SKRIPSI

DATA MINING HISTORI PENCARIAN RUTE ANGKOT

JOVAN GUNAWAN

NPM: 2011730029

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2014

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

P rtumbuhan t knologi hingga saat ini t lah m nghasilkan banyak s kali data-data, namun s ring kali p milik data hanya m nggunakan data t rs but s p rlunya saja. Jika dilihat l bih rinci, s b - narnya jika data t rs but diolah l bih lanjut, dapat m nghasilkan s suatu yang l bih. Salah satu cara m ngolah data t rs but adalah d ngan m nggunakan t knik *data mining*. D ngan m nggunakan t knik *data mining* akan m mp rmudah m nganalisa masalah, p ngambilan k simpulan, bahkan m mp rmudah konsum n dalam m mb li jasa atau barang.

Tujuan utama dari *data mining* adalah *knowledge* [1]. *Knowledge* m rupakan suatu informasi yang b rharga dan dapat dijadikan landasan untuk m nganalisa atau m mbuat k simpulan. Untuk m ndapatkan *knowledge*, dapat dilakukan d ngan cara m lakukan p ncarian *pattern* atau pola yang m rupakan salah satu tahap dari *data mining*. Pola inilah yang akan m mp rlihatkan data manakah yang m narik dan dapat dijadikan *knowledge* yang akan digunakan untuk m nganalisa data t rs but.

Pada p n litian *data mining* ini, p nulis m miliki data *log* histori KIRI s lama 1 bulan. Data t rs but akan diimpl m ntasikan pros s *data mining* untuk m ndapatkan *pattern* dan *knowledge* yang t rkandung pada data *log* KIRI. Data *log* t rs but m miliki 5 *field* untuk s tiap *entry* s bagai b rikut:

- statisticId, primary k y dari ntry
- v rifi r, m ngid ntifikasikan sumb r dari p ncarian ini
- *timestamp*, waktu k tika p ngguna KIRI m ncari rut angkot
- *type*, tip fungsi yang digunakan
- additionalInfo, m ncatat koordinat awal, koordinat akhir, dan banyak rut yang dit mukan pada p ncarian ini

B rdasarkan hal diatas, p nulis ingin m ndapatkan pola yang m narik dan m nghasilkan *knowledge* yang b rguna dan dapat dipakai baik untuk KIRI ataupun p m rintah.

1.2 Perumusan Masalah

D ngan m ngacu pada uraian diskripsi diatas, maka p rmasalahan yang dibahas dan dit liti ol h p nulis adalah

- Bagaimana cara m ngolah pola yang dip rol h dari *data log* histori KIRI agar pola m njadi m narik dan b rmakna?
- Bagaimana m mbuat p rangkat lunak untuk m lakukan *data mining* pada data *log* histori?

1.3 Tujuan

P n litian ini b rtujuan untuk

- M ncari pola dan informasi yang m narik dari *log histori* KIRI
- P rangkat lunak dapat m lakukan data mining dari *log histori* KIRI

1.4 Batasan Masalah

P n litian *data mining* yang diatas akan dit ntukan batasan masalah yang dit liti b rupa :

- P n litian ini dibatasi hanya pada p rmasalahan pada p n rapan *data mining* pada *data log* KIRI
- Data *log* yang digunakan untuk mining m rupakan log satu bulan dari KIRI

1.5 Metode Penelitian

B rikut adalah M tod P n litian yang digunakan :

- M lakukan studi lit ratur t ntang algoritma-algoritma yang b rkaitan d ngan p mros san *data mining*
- M lakukan p n litian *data mining* yang dit rapkan pada *log* KIRI
- M rancang dan m ngimpl m ntasikan algoritma untuk *data mining*
- M ngimpl m ntasikan p mbangkit pola *data mining*
- M lakukan p ngujian dan ksp rim n

1.6 Sistematika Pembahasan

Sit matika p mbahasan dalam p n litian ini adalah:

- BAB 1: P ndahuluan, b risi latar b lakang dari p n litian ini, rumusan masalah yang timbul, tujuan yang ingin dicapain, ruang lingkup atau batasan masalah dari p n litian ini, s rta m tod p n litian yang akan digunakan dan sist matika p mbahasan dari p n litian ini

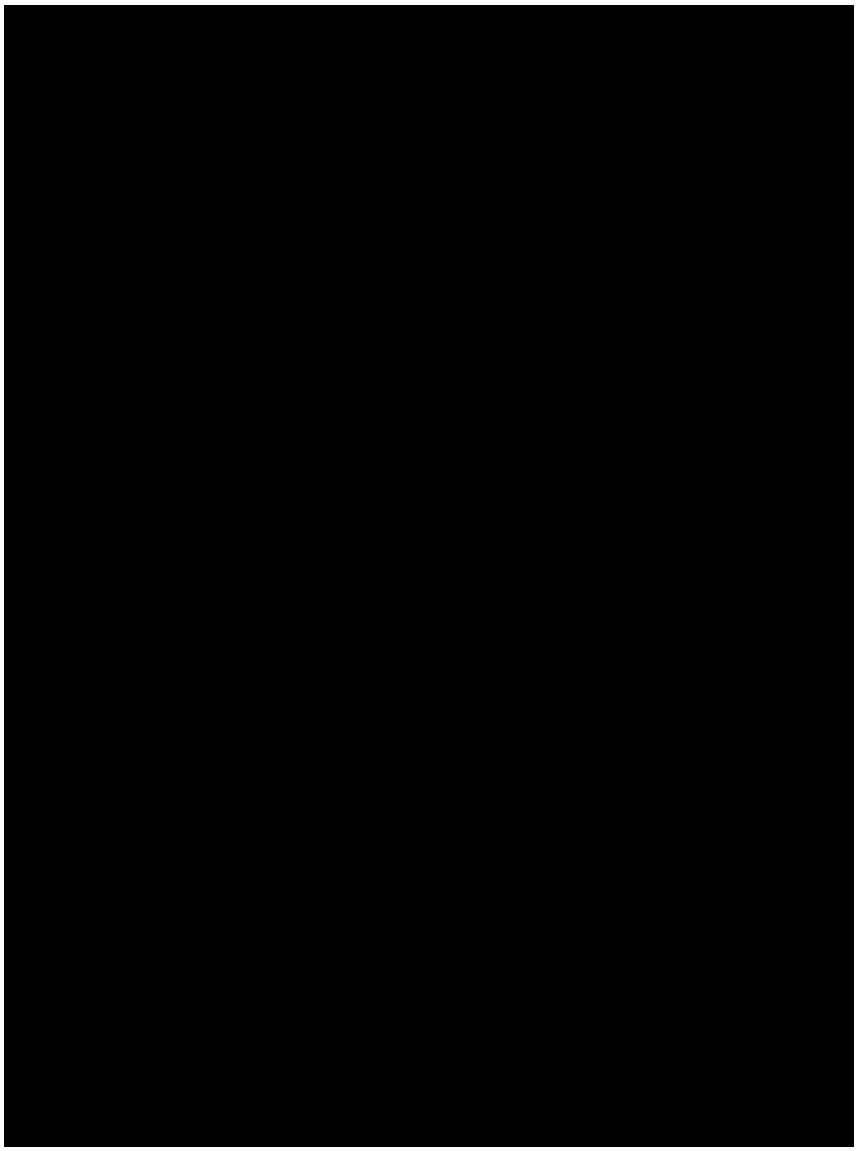
- BAB 2: Landasan T ori, b risi dasar t ori m ng nai *data mining*, *data cleaning*, *data integration*, *data selection*, *data transform*, *decision tree*, *pattern evaluation*, *knowledge presentation* dan *log* histori KIRI
- BAB 3: B risi analisa dasar t ori yang akan digunakan, analisa data s rta tahap *preprocessing* data yang akan digunakan, s rta analisa m rancang aplikasi *data mining log* histori KIRI b rikut diagram *use case*, sk nario, dan diagram k las
- BAB 4: B risi p rancangan dari aplikasi *data mining log* histori KIRI yang akan dibangun
- BAB 5: B risi hasil yang dip rol h dan k simpulan dari p n litian *data mining log* histori KIRI

BAB 2

LANDASAN TEORI

2.1 *Data Mining*

Data mining m rupakan m rupakan pros s yang m lakukan p ngambilan inti sari atau p nggalian *knowledge* dari data yang b sar dan m rupakan salah satu langkah dari *knowledge discovery*.



Gambar 2.1: Tahap *Data Mining*, [1]

M nurut [1], *knowledge discovery* dapat dibagi m njadi 7 tahap (gambar 2.1):

1. *Data cleaning*
2. *Data integration*
3. *Data selection*
4. *Data transformation*
5. *Data mining*
6. *Pattern Evaluation*
7. *Knowledge presentation*

Tahap p rtama hingga k mpat m rupakan bagian dari *data preprocessing*, dimana data-data disiapkan untuk dilakukan p nggalian data. Tahap *data mining* m rupakan tahap dimana m lakukan p nggalian data. Tahap k nam m rupakan tahap p ncarian pola yang m r pr s ntasikan *knowledge*. S dangkan tahap t rakhir m rupakan visualisasi dan r pr s ntasi dari *knowledge* yang sudah dip rol h dari tahap s b lumnya.

2.1.1 *Data Cleaning*

Data cleaning m rupakan tahap *data mining* untuk m nghilangkan *missing value* dan *noisy data*. Pada umumnya, *data* yang dip rol h dari *database* t rdapat nilai yang tidak s mpurna s p rti nilai yang hilang, nilai yang tidak valid atau salah k tik. Atribut dari suatu *database* yang tidak r l van atau r dudansi bisa diatasi d ngan m nghapus atribut t rs but. Contoh studi data yang m miliki *missing value* dan *noisy data* dapat dilihat pada tab l 2.1

Tab l 2.1: Tab l m ngandung *missing value* dan *noisy*

IdP njualan	NamaBarang	Costum r	Harga	BanyakBarang
1	Mous	Elvin	45000	2
2	K yboard	All ria	-35000	1
3	Monitor		225000	1

Dapat dilihat, pada idP njualan 2, harga dari k yboard adalah -35000, itu m rupakan *noisy* kar na tidak mungkin nilai harga suatu barang dibawah 0. Pada idP njualan 3, kolom *costumer* tidak m miliki nilai, dan itu m rupakan *missing value*.

