cut point* (titik percabangan) yang terbaik adalah yang memberikan *class information entropy* yang paling kecil diantara semua kandidat *cut point*. Diantara semua kandidat *split point* di atas, nilai *class information entropy* yang paling kecil adalah *split point* 25 dengan nilai *Ent*(<25, >=25) = 1.207963

Tahap 2. Menghitung nilai *gain*, yang nantinya akan digunakan untuk pemilihan atribut tebaik.

G(Umur) = I (S_1, S_2,S_3,S_4) – E(Umur)

= 1.417- 1.207963

= 0.20868

Dengan menggunakan cara yang sama, perhitungan *Gain* semua atribut dapat dicari.

G (Jns_kons) = 0.690039

G (Budget) = 0.464373

G (kebutuhan) = 0.771812

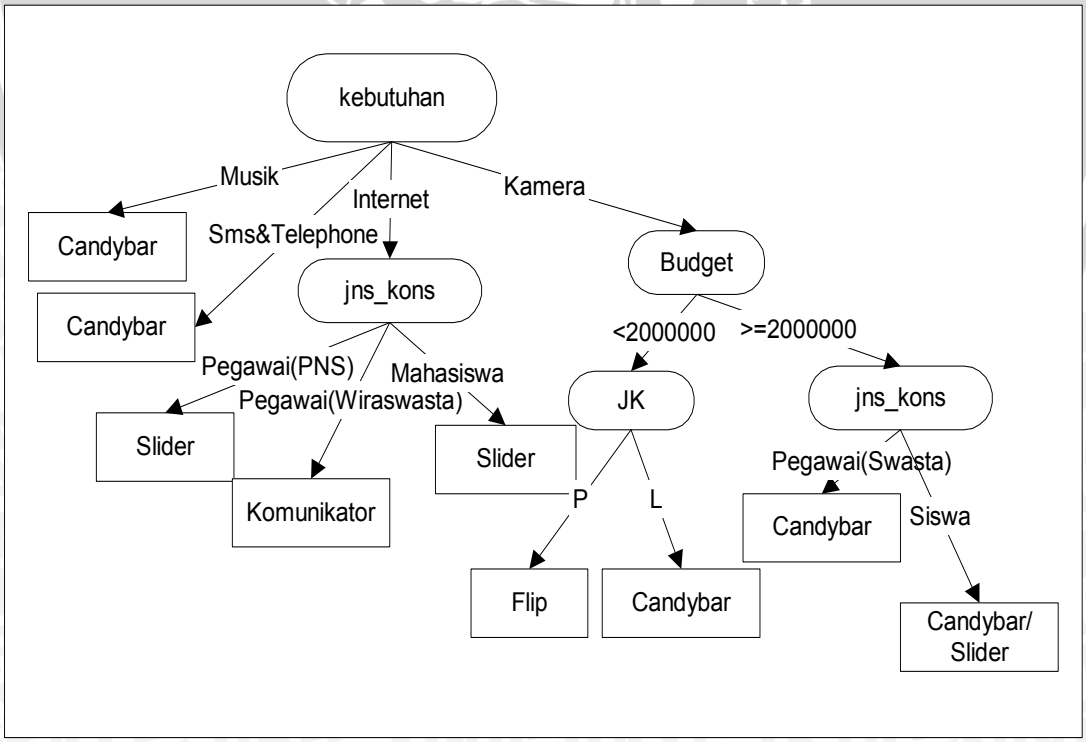
Setelah nilai *information gain* pada semua atribut dihitung, maka atribut yang mempunyai nilai *information gain* terbesar yang dipilih menjadi atribut tes. *Gain* terbesar adalah *gain* kebutuhan, yang dimana nantinya digunakan untuk atribut tes.

Dari hasil perhitungan diatas maka diperoleh suatu bentuk *decision tree* (pohon keputusan), dapat dilihat pada gambar 3.8.

Proses ekstraksi *decision tree* ke dalam *rule*

Setelah terbentuknya *decision tree* proses selanjutnya adalah mengekstrak *tree* ke dalam aturan klasifikasi. Maka aturan klasifikasi yang terbentuk adalah sebagai berikut :

1. **IF** kebutuhan = ’Musik’ **THEN** jenis HP = ’Candybar’
2. **IF** kebutuhan = ’Sms&telephone’ **THEN** jenis HP = ’Candybar’
3. **IF** kebutuhan = ’Internet’ ∧ Jns_kons = ’Pegawai(PNS)’ **THEN** jenis HP = ’Slider’
4. **IF** kebutuhan = ’Internet’ ∧ Jns_kons = ’Pegawai(Wiraswasta)’ **THEN** jenis HP = ’Komunikator’
5. **IF** kebutuhan = ’Internet’ ∧ Jns_kons = ’Mahasiswa’ **THEN** jenis HP = ’Slider’
6. **IF** kebutuhan = ’kamera’ ∧ budget < ’2jt’ ∧ JK = ’ P’ **THEN** jenis HP = ’Flip’
7. **IF** kebutuhan = ’kamera’ ∧ budget <’2jt’ ∧ JK = < ’L’ **THEN** jenis HP = ’Candybar’
8. **IF** kebutuhan = ’kamera’ ∧ budget >=’2jt’ ∧ Jns_kons = ’Pegawai(Swasta)’ ∧ **THEN** jenis HP = Candybar
9. **IF** kebutuhan = ’kamera’ ∧ budget >=’2jt’ ∧ Jns_kons = ’Siswa’ ∧ **THEN** HP = Candybar ∨ Slider



Gambar 3.8. *Decision Tree*

Proses rekomendasi

Berikut ini merupakan contoh data baru yang akan diberikan rekomendasi terhadap suatu produk HP. Berdasarkan data di bawah ini produk apa yang paling direkomendasikan?

Tabel 3.16 Contoh data yang akan dicari produk HP yang sesuai

No	Nama	JK	Umur	Jns_kons	Budget	Kebutuhan	Tipe HP	Jenis HP
1	Oncy	L	23	M	1.800.000	musik	?	?

Dari data di atas akan direkomendasikan sebuah produk dengan membaca aturan klasifikasi yang telah terbentuk berdasarkan jenis HP. Hasil rekomendasinya yaitu *rule* no.1.

IF kebutuhan = ”Musik” **THEN** jenis HP = Candybar

Kemudian langkah selanjutnya adalah mencari jenis HP dan tipe HP yang termasuk dalam aturan klasifikasi jenis HP. Pencarian ini dilakukan dengan cara membaca data hasil rekomendasi pada *rule* no.1.

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dijelaskan seluruh proses yang sudah dirancang pada bab sebelumnya, tampilan antarmuka dan bagian-bagian *source code* yang dibuat, serta analisis terhadap data yang dihasilkan sistem.

4.1 Lingkungan Implementasi

Implementasi merupakan proses transformasi representasi rancangan ke dalam bahasa pemrograman yang dapat dimengerti oleh komputer. Pada bab ini, lingkungan implementasi yang akan dijelaskan meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan perangkat keras

Perangkat keras yang digunakan dalam pengembangan sistem rekomendasi pembelian *handphone* ini adalah sebuah PC*(Personal Computer)* dengan spesifikasi sebagai berikut :

1. *Processor* Genuine Intel® 1.60 GHz
2. 512 MB RAM
3. 80 GB HDD
4. Monitor 12”
5. Keyboard
6. Mouse

Perangkat keras ini akan difungsikan sebagai tempat perangkat lunak untuk penelitian dijalankan.

4.1.2 Lingkungan perangkat lunak

Perangkat lunak yang digunakan dalam pengembangan sistem rekomendasi pembelian *handphone* ini adalah :

1. Sistem operasi *Microsoft Windows XP Professional Edition Service Pack 2* sebagai tempat aplikasi dijalankan.
2. *Borland Delphi 7.0* sebagai *software development* dalam mengembangkan aplikasi dan *Database Management System* (DBMS) SQL Server untuk menyimpan data.

4.2 Implementasi Program

Berdasarkan analisis dan perancangan sistem diatas, proses yang terdapat pada Sub bab 3.2 maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut. Dalam rekomendasi penjualan *handphone* terdapat dua tahap yaitu tahap pembelajaran dan tes data (pemberian rekomendasi dari aturan klasifikasi yang telah terbentuk). Tahap pembelajaran digunakan untuk proses pembentukan *tree* kemudian mengekstrak *tree* aturan klasifikasi, dan proses selanjutnya adalah *pruning tree*. Pada tahap tes data yang digunakan untuk merekomendasikan sebuah produk.

