

Implementasi Data Mining Classification untuk Mencari Pola Prediksi Hujan dengan Menggunakan Algoritma C4.5

Angga Raditya

Email: anggawasito@gmail.com

**Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma
JL. Margonda Raya No. 100 Depok**

ABSTRAK

Berkembangnya teknologi dan Informasi saat ini telah melahirkan “gunungan” data di bidang ilmu pengetahuan, bisnis dan pemerintah. Data mining mampu menjadi solusi untuk masalah ini. Data mining adalah kegiatan meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Klasifikasi merupakan salah satu fungsi dalam data mining. Metode decision tree mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. C4.5 adalah algoritma yang sudah banyak dikenal dan digunakan untuk klasifikasi data yang memiliki atribut-atribut numerik dan kategorial. World Meteorological Organization merupakan organisasi pengawas cuaca dunia. Didalamnya terdapat gunung data cuaca yang berpotensi untuk diolah. Data cuaca yang tersimpan tersebut memiliki atribut-atribut yang cukup lengkap untuk dibuat pohon keputusan. Untuk itu tujuan penelitian ini adalah untuk melihat pola prediksi dari setiap atribut-atribut yang terdapat pada data cuaca tersebut dengan menggunakan algoritma C4.5. Penelitian ini menggunakan bahasa pemrograman java serta DBMS MySQL untuk membangun aplikasinya. Akurasi pola prediksi yang didapat mampu mencapai 79%. Akurasi tersebut dihasilkan dari uji coba dengan menggunakan data cuaca tahun 2007 sebagai data training nya serta data cuaca tahun 2008 dan 2009 sebagai data testingnya

Kata Kunci: Data Mining, Classification, Algoritma C4.5, Cuaca

PENDAHULUAN

Cuaca merupakan faktor Alam yang sangat berpengaruh bagi kehidupan Manusia. Banyak kegiatan dan aktifitas manusia yang bergantung pada faktor dan kondisi cuaca, seperti pertanian, transportasi darat maupun udara. Prakiraan cuaca menjadi kebutuhan manusia untuk dapat menjalankan aktifitasnya dengan baik. Adapun faktor yang tidak lepas dari cuaca adalah suhu, kelembapan, tekanan udara, kecepatan angin, dan sebagainya.

World Meteorological Organization (WMO) dibawah naungan World Weather program Watch menampung data cuaca dari Negara-negara di dunia. Dimana Indonesia (BMKG) merupakan salah satu anggotanya. Data cuaca Indonesia yang disimpan di WMO merupakan data history cuaca setiap harinya dari setiap stasiun pemantau di setiap wilayah di Indonesia. Dan data cuaca tersebut semakin bertambah sampai sekarang.

Berkembangnya teknologi dan Informasi saat ini telah melahirkan “gunungan” data di bidang ilmu pengetahuan, bisnis dan pemerintah. Kemampuan teknologi informasi untuk mengumpulkan dan menyimpan berbagai tipe data jauh meninggalkan kemampuan untuk menganalisis, meringkas dan mengekstraksi “pengetahuan” dari data. Metodologi tradisional untuk menganalisis data yang ada, tidak dapat menangani data dalam jumlah besar. Para peneliti melihat peluang untuk melahirkan sebuah teknologi baru yang menjawab kebutuhan ini, yaitu *data mining*. Teknologi ini sekarang sudah ada dan diaplikasikan oleh perusahaan-perusahaan untuk memecahkan berbagai permasalahan bisnis. Secara garis besar Data mining berfungsi untuk mencari pola dari data dengan jumlah yang sangat besar.

Ada bermacam Fungsi yang digunakan dalam Data Mining dalam mengolah data pada umumnya, seperti Description, Clustering, Classification, Association Rules, Estimation. Dan setiap Fungsi memiliki algoritma tersendiri. Namun pada penulisan ini penulis menggunakan algoritma Decision Tree C4.5 yang masuk dalam Fungsi Classification. Berdasarkan permasalahan yang telah diuraikan diatas maka penulis mengangkat judul “Implementasi Data Mining Classification untuk Mencari Pola Prediksi Hujan dengan Menggunakan Algoritma C4.5”.

TINJAUAN PUSTAKA

Pengenalan Data Mining

Dalam perkembangannya Data Mining memiliki banyak definisi yang cukup beragam. Berikut adalah beberapa definisi Data Mining pada umumnya:

- Data mining adalah proses yang menggunakan statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan terkait dari berbagai database besar [Turban, dkk.2005].
- Menurut Gartner Group data mining adalah suatu proses menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [Daniel T. Larose, 2005].

Jadi dapat disimpulkan bahwa Data mining adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basisdata. Informasi yang dihasilkan diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basisdata.

Decision Tree

Decision Tree merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode decision tree mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Selain itu aturan juga dapat diekspresikan dalam bentuk bahasa basis data seperti Structured Query Language (SQL) untuk mencari record pada kategori tertentu. Decision tree juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Karena decision tree memadukan antara eksplorasi data dan permodelan Decision tree digunakan untuk kasus-kasus dimana outputnya bernilai diskrit.

Sebuah decision tree adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan record yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masing-masing rangkaian pembagian, anggota himpunan hasil menjadi mirip dengan yang lain [Berry & Linoff, 2004].