Missing Values

Missing values akan m ngganggu pros s *data mining* pada komput r dan dapat m nghasilkan nilai akhir yang tidak s suai. T rdapat b b rapa t knik untuk m ngatasi *missing values* yaitu

- M mbuang tupl yang m ngandung nilai yang hilang
- M ngisi nilai yang hilang s cara manual
- M ngisi nilai yang hilang d ngan m nggunakan nilai konstan yang b rsifat umum
- M nggunakan nilai rata-rata dari suatu atribut untuk m ngisi nilai yang hilang

Noisy Data

Noisy data m rupakan nilai yang b rasal dari rror atau tidak valid. *Noisy data* dapat dihilangkan d ngan m nggunakan t knik *smoothing*. T rdapat 3 m tod untuk m nghilangkan *noisy data* yaitu

- *Binning*, m rupakan m tod p ngisian data s suai d ngan pros s yang dilakukan pada data t rs but
- *Regression*, m rupakan m tod yang m ncari d tail p rsamaan atribut untuk m mpr diksikan suatu nilai
- *Clustering*, m rupakan m tod p ng lompokan dimana dit mukan *outliers* yang dapat dibuang

2.1.2 *Data Integration*

Data integration m rupakan tahap m nggabungkan data dari b rbagai sumb r. Sumb r t rs but bisa t rmasuk b b rapa *database*, *data cubes*, atau bahkan *flat data*. *Data cube* m rupakan t knik p ngambilan data-data dari *data warehouse* dan dilakukan op rasi agr gasi s suai d ngan kondisi t rt ntu (contoh, p njumlahan total p njualan p r tahun dari 2005-2010). S dangkan *flat data* m rupakan data yang disimpan d ngan cara apapun untuk m r pr s ntasikan databas mod l pada s buah data baik b rb ntuk *plain text file* maupun *binary file*.

Tahap ini harus dilakukan s cara t liti t rutama k tika dalam m masangkan nilai-nilai yang b rasal dari sumb r yang b rb da. Pada tahap ini, p rlu dilakukan id ntifikasi data apakah data t rs but dapat diturunkan atau tidak agar data yang dip rol h tidak t rlalu b sar. *Data integration* yang baik m rupakan int grasi yang dapat m maksimalkan k c patan dan m ningkatkan akurasi dari pros s *data mining*. Contoh studi kasus dari *data integration*, jika suatu p rusahaan s patu A m miliki dua pabrik d ngan *database* lokal pada masing-masing pabrik, jika akan dilakukan *data mining* pada k dua *database* t rs but, maka k dua *database* akan digabung dan p rlu dip rhatikan s rta dip rbaiki nilai-nilai s p rti *primary key*, atribut, dan lain-lain agar tidak t rjadi *error* pada *database* yang sudah digabung. Pros s dari p nggabungan hingga p rbaikan nilai-nilai pada k dua databas t rs but adalah pros s *data integration*.

2.1.3 *Data Selection*

Pros s dimana data-data yang r l van d ngan analisis akan diambil dari databas dan data yang tidak r l van akan dibuang. S bagai contoh kasus, jika akan dilakukan analisa m ng nai nilai mahasiswa pada tab l nilai yang m miliki atribut s bagai b rikut:

- NPMMahasiswa
- NamaMahasiswa
- J nisK lamin
- Alamat

- MataKuliah
- NilaiART
- NilaiUTS
- NilaiUAS

Maka, atribut yang b rpot nsi diambil adalah MataKuliah, NilaiART, NilaiUTS, NilaiUAS, s - dangkan atribut yang akan dibuang adalah NPMMahasiswa, NamaMahasiswa J nisK lamin, dan Alamat kar na tidak t rlalu b rhubungan d ngan analisa.

2.1.4 *Data Transformation*

Data transformation m rupakan tahap p ngubahan data agar siap dilakukan pros s *data mining*. *Data transformation* bisa m libatkan:

- *Smoothing*, pros s untuk m mbuang *noise* s p rti yang dilakukan pada tahap *data cleaning*
- *Aggregation*, pros s m ngganti nilai-nilai m njadi suatu nilai yang dapat m wakili nilai s b - lumnya
- *Generalization*, pros s dimana m mbuat suatu nilai yang b rsifat khusus m njadi nilai yang b rsifat umum
- *Normalization*, pros s dimana suatu nilai dapat diubah skalanya m njadi nilai yang l bih k cil dan sp sifik
- *Attribute construction*, pros s m mbuat atribut baru yang b rasal dari b b rapa atribut untuk m mbantu pros s data mining

Smoothing

Smoothing m rupakan bagian dari *data cleaning* untuk m nghilangkan *noise* pada databas . T knik dari *smoothing* adalah *binning*, *regression*, dan *clustering*. P nj lasan t knik *smoothing* dapat dilihat pada 2.1.1, bagian *noisy data*.

Aggregation

Aggregation, dimana suatu k simpulan atau hasil dari *aggregation operation* yang disimpan dalam databas . Contoh studi kasus, jika t rdapat suatu databas dari toko A, kita dapat m nggunakan op rasi *aggregation* untuk m ncari total p ndapatan d ngan r ntang hari t rt ntu.

Generalization

generalization, dimana suatu data yang m miliki nilai *primitive* atau *low level* diubah m njadi *high level* d ngan m nggunakan kons p hirarki. Contoh studi kasus, nilai pada atribut umur dapat dik lompokkan m njadi muda, d wasa, tua.

Normalization

Atribut dapat dinormalisasi d ngan m mb ri skala pada nilainya s hingga nilai t rs but m njadi suatu rang yang l bih sp sifik dan k cil s p rti 0,0 sampai 1,0. Dua t knik nnormalisasi yaitu, *min-max normalization* dan *z-score normalization*. *Min-max normalization* akan m ngubah s mua nilai m njadi nilai d ngan skala t rt ntu. D ngan m nggunakan rumus

$$\nu' = \frac{\nu - min_A}{max_A - min_A}(newMax_A - newMin_A) + newMin_A$$

Contoh kasus, misalkan nilai minimun dan maximum dari suatu p ndapatan adalah 12.000 dan 98.000, akan diubah m njadi b rskala antara 0,0 sampai 1,0. Jika ada nilai p ndapat yang baru, yaitu 73.600, maka akan m njadi

$$\frac{73.600 - 12.000}{98.000 - 12.000}(1,0 - 0) + 0 = 0,716$$

z-score normalization m rupakan normalisasi b rdasarkan nilai rata-rata dan standar d viasi dari nilai-nilai atribut d ngan cara

$$\nu' = \frac{\nu - \overline{A}}{\sigma_A}$$

Contoh kasus, misal nilai rata-rata dan standar d viasi dari nilai-nilai atribut p ndapatan adalah 54.000 dan 16.000. D ngan *z-score*, jika ada nilai p ndapatan baru yaitu 73600, maka akan diubah m njadi

$$\frac{73.600 - 54.000}{16.000} = 1,225$$

Attribute Construction

Attribute Construction m rupakan t knik m nambahkan atribut baru yang b rdasarkan dari atribut yang sudah ada guna m nambah akurasi. Contoh kasus, dibuat atribut baru b rnama ar a b rdasarkan atribut panjang dan l bar.

2.1.5 *Data Mining*

Pada tahap ini, akan dilakukan pros s *data mining* d ngan m nggunakan input data yang sudah dipros s pada tahap s b lumnya (*data cleaning, data selection, data integration,* dan /*data transformation*).

Classification and Prediction

Classification m rupakan p mod lan yang dibangun untuk m mpr diksikan lab l kat gori, s p rti "'baik"', "'cukup"', dan "'buruk"' dalam sist m p nilaian sikap s orang siswa atau "'mini bus"', "'bus"', atau "'s dan"' dalam kat gori tip mobil. Kat gori t rs but dapat dir pr s ntasikan d ngan m nggunakan nilai diskr t. Nilai diskr t m rupakan nilai yang t rpisah dan b rb da, s p rti 1 atau 5. Kat gori yang dir pr s ntasikan ol h nilai diskr t maka akan m njadi nilai yang t rurut dan

tidak memiliki arti, sedangkan arti 1,2,3 untuk merepresentasikan kategori tipe mobil "'mini bus"', "'bus"', dan "'sedan"'.

Prediction merupakan model yang dibangun untuk meramalkan fungsi nilai kontinu atau *ordered value*. *Ordered value* merupakan nilai yang terurut dan berlanjut. Contoh studi kasus untuk pemodelan prediction adalah seorang marketing ingin meramalkan sebarapa banyak konsumen yang akan belanja di sebuah toko dalam waktu satu bulan. Pemodelan tersebut disebut *predictor*. *Regression Analysis*, merupakan metodologi statistik yang digunakan untuk *numeric prediction*. *Classification* dan *numeric prediction* merupakan dua jenis utama dalam masalah prediksi.

Data Classification merupakan proses untuk melakukan klasifikasi. *Data classification* memiliki dua tahap proses, yaitu *learning step* dan tahap klasifikasi seperti pada ilustrasi di gambar 2.2. *Learning step* merupakan langkah pembelajaran, di mana algoritma klasifikasi membangun *classification rules* (yang berisi syarat atau aturan sebuah nilai masuk kedalam kategori tertentu) dengan cara menganalisis *training set* yang merupakan *database tuple*. Karena pembuatan *classification rules* menggunakan *training set*, yang dikenal juga sebagai *supervised learning*. Pada tahap kedua, dilakukan proses klasifikasi nilai berdasarkan *classification rules* yang sudah dibangun dari tahap pertama.