4.2.1 Struktur data

Struktur data yang diterapkan dalam aplikasi ini dibagi dalam 2 bagian, yaitu struktur data cari atribut terbaik untuk pembentukan *tree* direpresentasikan pada gambar 4.1, diterangkan pada tabel 4.1. Sedangkan struktur data *pruning* direpresentasikan pada gambar 4.2 dan diterangkan pada tabel 4.2.

```
TSplit = array [0..2, 0..1] of string;

const
  splitUmur : TSplit =(('19', '20'),('24', '25'),
    ('29', '30'));

const
  splitBudget : TSplit =(('999999', '1000000'),
    ('1999999', '2000000'),('2999999', '3000000'));

diskritGeneralType = record
    kategori : string;
    jumlah  : integer;
    probability : real;

end;

arrDiskrit = array of diskritGeneralType;
arrKontinyu = array of array of diskritGeneralType;

probType = record
    bufS : string;
    targetS : string;
    buf : integer;
    target : integer;
```

```
    probability : real
end;

arrProbDiskrit = array of array of probType;
arrProbKontinyu = array of array of array of probType;

TInformation = record
    name : string;
    usable : boolean;
    arrData : arrProbDiskrit;
    entropy : real;
    gain : real;
end;

arrInfo = array of TInformation;

TRecData = record
    field : string;
    data : array of string;
end;

arrRec = array of TRecData;
```

Gambar 4.1 Struktur Data Cari Atribut Terbaik

Tabel 4.1 Keterangan Struktur Data Cari Atribut Terbaik

`TSplit`	Array `TSplit` dua dimensi untuk menyimpan setiap split atribut bertipe kontinyu.
`arrDiskrit`	Untuk menyimpan jumlah setiap kategori suatu atribut bertipe diskrit pada data pembelajaran.
`arrKontinyu`	Untuk menyimpan jumlah setiap kategori suatu atribut bertipe kontinyu pada data pembelajaran.
`diskritGeneralType`	Merupakan tipe data *record* yang digunakan untuk menyimpan data pembelajaran. `diskritGeneralType` berisi tiga *field*, yaitu `Kategori` yang bertipe string dan `jumlah` yang bertipe integer serta `probability` yang bertipe real.
`Kategori`	*Field* bertipe string yang digunakan untuk menyi mpan nama kategori dari setiap atribut.

`Jumlah`	*Field* bertipe integer yang digunakan untuk menyimpan jumlah suatu kategori dari suatu atribut.
`Probability`	*Field* bertipe real yang digunakan untuk menyimpan nilai kategori terhadap keseluruhan data pembelajaran.
`probType`	Merupakan tipe data *record* yang digunakan untuk menyimpan data yang nantinya digunakan untuk proses perhitungan probabilitas kategori terhadap target *class*. `probType` berisi 2 *field* yang bertipe string yaitu `bufS` dan `targetS`, serta 2 *field* yang bertipe integer yaitu `buf` dan `target` kemudian *field* `probability` yang bertipe real.
`bufS`	*Field* yang digunakan untuk menyimpan nama kategori dari suatu atibut.
`Targets`	*Field* yang digunakan untuk menyimpan nama target *class*.
`Buf`	*Field* yang digunakan untuk menyimpan jumlah suatu kategori terhadap target *class*.
`Target`	*Field* yang digunakan untuk menyimpan jumlah suatu kategori terhadap target *class*.
`Probability`	*Field* bertipe real yang digunakan untuk menyimpan nilai kategori terhadap target *class*.
`arrProbDiskrit`	Array `arrProbDiskrit` dinamis berdimensi dua untuk menyimpan probabilitas antara kategori atribut bertipe diskrit dengan target *class*.
`arrProbKontinyu`	Array `arrProbKontinyu` dinamis berdimensi tiga untuk menyimpan probabilitas antara kategori atribut bertipe kontinyu dengan target *class*.
`Tinformation`	Merupakan tipe data *record* yang digunakan untuk menyimpan data yang nantinya digunkan untuk pencarian *information gain* tertinggi. `probType` berisi *field* `name` yang bertipe string, *field* `usable` yang bertipe

	boolean, *field* `arrData` yang bertipe arrProbDiskrit, dan 2 *field* yang bertipe real yaitu `entropy` dan `gain`.
`Name`	*Field* yang digunakan untuk menyimpan nama atribut.
`Usable`	Merupakan suatu variable yang mempunyai nilai balik berupa *boolean* `True` atau `False`
`arrData`	*Field* yang digunakan untuk menyimpan nilai probabilitas suatu atibut .
`Entropy`	*Field* yang digunakan untuk menyimpan nilai *entropy* suatu atibut.
`Gain`	*Field* yang digunakan untuk menyimpan nilai *gain* suatu atibut.
`arrInfo`	Array `arrRec` dinamis berdimensi satu untuk menyimpan *information* pada `setipa atribut.`
`TRecData`	Merupakan tipe data *record* yang berisi *field* `Field` betripe string serta `Data bertipe array.`
`arrRec`	Array `arrRec` dinamis berdimensi satu untuk menyimpan keseluruhan data.

```
TPruning = record
    KlaxMaxStr: string;
    klasMax,    Count,    slider,    flip,    candybar,
       komunikator : integer;
    e:real;
end;

arrPrun = array of TPruning;
```

Gambar 4.2 Struktur Data *Pruning*

Tabel 4.2 Keterangan Struktur Data *Pruning*

`TPruning`	Merupakan tipe data *record* yang digunakan untuk menyimpan data jenis_HP beserta jumlahnya. `TPruning` berisi 2 *field* yang bertipe string, yaitu `KlaxMaxStr` dan `jumlah` serta 6 *field* yang bertipe integer yaitu `klasMax, Count, slider, flip, candybar, dan komunikator.` Kemudian

	field yang terakhir adalah `e` bertipe real.
`arrPrun`	Array `arrRec` dinamis berdimensi satu untuk menyimpan hasil *rule* dari proses *pruning*.

Pola pembelajaran merupakan pasangan data yaitu antara atribut_*list* (kategori) dan target *class*. *Atribut* yang digunakan dalam proses pembelajaran yaitu yaitu jenis kelamin, umur, jenis konsumen, budget dan kebutuhan. Data ini merupakan hasil inputan dari *user*, hasil inputan ini disimpan dalam suatu variabel yang dapat dilihat pada tabel 4.3.

Tabel 4.3 Deskripsi Variabel Pada `prosedure buatTree`

Nama variable	Deskripsi
`dataJenisHP`	Merupakan tipe data `arrDiskrit` yang digunakan untuk menyimpan data jenis HP, data ini merupakan target *class*.
`dataJk`	Merupakan tipe data `arrDiskrit` yang digunakan untuk menyimpan data jenis kelamin.
`dataJenisKons`	Merupakan tipe data `arrDiskrit` yang digunakan untuk menyimpan data jenis konsumen.
`dataKebutuhan`	Merupakan tipe data `arrDiskrit` yang digunakan untuk menyimpan data kebutuhan
`dataUmur`	Merupakan tipe data `arrKontinyu` yang digunakan untuk menyimpan data umur.
`dataBudget`	Merupakan tipe data `arrKontinyu` yang digunakan untuk menyimpan data budget yang bertipe kontinyu
`probJk`	Merupakan tipe data `arrProbDiskrit` yang digunakan untuk menghitung probabilitas jenis kelamin
`probJenisKons`	Merupakan tipe data `arrProbDiskrit` yang digunakan untuk menghitung probabilitas umur
`probKebutuhan`	Merupakan tipe data `arrProbDiskrit`

	yang digunakan untuk menghitung probabilitas kebutuhan
`probUmur`	Merupakan tipe data `arrProbKontinyu` yang digunakan untuk menghitung probabilitas jenis konsumen
`probBudget`	Merupakan tipe data `arrProbKontinyu` yang digunakan untuk menghitung probabilitas budget
`Info`	Digunakan untuk menampung nilai informasi dari setiap atribut
`dataInfo`	Digunakan untuk menampung data informasi dari setiap atribut

4.2.2 Implementasi proses pembelajaran

Langkah pertama adalah *user* memasukkan tanggal transaksi penjualan HP yang akan dipakai untuk proses pembelajaran. Kemudian memilih merk produk untuk klasifikasi berdasarkan merk produk. Pengambilan data ini mengunakan ADOQuery yang nantinya dipakai untuk perhitungan jumlah sampel, contoh penggunaannya dapat dilihat pada gambar 4.3.

```
  s := 'select * from vw_klasifikasi where tanggal
between'+QuotedStr(DateToStr(DTPAwal.Date))+'and
'+QuotedStr(DateToStr(DTPAkhir.Date))+' and merk_HP ='
+QuotedStr(CBMerk.Text)+'and ' + strQuery[0];
  DBNavigator1.DataSource.DataSet.First;
  DM.QHP.close;
  DM.QHP.SQL.Text := s;
  DM.QHP.open;
```

Gambar 4.3 Pengambilan Data Dari Database

Setelah proses pengambilan data pembelajaran, sistem akan menyimpan data pembelajaran yang terpilih disimpan ke dalam suatu struktur data berupa *array*. Selanjutnya data pembelajaran tersebut akan diproses untuk pembentukan sebuah *tree*.

Langkah selanjutnya adalah proses pembentukan *tree* (*decision tree),* yang dibangun secara rekursif hingga tiap bagian pada node terdiri dari beberapa data yang berasal dari kelas yang sama. Pada

Pada proses pembangunan *tree*, terdapat dua tahap yaitu : pemilihan atribut terbaik dan pembentukan *tree*.

