BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

1.1. Tinjauan Pustaka

"Penerapan algoritma *C4.5* untuk klasifikasi predikat kelulusan mahasiswa Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta" [1]. Penelitian yang dilakukan bertujuan untuk menganalisa data yang telah bertambah banyak didalam *database* yang dimiliki. Sementara itu, data-data yang melimpah ini bisa dimanfaatkan sebagai sumber informasi strategis bagi program studi untuk memprediksi masa studi dan predikat kelulusan mahasiswa dengan menggunakan teknik teknik data mining. Penelitian ini menggunakan algoritma *C4.5* guna mendukung proses penelitian. Ada 7 tahap yang dilakukan dalam penelitian ini yang pertama yaitu, studi literatur atau kepustakaan yang dilakukan dengan menelusuri literatur untuk menggali teori-teori yang sedang berkembang, mencari metode penelitian yang digunakan terdahulu dan untuk memperoleh orientasi yang ada dalam permasalahan. Kedua, pemilihan obyek penelitian yang dilakukan untuk memprediksi dan mengklasifikasikan indeks prestasi komulatif mahasiswa Fakultas Komunikasi dan Informatika UMS. Ketiga, penentuan variabel *data mining*. Variabel-variabel yang akan digunakan untuk proses *data mining* ini ditentukan berdasarkan tujuan penelitian. Keempat, penentuan nilai *class* variabel. Kelima, pengumpulan data. Sesudah data terkumpul langkah selanjutnya yaitu, mengelolah data. Olah data yang dilakukan menurut meliputi pemisahan atribut-atribut yang diperlukan untuk proses *data mining*. Tahap terakhir yaitu, menganilisis data.

Berikutnya penulis mengutip dari artikel dengan judul "*Data mining* menggunakan algoritma *Naïve bayes* untuk klasifikasi kelulusan mahasiswa Universitas dian nuswantoro" [2]. Data mahasiswa dan data kelulusan mahasiswa Dian Nuswantoro menghasilkan data yang sangat berlimpah berupa data profil mahasiswa dan data akademik. Hal tersebut terjadi secara berulang dan menimbulkan penumpukan terhadap data mahasiswa sehingga mempengaruhi pencarian informasi terhadap data tersebut. Penelitian ini bertujuan untuk melakukan klasifikasi terhadap data mahasiswa Universitas Dian Nuswantoro Fakultas Ilmu Komputer angkatan 2009 berjenjang DIII dan S1 dengan memanfaatkan proses *data mining* dengan menggunakan teknik klasifikasi. Metode yang digunakan adalah *CRISP-DM* dengan melalui proses *business understanding*, *data understanding*¸ data *preparation*, *modeling*, *evaluation* dan *deployment*. Algoritma yang digunakan untuk klasifikasi kelulusan adalah algoritma *Naïve Bayes*. *Naïve Bayes* merupakan teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema atau aturan bayes dengan asumsi independensi yang kuat pada fitur, artinya bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama. Implemtasi menggunakan *RapidMiner* 5.3 digunakan untuk membantu menemukan nilai yang akurat. Atribut yang digunakan adalah NIM, Nama, Jenjang, Progdi, Provinsi Asal, Jenis Kelamin, SKS, IPK, dan tahun lulus. Hasil dari penelitian ini digunakan sebagai salah satu dasar pengambilan keputusan untuk menentukan kebijakan oleh pihak Fasilkom.

Selanjutnya penulis mengambil referensi jurnal dari internet dengan judul " Implementasi *data mining* dengan algoritma *C4.5* untuk memprediksi tingkat kelulusan mahasiswa" [3]. Pada penelitian ini penulis menggunakan algoritma *C4.5* dalam menentukan prediksi kelulusan berdasarkan attribute jenis kelamin, asal sekolah SMA dan IP semester satu sampai dengan semester enam. Algoritma *C4.5* merupakan algoritma klasifikasi pohon keputusan yang banyak digunakan karena memiliki kelebihan utama dari algoritma yang lainya. Kelebihan algoritma *C4.5* dapat menghasilkan pohon keputusan yang mudah diinterprestasikan, memiliki tingkat akurasi yang dapat diterima,efisien dalam menangani atribut bertipe diskrit dan numeric. Dalam mengkontruksi pohon, Algoritma *C4.5* membaca seluruh sampel *data training* dari *stoage* dan memuatnya ke memori. Hal ini lah yang menjadi salah satu kelemahan algoritma *C4.5* dalam kategori "skalabilitas" adalah algoritma ini hanya dapat digunakan jika *data training* dapat disimpan secara keseluruhan dan pada waktu yang bersamaan dimemori. *Data training* yang akan digunakan oleh peneliti adalah data alumni mahasiswa program studi teknik informatika universitas multimedia nusantara angkatan 2007dan 2008 sedangkan untuk data testing akan digunakan data alumni angkatan 2009. Dari kumpulan *data training* dan data testing, dapat diketahui informasi kelulusan yang dapat mempengaruhi beberapa keputusan program studi menggunakan data mining algoritma *C4.5.*