Algoritma C4.5

C4.5 adalah algoritma yang sudah banyak dikenal dan digunakan untuk klasifikasi data yang memiliki atribut-atribut numerik dan kategorial. Hasil dari proses klasifikasi yang berupa aturan-aturan dapat digunakan untuk memprediksi nilai atribut bertipe diskret dari record yang baru.

Algoritma C4.5 sendiri merupakan pengembangan dari algoritma ID3, dimana pengembangan dilakukan dalam hal : bisa mengatasi missing data, bisa mengatasi data kontinyu, pruning. Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

- a. Pilih atribut sebagai akar.
- b. Buat cabang untuk tiap-tiap nilai.
- c. Bagi kasus dalam cabang.
- d. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Dalam algoritma C4.5 digunakan information gain untuk memilih atribut yang akan digunakan untuk pemisahan obyek. Atribut yang mempunyai information gain paling tinggi dibanding atribut yang lain relatif terhadap set y dalam suatu data, dipilih untuk melakukan pemecahan.

Pada algoritma ini, pemilihan atribut mana yang akan menempati suatu simpul dilakukan dengan melakukan perhitungan entropi informasi (*information entropy*) dan mencari nilai yang paling minimum. Pemilihan atribut pada algoritma ini berdasarkan pada asumsi bahwa kompleksitas yang dimiliki oleh pohon keputusan sangat berkaitan erat dengan jumlah informasi yang diberikan oleh nilai-nilai atributnya. Dengan kata lain, teknik heuristik berdasarkan informasi ini memilih atribut yang memberikan perolehan informasi terbesar (*highest information gain*) dalam menghasilkan subpohon (*subtree*) untuk mengklasifikasikan sampel.

Entropy

Entropi merupakan distribusi probabilitas dalam teori informasi dan diadopsi ke dalam algoritma C4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah himpunan data (*data set*). Sebagai ilustrasi, semakin tinggi tingkat entropi dari sebuah *data set* maka semakin homogen distribusi kelas pada *data set* tersebut. Jika distribusi probabilitas dari kelas didefinisikan dengan $P = (p_1, p_2, p_3, \dots, p_k)$ maka entropi dapat dituliskan sebagai persamaan dari [12]:

$$(p_1, p_2, \dots, p_k) = - \sum_{i=1}^k (p_i \cdot \log_2(p_i)) \quad (2.1)$$

Persamaan 2.1 sama dengan persamaan $Info(T)$ sebagai berikut:

$$H(T) = - \sum_{i=1}^k \frac{f(C_i, T)}{n} \cdot \log_2 \frac{f(C_i, T)}{n} \quad (2.2)$$

Dimana *frequency* (C_i, T) adalah jumlah sampel di himpunan T yang memiliki kelas $C_1, C_2, C_3, \dots, C_k$.

Sebagai contoh, distribusi kelas (0.5, 0.5) lebih homogen bila dibandingkan dengan distribusi (0.67, 0.33) sehingga distribusi (0.5, 0.5) memiliki entropi yang lebih tinggi dari distribusi (0.67, 0.33). Hal ini dapat dibuktikan sebagai berikut:

$$E(0.5, 0.5) = -0.5 \times \log_2(0.5) - 0.5 \times \log_2(0.5) = 1$$

$$E(0.67, 0.33) = -0.67 \times \log_2(0.67) - 0.33 \times \log_2(0.33) = 0.91$$

Setelah T dipartisi ke dalam sejumlah subset $T_1, T_2, T_3, \dots, T_n$ berdasarkan atribut X maka perhitungan $Info$ dilakukan dengan menggunakan himpunan *training data* yang merupakan hasil partisi sebagai berikut:

$$Info(T) = \sum_{i=1}^n \left(\left(\frac{|T_i|}{|T|} \right) \cdot info(T_i) \right) \quad (2.3)$$

Gain

Setelah membagi *data set* berdasarkan sebuah atribut kedalam subset yang lebih kecil, entropi dari data tersebut akan berubah. Perubahan entropi ini dapat digunakan untuk menentukan bagus tidaknya pembagian data yang telah dilakukan. Perubahan entropi ini disebut dengan *information gain* dalam algoritma C4.5. *Information gain* ini diukur dengan menghitung selisih antara entropi *data set* sebelum dan sesudah pembagian (*splitting*) dilakukan. Pembagian yang terbaik akan menghasilkan entropi subset yang paling kecil, dengan demikian berdampak pada *information gain* yang terbesar [18].

Jika sebuah *data set* D dipartisi berdasarkan nilai dari sebuah atribut X sehingga menghasilkan subset (T_1, T_2, \dots, T_n) maka *information gain* dapat dihitung dengan persamaan:

$$Gain(T, X) = H(T) - \sum_{i=1}^n \left(\frac{|T_i|}{|T|} \right) H(T_i) \quad (2.4)$$

Dalam persamaan 2.4, $Info(T)$ adalah entropi dari *data set* sebelum dipartisi berdasarkan atribut X , dan $Info_x(T)$ adalah $Info$ dari *subset* setelah dilakukan pemartisian berdasarkan atribut X .