1. Pemilihan atribut terbaik

Sebelum dilakukan perhitungan atribut terbaik, terlebih dahulu dilakukan mendaftar k atribut beserta atribut_*list* (kategori), perhitungan jumlah atribut_*list* setelah itu dilakukan perhitungan probabilitas atribut_*list* terhadap target *class*. Untuk memilih atribut terbaik, dilakukan beberapa perhitungan informasi distribusi kelas terhadap atribut_*list*. Perhitungan-perhitungan yang dilakukan bertujuan untuk mendapatkan nilai gain tertinggi dari beberapa atribut, yang nantinya akan menentukan atribut mana yang dijadikan atribut tes.

a. Mendaftar setiap atribut_*list* (kategori)

Proses ini bertujuan untuk menghitung jumlah kategori setiap atribut diskrit maupun atribut kontinyu, *sourcecode* dapat dilihat pada gambar 4.5.

```
function                cekDataDiskrit(arrBuf:arrDiskrit;
data:string) : arrDiskrit;
var
  i : integer;
begin
  i := 0;
  while i<=Length(arrBuf) do
  begin
    if arrBuf[i].kategori = '' then
    begin
      arrBuf[i].kategori := data;
      arrBuf[i].jumlah := 1;
      i := length(arrBuf) + 1;
    end
      ---------------
  end;
  result := arrBuf;
end;

function    cekDataKontinyu(arrBuf:arrKontinyu;    data:
string) : arrKontinyu;
var
  i : integer;
begin
  for i:= low(arrBuf) to high(arrBuf) do
  begin
```

```
    if(StrToInt(data) <= StrToInt(arrBuf[i,0].kategori))
    then
      ----------
  end;
  Result := arrBuf;
end;
```

Gambar 4.4 Proses Perhitungan Jumlah Kategori Atribut Diskrit Dan Atribut Kontinyu Pada Data Pembelajaran.

b. Perhitungan probabilitas setiap kategori dari suatu atribut.

Proses ini bertujuan untuk menghitung probabilitas setiap kategori dari suatu atribut yang bertipe diskrit dan kontinyu. *Sourcecode* dari proses perhitungan atribut bertipe diskrit, dapat dilihat pada gambar 4.5 dan proses perhitungan atribut bertipe kontinyu, dapat dilihat pada gambar 4.6.

```
function hitungProbDisk(arrBuf : arrDiskrit; total :
integer) : arrDiskrit;
var
  i : integer;
begin
  for i:= low(arrBuf) to high(arrBuf) do
  begin
    arrBuf[i].probability := arrBuf[i].jumlah / total;
  end;
  Result := arrBuf;
end;
```

Gambar 4.5 Proses Perhitungan Probabilitas Atribut Bertipe Diskrit

```
function hitungProbKontinyu(arrBuf : arrKontinyu;
total : integer) : arrKontinyu;
var
  i : integer;
begin
  for i := low(arrBuf) to high(arrBuf) do
  begin
    arrBuf[i,0].probability := arrBuf[i,0].jumlah /
total;
    arrBuf[i,1].probability := arrBuf[i,1].jumlah /
total;
  end;
  Result := arrBuf
end;
```

Gambar 4.6 Proses Perhitungan Probabilitas Atribut Bertipe Kontinyu

c. Perhitungan probabilitas setiap kategori terhadap target *class*.

Perhitungan probabilitas atribut bertipe diskrit diproses menggunakan array dua dimensi yang digunakan untuk menyimpan nilai probabilitas kategori dan *class* target, sedangkan perhitungan artibut kontinyu diproses menggunakan array tiga dimensi yang nantinya digunakan untuk menyimpan nilai probabilitas *split*, *spilt* terbaik dan *class* target. *Sourcecode* dari proses perhitungan atribut bertipe diskrit, dapat dilihat pada gambar 4.7 dan proses perhitungan atribut bertipe kontinyu, dapat dilihat pada gambar 4.8.

```
function diskritProbVsTarget(navigator : TDBNavigator;
prob, target : TDBText;
 arrBuf : arrProbDiskrit; arrProb, arrTarget :
arrDiskrit) : arrProbDiskrit;
var
  i,j,n : integer;
begin
  navigator.DataSource.DataSet.First;
  for i := low(arrBuf) to high(arrBuf) do
  begin
    for j := low(arrBuf[i]) to high(arrBuf[i]) do
    begin
      arrBuf[i,j].targetS := arrTarget[i].kategori;
      arrBuf[i,j].bufS := arrProb[j].kategori;
      for n := 1 to navigator.DataSource.DataSet.
            RecordCount do
      begin
        ----------------------
      end;
      if arrBuf[i,j].target <> 0 then
        arrBuf[i,j].probability := arrBuf[i,j].buf /
         arrBuf[i,j].target
      else arrBuf[i,j].probability := 0;
        navigator.DataSource.DataSet.First;
    end;
  end;
  Result := arrBuf;
end;
```

Gambar 4.7 Proses Perhitungan Probabilitas Atribut Bertipe Diskrit

```
function kontinyuProbVsTarget(navigator : TDBNavigator;
prob, target : TDBText;
 arrBuf : arrProbKontinyu; arrProb : arrKontinyu;
arrTarget : arrDiskrit) : arrProbKontinyu;
var
  i,j,k,n : integer;
```

```
begin
  navigator.DataSource.DataSet.First;
  for i := low(arrBuf) to high(arrBuf) do
  begin
    for j := low(arrBuf[i]) to high(arrBuf[i]) do
    begin
      k := 0;
      begin
        arrBuf[i,j,k].targetS:=arrTarget[i]. kategori;
        arrBuf[i,j,k].bufS:= arrProb[j,k].kategori;
        for    n    :=    1    to    navigator.DataSource.
         DataSet.RecordCount do
        begin
          --------------
          end;
          navigator.DataSource.DataSet.Next;
        end;
      end;
      if arrBuf[i,j,k].target <> 0 then
        arrBuf[i,j,k].probability  :=  arrBuf[i,j,k].  buf
         / arrBuf[i,j,k].target
      else arrBuf[i,j,k].probability := 0;
      navigator.DataSource.DataSet.First;

       --------
    end;
  end;
  Result := arrBuf;
end;
```

Gambar 4.8 Proses Perhitungan Probabilitas Atribut Bertipe Kontinyu

d. Perhitungan *information* untuk setiap atribut

Proses perhitungan *information* pada setiap atributnya dilakukan setelah perhitungan probabilitas. Proses perhitungan *information* ini dijelaskan pada gambar 4.9.

```
function infoValue(arrTarget : arrDiskrit) : real;
var
  i : integer;
  info : real;
begin
  info:= 0;
  for i := 0 to length(arrTarget)-1 do
  begin
    if (arrTarget[i].probability <> 0) then
    info := info - arrTarget[i].probability
```

```
      * (Log2(arrTarget[i].probability));
  end;
  Result := info;
end;
```

Gambar 4.9 Proses Perhitungan *Information* Setiap Atribut

e. Perhitungan *entropy* pada setiap atribut

Pada impelementasi program, perhitungan terhadap tiap tipe atribut berbeda. Untuk atribut dengan tipe diskrit, *Sourcecode* dari proses perhitungannya dapat dilihat pada gambar 4.10. Pada atribut kontinyu, dipilih *entropy* yang memiliki nilai terkecil dari semua kandidat *split point*. *Sourcecode* dari proses ini, dapat dilihat pada gambar 4.11.

```
function      entropyDiskrit(arrBuf      :      arrDiskrit;
bufVsTarget : arrProbDiskrit) : real;
var
  i,j : integer;
  buf, ent : real;
begin
  ent := 0;
  for i := low(arrBuf) to high(arrBuf) do
  begin
    buf := 0;
    for j := low(BufVsTarget) to high(BufVsTarget) do
    begin
      if (bufVsTarget[j,i].probability = 0) then
        buf := buf
      else
        buf := buf -(bufVsTarget[j,i].probability
         * (Log2(bufVsTarget[j,i].probability)));
    end;
    buf := buf * arrBuf[i].probability;
    ent := ent + buf;
  end;
  Result := ent;
end;
```

Gambar 4.10 Proses Perhitungan *Entropy* Atribut Diskrit

```
function      entropyKontinyu(arrBuf      :      arrKontinyu;
bufVsTarget : arrProbKontinyu) : real;
var
  i,j,k : integer;
  buf1, buf2 : real;
  ent : array of real;
begin
```

```
  SetLength(ent,length(arrBuf));
  for i := low(ent) to high(ent) do
    ent[i] := 0;

  buf1 := 0; buf2 := 0;
  for i := low(arrBuf) to high(arrBuf) do
  begin
    for j := low(arrBuf[i]) to high(arrBuf[i]) do
    begin
      for k := low(bufVsTarget) to high(bufVsTarget) do
      begin
        if bufVsTarget[k,i,j].probability <> 0 then
        begin
          buf1 := buf1 - (bufVsTarget[k,i,j].probability
           * (log2(bufVsTarget[k,i,j].probability)));
        end
        -------------------
  end;

------------------
    if ent[i] < buf1 then
      buf1 := ent[i];
  end;

  result := buf1;
end;
```

Gambar 4.11 Proses Perhitungan *Entropy* Atribut Kontinyu

f. Perhitungan Gain tertinggi

Proses ini merupakan tahap terakhir dari proses pencarian atribut terbaik, atribut yang memiliki gain tertinggi maka atribut tersebut yang nantinya dijadikan sebagai atribut tes (atribut terbaik). Proses perhitungan gain terbaik langsung di proses pada `prosedure buatTree`.