Hasil penelitian yang telah diuraikan oleh para peneliti diatas, memiliki penerapan *data mining* yang beragam. Penelitian pertama, kedua dan ketiga memiliki tujuan yang hampir sama yaitu memfokuskan mencari sebuah informasi

pada data yang telah menumpuk di dalam *database* sebuah Universitas. Informasi yang dicari di dalam *database* ini yaitu tentang faktor yang mempengaruhi tingkat kelulusan mahasiswa agar informasi yang di dapat bisa dijadikan sebagai salah satu bahan evaluasi bagi Universitas untuk selanjutnya bisa menjadi strategi dalam proses perkuliahan, supaya tingkat kelulusan semakin meningkat. Perbedaan penelitian yang dilakukan oleh peneliti diatas terletak pada atribut dan algoritma yang digunakan, peneliti yang pertama dan ketiga menggunakan algoritma *C4.5* sedangkan peneliti kedua menggunakan algoritama *Naïve bayes.* Penelitian yang pertama dan ketiga hampir sama kasusnya dengan yang dibuat oleh penulis, perbedaan terletak pada atribut dan algoritma yang digunakan.

2.2. Landasan Teori

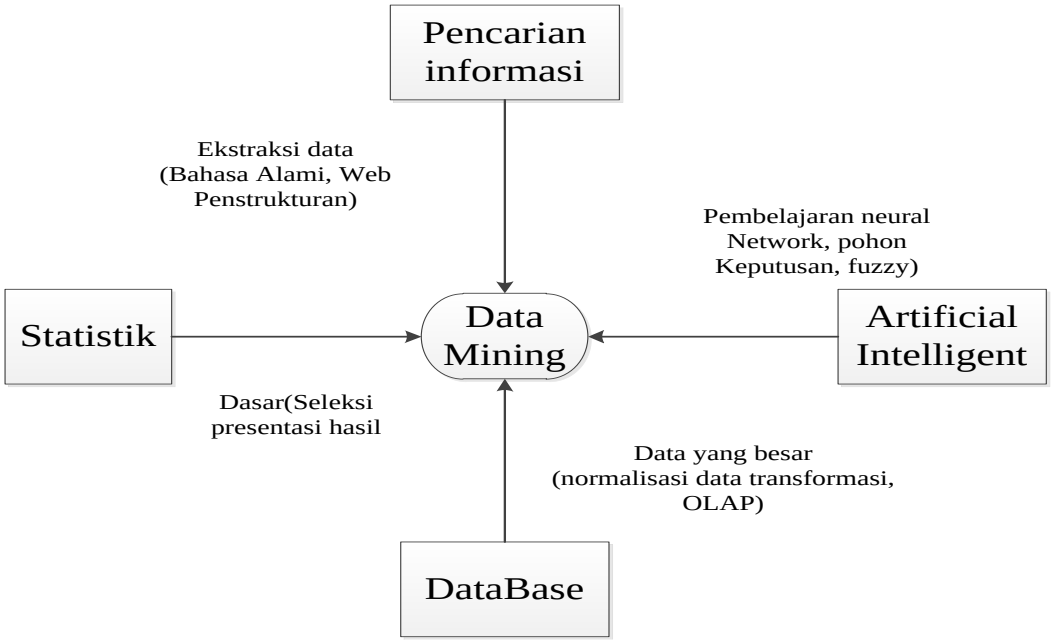
2.2.1. *Data Mining*

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam *database. Data mining* adalah prosses yang menggunakan teknik *statistic*, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai *databas*e besar [4].

Kemampuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa *factor*, antara lain [4]:

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam *data warehouse*, sehingga seluruh perusahaan memiliki akses kedalam *database* yang andal.
3. Adanya peningkatan akses data melalui navigasi *web* dan *internet*.

4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersedian Teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.



Gambar 2.1 Bidang ilmu *data mining.*

Data mining bukanlah suatu bidang yang sama sekali baru. Salah satu kesulitan untuk mendefinisikan *data mining* adalah kenyataan bahwa *data mining* mewarisi banyak aspek dan teknik dari bidang-bidang ilmu Yang sudah mapan terlebih dahulu. Gambar 2.1 menunjukan bahwa *data mining* memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent), machine learning,* statistic, database dan juga *information retrieval* [4].