Penanganan Attribut Continue

Penanganan atribut dengan nilai kontinu merupakan salah satu kelebihan yang dimiliki algoritma C4.5 bila dibandingkan dengan pendahulunya, ID3. *Distinct value* dari *training data T* harus diurutkan terlebih dahulu sehingga diperoleh *value* dengan urutan $\{v_1, v_2, \dots, v_m\}$. Cari setiap *threshold* yang berada di antara v_i dan v_{i+1} dengan menggunakan persamaan — Dengan begitu hanya akan ada $m-1$ kemungkinan *threshold*. Untuk masing-masing *threshold* ini dilakukan perhitungan *gain ratio*. *Threshold* yang terpilih adalah *threshold* dengan *gain ratio* terbesar. Langkah ini dilakukan untuk setiap atribut dengan tipe data numerik atau memiliki nilai yang kontinu.

Weka(Class J48)

Weka adalah API java yang menyediakan API untuk data mining. *J48* adalah salah satu kelas di paket *classifiers* pada sistem Weka yang mengimplementasikan C4.5. Weka mengorganisasi kelas-kelas ke dalam paket-paket dan setiap kelas di paket dapat mereferensi kelas lain di paket lain. Paket *core* merupakan kelas inti dan mengandung banyak kelas yang diakses dari hampir semua kelas yang lain. Kelas-kelas kunci di *core* adalah *Attribute*, *Instance*, dan *Instances*. Sebuah obyek dari kelas *Attribute* merepresentasikan sebuah atribut. Kelas ini membungkus nama, tipe dan, dalam kasus atribut diskret, nilai-nilai unik atribut. Sebuah obyek dari kelas *Instance* mengandung nilai atribut dari sebuah instansiasi (rekord yang dijadikan obyek). Sedangkan sebuah obyek *Instances* membungkus semua instansiasi-instansiasi atau himpunan data (Kirby, 2002; Witten *et al.*, 2002).

Paket *classifiers* berisi implementasi dari hampir semua algoritma untuk klasifikasi dan prediksi. Kelas yang paling penting di paket ini adalah *Classifier*, yang mendeklarasikan struktur umum dari skema klasifikasi dan prediksi. Kelas ini memiliki dua metoda, yaitu *buildClassifier* dan *classifyInstance*, yang harus diimplementasikan oleh kelas-kelas yang menginduk ke kelas ini. Semua kelas yang mengimplementasikan algoritma klasifikasi menginduk ke kelas *Classifier*, termasuk kelas *J48*. *J48*, yang menangani himpunan data dalam format ARFF, tidak mengandung kode untuk mengkonstruksi pohon keputusan. Kelas ini mereferensi kelas-kelas lain, kebanyakan di paket *weka.classifiers.j48*, yang mengerjakan semua proses konstruksi pohon.

METODE PENELITIAN

Pengambilan Data

Data yang digunakan untuk penelitian ini adalah data yang dipertukarkan di bawah Organisasi Meteorologi Dunia (WMO) World Weather Program Watch berdasarkan WMO Resolusi 40 (Cg-XII) dimana Indonesia (qq. BMKG) merupakan salah satu anggotanya. Data-data tersebut telah dipublikasikan oleh NCDC-NOAA-USA dalam format file ascii text dengan unit dan satuan data yang digunakan di negara tersebut (USA). Penulis telah melakukan perubahan format file data menjadi *.xls dan unit satuan data menjadi metric sesuai dengan penggunaan untuk Indonesia.

Data yang digunakan adalah data cuaca harian pertahun dari tahun 2005 sampai dengan tahun 2009 pada salah satu stasiun pemantau yang ada di jakarta. Saat pengumpulan data terdapat beberapa atribut data yang tidak dibutuhkan untuk proses

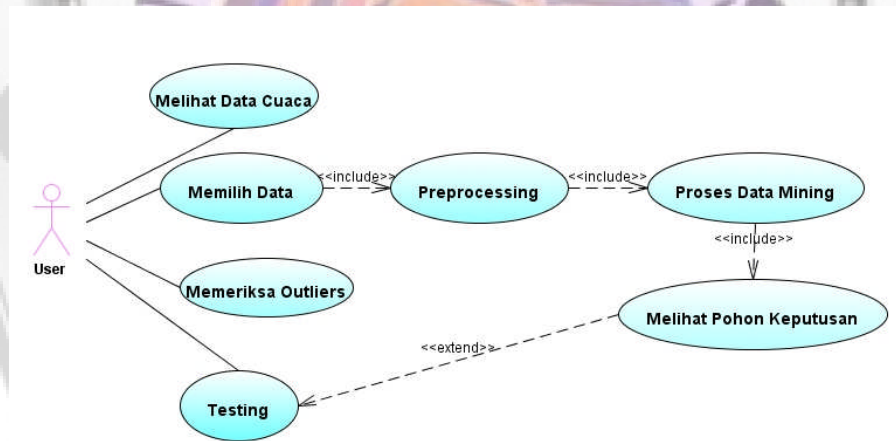
penelitian, seperti *wind gust* , *Temperatu Minimum* dan *visibility*. Sehingga kembali dilakukan proses pemilihan atribut sehingga sesuai dengan tujuan awal penelitian. Setelah dilakukan proses pemilihan atribut, maka kembali di dapat atribut-atribut yang diharapkan yaitu : *Tmean(suhu rata-rata)*, *Tmax(suhu maximum)*, *Td(Suhu titik embun pada hari tersebut)*, *RH (Relative Humidity)*, *SLP (Mean sea level pressure)*, *STP (Mean station pressure)*, *Wind_ave (kecepatan angin rata-rata)*, *Wind_max (kecepatan angin Maksimum)*, *Rain (Curah hujan)*.

