

BAB II

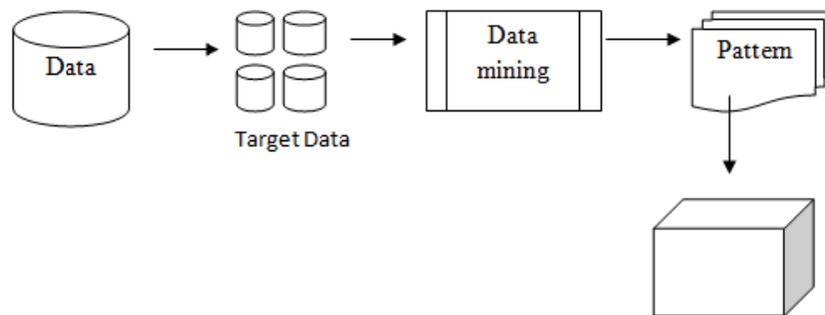
LANDASAN TEORI

2.1 Data mining

Bisnis yang ada saat ini memerlukan informasi yang dapat digunakan sebagai data untuk proses pengambilan keputusan. Data apabila digunakan secara efektif akan dapat sangat bernilai dan berharga dalam dunia bisnis yang sangat kompetitif. Di sisi lain ketika data tidak digunakan secara efektif, data tersebut akan menjadi tidak kompetitif dalam persaingan yang ada didalam dunia bisnis. *Data mining* merupakan sebuah proses yang berguna untuk mengolah data dan kemudian mengekstrak pola yang akan sangat berharga apabila digunakan untuk proses pengambilan keputusan. Misalnya, pola pelanggan yang bisa didapatkan dari data merupakan informasi yang sangat berharga bila diterapkan pada sistem perbankan untuk mengamankan pinjaman bank dan juga hal ini merupakan pengetahuan yang berharga bagi sistem retail untuk meningkatkan keuntungan dan kelayakitan pelanggan. Seperti yang ditunjukkan pada gambar 2.1 bahwa *data mining* merupakan proses untuk menemukan pola dan model dari data untuk pengambilan keputusan. *data mining* sebagai langkah tertentu dalam KDD (*Knowledge Discovery in Database*) adalah proses yang tidak mudah dalam mengidentifikasi pola data yang valid, baru, berpotensi untuk berguna dan dapat dimengerti (Fayyeh, 1996). *Data mining* menggunakan algoritma tertentu yang spesifik misalnya *discrimination*, *classification*, *association*, *clustering*, dan lain-

lain untuk menghasilkan pola dan model yang berbeda. Pola-pola yang ditemukan harus memiliki kriteria sebagai berikut (Fayyeh, 1996):

1. Berlaku pada data baru dengan beberapa derajat kepastian.
2. Baru yang setidaknya dapat berpengaruh pada sistem dan sebaiknya lebih codong kepada pengguna/*user*.
3. Kegunaan yang dapat menyebabkan beberapa manfaat bagi pengguna atau untuk tugas-tugas dari pengguna itu sendiri.
4. Dapat dimengerti yang dapat diperkirakan melalui kesederhanaan, dan jika tidak langsung dapat dimengerti, akan dimengerti setelah melalui beberapa pengolahan terlebih dahulu.



Gambar 2.1 Transformasi data menjadi pola atau model menggunakan data mining

Proses menemukan pola ataupun pengetahuan akan menghasilkan banyak pola-pola yang banyak. Dan dalam jumlah pola yang banyak ini akan menyebabkan kesulitan untuk dipahami, maka dari itu ada proses pengeliminasian yang diterapkan dengan menentukan nilai ambang batas (*threshold*), untuk menemukan pola yang lebih menarik. Proses yang menarik dari pencarian pola ini

adalah dengan menggabungkan empat pola kriteria seperti validitas, baru, kegunaan, dan kesederhaan untuk dapat dimengerti. Dua metode yang umum untuk digunakan dalam mencari pola yang menarik yaitu (Freitas, 2003):

1. Metode Subyektif, yaitu pengguna didorong dan tergantung pada domain misalnya, pengguna harus menentukan secara spesifik aturan yang dianggap menarik.
2. Metode Objektif, yaitu data didorong dan tidak tergantung pada domain misalnya, aturan pencarian pola yang menarik tergantung pada kualitas dari aturan dan kesamaannya pada aturan lain.

Metode subjektif dan objektif mirip dengan dua tujuan dari KDD (*Knowledge Discovery in Database*) yaitu:

1. Verifikasi, dimana sistem terbatas untuk memverifikasi hipotesis dari pengguna.
2. *Discovery*, dimana sistem secara otomatis menemukan pola-pola baru.

Tujuan dari *discovery* ini kemudian dibagi lagi yaitu:

- a. Prediksi, dimana sistem menemukan pola untuk memprediksi perilaku masa depan dari beberapa entitas.
- b. Deskripsi, dimana sistem menemukan pola presentasi kepada pengguna kedalam bentuk yang dapat dimengerti manusia. Model prediksi dapat menjadi pola deskripsi dan sebaliknya.

Sebagai sebuah disiplin ilmu, *data mining* bersinggungan dengan disiplin ilmu yang lainnya yaitu database, statistik, *machine learning*, kecerdasan buatan, sistem pakar dan pengenalan pola, *neural network*, visualisasi data, pencarian informasi, pengolahan gambar dan sinyal, dan analisis data spasial. *data mining*

telah diimplementasikan atau diterapkan kepada beberapa masalah yang ada pada dunia industri yang ada pada dunia nyata seperti seperti masalah *data mining* spasial, *data mining* bagian music, *data mining* teks, *data mining* visual, melestarikan privasi *data mining*, dan lain lain. Di tahun 2011 KDnuggets melakukan survey terhadap penerapan *data mining* yang diterapkan didalam operasional yang ada pada dunia industri, dan sepuluh besar industri yang menggunakan *data mining* dalam pengoperasionalnya yaitu *Customer Relationship Management* (CRM), perbankan, kesehatan, pendidikan, pendeteksian penipuan, ilmu pengetahuan, sosial media, *direct scoring*, pemasaran langsung / penggalangan dana dan asuransi.

Data mining spasial adalah proses untuk mencari pengetahuan dari data spasial seperti gambar dan film dalam rangka untuk mencari pola yang ada. Hal ini memiliki jangkauan luas meliputi Sistem Informasi Geografis (GIS), *remote sensing*, *database* yang meliputi gambar dan video, *medical imaging*, navigasi robot, dan lain-lain (Koperski, 1996). *Musical data mining* menggunakan teknik *data mining*, meliputi analisis konkurensi dalam rangka untuk menemukan kesamaan antara lagu dan mengklasifikasi lagu ke genre yang benar atau mengklasifikasikan lagu tersebut ke artis yang benar (Liu, 1999). Sementara itu, apabila *data mining* teks menggunakan koleksi data teks yang sangat besar secara *online* maka yang diklasifikasikan yaitu pencarian fakta-fakta baru dan tren mengenai kata-kata itu sendiri. *Data mining* secara visual menerapkan teknik *data mining* dengan menggunakan teknologi visualisasi informasi untuk meningkatkan keakuratan analisis data. Akhirnya dalam prosesnya untuk mempertahankan privasi itu sendiri maka *data mining* diperpanjang atau mempertahankan

keprivasian dari pengguna untuk membiarkan pengguna untuk memberikan nilai dimodifikasi untuk memberikan nilai pada atribut yang sensitif, dimana nilai yang dimodifikasi tersebut dapat dihasilkan dengan menggunakan sebuah *plug-in browser or extention* untuk produk dalam rangka menutupi informasi yang sensitif.