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yag besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining.* Proses KDD secara garis besar dapat dijelaskan sebagai berikut [4]:

1. *Data selection*

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing*

   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi *focus* KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa yang inkonsiste, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD

merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

   Data mining adalah proses mencari pola atau informasi menarik dalam data teripilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretationall*

   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.2.1.1. Pengelompokan data mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan yaitu [4]:

1. Deskripsi

   Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup professional akan sedikit didukung dalam pemilihan presiden. Deskripsi

dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali *variable* target estimasi lebih kearah *numeric* dari pada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari *variable* target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari *variable* target dibuat berdasarkan nilai *variable* prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai *variable* prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainya. Contoh lainya yaitu estimasi nilai indeks prestasi komulatif mahasiswa program pascasarjana dengan melihat nilai indeks prestasi mahasiswa tersebut pada saaat mengikuti program sarjana.

3. Prediksi

   Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada dimasa mendatang.

   Contoh prediksi dalam bisnis dan penelitian:

   - Prediksi harga beras dalam tiga bulan yang akan datang.
   - Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi

   Dalam klasifikasi, terdapat target *variable* kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi pendapatan sedang, pendapatan rendah.

   Contoh lain dalam klasifikasi dalam bisnis dan penelitian adalah:

   - Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang apa bukan.
   - Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
   - Mendiagnosis penyakit seseorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. *Clustering*

   Clustering merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain.

   Clustering berbeda dengan klasifikasi yaitu tidak adanya variable target dalam *clustering*. *Clustering* tidak mencoba untuk melakuakn klasifikasi, mengestimasi, atau memprediksi nilai dari variable target. Akan tetapi, algoritma *clustering* mencoba untuk melakukan pembagian terhadap

keseluruhan data menjadi kelompok-kelompokan yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

Contoh *clustering* dalam bisnis dan penelitian adalah:

- Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap prilaku finansial dalam baik dan mencurigakan.
- Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian adalah :

- Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.
- Menemukan barang dalam supermarket yang dibeli bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

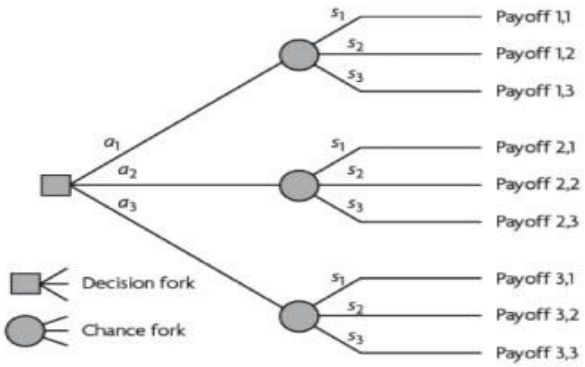
2.2.2. Pohon keputusan (*Decision Tree*).

Seiring dengan perkembangan kemajuan pola pikir manusia, manusia mulai mengembangkan sebuah sistem yang dapat membantu manusia dalam menghadapi masalah-masalah yang timbul sehingga dapat menyelesaikannya dengan mudah. Pohon keputusan atau yang lebih dikenal dengan istilah *Decision Tree* ini merupakan implementasi dari sebuah sistem yang manusia kembangkan dalam mencari dan membuat keputusan untuk masalah-masalah tersebutdengan memperhitungkan berbagai macam faktor yang berkaitan di dalam lingkup masalah tersebut. Dengan pohon keputusan, manusia dapat dengan mudah mengidentifikasi dan melihat hubungan antara faktor-faktor yang mempengaruhi suatu masalah sehingga dengan memperhitungkan faktor-faktor tersebut dapat dihasilkan penyelesaian terbaik untuk masalah tersebut. Pohon keputusan ini juga dapat menganalisa nilai resiko dan nilai suatu informasi yang terdapat dalam suatu alternatif pemecahan masalah[5].

Pohon keputusan dalam analisis pemecahan masalah pengambilan keputusan merupakan pemetaan alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah tersebut. Pohon keputusan juga memperlihatkan faktor-faktor kemungkinan yang dapat mempengaruhi alternative-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila kita mengambil alternatif keputusan tersebut. Secara umum, pohon keputusan adalah suatu gambaran permodelan dari suatu persoalan yang terdiri dari serangkaian keputusan yang mengarah kepada solusi yang dihasilkan. Peranan pohon keputusan sebagai alat bantu dalam mengambil keputusan telah dikembangkan

oleh manusia sejak perkembangan teori pohon yang dilandaskan pada teori graf. Seiring dengan perkembangannya, pohon keputusan kini telah banyak dimanfaatkan oleh manusia dalam berbagai macam sistem pengambilan keputusan[5].

Decision tree adalah struktur *flowchart* yang menyerupai tree (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. Alur pada *decision tree* di telusuri dari simpul akar ke simpul daun yang memegang prediksi.



Gambar 2.2 Bentuk *Decision Tree* Secara Umum[5].