Rancangan Sistem

Pada bagian ini akan dijelaskan mengenai alur sistem aplikasi yang dibuat. Beberapa proses yang akan terjadi dalam aplikasi ini dipresentasikan dengan diagram UML diantaranya Use case diagram dari aplikasi untuk menjelaskan bagaimana keterhubungan user dengan system dari aplikasi ini, activity diagram menjelaskan urutan proses yang dilakukan dalam aplikasi ini, serta class diagram untuk menjelaskan bagaimana implementasi algoritma c4.5 dalam aplikasi ini.

Use Case Diagram

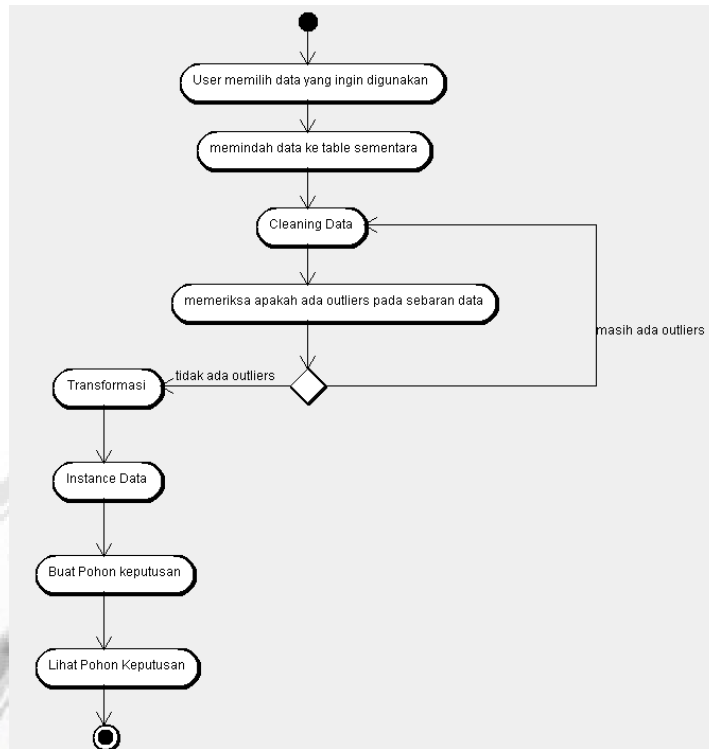
Use Case diagram adalah suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna diluar sistem. Dari gambar diatas terlihat bahwa ada satu actor yaitu user.



Gambar1 Use Case Diagram Sistem

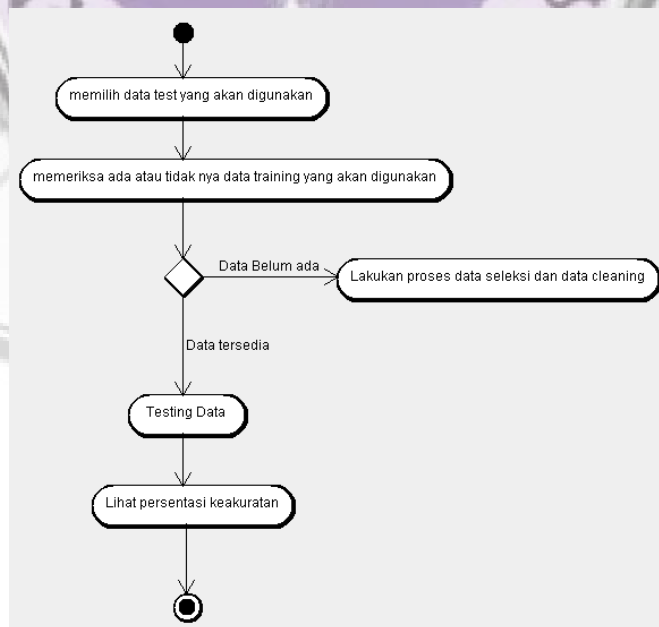
Activity Diagram

Activity Diagram merupakan suatu diagram yang dapat menampilkan secara detail urutan proses dari aplikasi. Perancangan aplikasi *data mining* ini dapat digambarkan dengan menggunakan *Activity Diagram* sebagai berikut :



Gambar2 Activity Diagram Pembentukan Pohon Keputusan

Activity Diagram diatas menggambarkan proses pembentukan pohon keputusan. Dari gambar *Activity Diagram* diatas dapat dilihat bahwa aplikasi *data mining* ini memiliki beberapa komponen-komponen utama yaitu Proses Pemilihan Data, Preprocessing , dan Pembentukan pohon Keputusan.



Gambar3 Activity Diagram Pengujian Pohon Keputusan

Proses pengujian pohon keputusan diatas berfungsi untuk menghitung keakuratan yang dihasilkan oleh pohon keputusan tersebut apabila diuji dengan suatu himpunan testing data tertentu.

