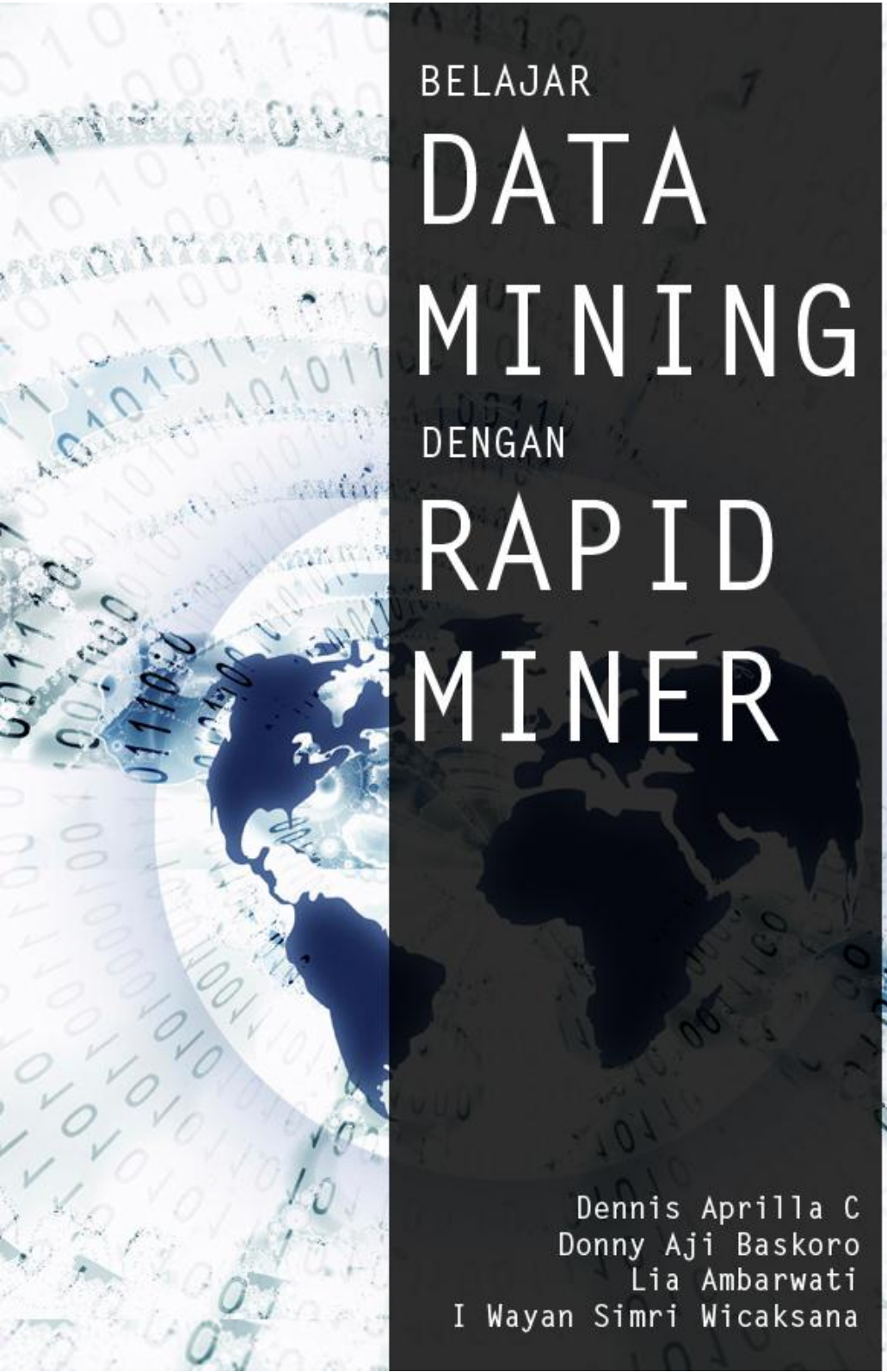
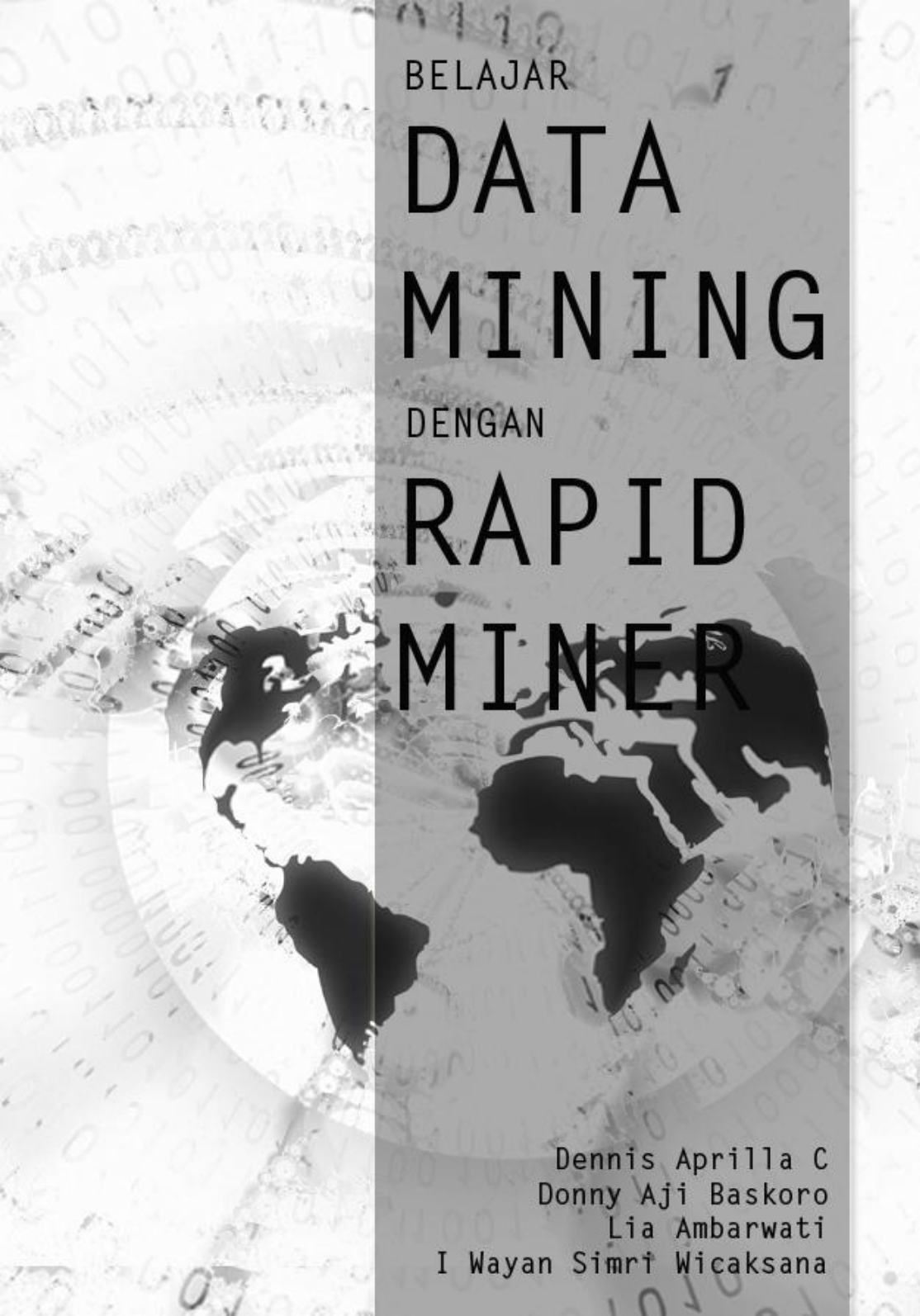


BELAJAR DATA MINING DENGAN RAPID MINER

Dennis Aprilla C
Donny Aji Baskoro
Lia Ambarwati
I Wayan Simri Wicaksana



BELAJAR DATA MINING DENGAN RAPID MINER

Dennis Aprilla C
Donny Aji Baskoro
Lia Ambarwati
I Wayan Simri Wicaksana

Identitas

Belajar Data Mining dengan RapidMiner

Penyusun:
Dennis Aprilla C
Donny Aji Baskoro
Lia Ambarwati
I Wayan Simri Wicaksana
Editor: Remi Sanjaya

Desain sampul: Dennis Aprilla C

Kata Pengantar

Dengan mengucapkan puji syukur kepada Tuhan YME atas Berkah Rahmat dan Hidayah-Nya, penulis dapat menyelesaikan buku yang berjudul Belajar Data Mining dengan RapidMiner.

Produk-produk perangkat lunak gratis (freeware) dan bersifat open source yang demikian banyak jumlahnya, telah memudahkan kita dalam melakukan proses pengolahan dan analisis data. Dalam melakukan analis terhadap data mining, RapidMiner merupakan salah satu solusi yang dapat kita gunakan. Keberadaan RapidMiner yang berupa freeware dan dapat dijalankan pada berbagai sistem operasi tidak hanya menguntungkan penyedia aplikasi karena tidak perlu mengeluarkan biaya untuk lisensi perangkat lunak, tetapi juga memudahkan pengembang maupun calon pengembang dalam mempelajari dan mencoba sendiri fitur-fitur yang ada.

Buku ini diharapkan dapat membantu pembaca mempelajari RapidMiner, melalui rangkaian tutorial bertahap mulai dari proses instalasi hingga pemrograman. Pada buku ini juga dibahas beberapa teori penunjang mengenai data mining seperti, decision tree, neural network dan market basket analysis untuk membuka wawasan pembaca mengenai data mining sebelum melakukan analisis data mining.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penyelesaian buku ini.

Akhir kata, penulis menyadari masih terdapat kekurangan dalam penyusunan buku ini baik pada teknis penulisan maupun materi, mengingat akan kemampuan yang dimiliki penulis. Untuk itu kritik dan saran dari semua pihak penulis harapkan demi penyempurnaan pembuatan buku ini. Semoga buku ini dapat bermanfaat bagi para pembaca.

Jakarta, April 2013

Penulis

Daftar Isi

Daftar Gambar

Daftar Tabel

Bagian Satu

Pendahuluan

Pengenalan Kecerdasan Buatan

Pengenalan RapidMiner

Chapter 1

Kecerdasan Buatan

Definisi Kecerdasan Buatan

Manusia memiliki kecerdasan, manusia memiliki kemampuan untuk menganalisa suatu masalah dengan menggunakan pengetahuan dalam otaknya dan

pengalaman yang pernah dilaluinya. Pengetahuan datang ketika manusia belajar, maka dari itu pembelajaran merupakan faktor penting bagi manusia untuk mencapai sebuah kecerdasan. Namun pengetahuan tidak akan cukup untuk menyelesaikan masalah jika tidak memiliki pengalaman, karena pengalaman akan selalu membawa pengetahuan baru. Tetapi akan sia sia, jika seseorang yang memiliki banyak pengalaman tetapi tidak memiliki akal untuk menalar

semua pengetahuan dan pengalaman yang ia miliki. Kombinasi dari pengetahuan, pengalaman, dan kemampuan menalar inilah yang membuat manusia menjadi cerdas dan dapat menyelesaikan permasalahan yang ia hadapi.