2.2.3. Algoritma *Induction Decision Tree* (ID3)

ID3 merupakan sebuah metode yang digunakan untuk membangkitkan pohon keputusan. Input dari algoritma ini adalah sebuah *database* dengan beberapa *variable* yang juga dikenal dengan atribut. Setiap masukan dalam *database* menyajikan sebuah objek dari domain yang disebut dengan *variable* bebas (*independent variable*). Sebuah atribut didesain untuk mengklasifikasikan objek yang disebut dengan *variable* tidak bebas (*dependent variable*).

Proses klasifikasi dilakukan dari node yang paling atas yaitu akar pohon (*root*). Dilanjutkan kebawah melalui cabang-cabang sampai dihasilkan node daun (*leaves*) dimana node daun ini menunjukan hasil akhir klasifikasi. Sebuah objek yang diklasifikasikan dalam pohon harus dites nilai *entropy*nya. *Entropy* adalah ukuran dari teori informasi yang dapat mengetahui karateristik *impurity* dan *homogeneity* dari kumpulan data. Dari nilai *entropy* tersebut kemudian dihitung nilai *information gain* (IG) masing-masing atribut *independent* terhadap atribut *dependent*-nya. IG merupakan nilai rata-rata *entropy* pada semua atribut[11].

2.2.3.1 Konsep *Entropy*

Entropy (S) merupakan jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada ruang sampel S. *Entropy* dapat dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil nilai *entropy* maka akan semakin *entropy* digunakan dalam mengekstrak suatu kelas. *Entropy* digunakan untuk mengukur ketidakaslian S[4].

Untuk menghitung nilai *entropy* harus menggunakan rumus *entropy* yang dapat dilihat pada persamaan 1 berikut.

$$Entropy(S) = \sum_{i=1}^{n} - Pi * \log_2 Pi$$

Keterangan :

- S : himpunan kasus
- A : fitur
- n : jumlah pasrtisi S

- pi : proporsi dari Si terhadap S

2.2.3.2 Konsep *Gain*

Gain (S,A) merupakan perolehan informasi dari atribut A relative terhadap output data S. Perolehan informasi didapat dari *output* data atau *variable* dependent S yang dikelompokan berdasarkan atribut A, dinotasikan dengan *gain* (S,A) [7]. Untuk menghitung nilai *gain* harus menggunakan rumus *gain* yang dapat dilihat pada persamaan 2 berikut.

$$Gain(S, A) = Entropy(\mathrm{S}) - \sum_{i=1}^{n} \frac{|Si|}{S} * Entropy(Si)$$

Keterangan:

- A : Atribut
- S : Sampel
- n : Jumlah partisi himpunan atribut A
- |Si| : jumlah sampel pada partisi ke –i
- |S| : jumlah sampel dalam S

2.2.4. *Software Development Life Cycle* (SDLC)

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle).* Metode ini merupakan siklus pengembangan perangkat lunak yang terdiri dari beberapa tahapan penting dalam membangun perangkat lunak yang dilihat dari segi pengembanganya. Metode ini ada 4 macam

model yaitu *waterfall, prototype, RAD, Agile Software Development.* Disini penulis menggunakan *waterfall.*

Menurut Pressman(2010) *Classic life cycle* atau model *waterfall* merupakan model yang paling banyak digunakan di dalam *software engginering.* Model ini melakukan pendekatan secara sistematis. Model ini disebut juga model berulang karena jika terjadi kesalahan dalam salah satu daftar tahapan maka dapat kembali ketahapan sebelumnya sampai selesai sehingga bisa melanjutkan ketahapan selanjutnya.

2.2.5 *RapidMiner*

RapidMiner merupakan perangakat lunak yang bersifat terbuka (open source). *RapidMiner* adalah sebuah solusi untuk melakukan analisis terhadap *data mining*, *text mining* dan analisis prediksi. *RapidMiner* menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. *RapidMiner* memiliki kurang lebih 500 operator *data mining*, termasuk operator untuk *input, output, data preprocessing* dan visualisasi. *RapidMiner* merupakan *software* yang berdiri sendiri untuk analisis data dan sebagai mesin *data mining* yang dapat diintegrasikan pada produknya sendiri. *RapidMiner* ditulis dengan munggunakan bahasa *java* sehingga dapat bekerja di semua sistem operasi [5].

RapidMiner sebelumnya bernama YALE (*Yet Another Learning Environment*), dimana versi awalnya mulai dikembangkan pada tahun 2001 oleh RalfKlinkenberg, Ingo Mierswa, dan Simon Fischer di Artificial Intelligence Unit

dari University of Dortmund. *RapidMiner* didistribusikan di bawah lisensi AGPL (GNU Affero General Public License) versi 3. Hingga saat ini telah ribuan aplikasi yang dikembangkan mengunakan *RapidMiner* di lebih dari 40 negara. *RapidMiner* sebagai *software open source* untuk *data mining* tidak perlu diragukan lagi karena *software* ini sudah terkemuka di dunia. *RapidMiner* menempati peringkat pertama sebagai *Software data mining* pada polling oleh KDnuggets, sebuah portal *data-mining* pada 2010-2011[5].