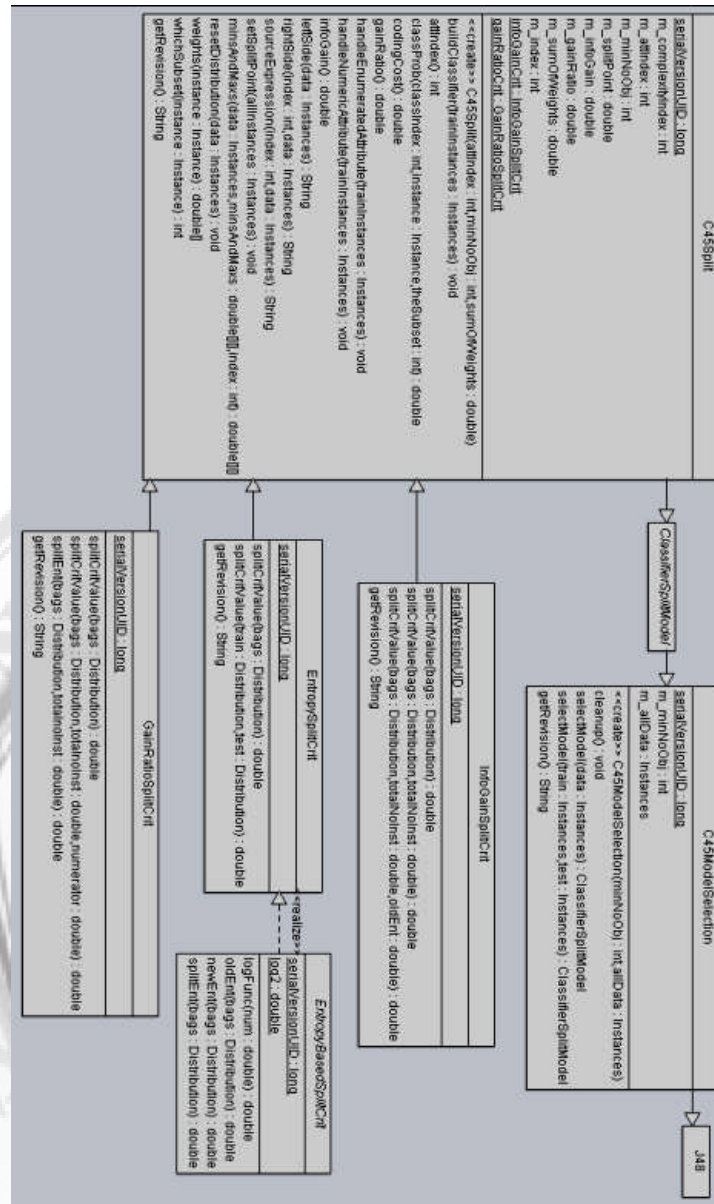
Class Diagram(Implementasi Algoritma C4.5 dalam Class J48)

Seperti yang telah dibahas pada bab dua. Bahwa penulis menggunakan data mining API yaitu Weka yang menyediakan Class J48 sebagai tools untuk membentuk *decission tree* yang berbasiskan algoritma C4.5. class J48 sendiri terdapat pada package classifier.

Paket *classifiers* berisi implementasi dari hampir semua algoritma untuk klasifikasi dan prediksi. Kelas yang paling penting di paket ini adalah Classifier, yang mendeklarasikan struktur umum dari skema klasifikasi dan prediksi. Kelas ini memiliki dua metoda, yaitu *buildClassifier* dan *classifyInstance*, yang harus diimplementasikan oleh kelas-kelas yang menginduk ke kelas ini. Semua kelas yang mengimplementasikan algoritma klasifikasi menginduk ke kelas Classifier, termasuk kelas *J48*.

J48 tidak mengandung kode untuk mengkonstruksi pohon keputusan melainkan menyediakan method yang dibutuhkan user untuk membangun pohon keputusan. Kelas ini mereferensi kelas-kelas lain, kebanyakan di paket *weka.classifiers.j48*, yang mengerjakan semua proses konstruksi pohon.


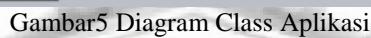




Gambar4 Diagram Class J48

HASIL DAN PEMBAHASAN

Sistem Aplikasi data mining yang dikembangkan dalam penelitian ini merupakan *Desktop Application* (aplikasi *desktop*) berbasis bahasa pemrograman Java Standard Edition dengan bantuan IDE Netbeans 6.9.1 sebagai tools Editornya. Aplikasi ini menggunakan DBMS MySQL untuk mengatur basis data selama proses seleksi data, pembersihan data, pembentukan pohon keputusan serta *testing* data. Berikut adalah diagram class aplikasi serta class-class yang dibuat untuk membangun aplikasi data mining ini.



```
graph TD;
    JA[Java Application] --> JAPI[JDBC API];
    JAPI --> JDM[JDBC Driver Manager];
    JDM --> JD1[JDBC Driver];
    JDM --> JD2[JDBC Driver];
    JDM --> JD3[JDBC Driver];
    JD1 --> O[(Oracle)];
    JD2 --> SS[(SQL Server)];
    JD3 --> CDS[(CDC Data Source)];
```