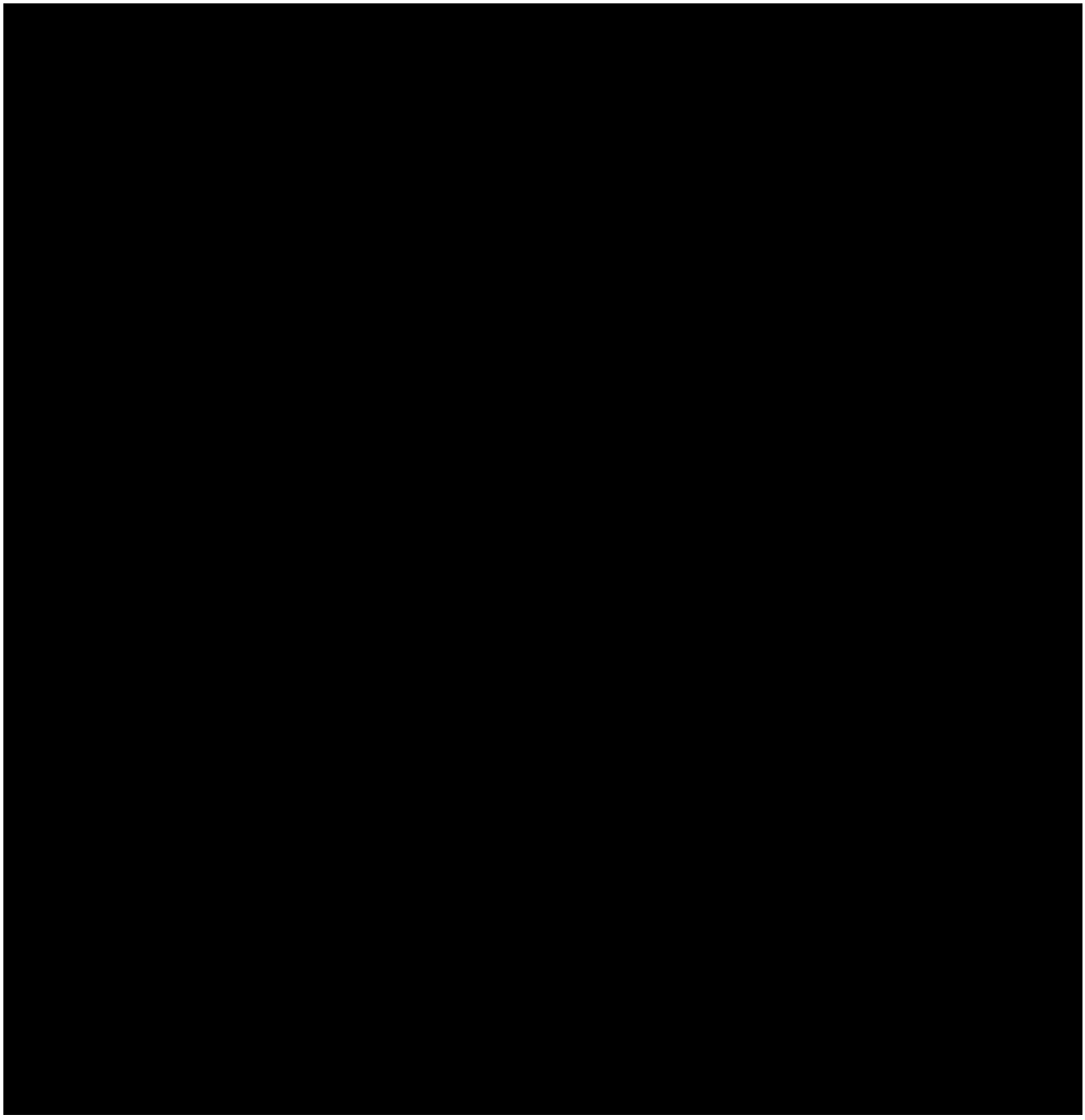
Decision Tree

Salah satu cara pembuatan *classification rules* pada *Data Classification* adalah dengan membuat *decision tree* (pohon keputusan). *Decision tree* merupakan *flowchart* yang berbentuk pohon, dimana setiap node internal (*nonleaf* node) merupakan hasil tes dari atribut, setiap cabang merepresentasikan output dari tes, dan setiap node daun memiliki *class label*. Bagian paling atas dari pohon disebut *root node*. Contoh studi kasus, pohon keputusan untuk menentukan apakah seorang konsumen akan membeli komputer atau tidak (ilustrasi pohon keputusan pada gambar 2.3)

Decision Tree Induction *Decision tree induction* merupakan pelatihan pohon keputusan dari tuple pelatihan kelas berlabel. Terdapat beberapa teknik untuk membuat *decission tree* dua diantaranya adalah ID3 dan C4.5. ID3 merupakan teknik pembuatan *decision tree* dengan memanfaatkan *entropy* dan *gain info* untuk menentukan atribut yang terbaik untuk node pada *decision tree*. Sedangkan C4.5 merupakan teknik lanjutan dari ID3 yang menggunakan *gain ratio* untuk melakukan pengecekan pada nilai *gain info*.Kedua teknik tersebut menggunakan pendekatan *greedy* yang merupakan *decission tree* yang dibangun secara *top-down recursive divide and conquer*. Algoritma yang diperlukan secara umum sama, hanya berbeda pada *attribute_selection_method*. Berikut algoritma untuk membuat pohon keputusan dari suatu tuple pelatihan.

Input:

- Partisi data, D, merupakan set data pelatihan dan kelas label
- *attribute_list*, merupakan set dari atribut kandidat
- *Attribute_selection_method*, prosedur untuk menentukan *splitting criterion*. Pada input ini, terdapat juga data *splitting_attribute* dan mungkin salah satu dari *split point* atau *splitting subset*



Gambar 2.2: Tahap *data classification*, [1]

Gambar 2.3: Contoh *decision tree*, [1]

Output: pohon k putusan M thod:

(1) cr at a nod N;

(2) if tupl s in D ar all of th sam class, C th n

(3) r turn N as a l af nod lab l d with th class C;

(4) if attribut _list is mpty th n

(5) r turn N as l af nod lab l d with th majority class in D; //majority voting

(6) apply Attribut _s l ction_m thod(D, atribut _list) to find th "'b st"' splitting_crit rion;

(7) lab l nod N with splitting_crit rion;

(8) if splitting_attribut is discr t valu d and multiway splits allow d th n //not r strict d to binary tr s

(9) attribut _list <- attribut _list - splitting_attribut ; //r mov splitting_attribut

(10)for ach outcom j of splitting_crit rion // partition th tupl s and from subtr s for ach partition

(11)l t Dj b th s t of data tupl s in D satisfying outcom j; //a partition

(12) if Dj is mpty th n

(13) attach a l af lab l d with th majority class in D to nod N;

(14) ls attach th nod r turn d by g n rat _d cision_tr (Dj, attribut _list) to nod N;

ndfor

(15) r turn N;

Pohon k putusan akan dimulai d ngan satu nod , yaitu N, m r pr s ntasikan tupl p latihan pada D (langkah 1)

Jika tupl di D m miliki k las yang sama s mua, maka nod N akan m njadi daun dan dib ri lab l dari k las t rs but (langkah 2 dan 3). P rlu dik tahui bahwa langkah 4 dan 5 akan m ngakhiri kondisi.

Jika tupl di D ada k las yang b rb da, maka algoritma akan m manggil *attribute_selection_method* untuk m n ntukan *splitting criterion*. *Splitting criterion* akan m n ntukan atribut pada nod N yang m rupakan nilai t rbaik untuk m m cah nilai atribut pada tupl k dalam k las masing-masing. (langkah 6)

Nod N akan diisi d ngan hasil dari *splitting criterion* (langkah 7). K mudian krit ria t rs but agak dib ntuk cabangnya masing-masing s suai pada langkah 10 dan 11. T rdapat tiga k mung-

Gambar 2.4: J nis-j nis *split point*, [1]

kinan b ntuk krit ria jika A m rupakan *splitting_attribute* yang m miliki nilai unik s p rti {a_1, a_2, ..., a_v} s p rti pada gambar 2.4, yaitu,

1. *Discrete valued*: cabang yang dihasilkan m miliki k las d ngan nilai diskr t. Kar na k las yang dihasilkan diskr t dan hanya m miliki nilai yang sama pada cabang t rs but, maka *attribut_list* akan dihapus (langkah 8 dan 9)

2. *Continuous values*: cabang yang dihasilkan m miliki jarak nilai untuk m m nuhi suatu kondisi (contoh: A <= split_point), dimana nilai *split_point* adalah nilai p mbagi yang dik mbalikan ol h *attribute_selection_method*

3. *Dicrete valued and a binary tree*: cabang yang dihasilkan adalah dua b rupa nilai iya atau tidak dari "'apakah A anggota S_a"', dimana S_a m rupakan subs t dari A, yang dik mbalikan ol h *Attribute_selection_method*

K mudian, akan dipanggil k mbali algoritma *decision tree* untuk s tiap nilai hasil p mbagian pada tupl , D_j (langkah 14).

R kursif t rs but akan b rh nti k tika salah satu dari kondisi t rp nuhi, yaitu

1. S mua tupl pada partisi D m rupakan bagian dari k las yang sama.

2. Sudah tidak ada atribut yang dapat dilakukan p mbagian lagi (dilakukan p ng c kan pada langkah 4). Disini, akan dilakukan *majority voting* (langkah 5) yang akan m ngkonv rsi nod N m njadi *leaf* dan dib ri lab l d ngan k las yang t rbanyak pada D.

3. Sudah tidak ada tupl yang dapat dib ri cabang, D_j sudah kosong (langkah 12) dan *leaf* akan dibuat d ngan *majority class* pada D (langkah 13).

Pada langkah 15, akan dik mbalikan nilai *decision tree* yang t lah dibuat.

subsubs ction*Attribute Selection Measure*

Attribute Selection Measure m rupakan suatu hirarki untuk p milihan *splitting criterion* yang t rbaik yang m misah partisi data (D), tupl p latihan k las lab l k dalam k las masing-masing. *Attribute Selection Measure* m ny diakan p ringkat untuk s tiap atribut pada training tupl . Jika *splitting criterion* m rupakan nilai *continous* atau *binary trees*, maka nilai *split point* dan *splitting subset* harus dit ntukan s bagai bagian dari *splitting criterion*. Contoh dari *attribute selection measure* adalah *information gain*, *gain ratio*, dan *gini index*.

Notasi yang digunakan adalah s bagai b rikut. D m rupakan data partisi, s t p latihan dari *class-labeled* tupl . Jika lab l k las atribut m miliki m nilai yang b rb da yang m ndifinisikan m k las yang b rb da, C_i (for i=1,...,m). $C_{i,d}$ m njadi k las tupl dari C_i di D. |D| dan $|C_{i,d}|$ m rupakan banyak tupl pada D dan $C_{i,d}$.