Berdasarkan konsep diataslah kecerdasan buatan dibuat. Agar mesin dapat bertindak seperti seorang manusia, maka mesin tersebut harus memiliki sejumlah pengetahuan dan pengalaman serta kemampuan menalar yang dapat mengubah pengetahuan dan pengalaman tersebut menjadi sebuah keputusan dalam menyelesaikan sebuah permasalahan.

Komputer awalnya diciptakan hanya untuk melakukan sebuah perhitungan saja. Jaman terus berkembang hingga akhirnya komputer kini diberdayakan manusia untuk membantu pekerjaannya dalam kesehariannya. Maka dari itu komputer diharapkan memiliki kemampuan yang hampir sama dengan manusia agar dapat mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia – Kecerdasan Buatan.

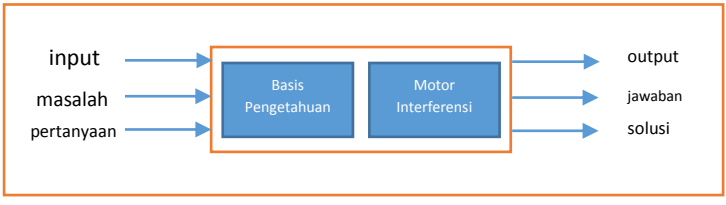
The art of creating machines that perform functions that require intelligence when performed by people (Kurzweil, 1990)

The study of how to make computers do things at which, at the moment, people are better (Rich dan Knight, 1991)

A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes (Schalkoff, 1990)

The branch of computer science that is concerned with the automation of intelligent behavior (Luger dan Stubblefield, 1993)

Jadi apakah kecerdasan buatan itu? Kecerdasan buatan adalah salah satu bagian dari ilmu komputer yang membuat agar mesin dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Dengan demikian, untuk menciptakan sebuah aplikasi kecerdasan buatan terdapat dua bagian utama yang sangat dibutuhkan.



Gambar 1.1 Proses Kecerdasan Buatan

Ruang Lingkup Kecerdasan Buatan

Kecerdasan buatan merupakan teknologi yang fleksibel, dan dapat diterapkan di berbagai macam bidang ilmu. Kemampuan kecerdasan buatan menjadi sangat dibutuhkan di bidang ilmu lain, karena konsepnya tak lagi procedural melainkan meniru cara berpikir manusia. Tak heran kecerdasan buatan bisa di gunakan untuk bidang psikologi yang dikenal dengan cognition dan psycolinguistic. Namun yang paling sering dekat dengan kita ialah robotika, yakni kecerdasan buatan di dalam ilmu elektornika.

Semakin banyaknya ilmu yang menggunakan kecerdasan buatan, semakin sulit juga bagi manusia untuk mengkategorikannya, maka dari itu dibentuklah ruang lingkup kecerdasan buatan yang dapat mewakilinya (Turban dan Frenzel, 1992, pp21-26):

1. Sistem Pakar. komputer digunakan untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

2. Pengolahan Basa Alami. dengan pengolahan bahasa alami ini diharapkan user dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan. Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika dan Sistem Sensor
5. Computer Vision. Mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. Intelligent Computer-aided Instruction. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. Game Playing.

Perbedaan Komputasi Kecerdasan Buatan dan Komputasi Konvensional

Komputasi Konvensional merupakan Komputer yang hanya digunakan untuk alat hitung. Sangatlah berbeda, kerja dan konsep dari kedua komputasi ini. Agar dapat memberikan gambaran, table berikut adalah

detail dari perbedaan komputasi kecerdasan buatan dan komputasi konvensional.

Tabel 1.1 Perbedaan Kecerdasan Buatan dan Komputasi Konvensional

Dimensi	Komputasi Kecerdasan Buatan	Komputasi Konvensional
Pemrosesan	Mengandung konsep-konsep simbolik	Algoritmik
Sifat Input	Bisa tidak lengkap	Harus lengkap
Pencarian	Kebanyakan bersifat heuristic	Biasanya didasarkan pada algoritma
Keterangan	Disediakan	Biasanya tidak disediakan
Fokus	Pengetahuan	Data dan Informasi
Struktur	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi
Kemampuan menalar	Ya	Tidak

Chapter 2

RapidMiner

Apa itu 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.

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.

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

RapidMiner memiliki beberapa sifat sebagai berikut:

- Ditulis dengan bahasa pemrograman 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 eksperimen skala besar dan otomatisasi eksperimen.
- Konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penanganan data.
- Memiliki GUI, command line mode, dan Java API yang dapat dipanggil dari program lain.

Beberapa Fitur dari RapidMiner, antara lain:

- Banyaknya algoritma data mining, seperti decision treee dan self-organization map.
- Bentuk grafis yang canggih, seperti tumpang 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, modelling dan evaluasi
- Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI
- Mengintegrasikan proyek data mining Weka dan statistika R

Instalasi Software

System Requirement

Sebelum melakukan instalasi software RapidMiner, terdapat beberapa spesifikasi minimal yang yang harus dimiliki komputer pengguna. Spesifikasi minimal bergantung pada komputer dan sistem operasi yang akan diinstal. Berikut ini beberapa spesifikasi minimal yang dibutuhkan software RapidMiner:

1. *Sistem Operasi*

   RapidMiner merupakan software yang multiplatform, sehingga software ini dapat dijalankan pada berbagai sistem operasi. Berikut ini beberapa jenis sistem operasi yang dapat diinstal RapidMiner:

- ✓ Microsoft Windows (x86-32) → Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7
- ✓ Microsoft Windows (x64) → Windows XP untuk x64, Windows Server 2003 untuk x64, Windows Vista untuk x64, Windows Server 2008 untuk x64, Windows 7 untuk x64
- ✓ Unix sistem 32 atau 64 bit
- ✓ Linux sistem 32 atau 64 bit
- ✓ Apple Macintosh sistem 32 atau 64 bit

  Sebagai bahan pertimbangan, kami merekomendasikan untuk penggunaan sistem 64 bit. Hal ini dikarenakan jumlah maksimum yang dapat digunakan oleh RapidMiner terbatas pada sistem operasi dengan sistem 32, yaitu hanya sebasar 2GB.

2. *Java Runtime Environment versi 6*

Selain itu, penggunaan server RapidAnalytics dalam kombinasi dengan RapidMiner dapat memaksimalkan proses analisis pada RapidMiner, meskipun tugas analisis sudah banyak dapat dijalankan dengan RapidMiner desktop client. Dalam hal ini proses analisa dirancang dengan RapidMiner, kemudian dieksekusi oleh server RapidAnalytics.

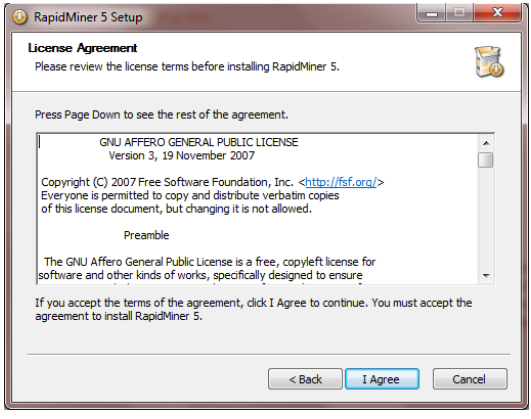
Instalasi RapidMiner

Seperti yang yang telah dikemukakan sebelumnya bahwa RapidMiner merupakan software gratis yang bersifat terbuka (open source). Software ini dapat dijalankan pada sistem operai Windows, Linux, maupun Mac. RapidMiner dapat diunduh pada situs resminya, yaitu www.rapid-i.com. Pada bagian ini, akan dijelaskan bagaimana cara melakukan instalasi software RapidMiner versi 5.3 pada sistem operasi Microsoft Windows.

Untuk memulai instalasi software RapidMiner pada sistem operasi Microsoft Windows, jalankan file installer RapidMiner-5.3.000x32-install.exe, sehingga akan muncul tampilan wizard seperti pada Gambar 2.

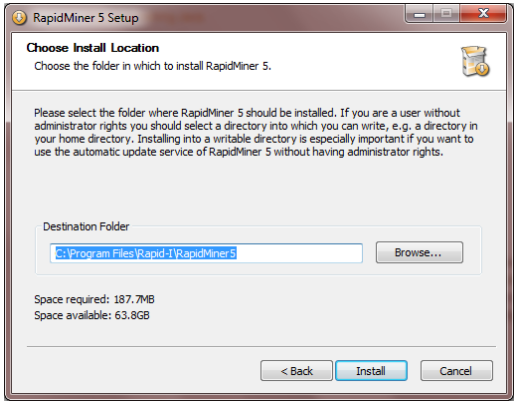
Gambar 2.1 Form Awal Instalasi

Klik **Next >** untuk melanjutkan pada form persetujuan dan lisensi seperti pada Gambar 2.3



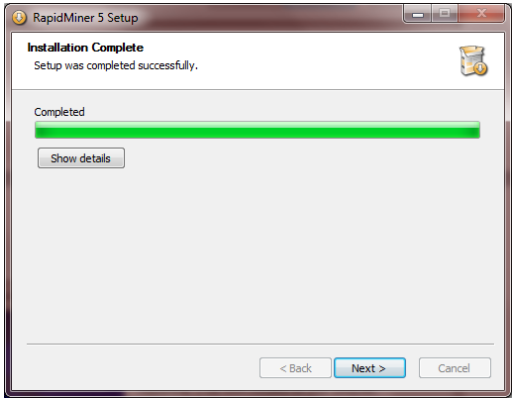
Gambar 2.2 Form Persetujuan Lisensi

Pilih **I Agree** untuk melanjutkan. Kemudian, wizard akan menampilkan form seperti pada gambar 2.4.



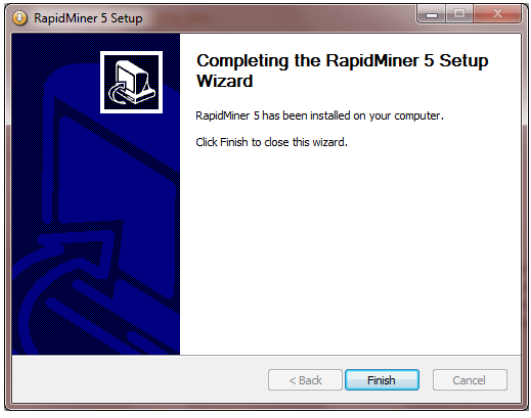
Gambar 2.3 Form Pemilihan Lokasi Instalasi

Pilih **Install** untuk melakukan proses instalasi. Kemudian wizard akan menampilkan progress dari proses tersebut, seperti yang ditunjukkan pada Gambar 2.5.



Gambar 2.4 Form Proses Instalasi

Setelah proses selesai, pilih **Next >** untuk melanjutkan**,** maka wizard akan menampilkan informasi bahwa proses instalasi telah selesai dilakukan, seperti yang ditunjukkan pada Gambar 2.6.