RapidMiner menyediakan GUI (*Graphic User Interface)* untuk merancang sebuah *pipeline analitis*. GUI ini akan menghasilkan file XML(*Extensible Markup Language*) yang mendefenisikan proses analitis keinginan pengguna untuk diterapkan ke data. File ini kemudian dibaca oleh *RapidMiner* untuk menjalankan analis secara otomatis[5].

RapidMiner memiliki beberapa sifat sebagai berikut[5]:

- Ditulis dengan bahasa pemegroman *java* sehingga dapat dijalankan di berbagai sistem operasi.
- Proses penemuan pengetahuan dimodelkan sebagai *operator trees.*
- Representasi XML internal untuk memastikan format standar pertukaran data.
- Bahasa *scripting* memungkinkan untuk eksperiman skala besar dan otomatisasi eksperimen.
- Konsep *multi-layer* untuk menjamin tampilan data yang efisien dan menjamin penangan data.

- Memiliki GUI, *command line mode* dan *Java API* yang dapat dipanggil dari program lain.

Beberapa fitur dari *RapidMiner*, antara lain [5]:

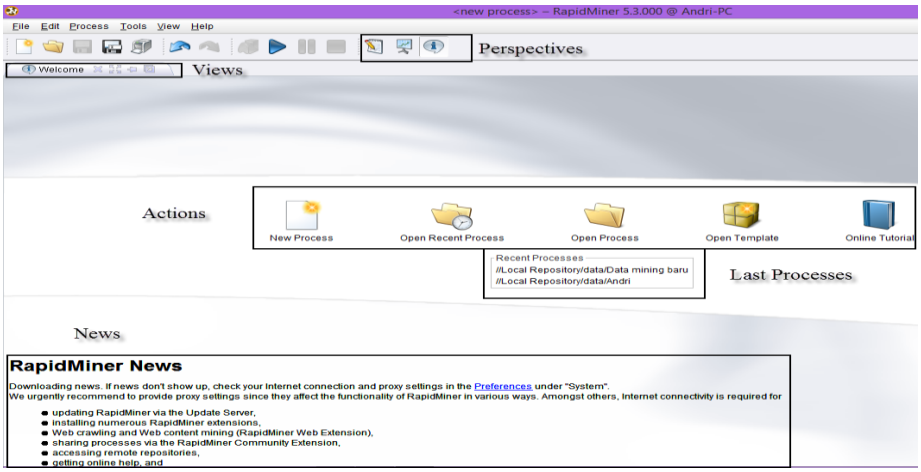
- Banyaknya algoritma data mining, seperti *decision tree* dan *self-organization map.*
- Bentuk grafis yang canggih, seperti tumbang tindih *diagram histogram, tree chart* dan 3D *scatter plots.*
- Banyaknya variasi *plugin*, seperti *text plugin* untuk melakukan analisis teks.
- Menyediakan prosedur *data mining* dan *machine learning* termasuk: ETL (*extraction, transformation, loading) data preprocessing*, visualisasi, modeling dan evalualisasi.
- Proses *data mining* tersusun atas operator-operator yang *nestable*, dideskripsikan dengan XML, dan dibuat dengan GUI.
- Mengintegrasikan proyek *data mining* Weka dan statistic R.

2.2.5.1. Pengenalan *Interface*

RapidMiner menyediakan tampilan yang user *friendly* untuk memudahkan penggunanya ketika menjalankan aplikasi. Tampilan pada *RapidMiner* dikenal dengan istilah Perspective, yaitu; *welcome perspective*, *design perspective* dan *result perspective.*[5]

a. Welcome Perspective

Ketika membuka aplikasi anda akan disambut dengan tampilan yang disebut dengan *welcome perspective*, seperti yang ditunjukan gambar 2.3. Pada bagian *toolbar*, terdapat *toolbar* ***perspective*** yang terdiri dari ikon-ikon untuk menampilkan *persepective* dari *RapidMiner*. Toolbar ini dapat dikonfigurasikan sesuai dengan kebutuhan Anda. Sedangkan ***Views*** menunjukkan pandangan (view) yang sedang Anda tampilkan