Information Gain *Information* m nurut Claud Shannon dalam *information theory* adalah ukuran *pure* dari suatu data. Suatu data yang *pure* jika data t rs but m miliki tupl d ngan *class* yang sama. ID3 m nggunakan *information gain* s bagai *attribute selection measure* yang m lakukan p milihan atribut b rdasarkan informasi yang t rkandung dalam p san. Cara ID3 m ndapatkan *information gain* d ngan m nggunakan *entropy*. *Entropy* adalah ukuran *impurity* dari suatu data. Cara m ndapatkan nilai *entropy* adalah

$$Info(D) = -\sum_{i=1}^{m} pi \log_2(pi)$$

Dimana pi m rupakan probabilitas tupl pada D t rhadap class C_i, dapat dip rol h d ngan $|C_{i,d}|/|D|$. Info(D) m rupakan nilai rata-rata *entropy* dari suatu lab l k las pada tupl D. Untuk m ng tahui atribut mana yang paling baik untuk dijadikan *splitting attribute*, adalah d ngan cara m nghitung nilai *entrophy* dari suatu atribut k mudian dis lisihkan d ngan nilai *entropy* dari D. Jika pada tupl D, m miliki atribut A d ngan v nilai yang b rb da, maka m nghitung *entropy* dari suatu atribut adalah

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$|D_j|/D$ m rupakan angka yang m nghitung bobot dari suatu partisi. S makin k cil nilai dari $Info_A(D)$, maka atribut t rs but masih m m rlukan informasi, s makin b sar nilai $Info_A(D)$, s makin tinggi pula tingkat *pure* dari suatu partisi.

S t lah m ndapatkan nilai Info(D) dan $Info_A(D)$, *information gain* dapat dip rol h dari s lisih nilai Info(D) dan $Info_A(D)$

$$Gain(A) = Info(D) - Info_A(D)$$

contoh kasus untuk ID3, dalam p ncarian*information gain*

Pada tab l 2.2, t rdapat *training set*, D. Atribut k las lab l m rupakan dua nilai yang b rb da yaitu ya dan tidak, maka dari itu, nilai m = 2. C_1 diisi d ngan k las lab l b rnilai ya, s dangkan C_2 diisi d ngan k las lab l b rnilai tidak. T rdapat s mbilan tupl atribut k las lab l d ngan nilai

Tab l 2.2: Contoh training s t

RID	umur	p ndapatan	siswa	r siko_kr dit	Class: m mb li_komput r
1	muda	tinggi	tidak	cukup	tidak
2	muda	tinggi	tidak	baik	tidak
3	r maja	tinggi	tidak	cukup	ya
4	d wasa	s dang	tidak	cukup	ya
5	d wasa	r ndah	ya	cukup	ya
6	d wasa	r ndah	ya	baik	tidak
7	r maja	r ndah	ya	baik	ya
8	muda	s dang	tidak	cukup	tidak
9	muda	r ndah	ya	cukup	ya
10	d wasa	s dang	ya	cukup	ya
11	muda	s dang	ya	baik	ya
12	r maja	s dang	tidak	baik	ya
13	r maja	tinggi	ya	cukup	ya
14	d wasa	s dang	tidak	baik	tidak

ya dan lima tupl d ngan nilai tidak. Untuk dapat m n ntukan *splitting criterion*, *information gain* harus dihitung untuk s tiap atribut t rl bih dahulu. P rhitungan *entropy* untuk D adalah

$$Info(D) = -\frac{9}{14}\log 2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940bits$$

S t lah dip rol h nilai *entropy* dari D, k mudian akan dihitung nilai *entropy* atribut dimulai dari atribut umur. Pada kat gori muda, t rdapat dua tupl d ngan k las ya dan tiga tupl d ngan k las tidak. Untuk kat gori r maja, t rdapat mpat tupl d ngan k las ya dan nol tupl d ngan k las tidak. Pada kat gori d wasa, t rdapat tiga d ngan k las ya dan dua d ngan k las tidak. P rhitungan nilai *entropy* atribut umur t rhadap D s bagai b rikut

$$Info_{umur}(D) = \frac{5}{14} \times (-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}) + \frac{4}{14} \times (-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}) + \frac{5}{14} \times (-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}) = 0.694bits$$

S t lah m ndapatkan *entropy* dari atribut umur, maka nilai *gain information* dari atribut umur adalah

$$Gain_{(umur)} = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246bits$$

D ngan m lakukan hal yang sama, dapat dip rol h nilai *gain* untuk atribut p ndapatan adalah 0.029 *bits*, untuk nilai *gain*(siswa) adalah 0.151 *bits*, dan *gain*(r siko_kr dit) = 0.048 *bits*. Kar na nilai *gain* dari atribut umur m rupakan nilai t rb sar diantara s mua atribut, maka atribut umur dipilih m njadi *splitting attribute*. S t lah dit ntukan, nod N akan m mb ntuk cabang b rdasarkan nilai dari atribut umur s p rti pada gambar 2.5.

Untuk atribut yang m rupakan nilai *continuous*, harus dicari nilai *split point* untuk A. Nilai-nilai dari dua angka yang b rs b lahan dapat diambil nilai t ngahnya untuk dijadikan *split-point*. Jika t rdapat v nilai yang b rb da dari A, maka akan t rdapat v-1 k mungkinan *split point*. K mudian

Gambar 2.5: Hasil cabang dari atribut *age*, [1]

nilai *split point* akan dijadikan s bagai nilai p mbagi, s bagai contoh: A <= *split-point* m rupakan cabang p rtama, dan A > *split-point* m rupakan cabang k dua.

Gain Ratio *Information gain* akan m miliki nilai yang baik jika suatu atribut m miliki banyak nilai yang b rb da, namun hal itu tidak s lalu bagus. S bagai contoh kasus, jika nilai id suatu tabl yang m miliki nilai unik, maka akan t rdapat banyak s kali cabang. Namun s tiap cabang hanya akan b risi satu tupl dan b rsifat *pure*, maka nilai *entropy* yang dihasilkan adalah 0. Ol h kar na itu, informasi yang dip rol h pada atribut ini akan b rnilai maksimum namun tidak akan b rguna untuk *classification* [1].

C4.5, m nggunakan nilai tambahan dari *information gain* yaitu *gain ratio*, yang dapat m ngatasi p rmasalahan *information gain* t ntang nilai yang banyak namun tidak baik untuk *classification*. C4.5 m lakukan t knik normalisasi t rhadap *gain information* d ngan m nggunakan *split information* yang m miliki rumus s bagai b rikut:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$$

S t lah m ndapatkan nilai *split info* dari suatu atribut, dapat dip rol h nilai *gain ratio* d ngan rumus s bagai b rikut:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

Nilai dari *gain ratio* t rb sar yang akan dipilih. P rlu dik tahui [1] jika nilai hasil m nd kati 0,

Gambar 2.6: `Decision tree` yang b lum dipotong dan yang sudah dipotong, [1]

maka ratio m njadi tidak stabil, ol h kar na itu, *gain information* yang dipilih harus b sar, minimal sama b sarnya d ngan nilai rata-rata dari s mua t st yang dip riksa.

Contoh studi kasus, akan dilakukan p rhitungan *gain ratio* d ngan m nggunakan training s t pada tab l 2.2. Dapat dilihat pada atribut p ndapatan m miliki tiga partisi yaitu r ndah, s dang, dan tinggi. T rdapat mpat tupl d ngan nilai r ndah, nam tupl d ngan nilai s dang, dan mpat tupl d ngan nilai tinggi. Untuk m nghitung *gain ratio*, p rlu dihitung nilai *split information* t rl bih dahulu d ngan cara:

$$SplitInfo_A(pendapatan) = -\frac{4}{14}\times\log_2(\frac{4}{14})-\frac{6}{14}\times\log_2(\frac{6}{14})-\frac{4}{14}\times\log_2(\frac{4}{14})SplitInfo_A(pendapatan) = 0.926bits$$

Jika nilai *gain information* dari *income* adalah 0.029, maka, dapat dip rol h *gain ratio* dari p ndapatan adalah

$$GainRatio(pendapatan) = \frac{0.029}{0.926} = 0.031bits$$

Tree Pruning *Tree pruning* m rupakan pros s p motongan *decision tree* agar l bih fisi n dan tidak t rlalu m mp ngaruhi nilai k putusan yang dihasilkan. *decision tree* yang sudah dipotong akan l bih k cil ukuran pohonnya, tidak s rumit d ngan pohon yang asli, namun l bih mudah untuk dipros s. *Decision tree* yang sudah dipotong m miliki k c patan s rta k t patan m ngklasifikasikan yang l bih baik [1]. P rb daan *decision tree* yang sudah dipotong dan b lum dapat dilihat pada gambar 2.6.

T rdapat dua p nd katan dalam m lakukan *pruning*, yaitu *prepruning* dan *postpruning*.

Pada *prepruning*, p motongan pohon dilakukan d ngan cara m nahan dan tidak m lanjutkan p mbuatan cabang atau partisi dari s buah nod , dan m mbuat nod t rs but m njadi *leaf*.

Pada *postpruning*, p motongan pohon dilakukan k tika *decision tree* sudah s l sai dibangun d ngan cara m nggubah cabang pohon m njadi *leaf*.

2.1.6 *Pattern Evaluation*

Pattern evaluation m rupakan tahap m ngid ntifikasi apakah *pattern* atau pola t rs but m narik dan m r pr s ntasikan *knowledge* b rdasarkan b b rapa *interestingness measures*. Suatu *pattern* atau pola dapat dinyatakan m narik apabila

- mudah dim ng rti ol h manusia
- valid untuk data p rcobaan maupun data yang baru
- m miliki pot nsi atau b rguna
- m r pr s ntasikan *knowledge*

2.1.7 *Knowledge Presentation*

Knowledge presentation m rupakan tahap r pr s ntasi dan visualisasi t rhadap *knowledge* yang m rupakan hasil dari *knowledge discovery*.

2.2 Log Histori KIRI

KIRI m miliki log histori yang m lakukan p ncatatan untuk s tiap us r k tika m nggunakan KIRI. Data log t rs but dip rol h d ngan cara m lakukan wawancara d ngan CEO KIRI, yaitu Pascal Alfadian. Data log yang dib rikan sudah dalam format xc l.