Data mining mampu mengatasi masalah data yang *overload* atau berlebihan sebagai data *continues* atau data yang terus bertumbuh. Survey yang dilakukan oleh KDnuggets menunjukkan bahwa *database* terbesar atau *datasets* terbesar yang telah digunakan yaitu 21% pemilih menggunakan lebih dari 1 *Terabyte database/dataset*. 4% pemilih menggunakan *database/dataset* lebih dari 1 *Petabyte* dan 19,5% pemilih menggunakan 1,1-10 *Gigabyte database/dataset* (Piatetsky-shapiro,2011). *Data mining* mengekstraksi pengetahuan melalui beberapa *database* yang berbeda misalnya *relational database, transaction database, object oriented database, eductive databases, spatial databases, temporal databases, multimedia databases, heterogeneous databases, active databases, legacy databases, and the internet information-base* (Chen, 1996). Survey yang dilakukan KDnuggets menunjukkan bahwa jenis data yang paling populer untuk dilakukan proses *data mining* pada tahun 2011 adalah data table, *time series*, data transaksi, data teks, data anonym, lokasi/geografi, data jaringan social, email

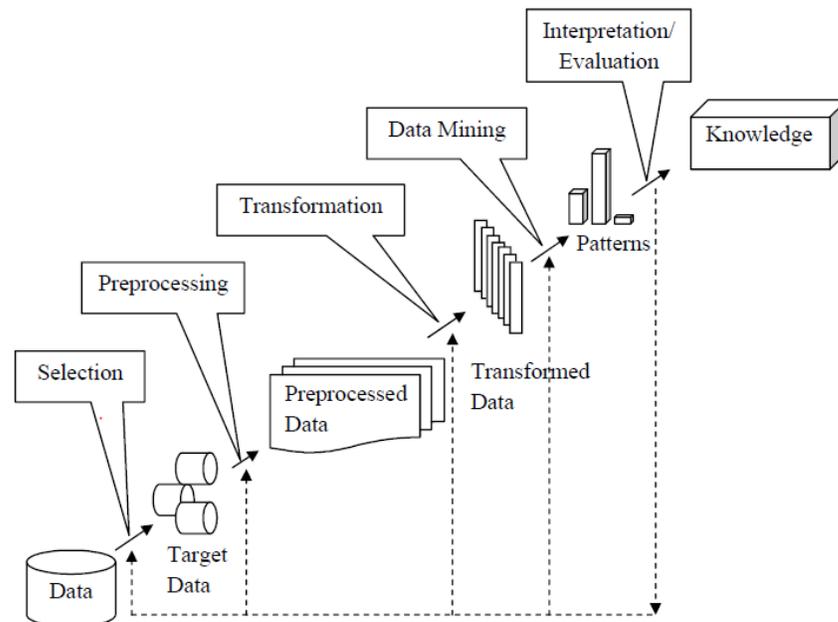
2.2 Knowledge Discovery in Database

Upaya didalam dunia industri lebih banyak tertuju kepada definisi dari metodologi yang dapat menjadi panduan didalam mengimplementasikan aplikasi *data mining*. KDD merupakan salah satu dari metodologi yang sangat populer

yang berfokus kepada proses secara keseluruhan proses penemuan ilmu / *knowledge* dari data, termasuk proses penyimpanan dan pengaksesan data, algoritma yang efektif dan efisien apabila berhubungan dengan data yang besar, interpretasi dan visualisasi dari hasil temuan pengetahuan. Istilah dari KDD ini diciptakan pertama kali pada tahun 1989 pada workshop KDD pertama (Piatetsky-shapiro, 1991). Seperti yang ditunjukkan pada gambar 2.2 bahwa, *data mining* sebagai sebuah langkah penting didalam proses KDD memerlukan 9 tahapan yaitu (Fayyad, 1996):

1. Mengembangkan pemahaman mengenai aplikasi domain dan mengidentifikasi tujuan dari proses KDD dari sudut pandang *user* / pelanggan.
2. Membuat sebuah dataset sasaran, memilih dataset, dan berfokus pada variable atau sampel data yang proses penemuannya akan dilakukan.
3. Pembersihan data dan *preprocessing*, dimana termasuk menghilangkan *noise*, mengumpulkan informasi yang berguna, menangani dalam hal *data fields* yang hilang dan menghitung untuk informasi *time sequence* sama halnya dengan isu-isu DBMS seperti isu tipe, skema, dan nilai-nilai / *values* yang tidak diketahui.
4. Reduksi data dengan menemukan fitur-fitur yang berguna untuk merepresentasi data.
5. Mencocokkan tujuan dari proses KDD dengan langkah-langkah tertentu pada metode *data mining* melalui *summarization*, *classification*, *regression*, *clustering* dan lain-lain.

6. Mengeksplorasi analisis dan model dan menyeleksi hipotesis dengan memilih algoritma *data mining* dan memilih metode yang akan digunakan untuk mencari pola dari data.
7. Melakukan *data mining* dengan mencari pola-pola yang menarik dalam bentuk representasional tertentu.
8. Menafsirkan pola yang akan dicari yang juga dapat melibatkan visualisasi dari pola yang telah diekstraksi dan model atau visualisasi dari data yang diberikan oleh model yang diekstraksi.
9. Bertindak terhadap pengetahuan yang berhasil ditemukan dengan menggunakan pengetahuan tersebut secara langsung, menggabungkan pengetahuan tersebut kedalam sistem yang lain untuk langkah lebih lanjut, atau hanya mendokumentasikannya saja dan melaporkan kepada pihak yang berkepentingan. Proses ini juga termasuk memeriksa dan menyelesaikan potensi konflik dengan pengetahuan yang telah ditemukan sebelumnya.



Gambar 2.2 Fase dari metodologi Knowledge Discovery in Database

Sebelum dilakukannya upaya untuk mengekstrak pengetahuan dari suatu data, penting untuk mengetahui pendekatan yang dilakukan secara keseluruhan. Dengan mengetahui beberapa algoritma yang digunakan untuk melakukan analisis data saja tidak cukup untuk melakukan proses *data mining* (DM). Untuk memformalkan *Knowledge Discovery Proccess* (KDP) dalam kerangka kerja yang umum. Ada beberapa alasan mengapa *KDP* dapat dijadikan sebagai suatu model proses yang standar:

1. Produk akhir harus dapat berguna bagi pengguna dari data.

Sebuah teknik untuk menginput data dalam data mining yang tidak terstruktur dapat disebut *data dredging* atau pengerukan data, sering menghasilkan hasil yang tidak berarti atau pengetahuan yang tidak berguna, contohnya menghasilkan suatu pengetahuan yang menarik tetapi tidak menyelesaikan masalah *user*, yang hasilnya akan mengakibatkan gagalnya suatu proyek. Hanya aplikasi yang menerapkan model KDP dengan baik akan dihasilkan produk yang baik, baru, valid, berguna, dan dapat dimengerti.

2. Sebuah model KDP yang baik harus memiliki struktur yang baik, logis, kohesif dan memiliki pendekatan yang dapat disajikan kepada pembuat keputusan yang mungkin kesulitan dalam memahami kebutuhan, nilai, dan mekanisme dibalik KDP.

Manusia sering gagal untuk memahami potensi pengetahuan yang terdapat pada data yang berjumlah besar. Kebanyakan orang tidak mau mencurahkan waktu yang banyak untuk dapat memahami atau mencari pengetahuan dari tumpukan data yang ada. Kebanyakan orang akan mengandalkan teknologi

untuk mengolah suatu data sehingga menghasilkan suatu informasi yang dapat berguna.

3. *Knowledge Discovery Project* memerlukan usaha yang signifikan pada bidang *project management* yang perlu didasarkan pada kerangka yang solid.

Sebagian besar dari *knowledge discovery project* melibatkan kerja sama tim sehingga membutuhkan perencanaan dan penjadwalan yang matang.

4. *Knowledge Discovery* harus mengikuti contoh dari disiplin ilmu lainnya yang telah memiliki model yang tetap.