Proses pencarian atribut terbaik akan berakhir jika setiap iterasi telah terbentuk suatu *class*. Proses ini akan membentuk suatu pembentukan *tree* (*decision tree*). *Sourcecode* dari proses pembentukan *tree*, dapat dilihat pada gambar 4.12.

```
function buatTree(dataAwal : arrRec; TInfo : arrInfo;
 treeView : TTreeView; strQuery : arrStr; pruning :
integer) : arrStr;
var
  rawData : arrRec;
  dataJenisHP,  dataJk,  dataJenisKons,  dataKebutuhan:
```

```
      arrDiskrit;
  dataUmur, dataBudget : arrKontinyu;
  probJk, probJenisKons, probKebutuhan: arrProbDiskrit;
  probUmur, probBudget : arrProbKontinyu;
  probUmurDiskrit, probBudgetDiskrit : arrProbDiskrit;

  info : real;
  dataInfo : arrInfo;
  test, totalData, totalField : integer;

  indeks,i,j : integer;
  buf : real;

  bufTree1, bufTree2 : TTreeNode;
  s : string;
  stat : boolean;
  bufStr : arrStr;

  winKontinyu:integer;
begin
  SetLength(bufStr,0);
  -----
  SetLength(dataJenisHP, countEntity3(rawData,0));
  SetLength(dataJk, countEntity3(rawData,1));
  SetLength(dataJenisKons, countEntity3(rawData,2));
  SetLength(dataKebutuhan, countEntity3(rawData,3));
  SetLength(dataUmur, 3);
  SetLength(dataBudget, 3);
  for i := 0 to 2 do
  begin
    SetLength(dataUmur[i],2);
    SetLength(dataBudget[i],2);
  end;

  SetLength(probJk, countEntity3(rawData,0));
  for i := low(probJk) to high(ProbJk) do
   SetLength(probJk[i], countEntity3(rawData,1));
  ---------
  dataJenisHP        :=        hitungProbDisk(dataJenisHP,
length(rawData[0].data));
  ------

  probJk  :=  diskritProbVsTarget3(rawData,1,0,  probJk,
dataJk, dataJenisHP);
  ----

  info := infoValue(dataJenisHP);

  dataInfo[0].arrData := probJk;
```

```
  //dataInfo[0].usable := true;
  dataInfo[0].entropy := entropyDiskrit(dataJk,probJk);
  dataInfo[0].gain:=gainValue(info,dataInfo[0].entropy);
  --------


 // gain max
 indeks := 0;
  buf := 0;
  for i:= low(dataInfo) to high(dataInfo) do
  begin
    if (buf < dataInfo[i].gain) and (i<> winKontinyu)
then
    begin
      buf := dataInfo[i].gain;
      indeks := i;
    end;
  end;


  if (treeView.Items.Count = 0) then
  begin
    bufTree1:=treeView.Items.Add(nil,dataInfo[indeks].
    name);
    s := '';
    stat := false;
    for i := low(dataInfo[indeks].arrData[0]) to
high(dataInfo[indeks].arrData[0]) do
    begin
      for j := low(dataInfo[indeks].arrData) to
high(dataInfo[indeks].arrData) do
      begin
      //keputusan apakah suatu node membentuk tree?
        if (dataInfo[indeks].arrData[j,i].probability =
1) then
        begin
           bufTree2 := treeView.Items.AddChild(bufTree1,
           dataInfo[indeks].arrData[j,i].bufS);

          treeView.Items.AddChild(bufTree2,
           '[' + dataInfo[indeks].arrData[j,i].targetS +
']');

          stat := true;
        end
        else if ((j = high(dataInfo[indeks].arrData))
and (stat=false)) then
        begin
          bufTree2 := treeView.Items.AddChild(bufTree1,
```

```
            dataInfo[indeks].arrData[j,i].bufS);

          bufStr := addDynString(bufStr,bufTree1.Text
           + dataInfo[indeks].arrData[j,i].bufS);
        end;
      end;
    stat := false;
    end;
  end
  else
  begin
    bufTree1 := treeView.TopItem;
    -----
  end;

  result := bufStr ;
end;
```

Gambar 4.12 Proses Pembentukan *Tree*

2. Mengekstrak aturan klasifikasi

Setelah proses pembentukan *tree* (pohon keputusan) berakhir, maka langkah selanjutnya adalah mengekstraknya menjadi aturan klasifikasi (IF-THEN) . *Sourcecode* dari proses ini, dapat dilihat pada gambar 4.13.

```
Procedure          TFHandphone.BuDecisionRuleClick(Sender:
TObject);
var
  i,limit : integer;
  keepSearch : boolean;
  bufNode, parentNode : TTreeNode;
  temp, s : string;
  buf: array of string;
begin
  Memo2.Clear;
  bufNode := TreeView1.Items.GetFirstNode;

  -----
  //parse and bwt decision rule
  while bufNode <> nil do
  begin
    if  (pos('=',  bufNode.Text)  <>  0)  or  (pos('>',
      bufNode.Text) <> 0)
      or (pos('<', bufNode.Text) <> 0) then
      s := s + bufNode.Text
    else if (pos('=', bufNode.Text) = 0) and (pos('[',
      bufNode.Text) = 0) then
```

```
      s := s+' and '+bufNode.Text
    else if (pos('[', bufNode.Text) <> 0) then
      s:= s+' then '+bufNode.Text;

    // bwt decision rule
    if (pos('[', bufNode.Text) <> 0) then
    begin
      parentNode := bufNode.GetPrev;
      -------
      end;
      parentNode.Delete;
      bufNode := TreeView1.Items.GetFirstNode;
    end;

    s := '';
    //write := false;
    else if(bufNode.HasChildren = false) then
    begin
      bufNode.Delete;
      bufNode := TreeView1.Items.GetFirstNode;
    end
    else
    begin
      bufNode := bufNode.GetNext;
    end;
  end;
end;
```

Gambar 4.13 Proses Mengekstrak Aturan Klasifikasi

3. *Pruning rule*

Setelah mengekstrak aturan klasifkasi, proses selanjutnya adalah *pruning rule.* Proses ini dilakukan jika jika *akurasi rate* pada *subtree* lebih besar jika dibandingkan dengan node *single. Sourcecode* dari proses ini, dapat dilihat pada gambar 4.14.

```
procedure            TFHandphone.BPruningRuleClick(Sender:
TObject);
var
  N, i,j,k,l, noRule :integer;
  f,x,e, z:real;
  sql,s,kategori,tanda,atribut : string;
begin
  Memo2.Clear;
  for i:=0 to (length(bufPrun)-1) do
  begin
    N:=bufPrun[i].count - bufPrun[i].klasMax;
    f:= N/bufPrun[i].count;
```

```
    z:=0.69;
    x:=z*(sqrt(((f*(1-
f))/N)+(Power(z,2)/(4*(power(N,2))))));
  e:=(f+(Power(z,2)/(2*N))+x)/(1+(power(z,2)/N));
  bufPrun[i].e:= e;
  end; //end for


  for i:= (length(bufPrun)-1) downto 1 do
  begin
    for j:=(i-1) downto 0 do
    begin
      //node 2 dan node 1
      if(StrPos(pchar(bufPrun[i].str),
        pchar(bufPrun[j].str)) <> nil) then
      begin
        ------

      end;
    end;
  end;  //end for
end;
```

Gambar 4.14 Proses *Pruning Rule*

4. Memasukkan aturan klasifikasi ke database

Proses terakhir dalam pembelajaran adalah memasukkan aturan klasifikasi ke dalam database. *Sourcecode* dari proses ini, dapat dilihat pada gambar 4.15.

```
procedure TFHandphone.Button2Click(Sender: TObject);
var
 id_rule, i,j,noRule : integer;
 atribut,tanda,kategori,s :string;
begin
  DM.ADOConnection1.Execute('insert  into  t_masterrule
     (tgl_awal,tgl_akhir,merk_hp)values('+QuotedStr
     (DateToStr(DTPAwal.Date))+','+QuotedStr(DateToStr(
      DTPAkhir.Date))+','+QuotedStr(CBMerk.Text)+')');

  DM.QIDRule.SQL.Clear;
  DM.QIDRule.SQL.Add('select  id_rule  from  t_masterrule
     order by id_rule desc');
  DM.QIDRule.Open;
  id_rule:=DM.QIDRule.fieldbyname('id_rule').AsInteger;

---------------
  DM.ADOConnection1.Execute('insert                    into
     t_ClassificationRule      values('+inttostr(id_rule)
```

```
    +','+inttostr(norule)+','+QuotedStr(atribut)+','+Qu
    otedStr(tanda)
     +','+QuotedStr(kategori)+')');

end;
```

Gambar 4.15 Proses Memasukkan Aturan Klasifikasi Ke Database

4.2.3 Implementasi Proses Perekomendasian

Proses pemberian rekomendasi merupakan proses tes data. Pemberian rekomendasi didapat dari proses pembelajaran yang membaca aturan klasifikasi yang telah terbentuk.