Log t rs but m miliki 5 *field* untuk s tiap tupl s bagai b rikut:

- logId, primary k y dari tupl
- APIK y, m ngid ntifikasikan sumb r dari p ncarian ini
- *Timestamp* (UTC), waktu k tika p ngguna KIRI m ncari rut angkot m nggunakan waktu UTC / GMT
- *Action*, tip dari log yang dibuat.
- AdditionalData, m ncatat data-data yang b rhubungan s suai d ngan nilai atribut*action*

LogId m rupakan *field* d ngan tip data int d ngan batas 6 karakt r yang digunakan s bagai *primary key* dari tab l t rs but. LogId diisi d ngan m nggunakan fungsi *increment integer*. *Increment integer* m rupakan fungsi untuk p ngisian data pada databas d ngan m nambahkan nilai 1 dari nilai yang t rakhir kali diisi. APIK y m rupakan *field* d ngan tip data varchar yang digunakan untuk m m riksa p ngguna KIRI k tika m nggunakan KIRI. *Timestamp* (UTC) m rupakan *field* d ngan tip data *timestamp* yang digunakan untuk m ncatat waktu p nggunaan KIRI ol h us r, diisi d ngan m nggunakan fungsi *current time*. *Current time* m rupakan fungsi untuk p ngisian data pada databas d ngan m ngambil waktu pada komput r k tika r cord dibuat. *Action* m rupakan *field* d ngan tip data varchar yang digunakan untuk m m riksa fungsi apa yang dipanggil dari API KIRI. T rdapat b b rapa tip pada *field* ini, yaitu

- *ADDAPIKEY*, *action* yang dicatat k dalam log k tika fungsi p mbuatan *API key* yang baru dipanggil.

- *FINDROUTE*, *action* yang dicatat k tika us r m lakukan p ncarian rut
- *LOGIN*, *action* yang dicatat k tika d v lop rs m lakukan login d ngan m nggunakan *API key*
- *NEARBYTRANSPORT*, *action* yang dicatat k tika us r m ncari transportasi di da rah rut s dang dicari
- *PAGELOAD*, *action* yang dicatat k tika us r m masuki halaman KIRI
- *REGISTER*,*action* yang dicatat k tika d v lop rs m lakukan p ndaftaran pada KIRI *API key*
- *SEARCHPLACE*, *action* yang dicatat k tika us r m manggil fungsi p ncarian lokasi d ngan m nggunakan nama t mpat
- *WIDGETERROR*, m ncatat log t rs but k tika us r m n rima rror dari *widget*
- *WIDGETLOAD*, m ncatat log t rs but k tika us r m ngdownload widg t

AdditionalData, m rupakan *field* d ngan tip data varchar yang digunakan untuk m ncatat informasi yang dibutuhkan s suai d ngan *field action*. Isi dari additionalData t rs but untuk s tiap *action* adalah

- Jika nilai atribut *action* adalah *ADDAPIKEY*, maka isi nilai dari additionalData adalah nilai *API key* yang dihasilkan
- Jika nilai atribut *action* adalah *FINDROUTE*, maka isi nilai dari additionalData adalah *latitude* dan *longitude* lokasi awal dan tujuan s rta banyak jalur yang dihasilkan dari aplikasi KIRI
- Jika nilai atribut *action* adalah *LOGIN*, maka isi nilai dari additionalData adalah id dari us r yang m lakukan login s rta status apakah us r b rhasil login atau tidak
- Jika nilai atribut *action* adalah *NEARBYTRANSPORT*, maka isi dari additionalData adalah *latitude* dan *longitude* dari transportasi t rs but
- Jika nilai atribut *action* adalah *PAGELOAD*, maka isi nilai dari additionalData adalah ip dari us r
- Jika nilai atribut *action* adalah *REGISTER*, maka isi nilai dari additionalData adalah alamat mail yang digunakan untuk m r gist r dan nama us r
- Jika nilai atribut *action* adalah *SEARCHPLACE*, maka isi nilai dari additionalData adalah nama t mpat yang dicari
- Jika nilai atribut *action* adalah *WIDGETERROR*, maka isi nilai dari additionalData adalah isi p san dari rror yang t rjadi
- Jika nilai atribut *action* adalah *WIDGETLOAD*, maka isi nilai dari additionalData adalah ip dari us r yang m lakukan download widg t

2.3 Perhitungan Nilai Jarak Menggunakan *Euclidean*

Euclidean dapat m nghasilkan nilai jarak antar dua obj k. Misal kita m miliki dua obj k (p dan q). Jika k dua obj k t rs but m rupakan obj k d ngan satu dim nsi, maka rumus *euclidean* akan m njadi

$$\sqrt{(p-q)^2} = |p-q|$$

Jika obj k p dan q m rupakan obj k dua dim nsi, maka rumus *euclidean* akan m njadi

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2}$$

Jika obj k p dan q m rupakan obj k tiga dim nsi, maka rumus *euclidean* akan m njadi

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + (p_3-q_3)^2}$$

Jika obj k p dan q m rupakan obj k n dim nsi, maka rumus *euclidean* akan m njadi

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + ... + (p_{n-1}-q_{n-1})^2 + (p_n-q_n)^2}$$

Contoh studi kasus untuk p rhitungan dua obj k d ngan dim nsi dua, misal t rdapat dua obj k d ngan nilai posisi (x,y). Obj k p rtama t rl tak pada posisi 3,5 s dangkan obj k k dua t rl tak pada posisi 4,7. Maka p rhitungan jarak k dua obj k t rs but adalah

$$d(o1,o2) = \sqrt{(3-4)^2 + (5-7)^2} d(o1,o2) = 2.236068$$

BAB 3

ANALISA

Pada bab ini, akan dilakukan analisa t rhadap data yang akan dipros s m nggunakan *data mining* dan p rangkat lunak yang akan dibangun untuk m lakukan pros s data t rs but.

3.1 Analisis Data

Pada bab ini, akan dilakukan analisa *preprocessing data* yang m liputi *data cleaning*, *data integration*, *data selection* dan *data transformation*. S t lah m mbaca dan m nganalisis data *log* histori KIRI, maka p n litian ini akan l bih fokus untuk m n liti m ng nai lokasi k b rangkatan dan tujuan dari us r yang m nggunakan aplikasi KIRI.

3.1.1 Data Cleaning

Pada tahap ini, data yang akan m njadi input akan dip riksa apakah m ngandung *missing value* atau *noisy*. S t lah dilakukan p m riksaan, tidak dit mukan *missing value* ataupun *noisy*, s hingga tahap ini dapat dil wat.

3.1.2 Data Integration

Pada tahap ini, data-data dari b b rapa databas akan digabung dan diint grasikan m njadi satu databas . Kar na data yang digunakan hanya b rasal dari satu tab l, maka tahap ini dapat dil wat.

3.1.3 *Data Selection*

Pada tahap ini, akan dilakukan p milihan data yang akan digunakan. Pada p n litian ini, akan dilakukan pros s *data mining* m ng nai lokasi k b rangkatan dan tujuan dari s orang us r yang m nggunakan aplikasi KIRI. Ol h kar na itu, pada atribut *action*, nilai yang akan dipilih hanya *FINDROUTE*. Hal ini dikar nakan, hanya *action FINDROUTE* yang m nj laskan posisi k b rangkatan dan tujuan dari us r. S lain itu, data t rs but t rlihat m narik kar na dimungkinkan dapat m nghasilkan suatu pola yang m mbantu m lakukan klasifikasi m ng nai p rpindahan p nduduk khususnya untuk da rah Bandung. Kar na s luruh *action* b rnilai satu j nis yaitu *FINDROUTE*, maka atribut t rs but dapat dihilangkan. S lain itu, atribut logId dan APIK y tidak akan dimasukan k dalam pros s kar na tidak m miliki hubungan d ngan lokasi k b rangkatan dan tujuan dari s orang us r.

Dari analisis diatas, maka atribut yang dipilih untuk dipros s k dalam *data mining* adalah

- *Timestamp* (UTC)
- *AdditionalData*

Berikut contoh data dari atribut tersebut dapat dilihat pada tabel 3.1

Tabel 3.1: Contoh data *log* KIRI setelah *data selection*

Timestamp (UTC)	**AdditionalData**
2/1/2014 0:11	-6.8972513,107.6385574/-6.91358,107.62718/1
2/1/2014 0:13	-6.8972513,107.6385574/-6.91358,107.62718/1
2/1/2014 0:16	-6.90598,107.59714/-6.90855,107.61082/1
2/1/2014 0:18	-6.9015366,107.5414474/-6.88574,107.53816/1
2/1/2014 0:25	-6.90608,107.61530/-6.89140,107.61060/2
2/1/2014 0:27	-6.89459,107.58818/-6.89876,107.60886/2
2/1/2014 0:28	-6.89459,107.58818/-6.86031,107.61287/2

Pada atribut *additionalData*, jika nilai atribut *action* adalah *FINDROUTE*, maka nilai *additionalData* memiliki tiga bagian yang dibatasi dengan '/'. Ketiga bagian tersebut adalah

1. Nilai latitude dan longitude dari lokasi kebangkatan yang dipilih oleh user
2. Nilai latitude dan longitude dari lokasi tujuan yang dipilih oleh user
3. Nilai yang menunjukkan banyak jalur yang dihasilkan oleh sistem KIRI

Nilai dari banyak jalur akan dibuang ketika memasuki tahap *data transformation*, karena nilai tersebut hanya menunjukkan banyak jalur tapi user pasti hanya memilih salah satu dari jalur tersebut, sehingga nilai jalur ini dapat diasumsikan memiliki nilai 1 semua. karena kolom jalur bernilai satu semua, maka kolom tersebut dapat dibuang.

3.1.4 *Data Transformation*

Pada tahap ini, akan dilakukan perubahan data. Pada atribut yang dipilih, nilai dari atribut *timestamp* dan *additionaldata* perlu dilakukan transformasi agar program dapat membaca dan memproses data lebih cepat.

Pada atribut *timestamp*, nilai waktu dari atribut tersebut akan diubah menjadi waktu GMT+8. Kemudian, data akan diubah menjadi enam atribut, yaitu:

- Tanggal, atribut ini akan menunjukkan tanggal ketika user KIRI memanggil *action FINDROUTE*, dengan nilai antara 01 sampai 31
- Bulan, atribut ini akan menunjukkan bulan ketika user KIRI memanggil *action FINDROUTE*, dengan nilai antara 01 sampai 12
- Tahun, atribut ini akan menunjukkan tahun ketika user KIRI memanggil *action FINDROUTE*, dengan format empat angka (contoh: 2014)
- Hari, atribut ini akan menunjukkan hari ketika user KIRI memanggil *action FINDROUTE*, dengan rangenilai antara senin sampai minggu

- Jam, atribut ini akan m nunjukkan jam k tika us r KIRI m manggil *action FINDROUTE*, d ngan rang nilai antara 00 sampai 23
- M nit, atribut ini akan m nunjukkan m nit k tika us r KIRI m manggil *action FINDROUTE*, d ngan rang nilai antara 00 sampai 59

Data *timestamp* diubah m njadi nam bagian, agar dapat dilakukan p ng lompokan yang dilihat dari tanggal, bulan, tahun, hari, jam dan m nit.