Sebuah contoh yang baik dapat dilihat dari bidang rekayasa piranti lunak, yang merupakan disiplin ilmu yang baru dan dinamis yang menunjukkan banyak karakteristik yang berhubungan dengan ilmu *knowledge discovery*.

5. Telah diketahui secara luas mengenai pemberian standar pada KDP.

Tantangan bagi para pengguna data mining yang ada saat ini yaitu menciptakan suatu standar yang dapat diterima secara luas. Standarisasi model dari KDP ini memungkinkan pengembangan metode dan prosedur standar sehingga memungkinkan pengguna untuk menyebarkan proyek-proyek mereka dengan lebih mudah. Hal ini juga mengarah langsung kepada proyek yang tentu lebih murah, cepat, dapat diandalkan, dan dapat lebih mudah dikelola.

2.2.1 Knowledge Discovery Process Model

1. Academic Research Model

Upaya yang dilakukan dalam membangun KDP dimulai dari sisi akademisi. Pada pertengahan tahun 1990-an, para peneliti mulai

mendefinisikan tahapan-tahapan untuk memandu pengguna *data mining* didunia ilmu pengetahuan. Penekanan utamanya adalah memberikan urutan kegiatan yang dapat membantu untuk melaksanakan proses KDP dalam domain lainnya. Ada dua proses model yang dikembangkan pada tahun 1996 dan 1998 adalah Sembilan langkah model yang dikemukakan oleh Fayyad dan delapan langkah oleh Anand dan Buchner. Ke Sembilan langkah yang dikemukakan Fayyad telah dibahas pada sesi sebelumnya.

2. Model Industri

Model industry secara garis besar mengikuti model dari akademik. Beberapa pendekatan yang berbeda telah dilakukan, mulai dari model yang diusulkan oleh individu dengan pengalaman industri yang luas untuk model yang diusulkan oleh industri besar. Dua model dalam industry adalah lima langkah model yang dikemukakan oleh Cabena, dan enam langkah model CRISP-DM yang dikembangkan oleh perusahaan-perusahaan yang ada di Eropa.

CRISP-DM pertama kali dibuat pada akhir 1990-an oleh empat perusahaan yaitu: Solusi Integral Ltd (penyedia solusi *data mining* komersial), NCR (penyedia database), Daimler Chrysler (produsen mobil), dan Ohra (perusahaan asuransi). Kedua perusahaan terakhir merupakan sumber dari data dan sumber studi kasus.

Model CRISP-DM terdiri dari enam langkah, yaitu:

1. *Business Understanding*. Langkah ini berfokus kepada pemahaman tujuan dan persyaratan dari perspektif bisnis. Hal ini juga mengubah definisi dari DM yang telah ada, dan mengubah desain dari rencana awal proyek. Hal ini kemudian dipecah menjadi beberapa step yaitu:
 - a. Penentuan tujuan bisnis,
 - b. Penilaian situasi,
 - c. Penentuan tujuan DM, dan
 - d. Generasi dari rencana proyek.
2. *Data understanding*. Langkah ini dimulai dengan pengumpulan data awal dan sosialisasi dengan data. Tujuannya meliputi identifikasi kualitas data, wawasan awal mengenai data, dan deteksi subset data yang menarik. *Data understanding* kemudian dibagi menjadi:
 - a. Pengumpulan data awal,
 - b. Deskripsi data,
 - c. Eksplorasi data, dan
 - d. Verifikasi kualitas data.
3. *Data preparation*. Langkah ini mencakup semua kegiatan yang dibutuhkan saat membangun dataset akhir, yang merupakan data yang akan dimasukkan ke dalam alat *data mining* pada langkah berikutnya. Yang termasuk pada bagian ini yaitu *table*, *record*, dan pemilihan atribut, pembersihan

data, pembangunan atribut baru, dan transformasi data. Hal ini terbagi menjadi:

- a. Pemilihan data,
- b. Pembersihan data,
- c. Konstruksi data,
- d. Integrasi data, dan
- e. Format data substep

4. *Modelling*. Pada langkah ini, berbagai teknik permodelan dipilih dan diterapkan. *Modelling* biasanya melibatkan penggunaan beberapa metode untuk jenis masalah *data mining* yang sama. Karena beberapa metode mungkin memerlukan format khusus untuk penginputan data, pengulahan langkah sebelumnya sering terjadi. Pembagian substepnya yaitu:

- a. Pemilihan teknik permodelan,
- b. Menguji desain,
- c. Pembuatan model,
- d. Penilaian model yang dihasilkan.

5. *Evaluation*. Setelah satu atau lebih model yang memiliki kualitas dari perspektif analisis data telah dibuat, model dievaluasi dari segi objektif bisnis. Sebuah tinjauan dari langkah-langkah yang dilakukan untuk membangun model juga dilakukan. Salah satu tujuan utamanya adalah untuk menentukan apakah masalah bisnis cukup penting untuk

dipertimbangkan. Pada akhir tahap ini, keputusan mengenai kegunaan dari *data mining* harus bisa tercapai. Substep dari langkah ini meliputi:

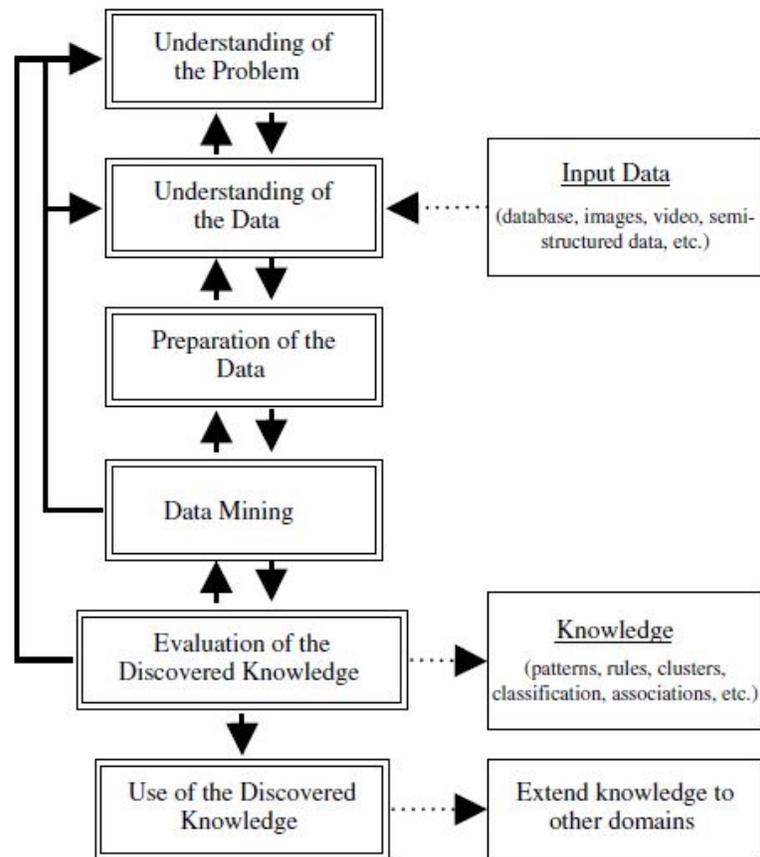
- a. Evaluasi hasil,
- b. Proses review,
- c. Penentuan langkah berikutnya.

6. *Deployment*. Pengetahuan yang telah ditemukan harus diorganisir dan disajikan dengan cara yang baik sehingga dapat dimengerti oleh user. Tergantung pada kebutuhan, langkah ini dapat menjadi sesederhana menghasilkan laporan atau serumit menerapkan KDD berulang kali. Hal ini terbagi menjadi:

- a. Rencana deployment,
- b. Rencana pemantauan dan pemeliharaan,
- c. Membuat laporan akhir,
- d. Review dari proses substep.