Pada proses ini, sistem akan menerima inputan data konsumen berupa umur, jenis kelamin, pekerjaan, budget yang dilimiliki oleh konsumen dan kebutuhan yang diinginkan konsumen. Sebelumnya telah dijelaskan pada sub bab 3.2.2, dari data-data tersebut langkah awalnya adalah menentukan jenis *handphone* yang sesuai berdasarkan aturan klasifikasi yang telah terbentuk oleh *decision tree* dari merk yang telah diinputkan oleh konsumen. Kemudian sistem akan mencari produk dari jenis *handphone* yang telah diketahui, yang dimana produk tersebut diperoleh dari transaksi penjualan sebelumnya. Berikut adalah potongan prosedur yang digunakan dalam proses rekomendasinya, dapat dilihat pada gambar 4.16.

```
procedure TFHandphone.BCariClick(Sender: TObject);
var
  i,j,k,id_rule:integer;
  sql,s,str : string;
  kriteria:array of string;
begin
  SetLength(kriteria,10);
  ------------

  //select no_rule
  DM.QMasterRule.SQL.Clear;
  DM.QMasterRule.SQL.Add('select   *   from   t_masterRule
where merk_hp='+QuotedStr(CBMrkHP.Text));
  DM.QMasterRule.Open;

  if(DM.QMasterRule.RecordCount >0) then
  begin
   while i< Length(kriteria)-2 do
   begin
    //select satu2
```

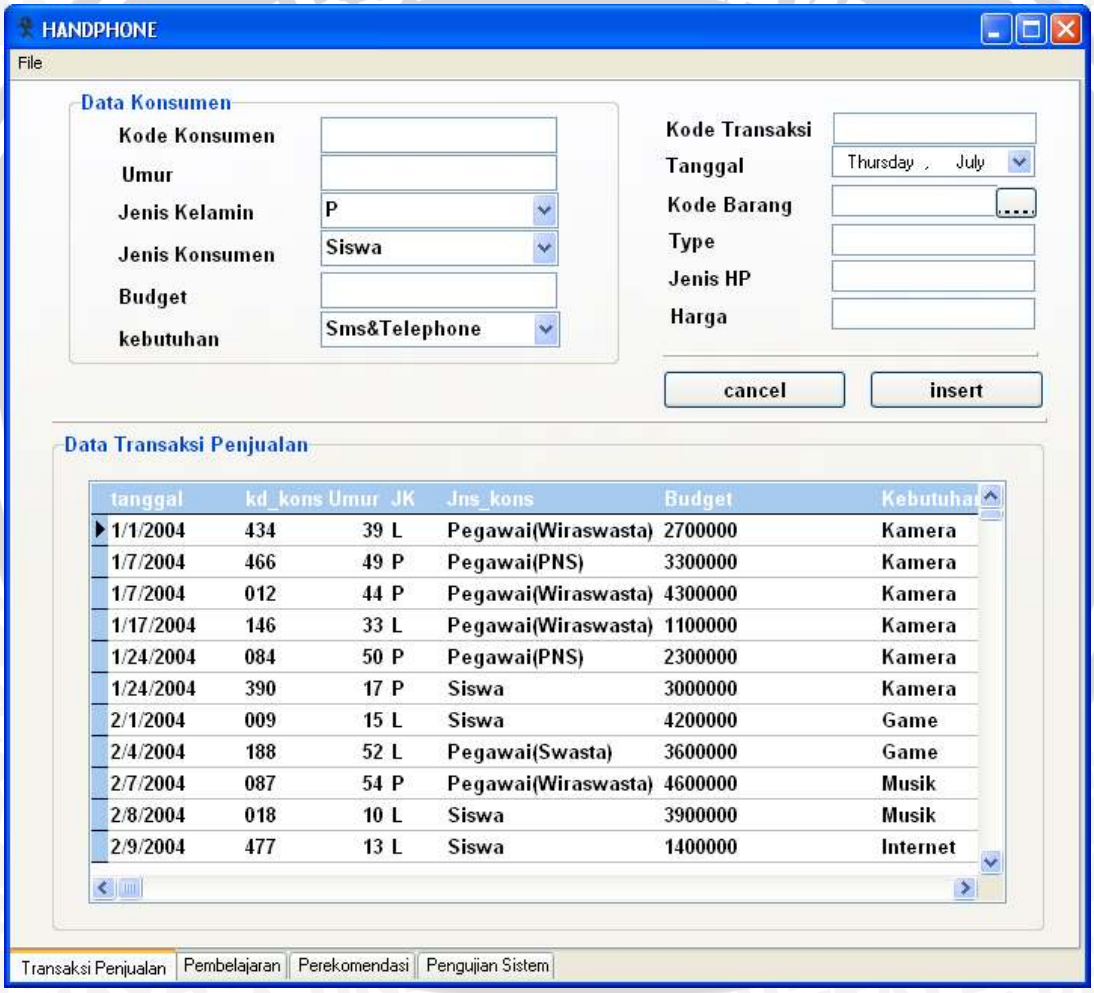
```
    sql:='select        distinct        aturan        from
      t_classificationrule  where id_rule='
      +DM.QMasterRule.FieldByName('id_rule').AsString+'
      and    atribut='    +QuotedStr(kriteria[i])+'    and
      kategori ='+QuotedStr(kriteria[i+1]);


    id_rule:=DM.QMasterRule.fieldbyname'id_rule').
          AsInteger;
    for k:=1 to j do
    begin
      sql:=sql+  ' or   aturan='+DM.QClassRule.fieldbyname
        ('aturan').asstring ;
      DM.QClassRule.Next;
    end;


  DM.QClassRule.Open;
  if DM.QClassRule.RecordCount >0 then
  begin
    DM.QMasterRule.SQL.Clear;
    DM.QMasterRule.SQL.Add('select      kategori      from
       t_classificationrule  where  id_rule  ='+inttostr
       (id_rule)+'and      aturan      ='+DM.QClassRule.
       fieldbyname('aturan').AsString+'and
       atribut=''kelas''');
    DM.QMasterRule.Close;
    DM.QMasterRule.Open;

    if DM.QMasterRule.RecordCount >0 then
    begin
      dm.QRekomendasi.SQL.Clear;
      dm.QRekomendasi.SQL.Add('SELECT          distinct
         h.jenis_HP, h.tipe_HP, h.Harga, d.Picture ');
      dm.QRekomendasi.SQL.Add('FROM      T_TransPemb  p
         INNER JOIN T_HP h ON p.kd_brg = h.kd_brg INNER
         JOIN  T_kons  k  ON  p.kd_kons  =  k.kd_kons  INNER
         JOIN T_detHP d ON h.kd_brg = d.kd_brg');
      dm.QRekomendasi.SQL.Add('where
         h.merk_hp='+QuotedStr(CBMrkHP.Text)    +'    and
         h.jenis_hp='+QuotedStr(DM.QMasterRule.fieldbyn
         ame('kategori').AsString)+'                    and
         h.harga<='+kriteria[1]);
      dm.QRekomendasi.Open;

    end;
  end
  else
    ShowMessage('tidak ada');
```

```
  end
  else
  begin
      ShowMessage('belum masuk rule');
  end;
end;
```

Gambar 4.16 Proses Perekomendasian

4.3 Penerapan Aplikasi

Aplikasi yang dibangun ini digunakan untuk melakukan berbagai proses rekomendasi, serta digunakan untuk proses pengujian sistem seperti yang disebutkan pada bab sebelumnya. Berikut merupakan tempilan *form* taransaksi penjualan.



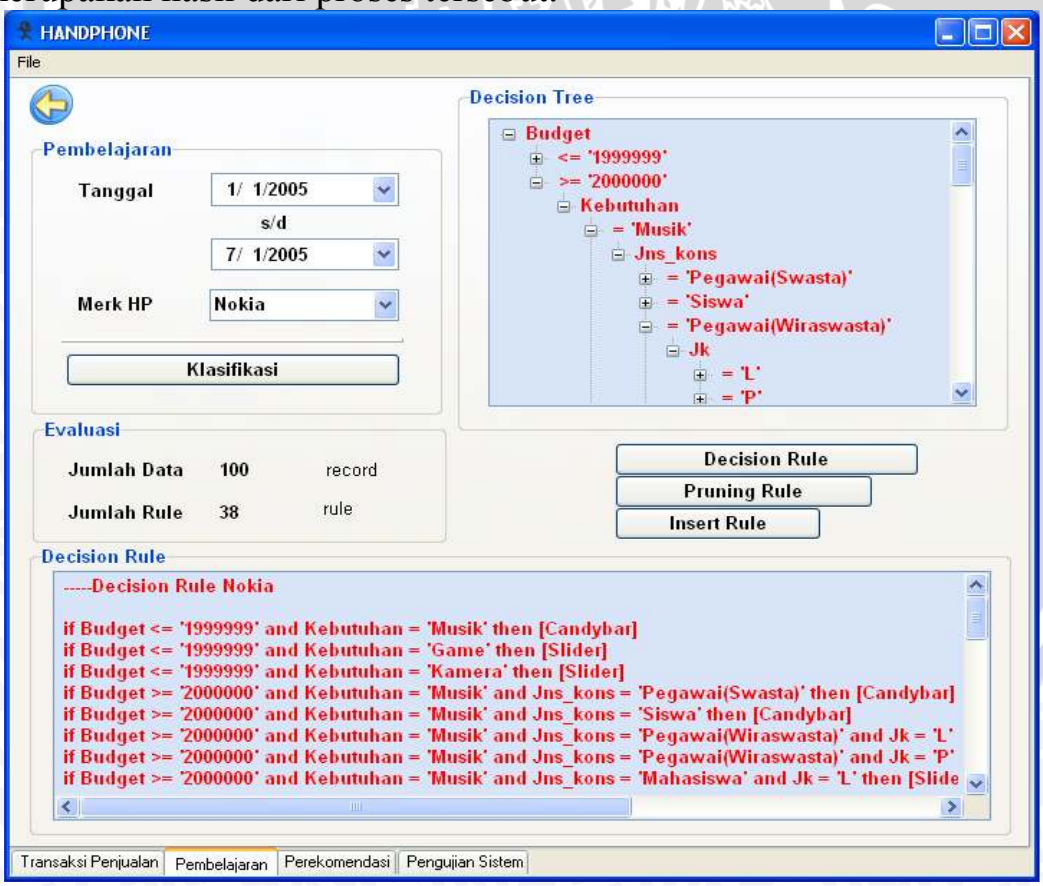
Gambar 4.17 Tampilan Menu Transaksi Penjualan