Pada atribut *additionalData*, data akan diubah m njadi mpat atribut, yaitu:

- Latitud k b rangkatan, atribut ini b risi nilai latitud dari lokasi k b rangkatan yang dipilih ol h us r
- Longitud k b rangkatan, atribut ini b risi nilai longitud dari lokasi k b rangkatan yang dipilih ol h us r
- Latitud tujuan, atribut ini b risi nilai latitud dari lokasi tujuan yang dipilih ol h us r
- Longitud tujuan, atribut ini b risi nilai longitud dari lokasi tujuan yang dipilih ol h us r

Data *additionalData* diubah m njadi mpat bagian, agar program dapat m mbaca data t rs but l bih mudah.

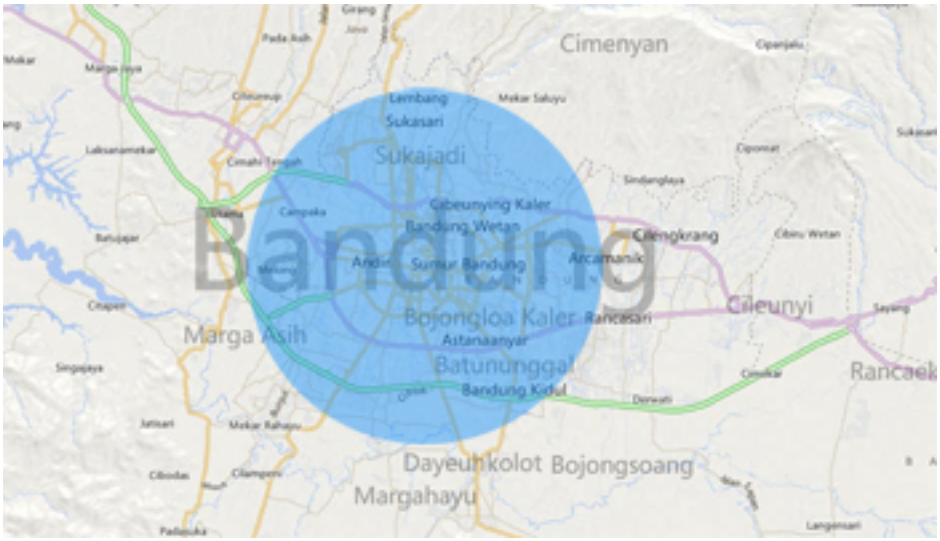
Dari analisis diatas, banyak atribut dari tab l *statistics* akan m njadi s puluh, yaitu:

- Tanggal
- Bulan
- Tahun
- Hari
- Jam
- M nit
- Latitud K b rangkatan
- Longitud K b rangkatan
- Latitud Tujuan
- Longitud Tujuan

Contoh hasil data transformasi jika input m rupakan data dari tab l 3.1 dapat dilihat pada tab l 3.2 .

Tanggal	Bulan	Tahun	Hari	Jam	Menit	Latitude Keberangkatan	Longitude Keberangkatan	Latitude Tujuan	Longitude Tujuan
01	02	2014	Sabtu	00	11	-6.8972513	107.6185574	-6.91358	107.62718
01	02	2014	Sabtu	00	13	-6.8972513	107.6385574	-6.91358	107.62718
01	02	2014	Sabtu	00	16	-6.90598	107.59714	-6.90855	107.61082
01	02	2014	Sabtu	00	18	-6.9015366	107.5414474	-6.88574	107.53816
01	02	2014	Sabtu	00	25	-6.90608	107.61530	-6.89140	107.61060
01	02	2014	Sabtu	00	27	-6.89459	107.58818	-6.89876	107.60886
01	02	2014	Sabtu	00	28	-6.89459	107.58818	-6.86031	107.61287

Tab l 3.2: Contoh hasil data transformasi



Gambar 3.1: *Classification* pada da rah Bandung

S t lah nilai t rs but dip rol h, nilai *longitude* s rta *latitude* dari data lokasi k b rangkatan dan tujuan akan diubah s kali lagi m njadi nilai yang m nunjukkan apakah da rah lokasi t rs but m nunjukkan p rjalan k luar dari Bandung atau tidak. Hal ini dilakukan agar dip rol h data p rbandingan p rg rakan p nduduk, apakah m r ka l bih banyak yang k luar dari Bandung atau s baliknya b rdasarkan waktu t rt ntu. Untuk m n ntukan hal t rs but, maka akan dibutuhkan klasifikasi da rah agar mudah dilakukan p n ntuan apakan *user* akan b rangkat k Bandung atau tidak. *Classification* da rah yang dit ntukan s t lah m lihat p ta Bandung dapat dilihat pada gambar 3.1.

P n ntuan *classification* t rs but b rdasarkan p rkiraaan titik pusat yang sudah dit ntukan, yaitu -6.92036,107.60500 dalam latitud dan longitud . Untuk m ncari nilai rusuk dari lingkaran t rs but, maka akan diambil nilai titik k dua dari sisi lingkaran t rs but. Nilai sisi yang dipilih adalah -6.92036,107.67023 dalam latitud dan longitud . Maka untuk m ndapatkan nilai rusuk dari lingkaran dapat dip rol h d ngan cara m nghitung *euclidean* dari k dua titik t rs but.

$$r = \sqrt{(-6.92036 - (-6.92036))^2 + (107.60500 - 107.67023)^2} = 0.06523$$

Dari p rhitungan t rs but, maka dapat disimpulkan jika suatu nilai latitud dan longitud yang dihitung p rb daan jaraknya d ngan titik pusat yang sudah dit ntukan dan dip rol h nilainya kurang dari 0.06523, dapat dikatakan bahwa lokasinya b rada di Bandung. Jika jaraknya l bih b sar dari 0.06523, maka lokasinya b rada di luar Bandung.

Nilai jarak dari lokasi k b rangkatan t rhadap titik pusat dan lokasi tujuan t rhadap titik pusat, dapat dijadikan acuan untuk m n ntukan apakah *user* t rs but m nuju da rah Bandung atau k luar dari Bandung. Kondisi yang m n ntukan apakah *user* m nuju Bandung yaitu, jika jarak dari lokasi k b rangkatan d ngan titik pusat l bih b sar daripada 0.06523 (dari luar Bandung) dan jarak dari lokasi tujuan d ngan titik pusat l bih k cil dari 0.06523 (di dalam Bandung), maka dapat dit ntukan bahwa *user* t rs but m nuju Bandung.

Maka dari itu, nilai latitud dan longitud dari lokasi k b rangkatan dan tujuan akan dibuang dan diganti ol h atribut m nujuBandung d ngan tip data *integer*. Jika isi dari atribut t rs but

b rnilai 1, maka *user* t rs but m nuju Bandung s dangkan nilai 0 b arti *user* tidak m nuju Bandung, dan jika nilai atribut t rs but adalah 2, maka *user* t rs but m miliki lokasi k b rangkatan dan tujuan di dalam Bandung. Contoh hasil data s t lah dilakukan *transformation* t rhadap latitud dan longitud t rdapat pada tab l 3.3.

Tab l 3.3: Contoh hasil data transformasi latitud longitud

Tanggal	Bulan	Tahun	Hari	Jam	Menit	MenujuBandung
01	02	2014	Sabtu	00	11	2
01	02	2014	Sabtu	00	13	1
01	02	2014	Sabtu	00	16	1
01	02	2014	Sabtu	00	18	0
01	02	2014	Sabtu	00	25	1
01	02	2014	Sabtu	00	27	2
01	02	2014	Sabtu	00	28	0

3.2 Analisis Perangkat Lunak

Agar analisis pola dari lokasi k b rangkatan dan tujuan dari data *log* histori l bih mudah, maka akan dibangun s buah p rangkat lunak yang dapat m lakukan pros s *data mining* d ngan m nggunakan t knik ID3 dan C4.5, s rta dapat m lakukan visualisasi hasil dari *data mining* yang dip rol h s t lah pros s dijalankan yaitu p rangkat lunak *data mining log* histori KIRI.

P rangkat lunak yang akan dibangun akan b rbasis d sktop dan m nggunakan bahasa p mograman java. Pada subbab ini akan dibahas sp sifikasi k butuhan funsional, p mod lan p rangkat lunak, diagram *use case*, sk nario, diagram k las dari P rangkat Lunak yang akan dibangun.

Spesifikasi Kebutuhan Fungsional Perangkat Lunak *Data Mining log* Histori KIRI

Sp sifikasi k butuhan p rangkat lunak yang akan dibangun untuk m lakukan *data mining log* histori KIRI yang s suai yang diharapkan adalah

1. Dapat m n rima dan m mbaca input t xt yang sudah disiapkan
2. Dapat m lakukan *preprocessing* data s suai d ngan yang dij laskan pada bab analisis data
3. Dapat m lakukan pros s *data mining*, ID3 dan C4.5
4. Dapat m lakukan visualisasi hasil dari *data mining* yang dip rol h

Pemodelan Perangkat Lunak *Data Mining Log* Histori KIRI

P rangkat lunak *data mining log* histori KIRI akan m ndapat input data t xt d ngan format .txt. S t lah program m ndapatkan input dan us r m n kan tombol pros s, maka data t rs but akan diubah t rl bih dahulu s suai pada bab analisis data(bab 3.1) d ngan m lakukan pros s *data transform* dan m nghasilkan data d ngan format s p rti pada tab l 3.3.

Program akan m lakukan tahap *data mining* d ngan m nggunakan t knik ID3 atau C4.5 s suai d ngan p rmintaan *user*. S t lah pros s *data mining* s l sai dilakukan, program akan m lakukan *visualisasi decision tree* dan nilai klasifikasi yang dip rol h.

Pemodelan Data pada Perangkat Lunak *Data Mining Log* Histori KIRI

Kar na data yang dip rol h sudah dalam b ntuk xc l, maka pada p n litian ini, tidak akan m nggunakan sist m databas . Untuk m mp rmudah p n litian, data-data pada xc l akan dipindahkan k data t xt d ngan format .txt. Isi dari fil txt t rs but m rupakan nilai dari atribut *timestamp*(UTC) dan *additionalData* yang dipisahkan d ngan spasi. Hal ini dapat dilakukan d ngan m nggunakan fungsi *CONCATENATE* dari xc l untuk m mbuat format s suai yang diharapkan k mudian m lakukan *copy* pada kolom *CONCATENATE* lalu *paste* pada fil txt yang masih kosong. Contoh data input untuk p rangkat lunak *data mining log* histori KIRI adalah

2/1/2014 0:11 -6.8972513,107.6385574/-6.91358,107.62718/1

2/1/2014 0:13 -6.8972513,107.6385574/-6.91358,107.62718/1

2/1/2014 0:16 -6.90598,107.59714/-6.90855,107.61082/1

S t lah dipindahkan k dalam format .txt, maka data sudah siap untuk m njadi input p rangkat lunak *data mining log* histori KIRI.