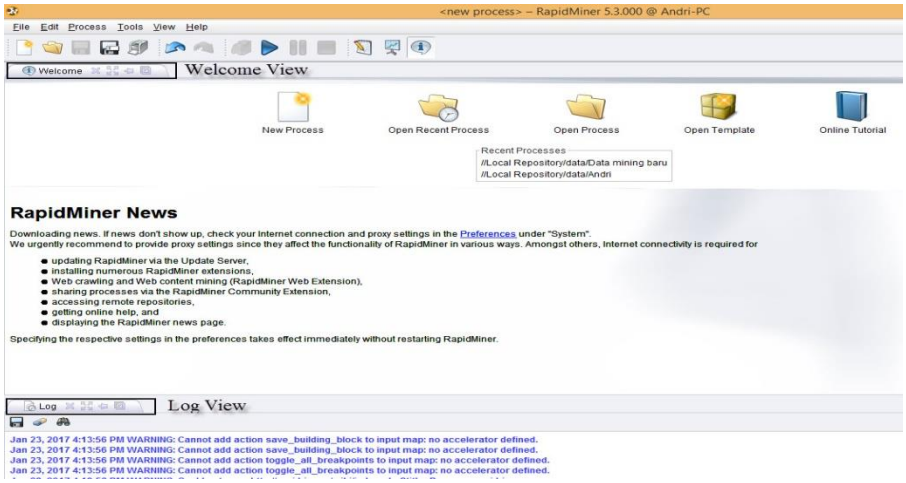
Gambar 2.3 Tampilan *welcome perpective*.

Jika komputer Anda terhubung dengan internet, maka pada bagian bawah *welcome perspective* akan menampilkan berita terbaru mengenai *RapidMiner*. Bagian ini dinamakan ***news***. Pada bagian tengah halaman terlihat daftar ***last processes*** (*Recent Processes*), bagian ini menampilkan daftar proses analisis yang baru saja dilakukan. Hal ini akan memudahkan Anda jika ingin melanjutkan proses sebelumnya yang sudah ditutup, dengan mengklik dua kali salah satu proses yang ada pada daftar tersebut. Bagian ***actions*** menunjukkan daftar aksi

yang dapat Anda lakukan setelah membuka *RapidMiner*. Berikut ini rincian lengkap daftar aksi tersebut:

- ***New*** : Aksi ini berguna untuk memulai proses analis baru. Untuk memulai proses analisis, pertama-tama Anda harus menentukan nama dan lokasi proses dan Data *repository*. Setelah itu, Anda bisa mulai merancang sebuah analisis baru.
- ***Open Recent Process***: Aksi ini berguna untuk membuka proses yang baru saja ditutup. Selain aksi ini, Anda juga bisa membuka proses yang baru ditutup dengan mengklik dua kali salah satu daftar yang ada pada *Recent Process*. Kemudian tampilan *welcome perspective* akan otomotasi beralih ke *design perspective.*
- ***Open Process*** : Aksi ini untuk membuka *repository browser* yang berisi daftar proses. Anda juga bisa memilih proses untuk dibuka pada *design perspective.*
- ***Open Template*** : Aksi ini menunjukkan pilihan lain yang sudah ditentukan oleh proses analisis.
- ***Online Tutorial*** : Aksi digunakan untuk memulai tutorial secara *online* (terhubung internet). Tutorial yang dapat secara langsung digunakan dengan *RapidMiner* ini, memberikan perkanalan dan beberapa konsep *data mining*. Hal ini direkomendasikan untuk Anda yang sudah memiliki pengetahuan dasar mengenai data mining dan sudah akrab dengan operasi dasar *RapidMiner*.

RapidMiner dapat menampilkan beberapa *view* pada saat bersamaan. Seperti yang ditunjukkan pada Gambar 2.4, pada tampilan *welcome perspective* terdapat ***welcome view*** dan ***log view***. Ukuran dari setiap *view* tersebut dapat diubah sesuai dengan kebutuhan Anda dengan mengklik dan menarik garis batas diantara keduanya ke atas atau ke bawah.



Gambar 2.4 *Welcome perspective.*

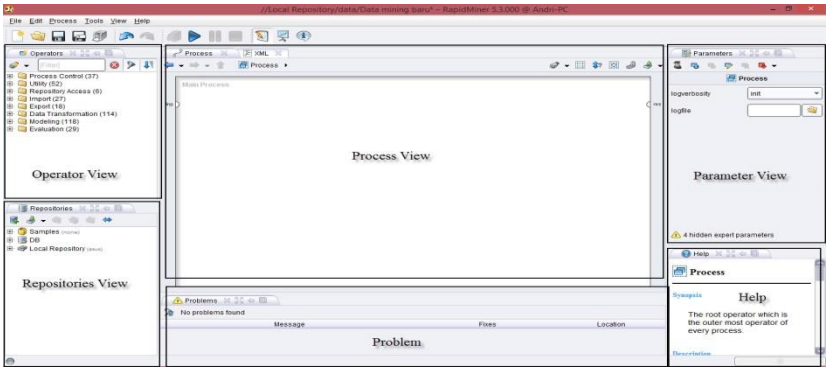
Gambar 2.5 *Header Tabm.*

Anda bisa melakukan beberapa aksi terhadap *view*, dengan mengklik salah satu ikon yang tampak pada bagian *view*, seperti yang ditunjukkan pada gambar 2.6. Berikut ini beberapa aksi yang dapat Anda lakukan:

- ***Close*** : Aksi ini untuk menutup *view* yang ditampilkan pada *perspective*. Anda bisa menampilkan *view* kembali dengan mengklik *menu view* dan memilih *view* yang ingin ditampilkan.
- ***Maximize*** : Aksi ini untuk memperbesar ukuran *view* pada perspective.
- ***Minimize*** : Aksi ini untuk memperkecil ukuran *view* pada perspective.
- ***Detach*** : Aksi ini untuk melepaskan *view* dari *perspective* menjadi jendela terpisah, kemudian Anda juga dapat memindahkannya sesuai dengan keinginan Anda.

b. Design Perspective

Design Perspective merupakan lingkungan kerja *RapidMiner*. Dimana *design perspective* ini merupakan *perspective* utama dari *RapidMiner* yang digunakan sebagai area kerja untuk membuat dan mengelola proses analisis. Seperti yang ditunjukkan pada Gambar 2.6, perspective ini memiliki beberapa *view* dengan fungsinya masing-masing yang dapat mendukung Anda dalam melakukan proses analisis data mining. Anda bisa mengganti perspective dengan mengklik salah satu ikon dari *tollbar perspective* yang sebelumnya telah dijelaskan. Selain dengan cara tersebut, Anda juga bisa mengganti *perspective* dengan mengklik menu *view*, kemudian pilih *perspective*, lalu pilih *perspective* yang ingin Anda tampilkan.



Gambar 2.6 Tampilan *Design Perspective*

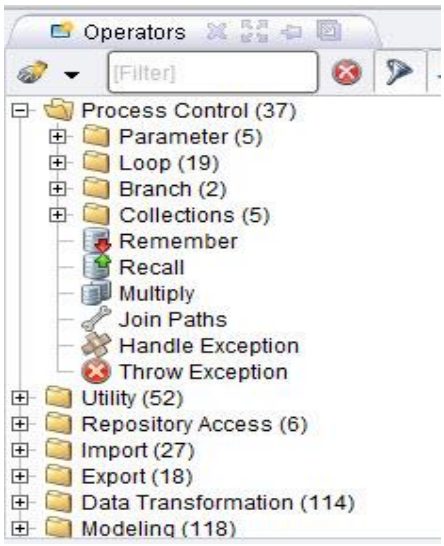
Sebagai Lingkungan kerja, *design perspective* memiliki beberapa view. Berikut ini beberapa view yang ditampilkan pada *design perspective*:

- ***Operator View***

Operator view merupakan view yang paling penting pada *perspective* ini. Semua operator atau langkah kerja dari *RapidMiner* disajikan dalam bentuk kelompok hierarki di *operator view* ini sehingga operator-operator tersebut dapat digunakan pada proses analisis, seperti yang ditunjukkan pada Gambar 2.7. Hal ini akan memudahkan Anda dalam mencari dan menggunakan operator yang sesuai dengan kebutuhan Anda. Pada *operator view* ini terdapat beberapa kelompok operator sebagai berikut:

1. *Process Control*: Operator ini terdiri dari operator perulangan dan percabangan yang dapat mengatur aliran proses.
2. *Utility*: Operator bantuan, seperti *operator macros, loggin, subproses*, dan lain-lain.
3. *Repository Access*: Kelompok ini terdiri dari operator-operator yang dapat digunakan untuk membaca atau menulis akses pada repository.

4. *Import*: Kelompok ini terdiri dari banyak operator yang dapat digunakan untuk membaca data dan objek dari format tertentu seperti file, *database*, dan lain-lain.
5. *Export*: Kelompok ini terdiri dari banyak operator yang dapat digunakan untuk menulis data dan objek menjadi format tertentu.
6. *Data Transformation*: kelompok ini terdiri dari semua operator yang berguna untuk transformasi data dan meta data.
7. *Modeling*: kolompok ini berisi proses data mining untuk menerapkan model yang dihasilkan menjadi set data yang baru.
8. *Evaluation*: kelompok ini berisi operator yang dapat digunakan untuk menghitung kualitas pemodelan dan untuk data baru.



Gambar 2.7 Kelompok operator dalam bentuk hierarki.

- ***Repository View***

Repository view merupakan komponen utama dalam *design perspective* selain *operator view*. *View* ini dapat Anda gunakan untuk mengelola dan menata proses Analisis Anda menjadi proyek dan pada saat yang sama juga dapat digunakan sebagai sumber data dan yang berkaitan dengan meta data.