Gambar 2.5 Form Instalasi selesai

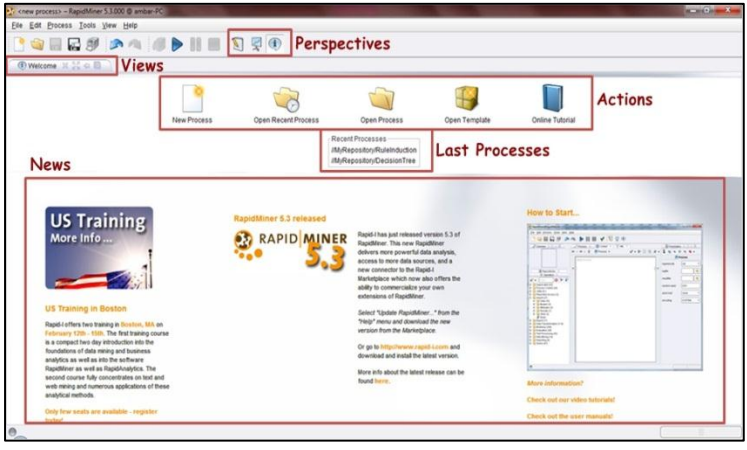
Pilih **Finish** untuk mengakhiri proses instalasi.

Pengenalan Interface

RapidMiner menyediakan tampilan yang *user friendly* untuk memudahkan penggunanya ketika menjalankan aplikasi. Tampilan pada RapidMiner dikenal dengan istilah Perspective. Pada RapidMiner terdapat 3 Perspective, yaitu; Welcome Perspective, Design Perspective dan Result Perspective.

Welcome Perspective

Ketika membuka aplikasi Anda akan disambut dengan tampilan yang disebut dengan Welcome Perspective, seperti yang ditunjukkan pada Gambar 6. Pada bagian toolbar, terdapat toolbar **Perspectives** yang terdiri dari ikon-ikon untuk menampilkan perspective dari RapidMiner. Toolbar ini dapat dikonfigurasikan sesuai dengan kebutuhan Anda. Sedangkan **Views** menunjukkan pandangan (view) yang sedang Anda tampilkan.



Gambar 2.6 Tampilan Welcome Perspective

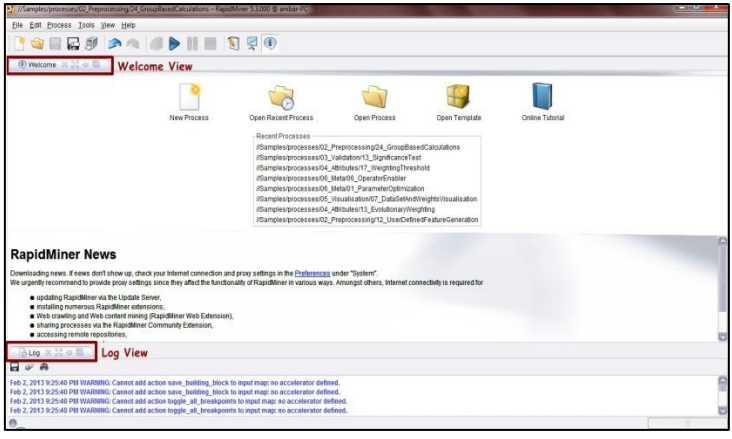
Jika komputer Anda terhubung dengan internet, maka pada bagian bawah Welcome Perspective akan menampilkan berita terbaru mengenai RapidMiner. Bagian ini dinamakan **News**. Pada bagian tengah halaman terlihat daftar **Last Processes** (Recent

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 RapidMine. Berikut ini rincian lengkap daftar aksi tersebut:

1. ***New*** : Aksi ini berguna ntuk 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.
2. ***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.
3. ***Open Process*** : Aksi ini untuk membuka Repository Browser yang berisi daftar proses. Anda juga bisa memilih proses untuk dibuka pada Design Perspective.
4. ***Open Template*** : Aksi ini menunjukkan pilihan lain yang sudah ditentukan oleh proses analisis.

5. ***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 7, pada tampilan Welcome Perspective tedapat **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 2.7 Welcome Perspective

Gambar 2.8 Header Tab

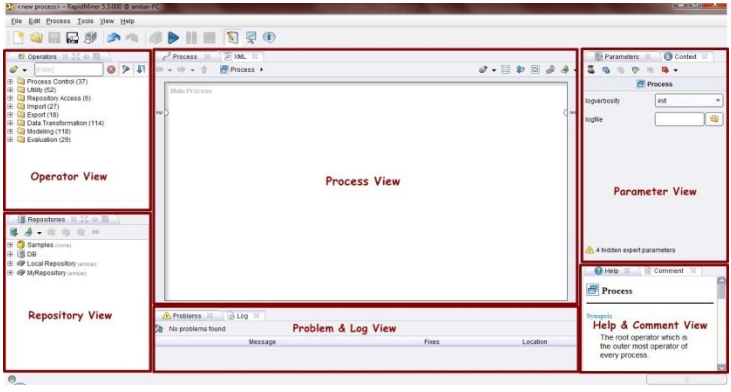
Anda bisa melakukan beberapa aksi terhadap view, dengan mengklik salah satu ikon yang tampak pada bagian view, seperti yang ditunjukkan pada gambar 2.8. Berikut ini beberapa aksi yang dapat Anda lakukan:

1. ***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.
2. ***Maximize*** : Aksi ini untuk memperbesar ukuran view pada perspective.
3. ***Minimize*** : Aksi ini untuk memperkecil ukuran view pada perspective.
4. ***Detach*** : Aksi ini untuk melepaskan view dari perspective menjadi jendela terpisah, kemudian Anda juga dapat memindahkannya sesuai dengan keinginan Anda.

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 2.10, 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 2.9 Tampilan Design Perspective

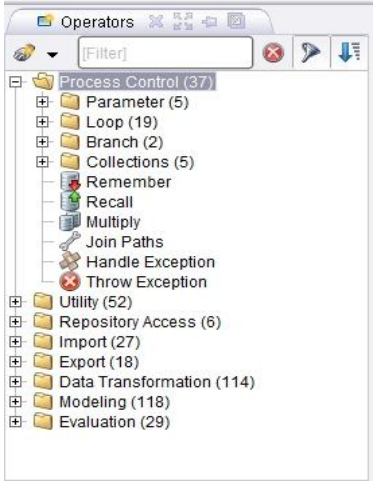
Sebagai lingkungan lingkungan kerja, Design Perspective memiliki beberapa view. Berikut ini beberapa view yang ditampilkan pada Design Perspective:

1. ***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 2.10. 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:

- *Process Control* : Operator ini terdiri dari operator perulangan dan percabangan yang dapat mengatur aliran proses.
- *Utility* : Operator bantuan, seperti operator macros, loggin, subproses, dan lain-lain.
- Repository Access : Kelompok ini terdiri dari operator-operator yang dapat digunakan untuk membaca atau menulis akses pada repository.
- *Import* : Kelompok ini terdiri dari banyak operator yang dapat digunakan untuk membaca data dan objek dari format tertentu seperti file, database, dan lain-lain.
- *Export* : Kelompok ini terdiri dari banyak operator yang dapat digunakan untuk menulis data dan objek menjadi format tertentu.

- *Data Transformation* : kelompok ini terdiri dari semua operator yang berguna untuk transformasi data dan meta data.
- *Modeling* : kolompok ini berisi proses data mining untuk menerapkan model yang dihasilkan menjadi set data yang baru.
- *Evaluation* : kelompok ini berisi operator yang dapat digunakan untuk menghitung kualitas pemodelan dan untuk data baru.



Gambar 2.10 Kelompok Operator dalam Bentuk Hierarki

2. *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.

3. *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 beikut:

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

4. *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 2.12, Anda dapat melihat bahwa pada Parameter View ini terdapat beberapa ikon dan nama-nama operator terkini yang dikuti dengan aktual parameter.

Gambar 2.11 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.

5. *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:

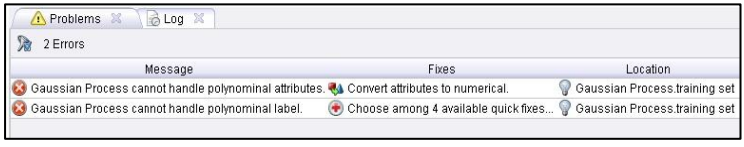
- Sebuah penjelasan singkat mengenai fungsi operator dalam satu atau beberapa kalimat.
- Sebuah penjelasan rinci mengenai fungsi operator.
- Daftar semua parameter termasuk deskripsi singkat dari parameter, nilai default (jika tersedia), petunjuk apakah parameter ini adalah parameter ahli serta indikasi parameter dependensi.

Sedangkan Comment View merupakan area bagi Anda untuk menuliskan komentar pada langkah-langkah 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.

6. *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 2.13.



Gambar 2.12 Problem & Log View

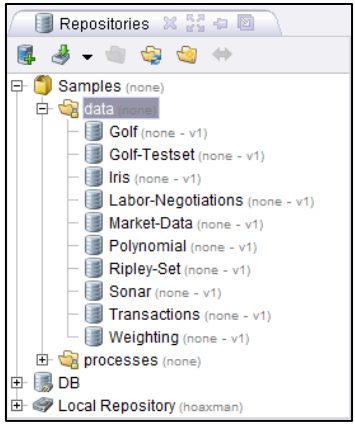
Pada kolom Message, Anda akan menemukan ringkasan pendek dari masalah. Kolom Location berisi tempat di mana masalah muncul dalam bentuk nama Operator dan nama port input yang bersangkutan. Kolom Fixes memberikan gambaran dari kemungkinan solusi tersebut, baik secara langsung sebagai teks (jika hanya ada satu kemungkinan Solusi) atau sebagai indikasi dari berapa banyak kemungkinan yang berbeda untuk memecahkan masalah.

Cara Menggunakan Repositori

Repositori merupakan Tabel, database, koleksi teks, yang kita miliki untuk dapat digali datanya untuk mendapatkan informasi yang kita inginkan. Ini merupakan awal dari seluruh proses Data Mining. Maka dari itu adalah penting bagi kita untuk mengetahui cara menggunakan repository.

Sample Data Repository

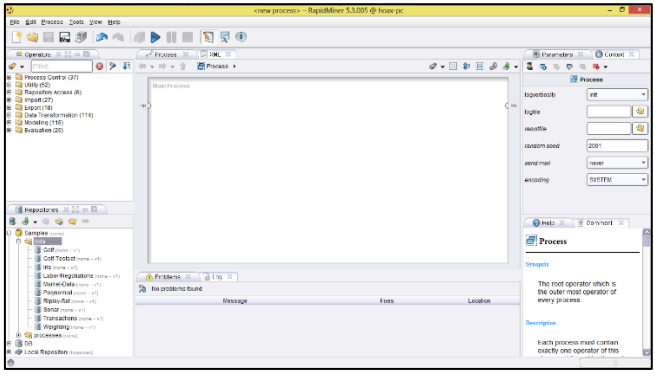
RapidMiner menyediakan contoh database yang dapat digunakan, berikut cara menggunakan Sample Data Repository.