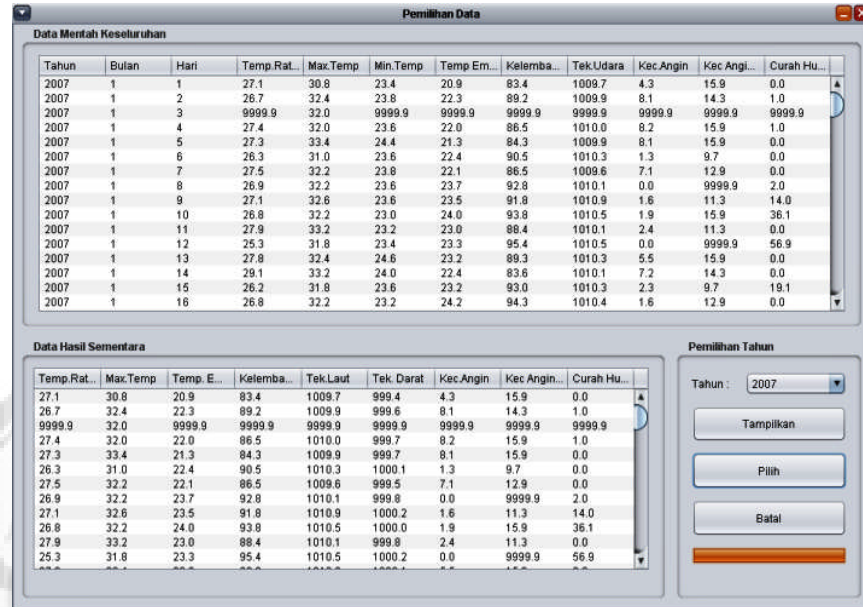
The diagram illustrates the JDBC architecture. At the top is the **Java Application**, which connects to the **JDBC API**. The **JDBC API** connects to the **JDBC Driver Manager**. The **JDBC Driver Manager** then connects to three separate **JDBC Driver** classes. Each **JDBC Driver** class connects to a specific database: **Oracle**, **SQL Server**, and **CDC Data Source**. A red box highlights the **JDBC API**, **JDBC Driver Manager**, and the three **JDBC Driver** classes, with the word **Class** written next to it.

Gambar6 Gambaran Umun Koneksi Aplikasi dengan JDBC

Dari gambar diatas dapat dilihat bahwa class DBConnection lah yang mengatur konksi aplikasi dengan DBMS. Contoh diatas menggambarkan bahwa JDBC dapat menghubungkan aplikasi Java dengan produk DBMS yang berbasis SQL.

Pemilihan Data

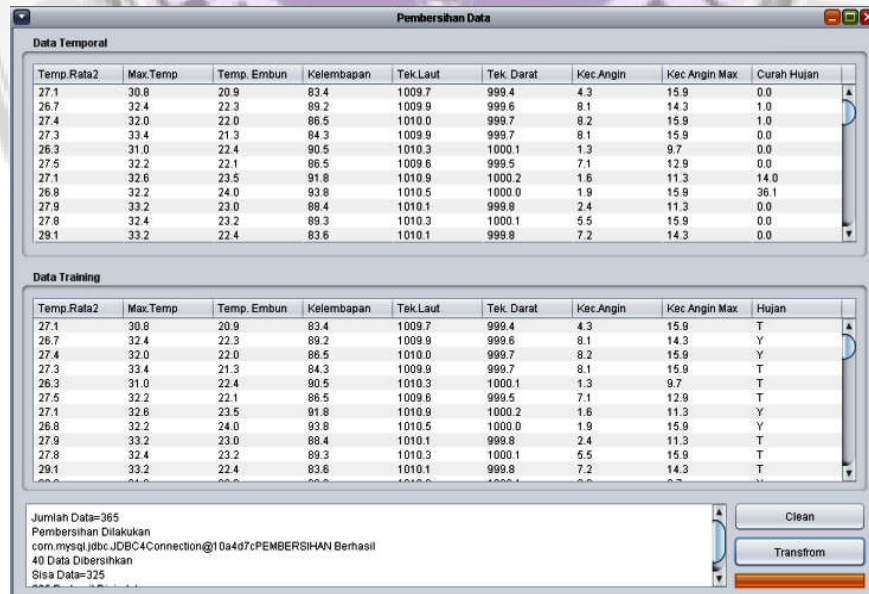
Untuk mendapatkan suatu pola Decision Tree perlu digunakan data training untuk membentuknya. Untuk uji coba aplikasi penulis menggunakan pemilihan data untuk uji coba aplikasi. Data yang dipilih adalah data cuaca tahun 2007.



Gambar7 Proses Pemilihan data

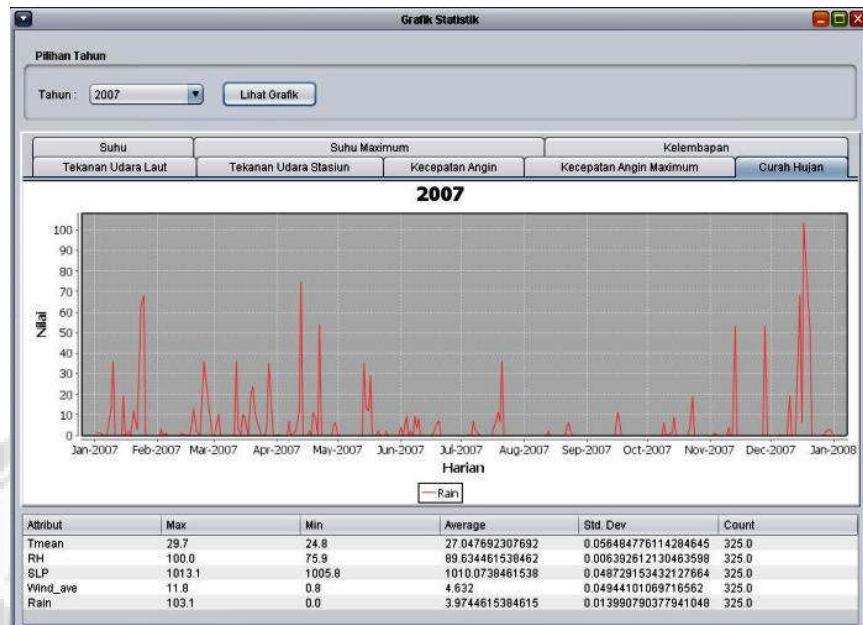
Preprocessing

Sebelum digunakan data cuaca ini melalui tahap *preprocessing* data yang berfungsi untuk menghilangkan missing value pada atribut-atribut, serta memeriksa ada tidaknya *outliers* pada himpunan data yang digunakan.