Menu pada gambar 4.17 digunakan untuk memasukan data konsumen dan barang yang ingin dibeli. Proses ini merupakan data transaksi penjualan yang nantinya akan dimasukkan ke dalam

database transaksi penjualan. Pada proses input data konsumen, *user* diminta untuk memasukan data konsumen dan data barang yang diinginkan. Data pembeli yang diinputkan berupa kode konsumen, jenis kelamin, jenis konsumen, budget serta kebutuhan (misal kamera, musik, dll). Sedangkan pada proses input data barang, *user* diminta untuk memasukan data barang yang sesuai dengan budget dan kebutuhan konsumen.Selanjutnya user harus memasukkan kode transaksi dan tanggal transaksi penjualan.

Jika proses transaksi penjualan telah selesai maka akan ditampilkan pada tabel data transaksi penjualan yang terdapat pada *database*. Data penjualan yang ditampilkan disesuaikan dengan tanggal transaksi penjualan yang dilakukan terlebih dahulu.

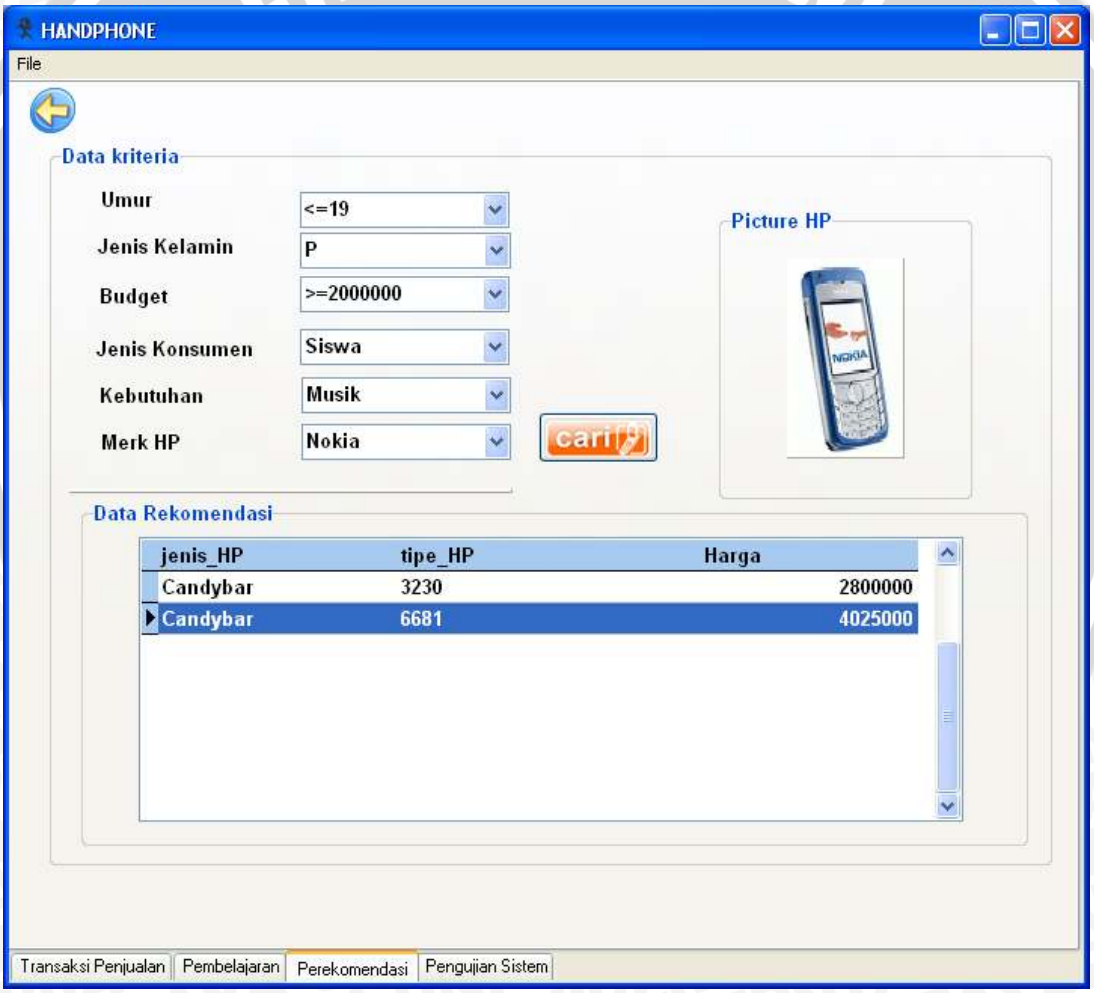
Apabila proses transaksi penjualan telah dilakukan, maka proses selanjutnya adalah proses pembelajaran. Dalam proses ini data yang digunakan adalah data transaksi penjualan. Sebelumnya *user* harus menginputkan rentang tanggal data transaksi penjualan, serta merk HP yang akan dilakukan proses pembelajaran. Gambar 4.18 merupakan hasil dari proses tersebut.



Gambar 4.18 Tampilan Menu Data Pembelajaran

Setelah dilakukan proses input data maka sistem akan mengklasifikasikan data pembelajaran dengan membentuk pohon keputusan (*decision tree*). Pada proses ini akan ditampilkan *decision tree*. Setelah proses klasifikasi berakhir, menu ini akan menampilkan informasi berupa jumlah data yang diklasifikasikan, jumlah *rule* yang terbentuk.

Pada bab-bab sebelumnya telah dijelahkan bahwa, setelah proses klasifikasi berakhir maka tahap selanjutnya adalah *pruning rule*. Maka untuk tahap selanjutnya *user* diminta untuk menekan tombol *Pruning Rule* untuk melakukan proses *pruning* tersebut. Setelah itu proses terakhir adalah memasukkan aturan klasifikasi ke dalam database, dimana *user* diminta untuk menekan tombol *insert rule.*

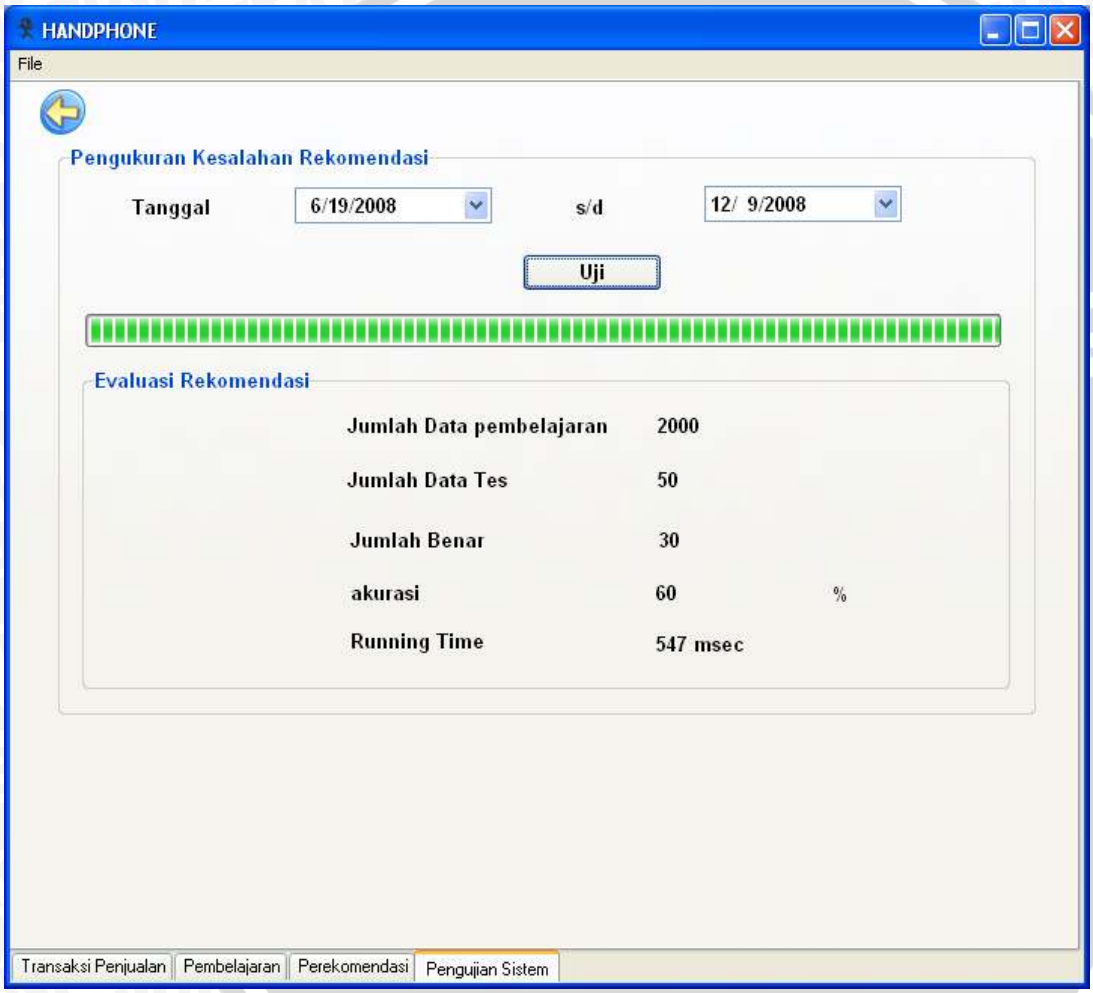


Gambar 4.19 Tampilan menu perekomendasian

Menu pada gambar 4.19 digunakan untuk perekomendasian. Pada ini digunakan untuk memberikan rekomendasi berupa produk

berdasarkan data konsumen yang diinputkan oleh *user*. Seteleh menginputkan data konsemen, *user* diminta untuk memilih merk HP yang diinginkan konsumen. Kemudian sistem akan memproses data-data tersebut dengan memberikan rekomendasi berupa produk dari merk HP yang telah ditentukan.