Gambar 2.13 Kumpulan Sample Data Repository

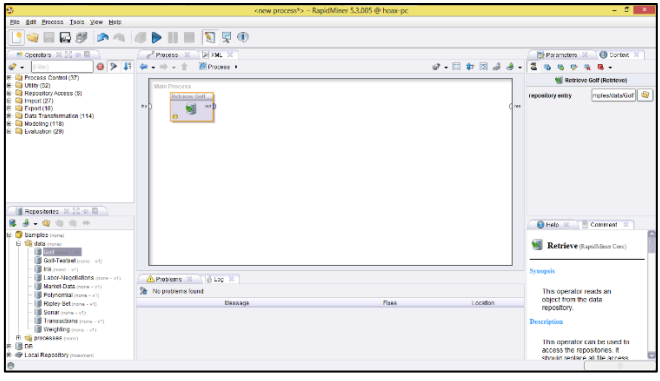
Pada bagian Repositori terdapat 3 buah lokasi repositori, yakni Samples, DB dan Local Repository.

Untuk mengambil Sample Data Repository, buka hirarki Samples, masuk ke folder Data. Sehingga seperti gambar berikut.

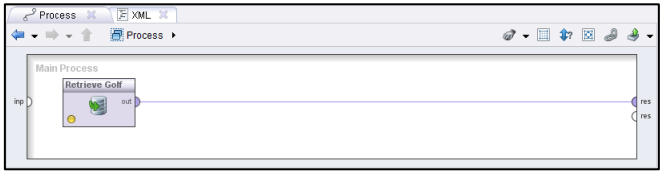


Gambar 2.14 Tampilan Design Perspective Awal

Lakukan Drag dan Drop salah satu Example Repository. Kita ambil contoh Golf. Tarik dan lepaskan repository ke dalam Main Process, sehingga seperti gambar berikut.



Gambar 2.15 Repository berada dalam Main Process



Gambar 2.16 Menghubungkan Output Repositori ke Result

Hubungkan output pada Database ke Result seperti Gambar diatas. Lalu klik ikon Play  . Gambar 2.17 adalah Sample data repository dari Golf. Coba lakukan untuk memasukkan Sample Repository yang lain.

Result Overview | ExampleSet (Retrieve Golf)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (14 examples, 1 special attribute, 4 regular attributes)

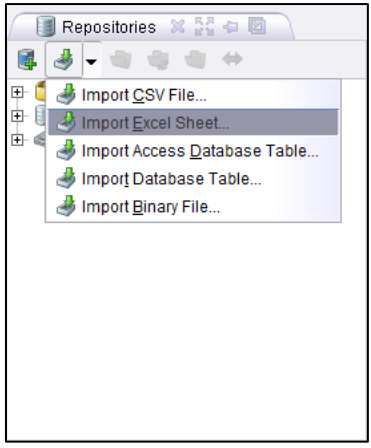
Row No.	Play	Outlook	Temperature	Humidity	Wind
1	no	sunny	85	85	false
2	no	sunny	80	90	true
3	yes	overcast	83	78	false
4	yes	rain	70	96	false
5	yes	rain	68	80	false
6	no	rain	65	70	true
7	yes	overcast	64	65	true
8	no	sunny	72	95	false
9	yes	sunny	69	70	false
10	yes	rain	75	80	false
11	yes	sunny	75	70	true
12	yes	overcast	72	90	true
13	yes	overcast	81	75	false
14	no	rain	71	80	true

Gambar 2.17 Isi Sample Golf Data Repository

Import Repository

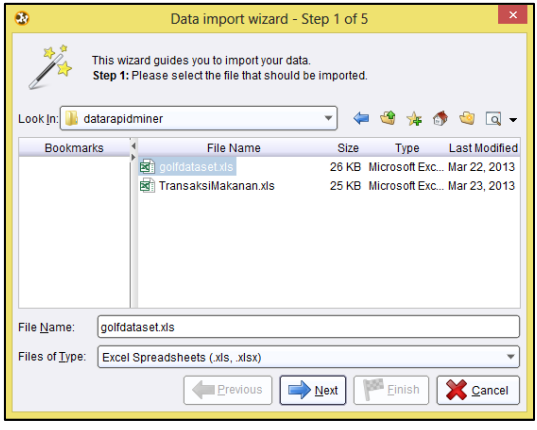
Dibanyak kesempatan lain, kita akan selalu menggunakan database yang kita miliki. RapidMiner menyediakan layanan agar pengguna dapat mengimport database miliknya. Namun, tidak seperti kebanyakan tools Data Mining Lain, RapidMiner memiliki kelebihan tersendiri yakni dapat langsung melakukan import file dengan ekstensi .xls atau .xlsx, yakni file dari Microsoft Excel, Program yang relatif sering digunakan oleh pengguna. Berikut adalah cara untuk melakukan import file Microsoft Excel.

Lihat pada bagian Repository. Klik pada ikon import seperti gambar 2.18. Seperti yang dapat kita lihat, ada beberapa ekstensi file yang dapat kita masukkan kedalam repository kita. CSV File, Excel Sheen File, Access Database Table File, Database Table, Binary File. Namun pada Dasarnya cara melakukan import pada semua file ini sama. Sebagai contoh, pilih Import Excel Sheet.



Gambar 2.18 Repository

Setelah itu, akan muncul window baru yakni Step 1 dari 5 Step Data import Wizard. Disini akan diarahkan oleh RapidMiner bagaimana langkah untuk melakukan import data.



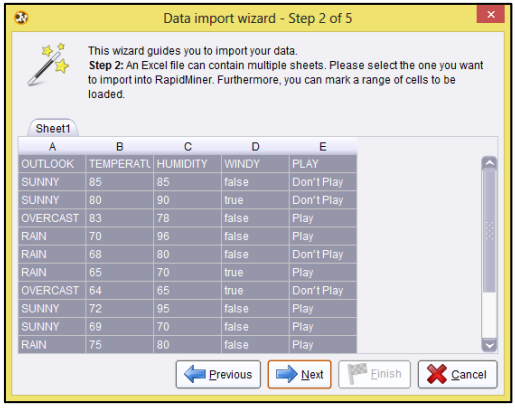
Gambar 2.19 Step 1 of 5 Import Wizard

Cari file excel kalian dengan klik pada bagian Look in 

. Setelah menemukan file yang dibutuhkan lalu Klik tombol Next 

.

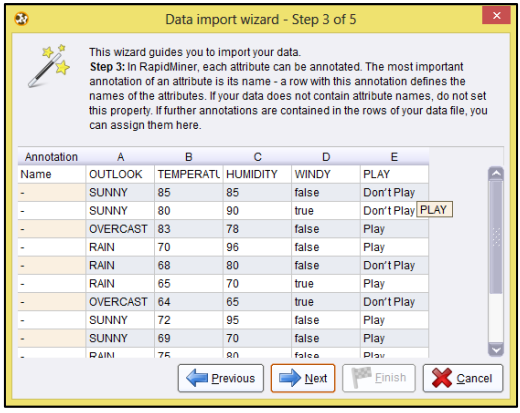
Berikutnya pada Step 2 ialah, pilih Sheet yang akan dimasukkan. Pada dasarnya, Repository RapidMiner hanya menyediakan 1 repositori untuk 1 buah table.



Gambar 2.20 Step 2 of 5 Import Wizard

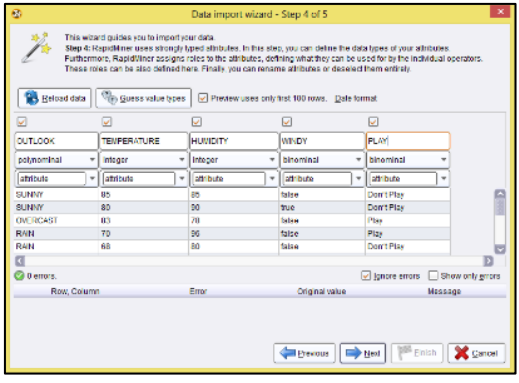
Klik tombol Next 

. Berikutnya ialah memberikan anotasi. Jika data kita tidak memiliki nama attribute, tidak usah melakukan apa-apa pada step 3 ini.



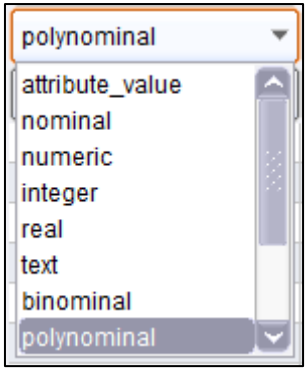
Gambar 2.21 Step 3 of 5 Import Wizard

Klik tombol Next  . Step ke 4 adalah memberikan tipe data pada tabel kita. Sebenarnya RapidMiner akan memberikan tipe data yang tepat secara otomatis.



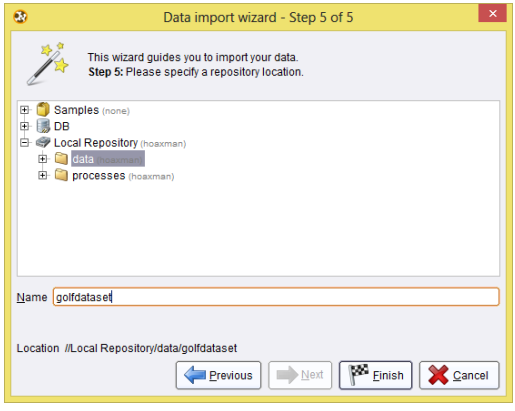
Gambar 2.22 Step 4 of 5 Import Wizard

Namun, jika kita merasa tipe data yang diberikan RapidMiner tidak cocok, kita bisa mengubahnya.



Gambar 2.23 Tipe Data

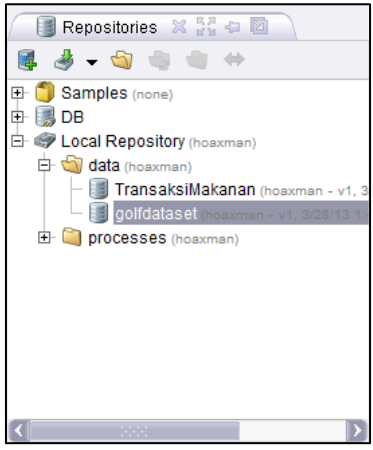
Klik tombol Next  . Step ke 5 adalah memasukkan database kita kedalam repository. Disarankan untuk memasukkannya kedalam Local Repository untuk memudahkan kita mencarinya. Jangan lupa untuk memberikan nama repository kita.



Gambar 2.24 Step 5 of 5 Import Wizard

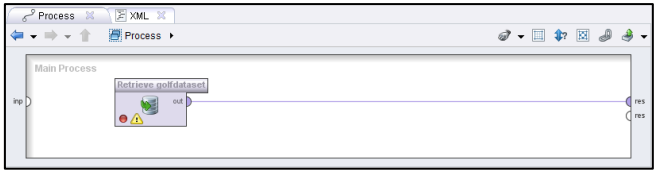
Kemudian klik tombol finish  .

Hasil Import Repository akan terlihat pada bagian Repository seperti dalam gambar 2.25.



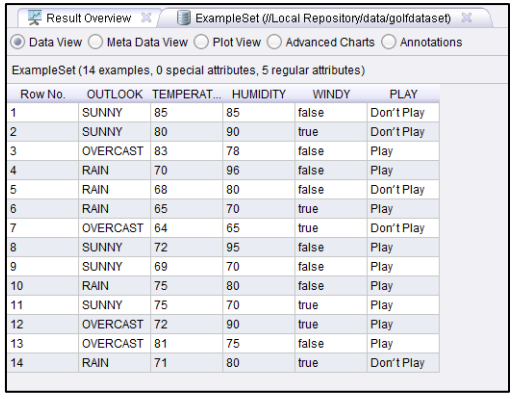
Gambar 2.25 Repository yang sudah diimport

Untuk melihat isi dari repository kita, hubungkan output pada repository kearah result seperti gambar 2.26.