3. Model *Hybrid*

Pengembangan model akademik dan model industri menyebabkan pengembangan model *hybrid* yaitu model yang menggabungkan aspek keduanya. Salah satu model tersebut adalah enam langkah model KDP yang dikembangkan oleh Cios (2000). Model ini dikembangkan berdasarkan kepada model CRISP-DM dengan mengadopsi model akademis.



Gambar 2.3 Enam langkah model KDD. Pal, N.R., Jain, L.C.,(Eds.) 2005.

Deskripsi dari enam langkah sebagai berikut:

1. Pemahaman mengenai domain masalah.

Langkah awal ini melibatkan bekerja sama dengan para ahli untuk mendefinisikan masalah dan menentukan tujuan proyek, mengidentifikasi orang-orang kunci, dan belajar mengenai solusi untuk masalah ini.

2. Pemahaman data.

Langkah ini meliputi pengumpulan data sampel dan memutuskan data yang akan digunakan, termasuk format dan

ukurannya yang akan digunakan. Latar belakang pengetahuan dapat digunakan untuk memandu upaya ini. Data diperiksa mengenai kelengkapan, redudansi, nilai-nilai yang hilang, dll. Langkah ini termasuk menverifikasi kegunaan data sehubungan dengan tujuan dari *data mining*.

3. Persiapan data.

Langkah ini memperhatikan saat menentukan data yang akan digunakan untuk input pada *data mining* pada langkah berikutnya. Hal ini meliputi sampel, menjalankan korelasi, dan pembersihan data, yang meliputi kelengkapan data, menghapus atau memperbaiki *noise* dan nilai-nilai yang hilang. Data yang telah dibersihkan akan diproses lebih lanjut dengan pemilihan fitur, dan ekstraksi algoritma (untuk mengurangi dimensi data). Hasil akhirnya data yang memenuhi persyaratan untuk *data mining* akan dipilih.

4. Data mining

Pada pengguna data mining yang menggunakan beberapa metode data mining untuk memperoleh pengetahuan dari data yang telah diproses.

5. Evaluasi terhadap data yang telah ditemukan

Evaluasi yang dilakukan meliputi pemahaman terhadap hasil, memeriksa apakah pengetahuan yang telah ditemukan tersebut baru dan menarik, dan memeriksa dampak dari pengetahuan yang telah ditemukan.

6. Menggunakan pengetahuan yang telah ditemukan.

Langkah final ini terdiri dari perencanaan dimana dan bagaimana cara untuk menggunakan data yang telah ditemukan. Domain yang termasuk pada penemuan pengetahuan dapat mempengaruhi domain lainnya. Rencana untuk memonitor implemementasi terhadap penemuan data dibuat dan keseluruhan proyek dibuat dokumentasinya. Sampai akhirnya, pengetahuan tersebut disebarkan.

Tabel 2.1 Pebandingan antara Model (Cios, K.J, 2007)

| Perangkat lunak pendukung beserta domain yang digunakan | Catatan | Langkah | Jumlah Langkah | Domain asal | Model |
|---|--|--|----------------|-------------|-----------------|
| Sistem komersil MineSet; pengobatan; <i>engineering</i> ; produksi; bisnis elektronik; perangkat lunak, Marketing | Merupakan model yang paling populer dan model yang paling banyak disitasi; menyediakan penjelasan teknis secara detail kepada data analis, tetapi lemah pada aspek bisnisnya | Membangun pemahaman mengenai aplikasi domain. Membuat sebuah dataset sasaran. Pembersihan data dan <i>preprocessing</i> . Reduksi data Mencocokkan tujuan dari proses KDD. Menjelajahi analisis dan model dan menyeleksi hipotesis. Melakukan <i>data mining</i> . Menafsirkan pola yang akan dicari. Bertindak terhadap pengetahuan yang berhasil ditemukan | 9 | akademik | Fayyad |
| Marketing; penjualan | Menyediakan rincian kunci langkah-langkah proses dari awal; Perhatian ditujukan kepada aplikasi pada pengetahuan yang ditemukan dan dokumentasi proyek | Identifikasi sumber daya manusia Spesifikasikan masalah <i>Data prospecting</i> <i>Knowledge elicitation</i> Identifikasi metodologi Preprocessing data Penemuan pola <i>Knowledge post-processing</i> | 8 | akademik | Anand & Buchner |
| Pengobatan; perangkat lunak | Gabungan dari model akademik dan model industri; mengidentifikasi dan menjelaskan beberapa umpun balik secara eksplisit | Memahami domain masalah yang ada. Memahami data Persiapan data <i>Data mining</i> Evaluasi terhadap penemuan pengetahuan yang ada Memahami pengetahuan yang telah ditemukan | 6 | hybrid | Cios et al. |
| Marketing; penjualan | Berorientasi pada bisnis dan mudah dipahami oleh spesialis non-data-mining | Penentuan tujuan bisnis Persiapan data <i>Data mining</i> Analisis hasil yang didapat Asimilasi masalah | 5 | industri | Cabena et al. |
| Sistem komersil Clementine; pengobatan; <i>engineering</i> ; marketing; penjualan | Menggunakan kosakata yang mudah dipahami; memiliki dokumentasi yang baik; membagi semua langkah menjadi sub langkah yang menyediakan semua detail yang dibutuhkan | Memahami terhadap bisnis yang sedang berjalan Memahami mengenai data Persiapan data Pembuatan model Evaluasi <i>Deployment</i> | 6 | industri | CRISP-DM |

Kebanyakan model memiliki langkah-langkah yang serupa, sementara beberapa langkah yang sama diantara kelima model tersebut yaitu: pemahaman domain, *data mining*, dan evaluasi dari pengetahuan yang telah ditemukan. Model yang memiliki sembilan langkah melakukan langkah-langkah yang memperhatikan tugas dari DM dan algoritma dari proses yang dijalankan. Model-model lain melakukan hal tersebut sebelum prepossessing data dilakukan untuk mendapatkan data yang benar untuk dipersiapkan untuk melakukan proses data mining tanpa harus mengulangi beberapa langkah sebelumnya. Dalam kasus model Fayyad, data yang disiapkan mungkin tidak cocok dan dengan demikian mungkin langkah akan berulang ke langkah kedua, ketiga, dan keempat mungkin diperlukan. Model yang memiliki lima langkah sangat mirip dengan model yang memiliki enam langkah, yang berbeda hanyalah langkah pemahaman data. Model yang memiliki delapan langkah memiliki rincian yang sangat rinci tentang fase awal yang terjadi pada langkah KDD, tetapi tidak ada langkah yang berkaitan dengan langkah yang akan dilakukan setelah pengetahuan telah ditemukan. Dari kelima model KDD yang disebutkan tidak bisa dipilih salah satu model yang terbaik diantara kelima model tersebut. Semua model memiliki kelebihan dan kekurangannya sendiri.

4. Masalah dalam penelitian

Tujuan utama dari model KDD adalah untuk dapat menintegrasikan keseluruhan proses untuk dapat mencapai standar prosedur industri.

Tujuan utama yang lainnya yaitu untuk menyediakan kompatibilitas

antara sistem dan platform yang berbeda yang digunakan selama proses berlangsung.

Metadata dan KDD

Tujuannya sendiri untuk memungkinkan pengguna untuk melakukan proses KDD tanpa memiliki latar belakang pengetahuan yang luas, tanpa manipulasi data secara manual, dan tanpa prosedur manual untuk pertukaran data dan pengetahuan antara metode *data mining*. Hasil ini membutuhkan kemampuan untuk menyimpan dan bertukar tidak hanya data namun juga pengetahuan yang dapat dinyatakan dalam bentuk model dan meta data yang dapat menjelaskan domain yang digunakan dalam prosesnya.