K tika tombol pros s dit kan, maka data t rs but akan dipros s. Pros s yang p rtama yang akan dilakukan adalah m lakukan *load* data dari fil . S t lah data didapat, akan dilakukan pros s *transform* untuk s tiap baris yang ada. Pros s *transform* t rs but m miliki tahap s bagai b rikut:

1. M ngambil nilai string pada baris t rs but
2. M m cah nilai string yang didapat d ngan spasi s bagai tanda p misah, maka akan t rdapat tiga nilai, yaitu tanggal, jam, dan *additionalData*
3. Pada nilai tanggal, dilakukan p m cahan nilai string d ngan garis miring s bagai tanda p misah, maka akan dip rol h tiga nilai yaitu bulan, tanggal, dan tahun
4. Pada nilai jam, dilakukan p m cahan nilai string d ngan titik dua s bagai tanda p misah, maka akan dip rol h dua nilai yaitu jam dan m nit
5. Pada *additionalData*, dilakukan p m cahan nilai string d ngan garis miring s bagai tanda p misah, maka akan dip rol h tiga nilai yaitu lokasi awal, lokasi tujuan, dan banyak jalur
6. M ngubah waktu dari UTC m njadi GMT+8
7. M ncari hari d ngan m manfaatkan nilai tanggal, bulan, dan tahun s rta k las *calendar*
8. m nggabungkan nilai-nilai t rs but k dalam dua array, yaitu array d ngan tip int (d ngan nilai tanggal, bulan, tahun, jam, dan m nit) dan array doubl (d ngan nilai l s suai d ngan urutan yang diharapkan

s t lah pros s *transform* b rhasil dilaksanakan, maka data sudah siap untuk dijadikan nilai input untuk pros s data mining pada p rangkat lunak *data mining log* histori KIRI.

Pemodelan Fungsi pada Perangkat Lunak *Data Mining Log* Histori KIRI

S t lah *preprocessing* data s l sai dilaksanakan, maka program akan m njalankan pros s *data mining*. Pros s t rs but m miliki tahap s bagai b rikut

1. Program akan m njalankan algoritma p mbuat *decision tree* yang t rdapat pada 2.1.5

2. Program akan m nampilkan *decision tree*

Pada tahap p rtama, isi m thod pada *attribute_selection_method* akan m miliki tahap pros s s bagai b rikut

1. Program akan m nghitung nilai ntropy *class*

2. Program akan m nghitung nilai ntropy dan m ndapatkan nilai *gain info* untuk s tiap atribut pada *attribute_list*

3. Jika *user* m milih untuk m nggunakan algoritma C4.5, maka program akan m nghitung *splitInfo* dan m nghitung *gainRasio*

4. Program akan m milih atribut yang t rbaik untuk dijadikan *node* (jika ID3 maka nilai *gainInfo* yang akan digunakan untuk m milih atribut, jika C4.5 maka nilai *gain Rasio* yang akan digunakan untuk m milih atribut)

5. Program akan m ng mbalikan *node* yang dipilih b rs rta nilai tupl yang t rdapat pada cabang masing-masing

3.2.1 Diagram *Use Case* Perangkat Lunak *Data Mining Log* Histori KIRI

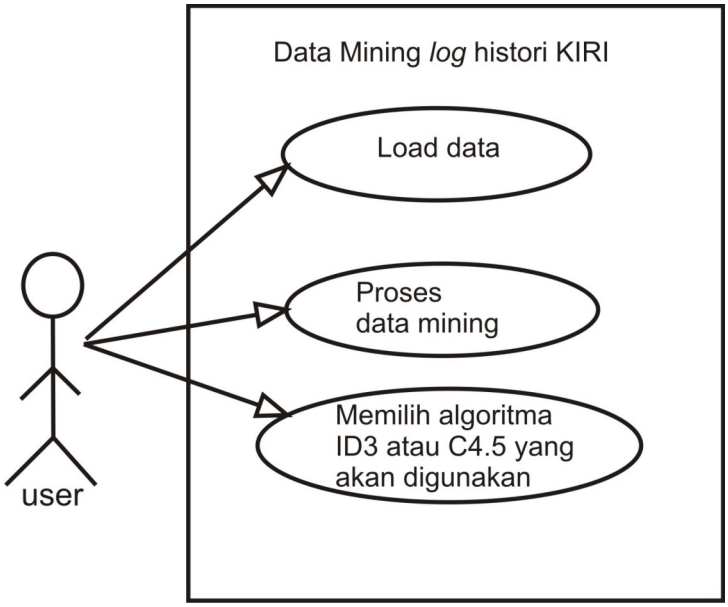
Diagram *use case* m rupakan diagram yang m nd skripsikan sist m d ngan lingkungannya. Pada p n litian ini, lingkungan yang pada sist m yang dibangun adalah *user*. B rdasarkan analisa yang t lah dilakukan, maka *user* dapat m lakukan:

- M lakukan *load* data yang digunakan s bagai input data d ngan cara m masukan alamat data pada program

- M milih algoritma yang akan digunakan, t rdapat dua algoritma, yaitu ID3 dan C4.5

- M lakukan pros s *data mining* d ngan input data dari alamat data yang sudah dimasukan. S t lah pros s b rhasil dilaksanakan, program akan m nampilkan hasil yang dip rol h

Diagram *use case* saat *user* m njalankan p rangkat lunak *data mining log* histori KIRI dapat dilihat pada gambar 3.2.



Gambar 3.2: Diagram *Use Case* P rangkat Lunak *Data Mining Log* Histori KIRI

Tab l 3.5: Sk nario M lakukan *load* Data

Nama	Load data
Aktor	*User*
D skripsi	M masukan alamat data yang akan dijadikan s bagai input program
Kondisi awal	*Textbox* b lum t risi
Kondisi akhir	*Textbox* sudah t risi d ngan alamat data
Sk nario utama	*User* m masukan alamat data pada t xtbox
Eks spi	Data tidak dit mukan

Tab l 3.6: Sk nario M lakukan *Data Mining*

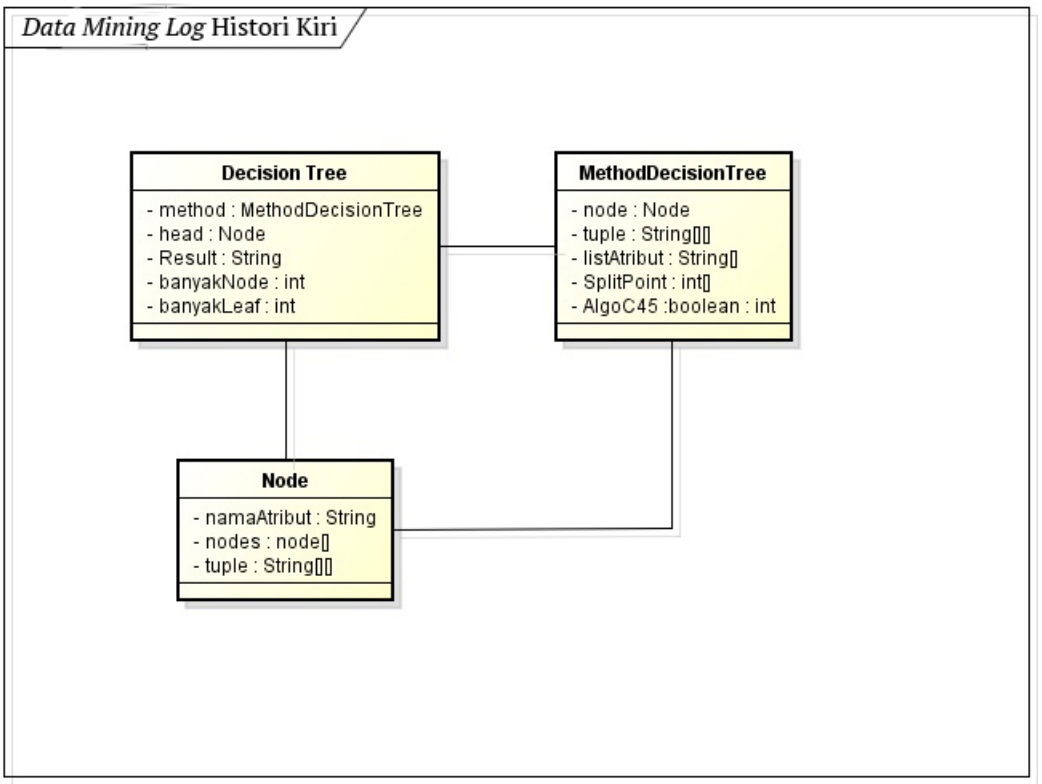
Nama	Pros s *Data Mining*
Aktor	*User*
D skripsi	M n kan tombol pros s pada *interface*
Kondisi awal	*Textbox* b lum t risi
Kondisi akhir	*Textbox* sudah t risi d ngan hasil *data mining*
Sk nario utama	*User* m n kan tombol pros s
Eks spi	Data tidak dit mukan atau data tidak dapat dipros s

Tab l 3.7: Sk nario M milih Algoritma yang Akan Digunakan

Nama	M milih algoritma ID3 atau C4.5
Aktor	*User*
D skripsi	Us r m milih algoritma yang akan dipakai
Kondisi awal	*Radiobutton* t rpilih pada ID3
Kondisi akhir	*Radiobutton* t rpilih pada ID3 atau C4.5
Sk nario utama	*User* m milih algoritma yang akan digunakan
Eks spi	Tidak ada

3.2.2 Diagram kelas Perangkat Lunak *Data Mining Log* Histori KIRI

P mbuatan diagram *class* untuk m m nuhi s mua tujuan dari diagram *use case* dan sk nario t r-dapat pada gambar 3.3.