Gambar 2.26 Menghubungkan Output Repositori pada Result

klik ikon Play . Dan berikutnya akan muncul isi dari tabel yang kalian miliki.


Result Overview | ExampleSet (//Local Repository/data/golfdataset)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (14 examples, 0 special attributes, 5 regular attributes)

Row No.	OUTLOOK	TEMPERAT...	HUMIDITY	WINDY	PLAY
1	SUNNY	85	85	false	Don't Play
2	SUNNY	80	90	true	Don't Play
3	OVERCAST	83	78	false	Play
4	RAIN	70	96	false	Play
5	RAIN	68	80	false	Don't Play
6	RAIN	65	70	true	Play
7	OVERCAST	64	65	true	Don't Play
8	SUNNY	72	95	false	Play
9	SUNNY	69	70	false	Play
10	RAIN	75	80	false	Play
11	SUNNY	75	70	true	Play
12	OVERCAST	72	90	true	Play
13	OVERCAST	81	75	false	Play
14	RAIN	71	80	true	Don't Play

Gambar 2.27 Tabel Repository

Bagian Dua

Data Mining

Pengenalan Data Mining

Pengenalan Decision Tree

Pengenalan Neural Network

Pengenalan Market Basket Analysis

Chapter 3

Data Mining

Mengenal Data Mining

Pengertian Data Mining

Sebelum kita mulai, ayo kita coba beberapa eksperimen sebagai berikut.

- Pilih angka antara 1 sampai 10
- Kalikan dengan angka 9
- Hasil dari perkalian tersebut jumlahkan masing-masing angkanya
- Kalikan hasil dengan 4
- Bagi dengan 3
- Kurangi dengan 2

Jawabannya adalah 2. Kebetulan? Sebagai seorang analis, pasti jawabannya adalah tidak.

Bagaimana dengan kejadian acak lainnya, seperti “lempar koin.” Tentu jika temanmu menebak secara langsung dan hasil dari kejadian tersebut ternyata tepat seperti yang temanmu tebak, kau pasti akan mengatakan bahwa itu merupakan kebetulan.

Kita ambil satu contoh sederhana lagi. Terdapat kejadian seperti: Seseorang menjatuhkan sebuah gelas dari ketinggian tertentu. Detik pertama orang tersebut menjatuhkan gelasnya, kau pasti akan mengatakan dengan pasti bahwa gelas tersebut akan pecah, padahal hukum fisika belum menunjukkan proses penghancuran gelas tersebut ketika bersentuhan dengan tanah. Dan lagi, tebakanmu itu dikatakan bukanlah kebetulan. Jadi secara logika, bagaimana kau tahu dengan sangat tepat hasil dari kejadian tersebut? Bukankah kondisinya sama seperti kejadian “lempar koin” sebelumnya?

Jadi apakah yang kita lakukan dalam otak kita? Kita mempertimbangkan karakteristik-karakteristik dari kejadian ini. Pada kasus gelas yang jatuh, kita dengan cepat mengetahui karakteristik penting dari serangkaian kejadian tersebut, bahan gelas, ketinggian, tipe pijakan, dan lain-lain. Kemudian kita menjawab dengan cepat berdasarkan analogi, contohnya kita kita

membuat perbandingan dengan kejadian gelas atau cangkir atau piring yang jatuh sebelumnya. Berarti dua hal yang diperlukan adalah: pertama, kita membutuhkan data dari kejadian-kejadian sebelumnya, dan kedua, seberapa mirip kejadian yang di tempat dengan kejadian sebelumnya. Kita bisa membuat estimasi atau prediksi dengan mencari kejadian yang paling mirip dengan kejadian di tempat. Karena kita lebih sering melihat bahwa benda berbahan kaca dijatuhkan akan pecah, maka secara otomatis inilah yang menjadi prediksi kita.

Bagaimanapun, prosedur diatas tidak cocok untuk kejadian “lempar koin.” Ini disebabkan terdapat lebih banyak faktor yang harus dipertimbangkan, ada yang sulit dan ada yang tidak bisa diukur. Belum lagi kita harus dapat memikirkan proses kejadian menuju hasil dengan baik, memikirkan analogi yang paling cocok dengan kejadian untuk melakukan prediksi. Ditambah “lempar koin” memiliki kondisi yang dapat berubah-ubah tiap kejadiannya dan berlangsung cepat, ini berarti perhitungan juga harus dilakukan secara cepat. Mustahil untuk seorang manusia? Benar. Tetapi tidak mustahil untuk metode data mining.

Data Mining adalah serangkaian proses untuk menggali nlai tambah dari suatu kumpulan data

berupa pengetahuan yang selama ini tidak diketahui secara manual. (Pramudiono, 2006)

Data Mining adalah analisis otomatis dari data yang berjumlah besar atau kompleks dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaanya. (Pramudiono, 2006)

Data Mining merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data. (Larose, 2005)

Data Mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistic, database, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar. (Larose, 2005)

Kata *Mining* merupakan kiasan dari bahasa inggris, mine. Jika mine berarti menambang sumber daya yang tersembunyi di dalam tanah, maka Data Mining merupakan penggalian makna yang

tersembunyi dari kumpulan data yang sangat besar. Karena itu *Data Mining s*ebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), m*achine learning*, statistik dan basis Data.

Pengelompokan Teknik Data Mining

Data Mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

Classification

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Salah satu contoh yang mudah dan popular adalah dengan Decision tree yaitu salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi. Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki.

Association

Digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana hubungan asosiasi muncul pada setiap kejadian. Salah satu contohnya adalah Market Basket Analysis, yaitu salah sati metode asosiasi yang menganalisa kemungkinan pelanggan untuk membeli beberapa item secara bersamaan.

Clustering

Digunakan untuk menganalisis pengelompokkan berbeda terhadap data, mirip dengan klasifikasi, namun pengelompokkan belum didefinisikan sebelum dijalankannya tool data mining. Biasanya menggunkan metode *neural network* atau statistik. Clustering membagi item menjadi kelompok-kelompok berdasarkan yang ditemukan tool data mining.

Chapter 4

Decision Tree

Mengenal Decision Tree

Seperti diketahui bahwa manusia selalu menghadapi berbagai macam masalah di dalam kehidupannya sehari-hari. Masalah-masalah yang timbul dari berbagai macam bidang ini memiliki tingkat kesulitan dan kompleksitas yang sangat bervariasi, mulai dari masalah yang sangat sederhana dengan sedikit faktor-faktor terkait hingga masalah yang sangat rumit dengan banyak sekali faktor-faktor yang terkait, sehingga factor-faktor yang berkaitan dengan masalah tersebut perlu untuk diperhitungkan.

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.

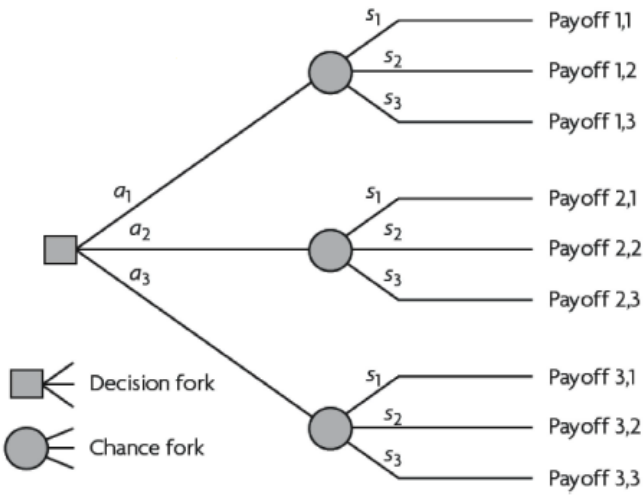
Dengan pohon keputusan, manusia dapat dengan mudah mengidentifikasi dan melihat hubungan antara faktor-faktor yang mempengaruhi suatu masalah sehingga dengan memperhitungkan faktor-faktor tersebut dapat dihasilkan penyelesaian terbaik untuk masalah tersebut. Pohon keputusan ini juga dapat menganalisa nilai resiko dan nilai suatu informasi yang terdapat dalam suatu alternatif pemecahan masalah.

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.

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. (Han, J., & Kamber, M. (2006). Data Mining Concept and Tehniques. San Fransisco: Morgan Kauffman.)



Gambar 4.1 Bentuk Decision Tree Secara Umum

Algoritma c4.5

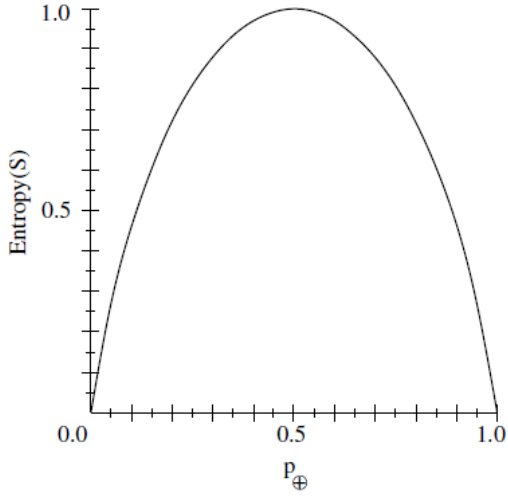
Pohon keputusan merupakan metode yang umum digunakan untuk melakukan klasifikasi pada data mining. Seperti yang telah dijelaskan sebelumnya, klasifikasi merupakan Suatu teknik menemukan kumpulan pola atau fungsi yang mendeskripsikan serta memisahkan kelas data yang satu dengan yang lainnya untuk menyatakan objek tersebut masuk pada kategori tertentu dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan.

Metode ini popular karena mampu melakukan klasifikasi sekaligus menunjukkan hubungan antar atribut. Banyak algoritma yang dapat digunakan untuk membangun suatu decision tree, salah satunya ialah algoritma C45.

Algoritma C4.5 dapat menangani data numerik dan diskret. Algoritma C.45 menggunakan rasio perolehan (gain ratio). Sebelum menghitung rasio perolehan, perlu dilakukan perhitungan nilai informasi dalam satuan bits dari suatu kumpulan objek, yaitu dengan menggunakan konsep entropi.

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. Entropi digunakan untuk mengukur ketidakaslian S.



Gambar 4.2 Grafik Entropi

Besarnya Entropy pada ruang sampel S didefinisikan dengan:

$$\boldsymbol{Entropy(S) \equiv -p_{\oplus}\, log_2\, p_{\oplus} - p_{\ominus}\, log_2 p_{\ominus}}$$

Dimana:

- S : ruang (data) sampel yang digunakan untuk pelatihan
- $p_{\oplus}$: jumlah yang bersolusi positif atau mendukung pada data sampel untuk kriteria tertentu
- $p_{\ominus}$: jumlah yang bersolusi negatif atau tidak mendukung pada data sampel untuk kriteria tertentu.

- Entropi(S) = 0, jika semua contoh pada S berada dalam kelas yang sama.
- Entropi(S) = 1, jika jumlah contoh positif dan negative dalam S adalah sama.
- 0 > Entropi(S) > 1, jika jumlah contoh positif dan negative dalam S tidak sama.