2.3 Metode Data Mining

Terdapat banyak metode/teknik/algorithm dan survei yang dilakukan oleh KDnuggets yang menunjukkan bahwa algoritma *data mining* yang banyak digunakan pada tahun 2011, dan algoritma tersebut adalah *decision tree/rules*, *regression*, *clustering*, *statistic (descriptive)*, *visualization*, *time series/sequence analysis*, *support vector (SVM)*, *association rules*, *ensemble methods and text*. Survei lainnya menunjukkan bahwa algoritma yang sering digunakan yaitu *C4.5*, *K-Means*, *Support Vector Machine (SVM)*, *Apriori*, *Expectation Maximization (EM)*, *PageRank*, *AdaBoost*, *k-Nearest Neighbors (kNN)*, *Naive Bayes and Classification and Regression Tress (CART)* (Wu, 2007).

Algoritma *data mining* dapat diklasifikasikan sebagai berikut: (Witten, 2011)

1. *Supervised learning*, dengan variabel *output* yang telah diketahui dan dikenal didalam dataset dan data *input* diberi label yang meliputi klasifikasi, klasifikasi *fuzzy*, *decision tree*, *Support Vector Machine (SVM)*, *artificial Neural Network*, *Naive Bayes* and *K-nearest Neighbor*.
2. *Unsupervised learning*, tanpa variabel *output* yang ada didalam dataset dan data *input* tidak diberi label yang meliputi *clustering*, *Expectation Maximization (EM)*, *association rule* and *Self- Organizing Map (SOM)*.

Algoritma *data mining* harus memiliki tiga komponen sebagai berikut:

1. Representasi Model (*Model Representation*), dimana bahasa tersebut digunakan untuk menggambarkan pola yang ditemukan.
2. Evaluasi Model (*Model Evaluation*), dimana laporan kuantitatif memenuhi tujuan dari proses KDD.
3. Metode pencarian yang memiliki dua komponen yaitu:
 - a. Pencarian parameter (*Parameter search*), dimana algoritma harus mencari parameter, yang mengoptimalkan criteria evaluasi model, yang berdasarkan data yang diobservasi dan model representasi yang tetap.
 - b. Pencarian model (*Model Search*), dimana terjadi pengulangan terus menerus terhadap parameter metode pencarian.

2.4 Atributed Oriented Induction

Metode *Attributed Oriented Induction (AOI)* pertama kali diperkenalkan pada tahun 1989 dimana metode ini mengintegrasikan paradigm *machine learning* terutama belajar dari contoh teknik dengan operasi *database*, mengekstraksi aturan umum dari sebuah set data yang menarik dan menemukan keteraturan data

level tingkat tinggi. AOI menyediakan mekanisme yang efisien dan efektif dalam menemukan berbagai macam aturan pengetahuan dari *database* atau *dataset*. Metode AOI ini sendiri telah diimplementasikan pada sebuah sistem *prototype* yang disebut *DBMINER* yang sebelumnya bernama *DBLEARN* dan telah melalui uji coba dengan sukses terhadap database relational yang memiliki ukuran besar. *DBLEARN* merupakan sebuah sistem *prototype* yang dikembangkan di Simon Fraser University. *DBMINER* dikembangkan dengan mengintegrasikan *database*, OLAP, dan teknologi *data mining*.

Pendekatan AOI dikembangkan untuk mempelajari berbagai jenis aturan pengetahuan seperti *characteristic rule*, *discrimination rules*, *classification rules*, *data evolution regularities association rules and cluster description rules* (Han, 1995).

1. *Characteristic rule* merupakan sebuah pernyataan yang mencirikan sebuah konsep yang dipenuhi data yang disimpan didalam *database*. *Rule* ini memberikan konsep umum yang dapat membantu orang dalam mengenali fitur-fitur umum dari data yang ada didalam kelas. Contohnya, gejala dari suatu penyakit tertentu.
2. *Discriminant rule* merupakan sebuah pernyataan, yang membedakan konsep dari satu kelas (target) dari yang lain (kontras). *Rule*/aturan ini memberikan kriteria diskriminan yang dapat digunakan untuk memprediksi keanggotaan kelas data baru, Contohnya untuk membedakan satu penyakit dengan penyakit lainnya.
3. *Classification rule* merupakan sejumlah aturan, yang mengklasifikasikan himpunan data yang relevan sesuai dengan satu atau lebih atribut yang

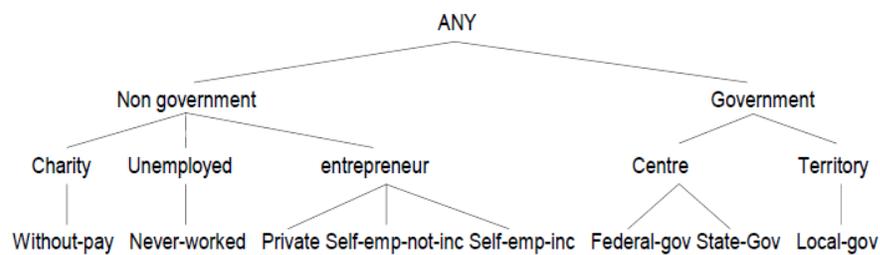
spesifik. Contohnya mengelompokkan penyakit ke dalam kelas dan memberikan gejalanya masing-masing.

4. *Association rule* merupakan sebuah hubungan asosiasi antara set-set data yang relevan. Contohnya, menemukan serangkaian gejala yang biasanya sering terjadi bersama-sama.
5. *Data evolution regularities rule* merupakan perilaku evolusi sebuah set data relevan yang umum. Misalnya, menjelaskan faktor-faktor utama yang mempengaruhi fluktuasi nilai saham seiring berjalannya waktu. Keteraturan evolusi *data* ini kemudian dapat diklasifikasikan menjadi *classification rule* dan *discrimination rule*.
6. *Cluster description rule* merupakan aturan yang digunakan untuk melakukan *cluster* data sesuai dengan data yang semantic. Contoh, pengelompokan mahasiswa berdasarkan dengan atribut yang berbeda.

2.4.1 Konsep Hirarki

Salah satu keuntungan dari AOI ini sendiri adalah dengan adanya konsep hirarki sebagai konsep dasar yang dapat disediakan oleh para ahli. Konsep hirarki menyimpan relasi pada suatu database yang menyediakan latar belakang pengetahuan penting untuk generalisasi data dan pada *data mining* yang memiliki beberapa level. Konsep hirarki ini merepresentasikan taksonomi dari suatu konsep nilai atribut. Konsep hirarki dapat dispesifikasikan berdasarkan hubungannya yang ada antara atribut *database* atau oleh sekelompok set dan disimpan dalam bentuk relasi didalam database yang sama (Han, 1996). Konsep hirarki dapat disesuaikan secara dinamis berdasarkan distribusi dari suatu set data yang

relevan dengan tugas-tugas *data mining*. Hirarki untuk atribut numerik dapat dibangun secara otomatis berdasarkan analisis distribusi data. Konsep hirarki untuk atribut numerik akan diperlakukan berbeda demi efisiensi. Contohnya, jika ada rentang nilai antara 0 dan 1,99, maka akan ada 199 nilai mulai dari 0,00 sampai 1,99, tetapi untuk masalah efisiensi maka hanya akan ada 1 *record* yang dibuat dengan 3 *fields* bukan dengan 200 *records* dengan 2 *fields*.