Data hasil rekomendasi akan ditampil, jika *user* ingin melihat gambar dari produk yang direkomendasikan maka *user* diminta untuk menekan daftar pada tabel data rekomendasi.



Gambar 4.20 Tampilan Menu Pengujian Sistem

Menu pada gambar 4.20 digunakan untuk melakukan pengujian sistem, *user* diharuskan untuk menetapkan rentang periode transaksi penjualan untuk data sampel yang akan diujikan. Proses ini akan menampilkan informasi jumlah data pembelajaran, jumlah benar dan juga meberikan informasi jumlah salah. Selain itu juga akan ditampilkan pula prosentase *akurasi* atau tingkat kebenaran yang

dinyatakan dalam persen serta waktu yang diperlukan sistem dalam proses rekomendasinya.

4.4 Pengujian sistem Rekomendasi

4.4.1 Pengujian Tingkat Kebenaran

Uji coba dilakukan untuk mengetahui kinerja sebuah metode dalam memecahkan suatu permasalahan. Dengan melakukan uji coba sebanyak 4 kali, maka didapatkan hasil pengujian tingkat kebenaran pada tabel 4.4.

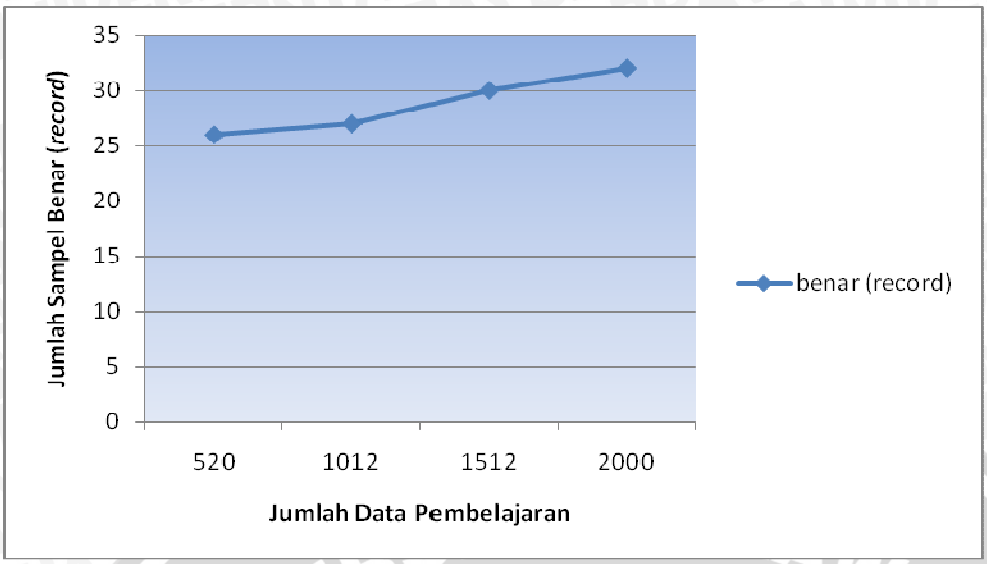
Tabel 4.4 Hasil Pengujian Tingkat Kebenaran

Data Pembelajaran ke-	Jumlah Aturan (Rule)	Jumlah Data Tes (*record*)	Jumlah benar (*record*)	Akurasi (%)
Data [1]	172	50	26	52
Data [2]	216	50	27	54
Data [3]	236	50	30	60
Data [4]	247	50	31	62

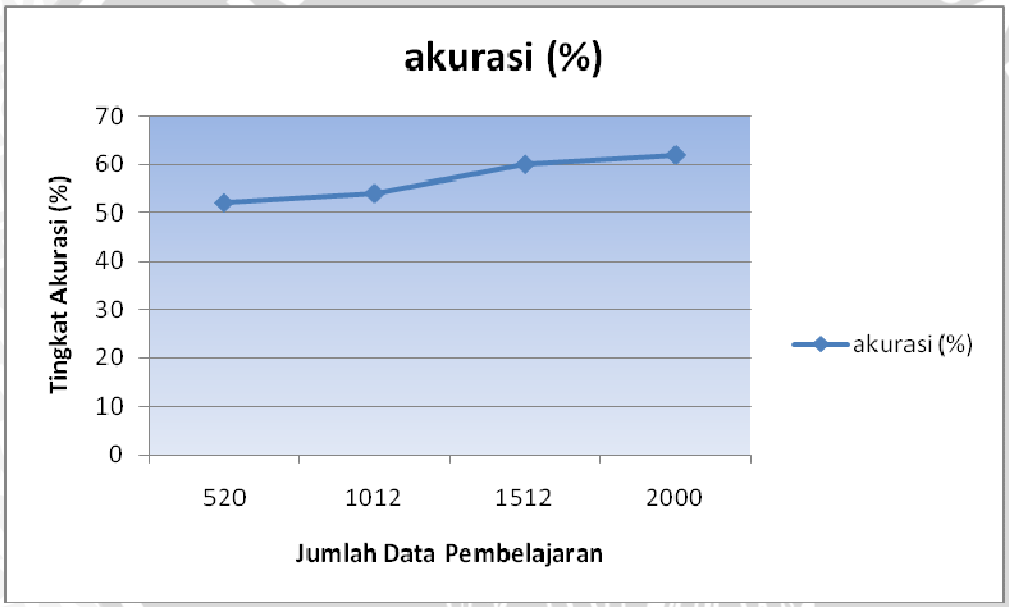
Tabel 4.4 merupakan tabel tingkat kebenaran dengan sejumlah data pembelajaran yang telah ditetapkan sebelumnya, yang dimana telah dijelaskan pada sub bab 3.3.4 dan data tes yang memiliki jumlah yang sama yaitu sebanyak 50 *record*. Setiap data pembelajaran menghasilkan tingkat kebenaran yang berbeda-beda, begitu juga akurasinya.

Berikut ini adalah penyajian hasil pengujian tingkat kebenaran rekomendasi dalam bentuk grafik berdasarkan jumlah benarnya.

Pada gambar 4.21, dapat diketahui bahwa jumlah data pembelajaran yang kecil menghasilkan jumlah sampel benar yang kecil juga pada saat tes data. Pada grafik jumlah sampel benar menujukkan bahwa jumlah benar terus meningkat seiring dengan bertambah jumlah data pembelajaran. Maka dapat diambil kesimpulan bahwa semakin banyak jumlah data pembelajaran yang digunakan, maka klasifikasi yang diberikan akan semakin benar begitu juga dengan akurasinya/ tingkat kebenaran yang semakin meningkat. Berikut merupakan bentuk grafik dari akurasi, yang dapat dilihat pada gambar 4.22.



Gambar 4.21 Grafik Jumlah Sampel Benar



Gambar 4.22 Grafik Tingkat Kebenaran

Pada gambar 4.22, dapat dilihat bahwa jumlah data pembalajaran mempengaruhi tingkat kebenaran yang dihasikan. Semakin meningkat data pembelajaran maka semakin meningkat juga tingkat kebenarannya. Semakin banyak data pembelajaran, maka semakin banyak juga informasi yang dikumpulkan oleh sistem. Hal ini tentunya akan meningkatkan akurasi hasil rekomendasi yang diberikan pada *user*.