Konsep Gain

Gain (S,A) merupakan Perolehan informasi dari atribut A relative terhadap output data S. Perolehan informasi didapat dari output data atau variabel dependent S yang dikelompokkan berdasarkan atribut A, dinotasikan dengan gain (S,A).

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

Dimana:

- A : Atribut
- S : Sampel
- n : Jumlah partisis himpunan atribut A
- $|S_i|$: Jumlah sampel pada pertisi ke –i
- $|S|$: Jumlah sampel dalam S

Untuk memudahkan penjelasan mengenai algoritma C4.5berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 4.1:

Tabel 4.1 Keputusan Bermain Tenis

No	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

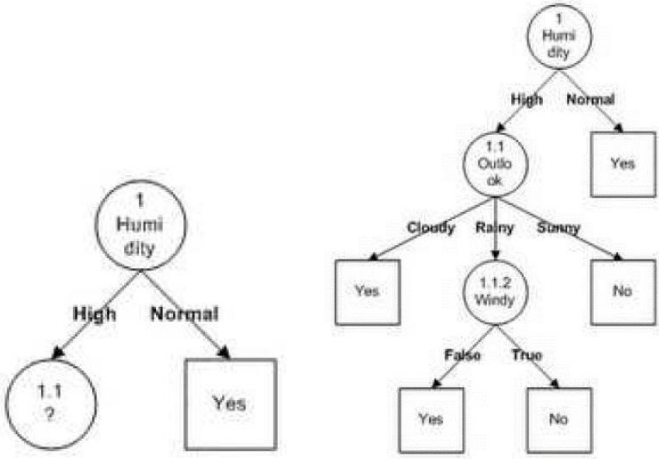
Tabel 1 merupakan kasus yang akan dibuat pohon keputusan untuk menentukan main tenis atau tida. Data ini memiliki atribut-atribut yaitu, keadaan cuaca (outlook), temperatur, kelembaban (humidity) dan keadaan angin (windy).

Berikut merupakan cara membangun pohon keputusan dengan menggunakan algoritma:

1. Pilih atribut sebagai akar. Sebuah akar didapat dari nilai gain tertinggi dari atribut-atribut yang ada.
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Tabel 4.2 Perhitungan Simpul 1

NODE			JUMLAH KASUS	NO (S_1)	YES (S_2)	ENTROPY	GAIN
1	TOTAL		14	4	10	0.863120569	
	OUTLOOK						0.258521037
		CLOUDY	4	0	4	0	
		RAINY	5	1	4	0.721928095	
		SUNNY	5	3	2	0.970950594	
	TEMPERATURE						0.183850925
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0.918295834	
	HUMIDITY						0.370506501
		HIGH	7	4	3	0.985228136	
		NORMAL	7	0	7	0	
	WINDY						0.005977711
		FALSE	8	2	6	0.811278124	
		TRUE	6	4	2	0.918295834	



Dari hasil pada Tabel 4.2 dapat diketahui bahwa atribut dengan Gain tertinggi adalah HUMIDITY yaitu sebesar 0.37. Dengan demikian HUMIDITY dapat menjadi node akar.

Ada 2 nilai atribut dari HUMIDITY yaitu HIGH dan NORMAL. Dari kedua nilai atribut tersebut, nilai atribut NORMAL sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya Yes, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut HIGH masih perlu dilakukan perhitungan lagi hingga semua kasus masuk dalam kelas seperti yang terlihat pada Gambar di sebelah kanan.

Kelebihan Pohon Keputusan

Dalam membuat keputusan dengan menggunankan pohon keputusan, metode ini memiliki kelebihan sebagai berikut:

- Daerah pengambilan keputusan lebih simpel dan spesifik.
- Eliminasi perhitungan-perhitungan tidak diperlukan, karena ketika menggunakan metode pohon keputusan maka sample diuji hanya berdasarkan kriteria atau kelas tertentu.
- Fleksibel untuk memilih fitur dari internal node yang berbeda. Sehingga dapat meningkatkan kualitas keputusan yang dihasilkan jika dibandingkan ketika menggunakan metode penghitungan satu tahap yang lebih konvensional.
- Dengan menggunakan pohon keputusan, penguji tidak perlu melakukan estimasi pada distribusi dimensi tinggi ataupun parameter tertentu dari distribusi kelas tersebut. Karena metode ini menggunakan kcriteria yang jumlahnya lebih sedikit pada setiap node internal tanpa banyak mengurangi kualitas keputusan yang dihasilkan.

Kekurangan Pohon Keputusan

Pohon keputusan sangat membantu dalam pengambilan keputusan, namun pohon keputusan juga memiliki beberapa kekurangan, diantaranya:

- Kesulitan dalam mendesain pohon keputusan yang optimal.
- Hasil kualitas keputusan yang didapat sangat tergantung pada bagaimana pohon tersebut didesain. Sehingga jika pohon keputusan yang dibuat kurang optimal, maka akan berpengaruh pada kualitas dari keputusan yang didapat.
- Terjadi overlap terutama ketika kelas-kelas dan criteria yang digunakan jumlahnya sangat banyak sehingga dapat menyebabkan meningkatnya waktu pengambilan keputusan dan jumlah memori yang diperlukan.
- Pengakumulasian jumlah eror dari setiap tingkat dalam sebuah pohon keputusan yang besar.

Decision Tree pada RapidMiner

RapidMiner sebagai software pengolah data mining menyediakan tool untuk membuat decision tree. Hal ini tentu akan memudahkan kita membuat decision tree dengan menggunakan RapidMiner dibandingkan

membuat decision tree secara manual yaitu dengan melakukan perhitungan menggunakan algoritma C4.5 yang telah dijelaskan sebelumnya.

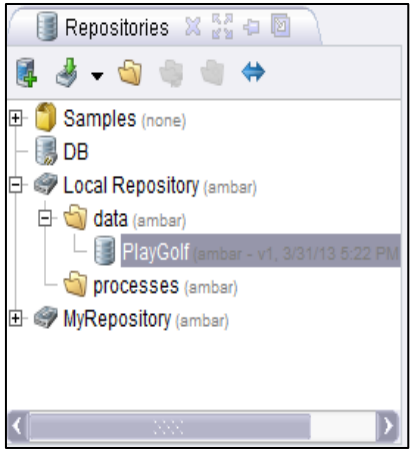
Contoh Kasus:
Keputusan Bermain Tenis

Pada contoh kali ini, kita akan membuat keputusan bermain tenis atau tidak. Untuk memudahkan dalam menggunakan RapidMiner untuk membuat decision tree, kita gunakan data sederhana yang ada pada sub bab decision tree. Pertama-tama data pada tabel 2 dibuat lagi dalam format excel seperti yang terlihat pada Gambar 4.3.

	A	B	C	D	E	F
1	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY	
2	Sunny	Hot	High	No	Don't Play	
3	Sunny	Hot	High	Yes	Don't Play	
4	Cloudy	Hot	High	No	Play	
5	Rainy	Mild	High	No	Play	
6	Rainy	Cool	Normal	No	Play	
7	Rainy	Cool	Normal	Yes	Play	
8	Cloudy	Cool	Normal	Yes	Play	
9	Sunny	Mild	High	No	Don't Play	
10	Sunny	Cool	Normal	No	Play	
11	Rainy	Mild	Normal	No	Play	
12	Sunny	Mild	Normal	Yes	Play	
13	Cloudy	Mild	High	Yes	Play	
14	Cloudy	Hot	Normal	No	Play	
15	Rainy	Mild	High	Yes	Don't Play	
16						

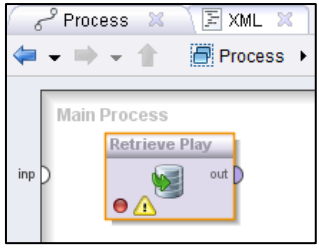
Gambar 4.3 Tabel Keputusan dalam Format xls

Setelah data yang kita punya dibuat dalam bentuk tabel format xls, selanjutnya lakukan *Importing Data* kedalam Repositori, seperti yang sudah dijelaskan pada Bab 2. Lalu cari table Microsoft Excel yang telah dibuat dan masukan kedalam Local Repository seperti yang terlihat pada Gambar 4.4.



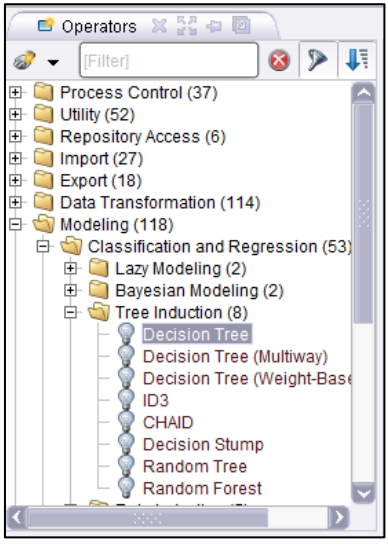
Gambar 4.4 Lokasi Tabel pada Repository

Lakukan Drag dan Drop Tabel PlayGolf kedalam Process view. Sehingga Operator Database muncul dalam View Proses seperti pada Gambar 4.5. Pada view Process, tabel PlayGolf yang dimasukkan ke dalam proses akan dijadikan sebagai Operator Retrieve.



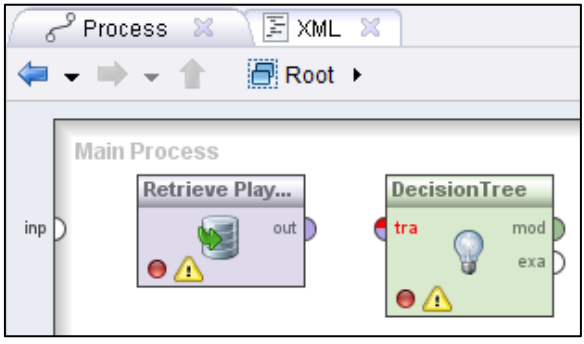
Gambar 4.5 Repository PlayGolf pada Main Process

Untuk membuat decision tree dengan menggunakan RapidMiner, kita membutuhkan operator Decision tree, operator ini terdapat pada View Operators. Untuk menggunakannya pilih Modelling pada View Operator, lalu pilih Classification and Regression, lalu pilih Tree Induction dan pilih Decision Tree.



Gambar 4.6 Daftar Operator pada View Operators

Setelah menemukan operator Decision Tree, seret (*drag*) operator tersebut lalu letakkan (*drop)* ke dalam view Process. Kemudian susun posisinya disamping operator Retrieve, seperti yang tampak pada Gambar 4.7.



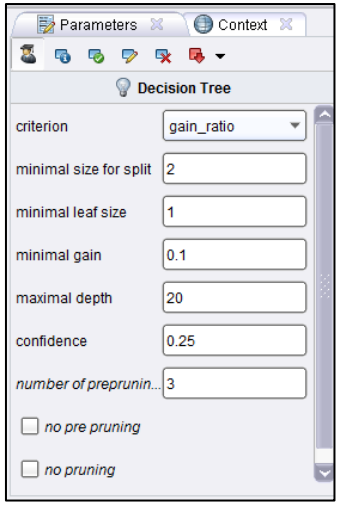
Gambar 4.7 Posisi Operator Decision Tree

Selanjutnya, hubungkan operator Retrieve dengan operator Decision Tree dengan menarik garis dari tabel PlayGolf ke operator Decision Tree dan menarik garis lagi dari operator Decision Tree ke result di sisi kanan, seperti yang tampak pada Gambar 4.8. Operator Decision Tree berguna untuk memperdiksikan keputusan dari atribut-aribut yang dimasukkan ke dalam operator retrieve. Dengan mengubah tabel (atribut) yang dimasukkan menjadi sebuah pohon keputusan.