Gambar8 Preprocessing data

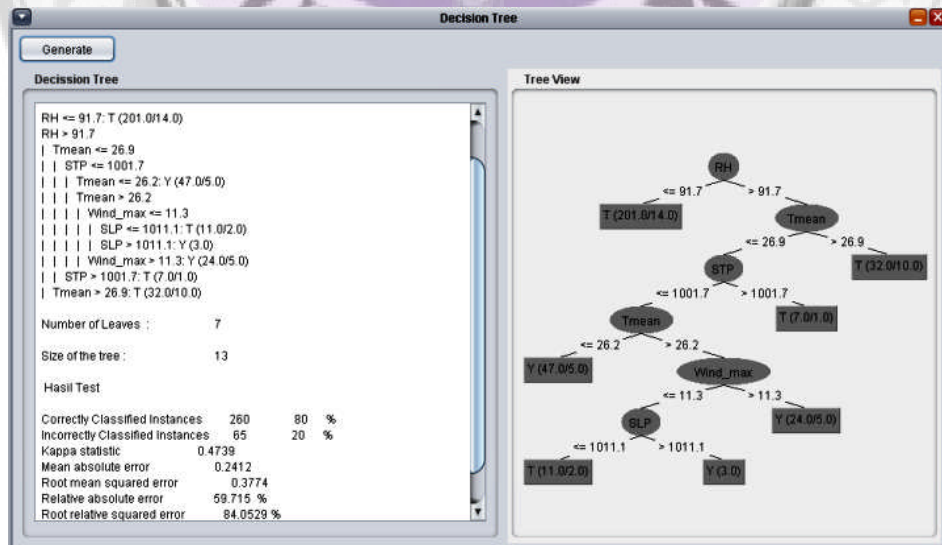
Dibutuhkan adanya grafik statistical untuk setiap atribut guna memeriksa ada tidaknya outliers atau missing value. Outliers itu sendiri merupakan nilai ekstrim yang terdapat didalam atribut itu sendiri.



Gambar9 Pemeriksaan Data

Decision Tree

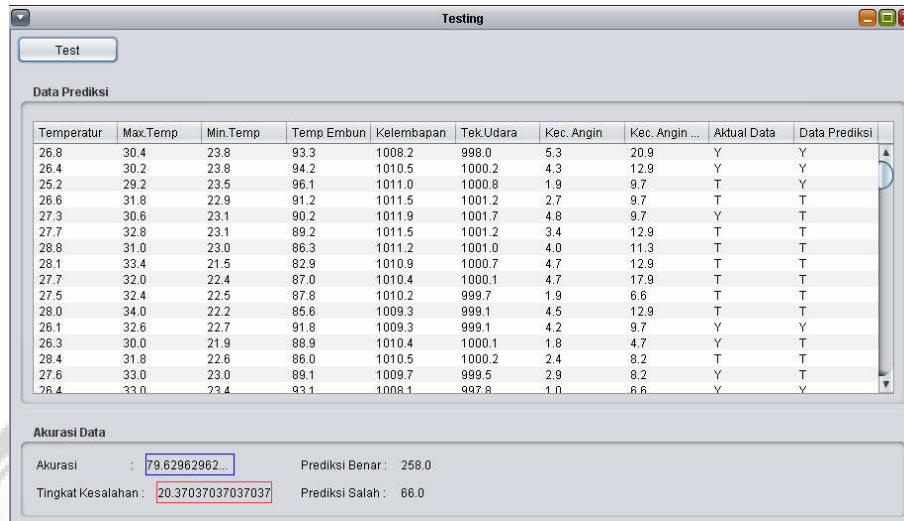
Dari proses prediksi yang dilakukan pada 2 data tes set, maka didapatkan hasil pola prediksi hujan menggunakan algoritma C4.5. Pohon keputusan yang didapat dari proses training terhadap data tahun 2007 adalah sebagai berikut:



Gambar10 Decision Tree

Testing Data

Proses selanjutnya adalah testing dengan menggunakan pohon keputusan yang telah dibentuk dari data training terhadap data testing yang telah dipilih pada langkah sebelumnya.