4.4.2 Pengujian *running time* rekomendasi sistem

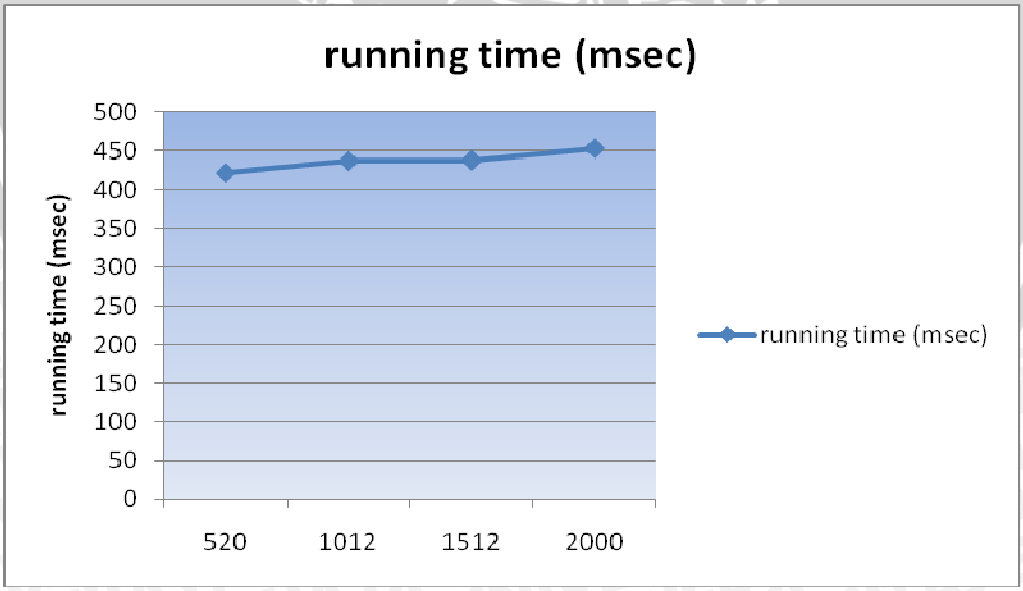
Pengujian ini akan mengukur waktu yang dibutuhkan oleh sistem dalam memberikan rekomendasi. Proses ini dilakukan untuk mengetahui perbandingan kecepatan dari pertambahan jumlah data pembelajaran.

Berikut ini adalah tabel *running time* rekomendasi sistem terhadap sejumlah data tes dengan jumlah data pembelajaran yang berbeda-beda berdasarkan hasil klasifikasi.

Tabel 4.5 Hasil Pengujian *Running Time* Rekomendasi

Data Pembelajaran ke-	Jumlah Data Tes (*record*)	*Running Time* (msec)
Data [1]	50	422
Data [2]	50	437
Data [3]	50	438
Data [4]	50	453

Grafik pengujian *running time* rekomendasi dapat dilihat seperti pada gambar 4.23.



Gambar 4.23 Grafik Hasil Pengujian *running time* Rekomendasi

Dari grafik diatas diketahui bahwa dengan bertambahnya jumlah data pembelajaran maka waktu yang dibutuhkan untuk pemberian

rekomendasi juga akan semakin lama. Hal ini disebabkan karena pola klasifikasi yang terbentuk semakin banyak, maka sistem juga membutuhakan waktu yang banyak.

4.5 Analisis Hasil

Berdasarkan hasil uji coba tingkat kebenaran yang dilakukan dengan jumlah data pembelajaran yang berbeda-beda dan jumlah data tes yang sama, uji coba tersebut bertujuan untuk mengetahui tingkat kebenaran sistem dalam memberikan rekomandasi. Pengujian tingkat kebenaran dapat dilihat pada grafik jumlah sampel benar. Grafik ini dapat di lihat pada gambar 4.21, dari gambar tersebut dapat diketahui bahwa semakin banyak data pembelajaran maka klasifikasi yang di berikan akan semakin benar. Hal ini juga berpengaruh pada akurasi/ tingkat kebenaran yang dihasilkan akan semakin meningkat juga. Akurasi/ tingkat kebenaran dapat dilihat pada gambar 4.22, yang menunjukkan bahwa tingkat kebenaran semakin meningkat, hal ini disebabkan oleh meningkatnya jumlah data pembelajaran. Semakin banyak data pembelajaran, maka semakin banyak juga informasi yang dikumpulkan oleh sistem. Hal ini tentunya akan meningkatkan akurasi hasil rekomendasi yang diberikan pada *user*.

Sedangkan untuk hasil pengujian *running time* rekomendasi, dapat dilihat pada gambar 4.23. Dari gambar 4.23 didapatkan hasil bahwa waktu yang dibutuhkan untuk mengahasilkan rekomendasi semakin banyak karena disebabkan data pembelajaran yang terus meningkat. Bertambahnya jumlah data pembelajaran mempengaruhi *running time* dalam memberikan rekomendasi, hal ini disebabkan karena pola klasifikasi yang terbentuk semakin banyak.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari tugas akhir ini adalah :

1. *Data mining* dengan metode *decision tree* menggunakan algoritma C4.5 dapat diimplementasikan untuk perekomendasian *handphone*.
2. Semakin bertambahnya data pembelajaran maka kebenaran sistem dalam pemberian rekomendasi akan semakin meningkat, hal ini disebabkan oleh banyaknya informasi yang dikumpulkan oleh sistem.
3. Bertambahnya data pembelajaran akan berpengaruh pada banyaknya waktu yang dibutuhkan untuk memberikan rekomendasi, karena pola klasifikasi yang terbentuk semakin banyak.

5.2 Saran

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

1. Untuk penelitian algoritma C4.5 selenjutnya diharapkan untuk menambah jumlah atribut, agar pembemtukan *tree* lebih beragam.
2. Meningkatkan tingkat kebenaran dalam pemberian rekomendasi HP dan memperbaiki waktu untuk rekomendasi HP.

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Andrew W. Moore. *Decision Trees*. Carnegie Mellon University. http://www.cs.cmu.edu/~awm, diakses tanggal 20 Maret 2008.

Baltes, Jacky. *COMP-4360 Machine Learning Lecture 2: Decision Trees*. http://www4.cs.umanitoba.ca/~jacky/Teaching/Courses/COMP-4360-MachineLearning/current/index.php, diakses tanggal 20 Februari 2008.

Berry, Michael J.A., Linoff, Gordon S. 2004. *Data Mining Techniques For Marketing, Sales, and Customer Relationship Management Second Edition,* Wiley Publishing, Inc., Indianapolis, Indiana.

Frank, Vanden Berghen. 2003. "*Classification Trees : C4.5*", Universit Libre de Bruxelles.

Fayyad, Usama. 1996. *Advances in Knowledge Discovery and Data Mining*. MIT Press.

Gunduz, Sule dan Ozsu, Tamer. 2004. *Recommendation Model for User Accesses to Web Pages.* http://www3.itu.edu.tr/~sgunduz/papers/icann.pdf, diakses tanggal 15 Februari 2008.

Han, Jiawei and Khamber, Micheline. 2006. *Data Mining* : *Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, USA.

Han, Jiawei and Khamber, Micheline. 2000. *Data Mining* : *Concepts and Techniques.* Morgan Kaufmann Publishers, Simon Fraser.

Haruechaiyasak, C., Mei-Ling Shyu, dan Shu-Ching Chen. 2004. *A Data Mining Framework for Building AWeb-Page Recommender System*. http://www.eng.miami.edu/~shyu/ Paper/2004/iri04-rec.pdf.

Heryati, Diah Noor. 2007. *Data Mining Classification Dengan Metode Decision Tree Menggunakan Algoritma C4.5*. Jurusan Teknologi Informatika, Sekolah Tinggi Teknologi Telkom Bandung, Bandung.

Hidayat, MCS, Taufiq *http://ptbb.uny.ac.id/download/ujicoba/Bab%2001%20-%20Pendahuluan.pdf*, diakses tanggal 18 April 2008.

http://www.google.co.id/search?hl=id&q=proses+knowledge+discovery+in+database&btnG=Telusuri+dengan+Google&meta=, diakses tanggal akses 28 September 2005.

http://lecturer.eepisits.edu/~tessy/lecturenotes/db2/bab10.pdf, diakses tanggal 24 Maret 2008.

http://www.dtreg.com/classregress.htm, diakses tanggal 24 Maret 2008.

J.R. Quinlan. 1993. *C4.5:Programs for Machine Learning*. Morgan Kaufmann.

Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya),* Graha Ilmu, Yogyakarta.

Larose, Daniel T. 2005. *Discovering Knowledge in Data. An Introduction to Data Mining*. John Wiley & Sons. New Jersey.

Ramakrishnan, Raghu. 2003. *Database Management System*", Mc Graw Hill.

Moertini, Veronica S. 2003. *Towards The Use Of C4.5 Algorithm For Classification Banking Dataset*. Fakultas Matematika dan ilmu Komputer Alam, Jurusan Ilmu Komputer, Universitas Katolik Parahyangan Bandung, Bandung. *http://home.unpar.ac.id/~integral/Volume%208/Integral%208%20No.%202/C45%20Algorithm.PDF,* diakses tanggal 1 Maret 2008.

Solorio, Thamar I. and Fuentes, Olac, "*Improving Classifer Accuracy Using Unlabled Data*", Departement of Comuter Science, Instituto Nacional de Astrofisica, Optica y Electronica, Luis Enrique Erro #1 Santa Maria Tonantzintla, Mexico.

Tan, Pang-Ning, dkk. 2006. *Introduction to Data Mining*. Pearson Addison Wesley, USA.

Thearling, Kurt. *An Introduction To Data Mining*. Whitepaper. *http://www3.shore.net/~kht/dmwhite/dmwhite.htm*, diakses tanggal 1 April 2008.

Yudi Agusta, Phd. 2006. *Data Warehouse and Data Mining,Lecture 9,* 2006. *http://yudiagusta.wordpress.com,* diakses tanggal 27 Maret 2008.