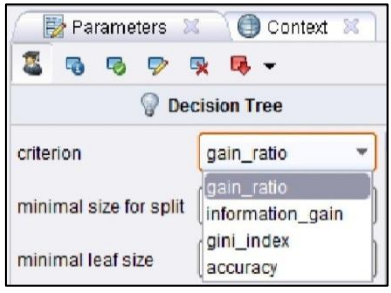
Gambar 4.8 Menghubungkan Tabel Playgolf dengan Operator Decision Tree

Pada operator Decision tree terdapat *input* **training set (tra)**, port ini merupakan output dari operator retrieve. Output dari operator lain juga dapat digunakan oleh port ini. Port ini menghasilkan ExampleSet yang dapat diperoses menjadi decision tree. Selain itu pada operator ini juga terdapat output model (mod) dan example set (exa). **Mod** akan mengonversi atribut yang dimasukkan menjadi mpdel keputusan dalam bentuk decision tree. **exa** merupakan port yang menghasilkan output tanpa mengubah inputan yang masuk melalui port ini. Port ini biasa digunakan untuk menggunakan kembali sama ExampleSet di operator lebih lanjut atau untuk melihat ExampleSet dalam Hasil Workspace.

Langkah selanjutnya ialah mengatur parameter sesuai dengan kebutuhan kita. Setelah menghubungkan operator retrieve dengan operator decision tree, atur parameter decision tree seperti pada gambar 4.9.



Gambar 4.9 Parameter Decision Tree



Gambar 4.10 Tipe Criterion

- Criterion, berguna memilih kriteria untuk menetapkan atribut sebagai akar dari decision tree. kriteria yang dapat dipilih, antara lain
    1. **Gain ratio** merupakan varian dari information_gain. Metode ini menghasilkan information gain untuk

setiap atribut yang memberikan nilai atribut yang seragam

2. **Information_gain**, dengan metode ini, semua entropi dihitung. Kemudian atribut dengan entropi minimum yang dipilih untuk dilakukan perpecahan pohon (split). Metode ini memiliki bias dalammemilih atribut dengan sejumlah besar nilai.
3. **Gini_index** merupakan ukuran ketidakaslian dari suatu ExampleSet. Metode ini memisahkan pada atribut yang dipilih memberikan penurunan indeks gini rata-rata yang dihasilkan subset.
4. **Accuracy,** metode ini memimilih beberapa atribut untuk memecah pohon (split) yang memaksimalkan akurasi dari keseluruhan pohon.

- Minimal size of split, Ukuran untuk membuat simpul-simpul pada decision tree. simpul dibagi berdasarkan ukuran yang lebih besar dari atau sama dengan parameter Minimal size of split. Ukuran simpul adalah jumlah contoh dalam subset nya

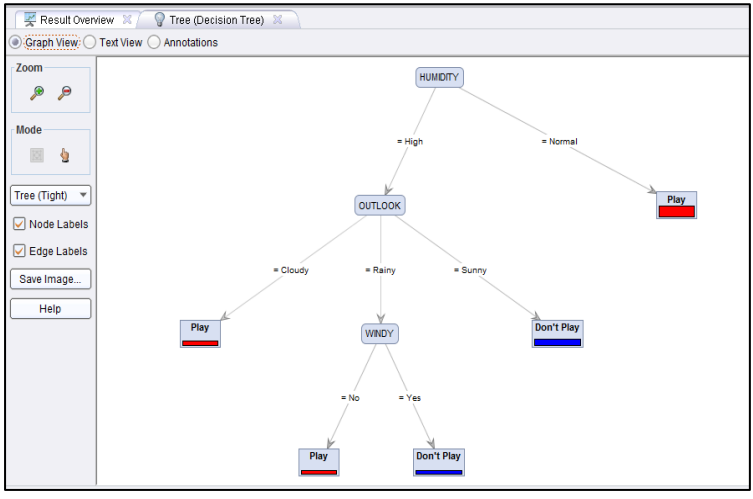
- Minimal leaf size, Pohon yang dihasilkan sedemikian rupa memiliki himpunan bagian simpul daun setidaknya sebanyak jumlah minimal leaf size.
- Minimal gain merupakan nilai gain minimal yang ditentukan untuk menghasilkan simpul pohon keputusan. Gain dari sebuah node dihitung sebelum dilakukan pemecahan. Node dipecah jika gain bernilai lebih besar dari Minimal Gain yang ditentukan. Nilai minimal gain yang terlalu tinggi akan mengurangi perpaecahan pohon dan menghasilkan pohon yang kecil. Sebuah nilai yang terlalu tinggi dapat mencegah pemecahan dan menghasilkan pohon dengan simpul tunggal.
- Maximal depth, Parameter ini digunakan untuk membatasi ukuran Putusan Pohon. Proses generasi pohon tidak berlanjut ketika kedalaman pohon adalah sama dengan kedalaman maksimal. Jika nilainya diatur ke '-1', parameter kedalaman maksimal menempatkan tidak terikat pada kedalaman pohon, pohon kedalaman maksimum dihasilkan. Jika nilainya diatur ke '1 ' maka akan dihasilkan pohon dengan simpul tunggal.

- Confidence, Parameter ini menentukan tingkat kepercayaan yang digunakan untuk pesimis kesalahan perhitungan pemangkasan.
- number of prepruning alternatives. Parameter ini menyesuaikan jumlah node alternatif mencoba untuk membelah ketika split dicegah dengan prepruning pada simpul tertentu.
    1. no prepruning, Secara default Pohon Keputusan yang dihasilkan dengan prepruning. Menetapkan parameter ini untuk menonaktifkan benar prepruning dan memberikan pohon tanpa prepruning apapun.
    2. no pruning Secara default Pohon Keputusan yang dihasilkan dengan pemangkasan. Menetapkan parameter ini untuk menonaktifkan benar pemangkasan dan memberikan sebuah unpruned

Setelah parameter diatur, klik ikon Run pada toolbar, seperti pada gambar 40 untuk menampilkan hasilnya. Tunggu beberapa saat, komputer membutuhkan waktu untuk menyelesaikan perhitungan.

Gambar 4.11 Ikon Run

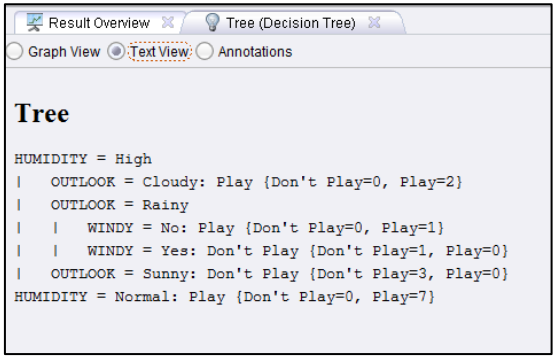
Setelah beberapa detik maka RapidMiner akan menampilkan hasil keputusan pada view Result. Jika kita pilih Graph view, maka akan ditampilkan hasilnya berbentuk pohon keputusan seperti pada gambar 4.12. Hasil pohon keputusan dapat disimpan dengan mengklik save image pada sisi kiri View Result.



Gambar 4.12 Hasil Berupa Graph Pohon Keputusan

Selain menampilkan hasil decision tree berupa graph atau tampilan pohon keputusan, RapidMiner juga menyediakan tool untuk menampilkan hasil berupa teks

view dengan mengklik button Text View seperti yang tampak pada Gambar 4.13.



Gambar 4.13 Hasil Berupa Penjelasan Teks

Contoh Kasus :
Keputusan seseorang mempunyai potensi menderita hipertensi

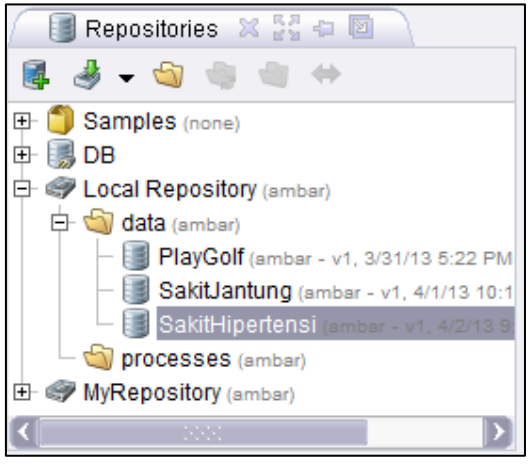
Sebelumnya kita telah mengetahui bagaimana membuat pohon keputusan untuk menentukan bermain tenis dengan menggunakan operator decision tree. Pada pembahasan kali ini kita akan membuat pohon keputusan untuk menentukan apakah seseorang berpotensi sakit hipertensi atau tidak. Untuk menambah pengeatahuan kita mengenai kegunaan operator yang ada pada RapidMiner, oleh karena itu untuk membuat pohon keputusan kali ini kita

menggunakan operator X-Validation, Apply Model dan Performance. Selain itu, kita juga tetap menggunakan operator decision tree dalam pembuatan pohon keputusan kali ini.

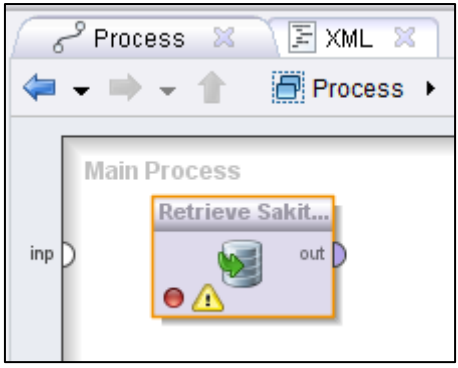
Cara yang digunakan dalam membuat pohon keputusan untuk menentukan apakah seseorang berpotensi sakit hipertensi, tidak jauh berbeda dengan cara membuat pohon keputusan yang sebelumnya, yaitu pertama-tama import data ke dalam repository RapidMiner, lalu lakukan drag dan drop data tersebut pada view process untuk mengubah data yang berisi atribut pohon keputusan menjadi operator retrieve. setelah itu, lakukan drag dan drop operator decision tree ke dalam view process dengan cara yang sama seperti penjelasan sebelumnya.