Temperatur	Max.Temp	Min.Temp	Temp Embun	Kelembapan	Tek.Udara	Kec. Angin	Kec. Angin...	Aktual Data	Data Prediksi
26.8	30.4	23.8	93.3	1008.2	998.0	5.3	20.9	Y	Y
26.4	30.2	23.8	94.2	1010.5	1000.2	4.3	12.9	Y	Y
25.2	29.2	23.5	96.1	1011.0	1000.8	1.9	9.7	T	Y
26.6	31.8	22.9	91.2	1011.5	1001.2	2.7	9.7	T	T
27.3	30.6	23.1	90.2	1011.9	1001.7	4.8	9.7	Y	T
27.7	32.8	23.1	89.2	1011.5	1001.2	3.4	12.9	T	T
28.8	31.0	23.0	86.3	1011.2	1001.0	4.0	11.3	T	T
28.1	33.4	21.5	82.9	1010.9	1000.7	4.7	12.9	T	T
27.7	32.0	22.4	87.0	1010.4	1000.1	4.7	17.9	T	T
27.5	32.4	22.5	87.8	1010.2	999.7	1.9	6.6	T	T
28.0	34.0	22.2	85.6	1009.3	999.1	4.5	12.9	T	T
26.1	32.6	22.7	91.8	1009.3	999.1	4.2	9.7	Y	Y
26.3	30.0	21.9	88.9	1010.4	1000.1	1.8	4.7	Y	T
28.4	31.8	22.6	86.0	1010.5	1000.2	2.4	8.2	T	T
27.6	33.0	23.0	89.1	1009.7	999.5	2.9	8.2	Y	T
26.4	33.0	23.4	93.1	1008.1	997.8	1.0	6.6	Y	Y

Akurasi Data

Akurasi : 79.62962962... Prediksi Benar : 258.0
 Tingkat Kesalahan : 20.37037037037037 Prediksi Salah : 66.0

Gambar11 Hasil Testing Data

Tahun/Prediksi	2008	2009
Benar	259	258
Salah	66	65
Total	325	323

Tabel1 Perhitungan Hasil Uji Coba.

Akurasi ketepatan hasil prediksi pohon keputusan terhadap data tahun 2008 adalah : $((259/325) \times 100\%) = 79,692 \%$. Sedangkan nilai kesalahan pada penelitian dengan pohon keputusan terhadap data tahun 2008 adalah : $((66/325) \times 100\%) = 20,302 \%$. Untuk akurasi ketepatan prediksi pohon keputusan terhadap data cuaca tahun 2009 adalah $((258/323) \times 100\%) = 79,876 \%$ dan untuk persentase kesalahan prediksinya adalah $((65/323) \times 100\%) = 20,123 \%$

KESIMPULAN

Dari penelitian dan pengujian yang telah dilakukan dapat disimpulkan bahwa data mining Classification dengan menggunakan metode pohon keputusan dengan algoritma C4.5 untuk membentuk pohon keputusan prediksi cuaca dapat dilakukan. Sejumlah kelebihan dalam penggunaan algoritma C4.5 dalam membangun pohon keputusan prediksi cuaca adalah kemampuannya menangani data kontinu maupun data nominal, mengingat bahwa hampir seluruh atribut cuaca yang digunakan bertipe data kontinu.

Selain itu juga algoritma C4.5 memiliki kelebihan dalam membangun keputusan dengan tingkat error yang lebih sedikit. Hal ini dibuktikan dengan uji coba terhadap data cuaca tahun 2008 dan 2009 dengan data training pembentuk pohon keputusan tahun 2007. Akurasi yang dihasilkan dari dua kali uji coba menghasilkan tingkat akurasi diatas 70% dan hampir mendekati 80 %.

Pengklasifikasian attribute tujuan sangatlah penting mengingat bahwa attrigut hujan merupakan atribut dengan tipe data kontinu sedangkan, algoritma C4.5 hanya bisa digunakan jika tipe data pada atribut tujuan bertipe diskrit. Kelengkapan data pada penelitian ini menjadi kelemahan dalam pembentukan pohon keputusan karena penanganan *outliers* pada data suatu hari seharusnya tidak dihilangkan karna dapat mengurangi volume data untuk prediksi.

DAFTAR PUSTAKA

- Daniel T Larose, 2005. *Discovering Knowledge in Data: An Introduction to Data Mining*. John Willey & Sons, Inc.
- Xindong Wu, 2009. *The Top Ten Algorithms in Data Mining*. CRC Press, Inc .
- Kusrini, Emha Taufiq Luthfi, 2009. *Algoritma Data Mining*. Yogyakarta. Andi.
- Ian H. Witten, Eibe Frank, Mark A. Hall, 2011. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Han, Jiawei and Micheline Kamber, 2001. *“Data Mining Concepts and Techniques”*, Morgan Kaufmann, California.
- Remco R. Bouckaert, Eibe Frank, Mark A. Hall, 2010. *WEKA Manual for Version 3-6-2s*. University of Waikato, Hamilton, New Zealand.
- Ronald Mak, 2002. *The Java Programmer's Guide to Numerical Computing*. Prentice Hall PTR.
- Mark Matthews, Jim Cole, Joseph D. Gradecki, 2010. *MySQL and Java Developer's Guide*. Wiley Publishing, Inc.
- Perdita Stevens, Rob Pooley, 2006. *Using UML Software engineering with objects and components*. Addison-Wesley, Inc.
- <http://www.cs.waikato.ac.nz/ml/weka/> (Waktu Akses : 12 Juli 2011, 14 : 17 WIB)
- <http://weka.wikispaces.com/Using+the+Experiment+API> (Waktu Akses : 12 Juli 2011, 14 : 17 WIB)
- <ftp://ftp.ncdc.noaa.gov/pub/data/gsod/> (Waktu Akses : 12 Juli 2011, 14 : 17 WIB)
- https://docs.google.com/present/view?id=dgnjdcv4_1046fp9k6qc6 (Waktu Akses : 12 Juli 2011, 14 : 17 WIB)