Gambar 2.4 Sebuah pohon hirarki konsep untuk atribut kerja dataset rentang usia dewasa(Frank, 2010)

| | |
|---|-------------------|
| Without-pay | ⊂ Charity |
| Never-worked | ⊂ Unemployed |
| {Private, self-emp-not-inc, self-emp-inc} | ⊂ Entrepreneur |
| {federal-gov, state-gov} | ⊂ Centre |
| Local-gov | ⊂ Territory |
| {Charity, Unemployed, Entrepreneur} | ⊂ Non government |
| {Centre, Territory} | ⊂ Government |
| {Non government, Government} | ⊂ ANY (workclass) |

Gambar 2.5 Hirarki konsep untuk atribut konsep pohon hirarki dengan rentang usia dewasa (Frank, 2010)

Dalam konsep hirarki, konsep diperintahkan sesuai dengan tingkatan mulai dari spesifik atau dari tingkatan bawah sampai dengan tingkatan umum atau tingkatan yang lebih tinggi. Generalisasi dapat dicapai dengan mengurutkan ke tingkatan yang memiliki tingkat lebih tinggi sesuai dengan jalur yang ada pada konsep hirarki. Konsep yang paling umum adalah dengan deksripsi *null* sebagai konsep yang paling spesifik sesuai dengan nilai-nilai tertentu dari atribut didalam *database*, yang dideksripsikan sebagai *ANY*. Konsep hirarki dapat seimbang ataupun tidak seimbang, dimana hirarki yang tidak seimbang harus dirubah menjadi hirarki yang seimbang. Gambar 2.4 menunjukkan bahwa pohon konsep hirarki untuk atribut kerja pada usia dewasa (Frank, 2010), mempunyai 3 tingkatan. Tingkatan pertama disebut sebagai tingkatan bawah yang memiliki 8 konsep yaitu *withoutpay*, *never-worked*, *private*, *self-emp-not inc*, *self-emp-inc*, *federal-gov*, *state-gov* and *localgov*. Tingkatan kedua memiliki 5 konsep yaitu *charity*, *unemployed*, *entrepreneur*, *centre* and *territory*. Tingkat ketiga memiliki 2 konsep yaitu *non government* dan *government*. Contohnya konsep dari *non government* berada pada tingkatan tinggi dan memiliki 3 sub konsep pada tingkatan kedua yaitu konsep *charity*, *unemployed* and *entrepreneur*. Konsep *entrepreneur* memiliki 3 sub konsep pada tingkatan rendah yaitu konsep: *private*, *self-emp-not-inc* and *self-emp-inc*. Konsep pohon hirarki pada gambar 2.4 dapat digambarkan menjadi gambar 2.5 dimana simbol \subset menunjukkan generalisasi. Contohnya *without-pay* \subset *charity* yang menunjukkan bahwa *charity* merupakan generalisasi dari *without-pay*.

2.4.2 AOI characteristic and discriminant rule

AOI dapat diimplementasikan dengan desain arsitektur dimana *characteristic rule* dan *discriminant rule* dapat dipelajari langsung dari database transaksional (OLTP) atau *data warehouse* (OLAP) (Han, 1999) dengan bantuan konsep hirarki sebagai generalisasi *knowledge*. Konsep hirarki dapat dibuat dari database OLTP sebagai sumber daya langsung.

Dari sebuah database, kita dapat mempelajari dua jenis pembelajaran:

1. Pembelajaran secara positif (*positif learning*) sebagai target dari kelas dimana data tuple ada didalam database, yang sesuai dengan konsep pembelajaran. Pembelajaran secara positif/kelas target akan dibangun saat belajar *characteristic rule*.
2. Pembelajaran secara negatif (*negative learning*) sebagai kelas kontras dimana data tidak dimiliki oleh kelas target. *Negative learning* akan dibangun ketika belajar *discriminant* atau *classification rule*.

Characteristic rule telah digunakan oleh AOI untuk mengenali, belajar, dan mencari karakter tertentu untuk tiap atribut sebagai karakter khusus mereka. Proses *characteristic rule* mengeneralisasi dengan bantuan dari konsep hirarki sebagai latar belakang untuk menemukan kelas target sebagai *positive learning*. Aturan proses *mining* tidak dapat dibatasi hanya sebatas satu aturan saja, karena semakin banyak aturan yang dapat dibuat maka akan semakin banyak *mining data* yang dapat dilakukan. Hal ini membuktikan bahwa ini merupakan suatu sistem yang cerdas, yang dapat

membantu manusia untuk membuat sistem yang memiliki kemampuan berpikir seperti manusia (Chen, 1996). Aturan akan sering ditemukan dengan mengeneralisasi dalam berbagai cara yang mungkin.

Database relasional sebagai sumber daya untuk *data mining* untuk AOI dapat dibaca dengan data *manipulation language select sql statement*. Dengan menggunakan *query* untuk membangun sebuah aturan dapat memberikan mekanisme yang efisien untuk memahami aturan yang dilakukan proses *mining*. Dalam proses AOI saat ini, sebuah *query* diproses dengan bahasa SQL. Hal ini mengumpulkan set data yang relevan dengan merubah *relational query*, mengeneralisasi data oleh AOI dan kemudian menyajikan output dalam bentuk yang berbeda.

AOI mengeneralisasi dan mengurangi hubungan yang kuat sampai akhir hubungan dapat memenuhi harapan pengguna berdasarkan *threshold* yang telah disiapkan. Satu atau dua *threshold* dapat diterapkan, dimana satu *threshold* digunakan untuk mengontrol jumlah atribut dan tupel yang berbeda dalam proses generalisasi, sementara dua *threshold* digunakan untuk mengontrol jumlah atribut dan tupel yang berbeda dalam proses generalisasi. *Threshold* digunakan sebagai pengontrol jumlah maksimal dari tupel untuk jumlah kelas target dapat digantikan oleh sejumlah grup operator SQL yang akan membatasi hasil akhir dari proses generalisasi. Dalam prakteknya bahwa menetapkan *threshold* yang berbeda akan menghasilkan tupel yang berbeda sesuai dengan yang diinginkan namun akan memakan waktu dan membosankan.

AOI dapat melakukan teknik datawarehouse dengan melakukan proses generalisasi berulang-ulang untuk menghasilkan aturan pada tingkat konsep yang berbeda dalam konsep hirarki, yang memungkinkan pengguna untuk menemukan tingkat penemuan dan aturan yang paling cocok.

Ada delapan langkah strategi yang harus dilakukan pada proses generalisasi yang diungkapkan Han (1992). Berikut dilampirkan langkah satu sampai tujuh untuk *characteristic rule* dan langkah satu sampai delapan untuk *discriminant rule*.

1. *Generalization on the smallest decomposable components*, generalisasi harus dilakukan pada komponen terurai terkecil dari hubungan data.
2. *Attribute removal*, apabila terdapat satu set nilai yang berbeda untuk atribut tetapi tidak ada tingkatan konsep yang lebih tinggi, maka atribut tersebut harus dihapus pada saat proses generalisasi sedang berlangsung
3. *Concept tree Ascension*, apabila pada konsep hirarki terdapat tingkatan konsep yang lebih tinggi untuk nilai dari tupel, substitusi nilai dengan nilai yang memiliki tingkatan konsep lebih tinggi akan mengeneralisasi tupel.
4. *Vote propagation*, nilai *vote* merupakan nilai akumulasi tupel dimana *vote* akan diakumulasikan saat penggabungan tupel-tupel yang identik dalam proses generalisasi.

5. *Threshold control on each attribute*, jika jumlah nilai yang berbeda dalam hubungan yang dihasilkan dapat lebih besar dibandingkan dengan nilai *threshold* yang telah ditentukan, proses generalisasi lebih lanjut harus dilakukan.
6. *Threshold control on generalized relations*, jika jumlah tuple lebih besar dibandingkan nilai *threshold* yang telah ditentukan sebelumnya, generalisasi lebih lanjut akan dilakukan berdasarkan atribut yang dipilih dan digabungkan berdasarkan tuple identik harus dilakukan.
7. *Rule transformation*, mengubah generalisasi akhir untuk aturan kuantitatif dan kualitatif dari sebuah tuple (*conjunctive*) dan beberapa tuples (*disjunctive*)
8. *Handling overlapping tuples*, jika terjadi tumpang tindih tupel pada kedua target dan kelas kontras, maka tuple ini harus ditandai dan dibuang dari hubungan generalisasi akhir.