- ***Process View***

Process view menunjukkan langkah-langkah tertentu dalam proses analisis dan sebagai penghubung langkah-langkah tersebut. Anda dapat menambahkan langkah baru dengan beberapa cara hubungan diantara langkah-langkah ini dapat dibuat dan dilepas kembali. Pada dasarnya bekerja dengan *RapidMiner* ialah mendefinisikan proses analisis, yaitu dengan menunjukkan serangkaian langkah kerja tertentu. Dalam *RapidMiner,* komponen proses ini dinamakan sebagai operator. Operator pada *RapidMiner* didefinisikan sebagai berikut:

1. Deskripsi dari input yang diharapkan.
2. Deskripsi dari output yang disediakan.
3. Tindakan yang dilakukan oleh operator pada *input*, yang akhirnya mengarah dengan penyediaan *output*.
4. Sejumlah parameter yang dapat mengontrol *action performed.*

- ***Parameter View***

Beberapa operator dalam *RapidMiner* membutuhkan satu atau lebih parameter agar dapat diindikasikan sebagai fungsionalitas yang benar. Namun terkadang parameter tidak mutlak dibutuhkan, meskipun eksekusi operator dapat dikendalikan dengan menunjukkan nilai parameter tertentu. *Parameter view*

memiliki *toolbar* sendiri sama seperti *view-view* yang lain. Pada Gambar 2.8, Anda dapat melihat bahwa pada *parameter view* ini terdapat beberapa ikon dan nama-nama operator terkini yang dikuti dengan aktual parameter.

Gambar 2.8 Tampilan *Parameter View*.

Huruf tebal berarti bahwa parameter mutlak harus didefinisikan oleh analis dan tidak memiliki nilai default. Sedangkan huruf miring berarti bahwa parameter diklasifikasikan sebagai parameter ahli dan seharusnya tidak harus diubah oleh pemula untuk analisis data. Poin pentingnya ialah beberapa parameter hanya ditunjukkan ketika parameter lain memiliki nilai tertentu.

- ***Help & Comment View***

Setiap kali Anda memilih operator pada *operator view* atau *process view*, maka jendela bantuan dalam *help view* akan menunjukkan penjelasan mengenai operator ini. Penjelasn yang ditampilkan dalam *help view* meliputi:

1. Sebuah penjelasan singkat mengenai fungsi operator dalam satu atau beberapa kalimat.
2. Sebuah penjelasan rinci mengenai fungsi operator.

3. Daftar semua parameter termasuk deskripsi singkat dari parameter, nilai default (jika tersedia), petunjuk apakah parameter ini adalah parameter ahli serta indikasi parameter dependensi.

Sedangkan *comment view* merupakan area bagi Anda untuk menuliskan komentar pada langkahlangkah proses tertentu. Untuk membuat komentar, Anda hanya perlu memilih operator dan menulis teks di atasnya dalam bidang komentar. Kemudian komentar tersebut disimpan bersama-sama dengan definisi proses Anda. Komentar ini dapat berguna untuk melacak langkah-langkah tertentu dalam rancangan nantinya.

- ***Problem & Log View***

Problem view merupakan komponen yang sangat berharga dan merupkan sumber bantuan bagi Anda selama merancang proses analisis. Setiap peringatan dan pesan kesalahan jelas ditunjukkan dalam *problem view*, seperti yang ditunjukkan pada Gambar 2.9

Gambar 2.9 *Problem & Log view.*

Pada kolom *Message,* Anda akan menemukan ringkasan pendek dari masalah. Kolom *location* berisi tempat di mana masalah muncul dalam bentuk nama operator dan nama port input yang bersangkutan. Kolom *fixes* memberikan gambaran dari kemungkinan solusi tersebut, baik secara langsung sebagai teks

(jika hanya ada satu kemungkinan solusi) atau sebagai indikasi dari berapa banyak kemungkinan yang berbeda untuk memecahkan masalah.

2.2.6. *Microsoft SQL Server*

SQL Server merupakan Relational Database Management System (RDMS) yang menghubungkan pengguna dengan data untuk pengelolaan basis data. *SQL Server* dapat digunakan untuk menghubungkan satu ataupun beberapa server. Bahasa basis data yang digunakan *SQL Server* adalah *Transact-SQL*. *Transact-SQL* merupakan bahasa *SQL* yang dimiliki oleh *SQL* Server yang berguna bagi pengguna untuk mendapatkan satu atau kumpulan data pada basis data dengan cara menjalankan perintah dari suatu pernyataan *SQL* [8].

2.2.7. *Microsoft Excel*

Microsoft excel adalah *software spreadsheet* paling terkenal di dunia bisnis dan perkantoran. *Excel* digunakan hampir semua bidang bisnis. *Excel* dapat dijumpai di mana-mana dan bisa dikatakan sebagai aplikasi yang *universal* dan dipakai semua orang. Aplikasi *excel* memiliki fitur kalkulasi dan pembuatan grafik, serta mudah dipakai sehingga *excel* menjadi salah satu program komputer yang populer digunakan di PC hingga saat ini. Bahkan, saat ini *excel* merupakan program *spreadsheet* paling banyak digunakan, baik *platform* PC berbasis *windows* maupun *platform macintosh* berbasis *Mac OS* semenjak versi 5.0 yang keluar di tahun 1993.