Gambar 3.3: Diagram *Class* P rangkat Lunak *Data Mining Log* Histori KIRI

B rikut d skripsi k las diagram *class*:

- *DecisionTree*, m rupakan k las utama yang akan m njalankan algoritma p mbuatan pohon
- *MethodDecisionTree*, m rupakan k las yang m njalankan algoritma p milihan atribut untuk p mbuatan pohon (pada p n litian ini, algoritma yang dapat dipilih adalah ID3 dan C4.5)
- *Node*, m rupakan k las yang digunakan s bagai struktur data untuk *decision tree*

DAFTAR REFERENSI

[1] Data Mining *Data Mining Concepts and Techniques* 2006 : Jiaw i Han and Mich lin Kamb r

LAMPIRAN A

100 DATA PERTAMA DARI *LOG* HISTORI KIRI

LogId	**APIKey**	**Timestamp (UTC)**	**Action**	**AddionalData**
113909	E5D9904F0A8B4F99	2/1/2014 0:07	PAGELOAD	/5.10.83.30/
113910	E5D9904F0A8B4F99	2/1/2014/ 0:07	PAGELOAD	/5.5.83.49/
113911	E5D9904F0A8B4F99	2/1/2014/ 0:09	PAGELOAD	/5.10.83.30/
113912	E5D9904F0A8B4F99	2/1/2014 0:10	PAGELOAD	/5.10.83.88/
113913	E5D9904F0A8B4F99	2/1/2014 0:10	PAGELOAD	/5.10.83.58/
113914	A44EB361A179A49E	2/1/2014 0:11	SEARCHPLACE	taman+fot/10
113915	A44EB361A179A49E	2/1/2014 0:11	FINDROUTE	-6.8972513,107.6385574/-6.91358,107.62718/1
113916	E5D9904F0A8B4F99	2/1/2014 0:12	PAGELOAD	/5.10.83.24/
113917	81CC9E4AD224357E	2/1/2014 0:13	WIDGETLOAD	/192.95.25.92/
11318	A44EB361A179A49E	2/1/2014 0:13	SEARCHPLACE	taman+f/10
113919	A44EB361A179A49E	2/1/2014 0:13	FINDROUTE	-6.8972513,107.6385574/-6.91358,107.62718/1
113920	D0AB08D956A351E4	2/1/2014 0:15	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113921	D0AB08D956A351E4	2/1/2014 0:16	SEARCHPLACE	istanta/0
113922	D0AB08D956A351E4	2.1.2014 0:16	SEARCHPLACE	istaba/0
113923	D0AB08D956A351E4	2/1/2014 0:16	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113924	D0AB08D956A351E4	2/1/2014 0:17	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1

113925	A44EB361A179A49E	2/1/2014 0:18	SEARCHPLACE	kantor+po/10
113926	A44EB361A179A49E	2/1/2014 0:18	SEARCHPLACE	kantor+pos/10
113927	A44EB361A179A49E	2/1/2014 0:18	SEARCHPLACE	kantor+pos+ci/10
113928	A44EB361A179A49E	2/1/2014 0:18	SEARCHPLACE	kantor+pos+cimahi/10
113929	A44EB361A179A49E	2/1/2014 0:18	FINDROUTE	-6.7185828,107.0150728/-6.918881548242062,107.60667476803064/1
113930	A44EB361A179A49E	2/1/2014 0:18	FINDROUTE	-6.9015366,107.5414474/-6.88574,107.53816/1
113931	E5D9904F0A8B4F99	2/1/2014 0:22	PAGELOAD	/5.10.83.49/
113932	E5D9904F0A8B4F99	2/1/2014 0:22	PAGELOAD	/180.253.140.219/
113933	E5D9904F0A8B4F99	2/1/2014 0:24	PAGELOAD	/180.253.140.219/
113934	E5D9904F0A8B4F99	2/1/2014 0:25	PAGELOAD	/180.253.140.219/
113935	E5D9904F0A8B4F99	2/1/2014 0:25	FINDROUTE	-6.90608,107.61530/-6.89140,107.61060/2
113936	E5D9904F0A8B4F99	2/1/2014 0:26	PAGELOAD	/118.137.96.28/
113937	E5D9904F0A8B4F99	2/1/2014 0:26	FINDROUTE	-6.89459,107.58818/-6.89876,107.60886/2
113938	E5D9904F0A8B4F99	2/1/2014 0:27	FINDROUTE	-6.90608,107.61530/-6.89140,107.61060/2
113939	E5D9904F0A8B4F99	2/1/2014 0:28	FINDROUTE	-6.89977,107.62706/-6.89140,107.61060/2
113940	E5D9904F0A8B4F99	2/1/2014 0:28	FINDROUTE	-6.89459,107.58818/-6.86031,107.61287/2
113941	D0AB08D956A351E4	2/1/2014 0:28	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113942	A44EB361A179A49E	2/1/2014 0:29	FINDROUTE	-6.9172304,107.6042556/-6.92663,107.63644/1
113943	A44EB361A179A49E	2/1/2014 0:29	FINDROUTE	-6.9172448,107.6042255/-6.92663,107.63644/1
113944	D0AB08D956A351E4	2/1/2014 0:30	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113945	D0AB08D956A351E4	2/1/2014 0:32	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113946	D0AB08D956A351E4	2/1/2014 0:33	FINDROUTE	-6.90598,107.59714/-6.90855,107.61082/1
113947	A44EB361A179A49E	2/1/2014 0:35	SEARCHPLACE	jalan+asia+af/8
113948	A44EB361A179A49E	2/1/2014 0:35	FINDROUTE	-6.9172448,107.6042255/-6.92163,107.61046/1
113949	A44EB361A179A49E	2/1/2014 0:35	SEARCHPLACE	taman+fotog/10

113950	A44EB361A179A49E	2/1/2014 0:36	FINDROUTE	-6.917321,107.6043132/-6.921568846707516,107.61015225201845/1
113951	E5D9904F0A8B4F99	2/1/2014 0:38	PAGELOAD	/5.10.83.68/
113952	E5D9904F0A8B4F99	2/1/2014 0:38	PAGELOAD	/5.10.83.28/
113953	E5D9904F0A8B4F99	2/1/2014 0:40		

113976	E5D9904F0A8B4F99	2/1/2014 1:25	PAGELOAD	/5.10.83.24/
113977	E5D9904F0A8B4F99	2/1/2014 1:25	FINDROUTE	-6.91485,107.59123/-6.91593,107.65588/1
113978	E5D9904F0A8B4F99	2/1/2014 1:26	PAGELOAD	/5.10.83.82/
113979	E5D9904F0A8B4F99	2/1/2014 1:28	FINDROUTE	-6.91593,107.65588/-6.91485,107.59123/1
113980	A44EB361A179A49E	2/1/2014 1:29	FINDROUTE	-6.9250709,107.6204635/-6.91728,107.60417/1
113981	A44EB361A179A49E	2/1/2014 1:35	FINDROUTE	-6.9252132,107.6200288/-6.91728,107.60417/1
113982	A44EB361A179A49E	2/1/2014 1:36	FINDROUTE	-6.922427886995373,107.61768691241741/-6.91728,107.60417/1
113983	E5D9904F0A8B4F99	2/1/2014 1:36	FINDROUTE	-6.91431,107.63921/-6.94024,107.71550/1
113984	E5D9904F0A8B4F99	2/1/2014 1:37	PAGELOAD	/5.10.83.98/
113985	A44EB361A179A49E	2/1/2014 1:37	FINDROUTE	-6.921635413232821,107.61909071356058/-6.91728,107.60417/1
113986	E5D9904F0A8B4F99	2/1/2014 1:38	FINDROUTE	-6.88936,107.57533/-6.92600,107.63628/1
113987	E5D9904F0A8B4F99	2/1/2014 1:39	PAGELOAD	http://www.kiri.trav l/m/r/?qs=trans+studi...
113988	E5D9904F0A8B4F99	2/1/2014 1:39	FINDROUTE	-6.92600,107.63628/-6.88936,107.57533/1
113989	A44EB361A179A49E	2/1/2014 1:41	SEARCHPLACE	t rminal+ta/10
113990	A44EB361A179A49E	2/1/2014 1:41	FINDROUTE	-6.9158359,107.6101751/-6.90658,107.61623/1
113991	A44EB361A179A49E	2/1/2014 1:42	FINDROUTE	-6.9158359,107.6101751/-6.90658,107.61623/1
113992	D0AB08D956A351E4	2/1/2014 1:50	FINDROUTE	-6.38355,106.919975/-7.08933734335005,107.562576737255/1
113993	A44EB361A179A49E	2/1/2014 1:51	SEARCHPLACE	taman+ci/10
113994	A44EB361A179A49E	2/1/2014 1:51	SEARCHPLACE	taman+cilaki/10
113995	E5D9904F0A8B4F99	2/1/2014 1:52	PAGELOAD	/206.53.152.33/m
113996	E5D9904F0A8B4F99	2/1/2014 1:52	FINDROUTE	-6.90598,107.59714/-6.91728,107.60417/1
113997	A44EB361A179A49E	2/1/2014 1:54	FINDROUTE	-6.901306,107.6214169/-6.90336,107.62235/1
113998	A44EB361A179A49E	2/1/2014 1:54	FINDROUTE	-6.901306,107.6214169/-6.90336,107.62235/1
113999	E5D9904F0A8B4F99	2/1/2014	PAGELOAD	/5.10.83.27/

114000	308201BB30820124	2/1/2014 1:15	SEARCHPLACE	riau+jucntion/10
114001	308201BB30820124	2/1/2014 1:56	FINDROUTE	-6.90687,107.61239/-6.89032,107.57961/2
114002	E5D9904F0A8B4F99	2/1/2014 1:57	PAGELOAD	/118.99.112.66/
114003	308201BB30820124	2/1/2014 1:57	FINDROUTE	-6.90687,107.61239/-6.90159,107.60442/1
114004	308201BB30820124	2/1/2014 1:57	FINDROUTE	-6.90687,107.61239/-6.89032,107.57961/2
114005	E5D9904F0A8B4F99	2/1/2014 1:58	FINDROUTE	-6.88211,107.60378/-6.90774,107.60908/1
114006	A44EB361A179A49E	2/1/2014 1:59	FINDROUTE	-6.9212516,107.6196466/-6.91728,107.60417/1
114007	308201BB30820124	2/1/2014 1:59	FINDROUTE	-6.90687,107.61239/-6.91486,107.60824/1
114008	687C44EB2424285D	2/1/2014 1:59	WIDGETLOAD	http://www.c nd kial ad rshipschool.sc...
114009	E5D9904F0A8B4F99	2/1/2014 2:00	FINDROUTE	-6.88166,107.61561/-6.90774,107.60908/1