Berikut merupakan karakteristik algoritma untuk mencari *characteristic rule* AOI (Han, 1992):

| |
|---|
| AOI characteristic rule algorithm: |
| <i>Input: dataset, concept hierarchies, learning task, attribute threshold, rule threshold</i> |
| <i>Output: characteristic rule of the learning task</i> |
| <ol style="list-style-type: none"> 1 For each of attribute A_i ($1 \leq i \leq n$, where $n = \#$ of attributes) in the generalized relation GR 2 { While $\#$_of_distinct_values_in_attribute_ $A_i >$ threshold 3 { If no higher level concept in concept hierarchy for attribute A_i 4 Then remove attribute A_i |

```

5 Else substitute the value of  $A_i$  by its corresponding minimal
generalized concept
6 Merge identical tuples
7 }
8 }
9 While #_of_tuples in GR > threshold
10 { Selective generalize attributes
11 Merge identical tuples
12 }

```

Algoritma AOI *characteristic rule* merupakan implementasi dari langkah-langkah strategi generalisasi. Algoritma ini menunjukkan dua sub proses yaitu *control number of distinct attributes* dan *control number of tuples*.

1. *Control number of distinct attributes* adalah proses vertikal yang memeriksa tiap-tiap atribut. Hal ini dilakukan dengan memeriksa semua atribut dari hasil belajar dataset sampai jumlah atribut yang berbeda kurang lebih sama dengan jumlah *threshold*. Tiap-tiap atribut yang memiliki atribut yang berbeda yang memiliki nilai lebih besar dibandingkan dengan nilai *threshold* akan diperiksa apakah tingkatnya lebih tinggi dalam konsep hirarkinya. Jika atribut tersebut tidak memiliki konsep tingkat yang lebih tinggi maka atribut tersebut tidak akan digunakan. Di sisi lain jika atribut tersebut memiliki atribut yang lebih tinggi maka nilai atribut tersebut akan diganti dengan nilai konsep yang lebih tinggi. Penggabungan tupel yang identik akan selesai dilakukan dalam rangka untuk menyelesaikan generalisasi dan mengumpulkan nilai

vote dari tuple dengan mengeliminasi tupel yang redundan. Setelah proses pertama ini selesai maka semua atribut akan memiliki jumlah yang atribut berbeda yang kurang lebih sama dengan jumlah *threshold*. Sub proses pertama ini merupakan implementasi dari langkah 1-5 strategi generalisasi.

2. *Control number of tuples* merupakan proses horizontal, dimana aturan dicek secara horizontal setiap aturannya. Hal ini dilakukan untuk atribut-atribut yang lulus pada sub proses pertama dimana setiap atribut akan memiliki jumlah atribut berbeda yang kurang lebih sama dibandingkan ambang *threshold*. Sub proses kedua ini hanya dilakukan apabila jumlah *rule* lebih besar dibandingkan dengan jumlah *threshold*. Proses generalisasi secara selektif dan beberapa penggabungan tupel yang identik akan mengurangi jumlah *rule* yang ada. Seperti sub proses pertama menggabungkan tupel yang identik akan dilakukan dalam rangka untuk menyelesaikan generalisasi dan mengumpulkan tupel yang identik untuk menghilangkan tupel yang berlebihan. Setelah subproses kedua ini selesai maka jumlah *rule* akan kurang lebih sama dengan *threshold*. Sub proses kedua ini merupakan proses implementasi dari tahap 3, 4, dan 6 dari tahap generalisasi.

Algoritma *AOI characteristic rule* merupakan implementasi dari langkah satu sampai delapan pada proses generalisasi. Karena algoritma *AOI discriminant rule* dan *AOI characteristic rule* mempunyai langkah-langkah strategi generalisasi yang sama antara langkah satu sampai tujuh

yang secara harafiah memiliki proses yang sama dan perbedaannya hanya dalam langkah kedelapan. Mereka juga memiliki proses yang samayaitu nomor control atribut yang berbeda sebagai sub proses pertama dan nomor dari tupel sebagai sub proses kedua. Langkah penanganan yang dilakukan apabila tupel saling bertumpang tindih adalah proses diawal sebelum subproses pertama dan kedua dalam proses pertama dan kedua sebelum menggabungkan tupel yang identical.

Berikut merupakan karakteristik algoritma untuk *discriminant rule AOI*(Han, 1993):

| |
|---|
| AOI discriminant rule algorithm: |
| <i>Input: dataset, concept hierarchies, learning task, attribute threshold, rule threshold</i> |
| <i>Output: discriminant rule of the learning task</i> |
| <ol style="list-style-type: none"> 1 For each of attribute A_i ($1 \leq i \leq n$, where $n = \#$ of attributes) in the generalized relation GR 2 { Mark the overlapping tuples 3 While <i>#_of_distinct_values_in_attribute_Ai</i> > threshold 4 { If no higher level concept in concept hierarchy for attribute A_i 5 Then remove attribute A_i 6 Else substitute the value of A_i by its corresponding minimal generalized concept 7 Mark the overlapping tuples 8 Merge identical tuples 9 } 10 } 11 While <i>#_of_tuples in GR</i> > threshold 12 { Selective generalize attributes 13 Mark the overlapping tuples 14 Merge identical tuples 15 } |

2.5 Data Warehouse

2.5.1 Konsep Data Warehouse

Dalam dunia computer data warehouse adalah sebuah database untuk menyimpan data *history* yang sangat besar. Berdasarkan definisi (Inmon, 2005, pp.29), Data Warehouse (DWH) adalah *subject-oriented*, *integrated*, *time-variant* dan *non-volatile* dari kumpulan data untuk membantu proses pengambilan keputusan oleh manajemen. Berikut ini adalah penjelasan secara detail dari definisi DWH (Connolly & Begg, 2009, pp.1197):

- *Subject-oriented* artinya DWH haruslah diorganisasikan berdasarkan subjek yang besar yang di dalam suatu lingkungan *enterprise* (seperti data pelanggan, produk dan sales) bukan berdasarkan aplikasi besar yang digunakan (seperti invoice pelanggan, kontrol stok, dan sales produk). Hal ini menggambarkan bahwa menyimpan data *decision-support* lebih tepat daripada data *application-oriented*
- *Integrated* karena menggabungkan beberapa data yang berasal dari sumber berbeda dari sebuah sistem *enterprise*. Sering kali data sumber tersebut terdapat inkonsistensi dikarenakan perbedaan format data. Integrasi sumber data-data tersebut haruslah menjadi konsisten untuk menampilkan tampilan data yang seragam kepada pengguna.

- *Time-variant* karena data didalam DWH hanya akurat dan valid pada beberapa waktu tertentu atau selama beberapa interval waktu. Data-data tersebut merupakan serangkaian daripada *snapshot*.
- *Non-volatile* karena data tidak diperbarui secara *realtime* melainkan *direfresh* secara *periodic* dari sistem operasional. Data baru selalu ditambahkan sebagai suplemen untuk database bukan mengganti data yang sudah ada. Database selalu menyimpan data baru ini dan secara bertahap mengintegrasikan dengan data yang sebelumnya.

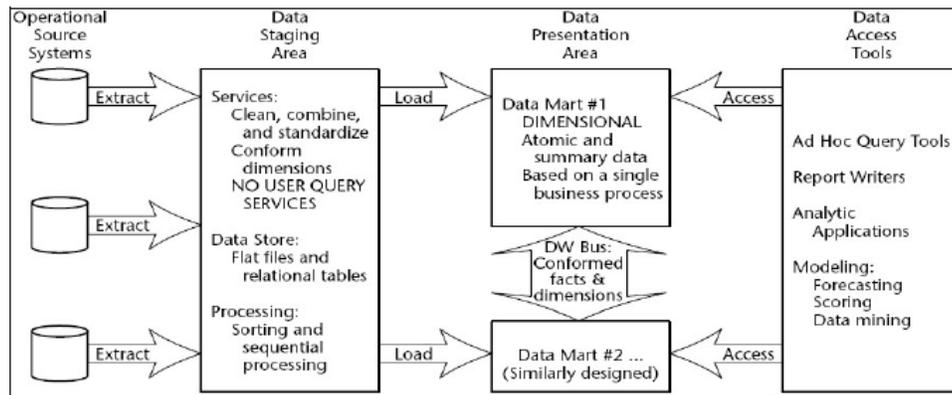
2.5.2 OLTP dan DWH

Secara umum sistem basis data dalam suatu perusahaan bisa dibedakan menjadi dua tipe yaitu *Online Transaction Processing* (OLTP) dan *Online Analytical Processing* (OLAP). Sistem OLTP didesain untuk memroses data dari suatu aplikasi korporat sehingga didalamnya terdapat banyak proses manipulasi data. Data transaksi tersebut akan digunakan oleh DWH sebagai sumber data yang kemudian diolah untuk digunakan dalam aktivitas analisa. Alat bantu analisa inilah yang biasa disebut OLAP. OLTP merupakan sistem untuk transaksi sehingga didalamnya perubahan data sangat dinamis sedangkan DWH merupakan data histori yang digunakan untuk analisis sehingga data didalamnya bersifat statis.

2.5.3 Arsitektur DWH

Arsitektur dari DWH secara garis besar dibagi menjadi empat elemen / komponen utama yaitu *operational source system*, *data staging*

area, data presentation area dan data access tools(Kimball & Ross, 2002, pp.7). Berikut adalah gambar arsitektur komponen-komponen dalam DWH.



Gambar 2.6 Arsitektur Data Warehouse(Kimball & Ross, 2002, pp.7)

Operational Source System (OSS) merupakan sistme yang menyimpan semua data transaksi dari suatu aplikasi bisnis. Sehingga dibutuhkan kemampuan proses yang bagus dan ketersediaan sistem yang handal. Data-data transaksi tersebut diolah (*extract*) kedalam data staging.

Data Staging Area (DSA) didalamnya terdapat proses yang sering dikenal sebagai *extract-transfomation-load* (ETL). Selain proses tersebut diperlukan juga media penyimpanan sementara ketika data tersebut diolah dan sebelum diload kedalam komponen selanjutnya

Data Presentation Area (DSP) merupakan tempat dimana data telah diaggregasi melalui proses ETL dan sudah siap untuk diquery dalam proses analisa. Model data yang digunakan dalam DSP ini berbeda dengan model data pada OSS, dimana model yang digunakan adalah dimensional model bukan *third-normal-form* (3NF). Dalam dimensional model terdapat

2 tipe tabel yaitu *fact* dan *dimension* yang digabungkan dalam suatu *data mart* berdasarkan proses bisnis yang ada

Komponen terakhir dalam arsitektur DWH adalah *Data Access Tools* (DAT). Alat bantu ini digunakan untuk mempermudah melakukan query dan analisa data dari DWH. Banyak tersedia aplikasi baik komersial maupun gratis untuk DAT.

2.6 Decision Support System

Decision Support System merupakan sebuah sistem yang interaktif, dan berbasis computer yang membantu dalam memilih dan menilai sebuah aktifitas. Sistem ini menyediakan penyimpanan dan pengambilan data lalu juga meningkatkan kemudahan dalam pengaksesan informasi dan pengambilan beberapa fungsi untuk membangun model berbasis dukungan. Sistem ini mendukung *framing*, *modeling*, dan pemecahan masalah.

Aplikasi khas dari DSS ini mencakup bagian manajemen dan membuat rencana dalam bisnis, kesehatan, militer, dan berbagai daerah dimana manajemen akan menghadapi situasi saat menentukan keputusan yang kompleks. *Decision support system* biasanya digunakan untuk membuat keputusan strategis dan keputusan yang bersifat taktis yang dihadapi oleh manajemen tingkat atas, selain itu juga digunakan saat untuk membuat suatu keputusan yang memiliki konsekuensi yang cukup tinggi, dimana waktu yang dibutuhkan untuk memikirkan suatu keputusan dapat dikurangi.

Ada tiga komponen fundamental dalam DSS (Andrew, 1991):

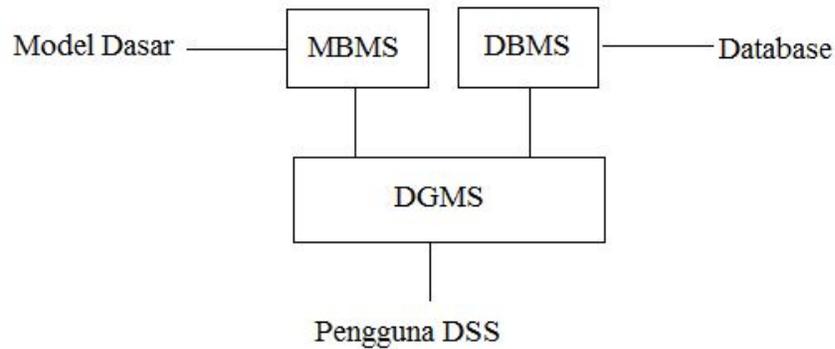
- *Database management system* (DBMS): Sebuah DBMS berfungsi sebagai sebuah bank data untuk sebuah DSS. Sebuah DBMS dapat menyimpan

sejumlah data yang sangat besar yang relevan dengan masalah yang sedang dihadapi dimana DSS telah dirancang dan menyediakan struktur data yang logis untuk pengguna dapat berinteraksi. Sebuah DBMS memisahkan pengguna dari aspek fisik dari struktur database dengan prosesnya. DBMS juga harus mampu menginformasikan pengguna mengenai jenis data yang tersedia dan bagaimana mendapatkan akses ke suatu data tersebut.

- *Model Base management System (MBMS)*: Tugas atau peran dari MBMS analog dengan DBMS. Fungsi utamanya adalah menyediakan kebebasan antara *specic model* dimana digunakan dalam proses DSS dari suatu aplikasi. Tujuan dari MBMS ini sendiri adalah untuk mengubah data dari MBMS menjadi suatu informasi yang berguna untuk pengambilan keputusan. Karena banyak masalah bahwa pengguna dari DSS, maka MBMS ini membantu para pengguna dalam membuat model dalam menyelesaikan masalah.
- *Dialog generation and management system (DGMS)*: Produk utama dari interaksi dengan sebuah DSS adalah pengetahuan. Pengguna yang banyak menggunakan produk DSS biasanya adalah *manager* yang tidak dilatih untuk menggunakan komputer, maka DSS perlu dilengkapi dengan *user interface* yang intuitif dan mudah digunakan. Tatap muka (*user interface*) ini membantu dalam pembangunan model, dan membantu dalam berinteraksi dengan model itu sendiri, misalnya dengan mendapatkan pengetahuan dan rekomendasi dari sistem tersebut. Tanggung jawab utama

dari DGMS adalah untuk meningkatkan kemampuan pengguna sistem untuk memanfaatkan dan mendapatkan manfaat dari DSS.

Ada banyak DSS yang beredar saat ini, ketiga komponen utama dari DSS yang dijelaskan diatas dapat banyak ditemukan pada banyak arsitektur DSS yang beredar saat ini, ketiga komponen tersebut memainkan peranan penting dalam struktur dari DSS ini. Interaksi dari ketiga komponen tersebut terlihat pada gambar 2.7.



Gambar 2.7 Arsitektur DSS(Andrew,1991)

Pada dasarnya, pengguna berinteraksi dengan DSS melalui DGMS. DSS juga berkomunikasi dengan DBMS dan MBMS