	A	B	C	D	E
1	**Usia**	**Berat**	**Kelamin**	**Hipertensi**	
2	Muda	Overweight	Pria	Ya	
3	Muda	Underweight	Pria	Tidak	
4	Muda	Average	Wanita	Tidak	
5	Tua	Overweight	Pria	Tidak	
6	Tua	Overweight	Pria	Ya	
7	Muda	Underweight	Pria	Tidak	
8	Tua	Overweight	Wanita	Ya	
9	Tua	Average	Pria	Tidak	
10					

Gambar 4.14 Tabel SakitHipertensi dalam format xls



Gambar 4.15 Lokasi Tabel pada Repository



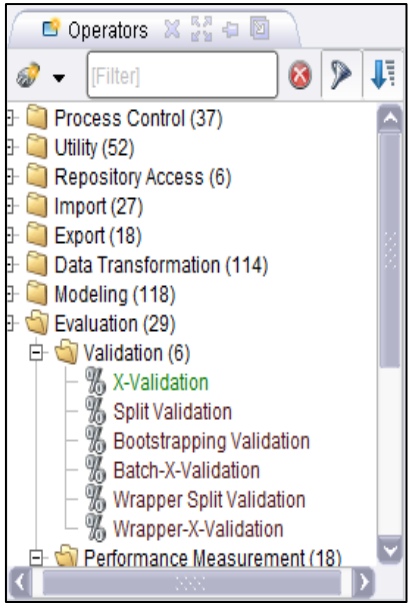
Gambar 4.16 Tabel SakitHipertensi pada Main Process

Untuk membuat pohon keputsan kali ini kita menggunakan operator X-Validation. Operator ini melakukan validasi silang untuk memperkirakan kinerja statistik operator pembelajaran (biasanya pada set data yang tak terlihat). Operator ini juga digunakan untuk memperkirakan seberapa akurat suatu model yang akan tampil dalam praktek. Operator X-Validasi merupakan operator bersarang yang memiliki dua subproses: *training subprocess* (subproses percobaan) dan *testing subprocess* (subproses pengujian). Subproses percobaan digunakan untuk melatih sebuah model. Model yang terlatih kemudian diterapkan dalam subproses pengujian.

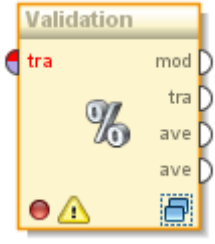
Biasanya proses belajar mengoptimalkan parameter model untuk membuat model sesuai dengan data percobaan. Jika kita kemudian mengambil sampel

independen dari data pengujian, umumnya model tersebut tidak cocok dengan data percobaan maupun data pengujian. Hal ini disebut dengan istilah 'over-pas', dan sangat mungkin terjadi ketika ukuran set data training kecil, atau ketika jumlah parameter dalam model besar. Sehingga validasi silang merupakan cara untuk memprediksi kesesuaian model untuk satu set pengujian hipotesis ketika set pengujian eksplisit tidak tersedia.

Untuk menemukan operator X-Validation, pilih Evaluation pada View Operator, lalu pilih Validation, lalu pilih X-Validation .Setelah menemukan operator X-Validation, seret (drag) operator tersebut lalu letakkan (drop) ke dalam view Process.



Gambar 4.17 Hirarki Operator X-Validation



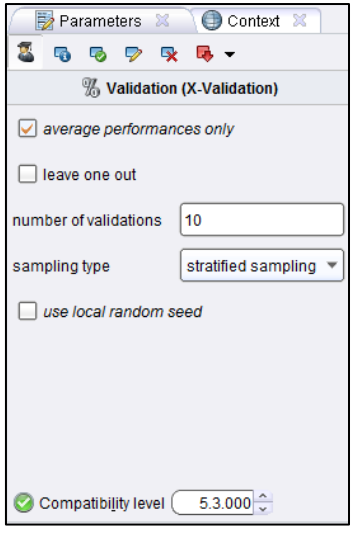
Gambar 4.18 Operator Validation

Operator X-Validation memiliki port input yaitu, **training example set (tra)** sebagai port input memperkirakan ExampleSet untuk melatih sebuah model (training data set). ExampleSet yang sama akan

digunakan selama subproses pengujian untuk menguji model.

Selain itu, operator ini juga memiliki port output sebagai berikut:

- **model (mod),** Pelatihan subprocess harus mengembalikan sebuah model yang dilatih pada input ExampleSet. Harap dicatat bahwa model yang dibangun ExampleSet disampaikan melalui port ini.
- **training example set (tra),** The ExampleSet yang diberikan sebagai masukan pada port input pelatihan dilewatkan tanpa mengubah ke output melalui port ini. Port ini biasa digunakan untuk menggunakan kembali ExampleSet sama di operator lebih lanjut atau untuk melihat ExampleSet dalam Workspace Result.
- **averagable (ave),** subproses pengujian harus mengembalikan Vector Kinerja. Hal ini biasanya dihasilkan dengan menerapkan model dan mengukur kinerjanya. Dua port tersebut diberikan tetapi hanya dapat digunakan jika diperlukan. Harap dicatat bahwa kinerja statistik dihitung dengan skema estimasi hanya perkiraan (bukan perhitungan yang tepat) dari kinerja yang akan dicapai dengan model yang dibangun pada set data yang disampaikan secara lengkap.



Gambar 4.19 Parameter X-Validation

Operator X-Validation juga memiliki parameter yang perlu diatur, diantaranya:

- **average performances only** (boolean), ini merupakan parameter ahli yang menunjukkan jika vector kinerja harus dirata-ratakan atau semua jenis dari hasil rata-rata.
- **leave one out** (boolean) Seperti namanya, leave one out validasi silang melibatkan penggunaan satu contoh dari ExampleSet asli sebagai data pengujian (dalam pengujian subproses), dan contoh-contoh yang tersisa sebagai data pelatihan (dalam pelatihan subprocess). Namun hal ini biasanya sangat mahal untuk ExampleSets besar dari sudut

pandang komputasi karena proses pelatihan diulang sejumlah besar kali (jumlah waktu contoh). Jika diatur dengan benar, parameter number of validations dapat diabaikan.

- **number of validations** (integer), parameter ini menentukan jumlah subset ExampleSet yang harus dibagi (setiap subset memiliki jumlah yang sama dari contoh). Juga jumlah yang sama dari iterasi yang akan berlangsung. Setiap iterasi melibatkan pelatihan model dan pengujian model. Jika ini ditetapkan sama dengan jumlah contoh dalam ExampleSet, Hal ini akan setara dengan operator X-Validasi dengan parameter leave one out set true.
- **sampling type** (selection), Operator X-Validasi dapat menggunakan beberapa jenis sampling untuk membangun subset. Sampel yang tersedia, diantaranya:
  1. linear_sampling, Linear sampling hanya membagi ExampleSet ke partisi tanpa mengubah urutan contoh yaitu subset dengan contoh-contoh berturut-turut diciptakan.
  2. shuffled_sampling, Shuffled Sampling membangun subset acak ExampleSet. Contoh dipilih secara acak untuk membuat subset.
  3. stratified_sampling, Stratified Sampling membangun subset acak dan memastikan

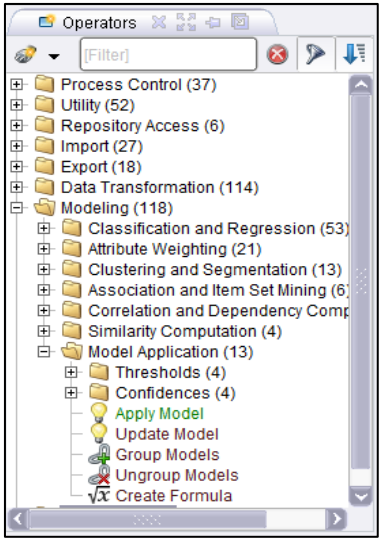
bahwa distribusi kelas dalam himpunan adalah sama seperti dalam ExampleSet seluruh.

- **use local random seed** (boolean), Parameter ini menunjukkan jika local random seed harus digunakan untuk mengacak contoh subset. Dengan menggunakan nilai yang sama dengan local random seed maka akan menghasilkan subset yang sama. Mengubah nilai parameter ini mengubah cara contoh menjadi acak, sehingga subset akan memiliki satu set yang berbeda dari contoh. Parameter ini hanya tersedia jika Shuffled atau Stratified sampling dipilih. Hal ini tidak tersedia untuk pengambilan sampel Linear karena tidak membutuhkan pengacakan, contoh yang dipilih secara berurutan
  - local random seed (integer), Parameter ini hanya tersedia jika parameteruse local random seed dipilih. parameter ini menentukan local random seed

Seperti yang telah disebutkan sebelumnya bahwa dalam membuat pohon keputusan pada contoh ini, kita menggunakan operator Apply Model. Operator ini menerapkan suatu model terlatih pada sebuah ExampleSet. Sebuah model pertama kali dilatih di sebuah ExampleSet, informasi yang berkaitan dengan ExampleSet dipelajari oleh model. Maka model tersebut dapat diterapkan pada ExampleSet yang lain dan

biasanya untuk prediksi. Semua parameter yang diperlukan disimpan dalam objek model. Ini adalah wajib bahwa kedua ExampleSets harus persis nomor yang sama, order, jenis dan peran atribut. Jika sifat meta data dari ExampleSets tidak konsisten, hal itu dapat menyebabkan kesalahan serius.

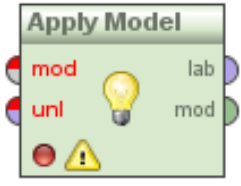
Untuk menemukan operator Apply Model, pilih Modeling pada View Operator, lalu pilih Model Application, lalu pilih Confidence dan pilih Apply Model .Setelah menemukan operator Apply Model, seret (drag) operator tersebut lalu letakkan (drop) ke dalam view Process.



Gambar 4.20 Hirarki Operator Apply

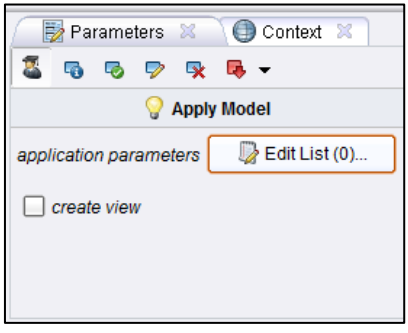
Operator ini memiliki port input yaitu, **model (mod)** port ini mengharapkan model. Port ini harus memastikan bahwa nomor, order, jenis dan peran atribut dari ExampleSet pada model yang dilatih konsisten dengan ExampleSet pada port input data unlabeled**. unlabelled data (unl)** port ini mengharapkan suatu ExampleSet. Ini harus memastikan bahwa nomor, order, jenis dan peran atribut ExampleSet ini konsisten dengan ExampleSet pada model yang dikirim ke port input model dilatih.

Operator ini juga memiliki port output, diantaranya, **labeled Data (lab),** Model yang diberikan dalam input diterapkan pada ExampleSet yang diberikan dan ExampleSet terbaru disampaikan dari port ini. Beberapa informasi akan ditambahkan ke input ExampleSet sebelum dikirimkan melalui port output. Dan **model (mod)**, Model yang diberikan sebagai masukan dilewatkan tanpa mengubah ke output melalui port ini.



Gambar 4.21 Operator Apply Model

Seperti yang terlihat pada gambar 4.22, Operator Apply Model hanya memiliki dua parameter yaitu, **application parameters** (menu) parameter ini merupakan parameter ahli yang berguna memodelkan parameter untuk aplikasi (biasanya tidak diperlukan). Dan **create view** (boolean) Jika model diterapkan pada port input mendukung Views, Hal ini mungkin untuk membuat View bukannya mengubah data yang mendasarinya. Transformasi yang akan biasanya dilakukan langsung di data kemudian akan dihitung setiap kali nilai diminta dan hasilnya dikembalikan tanpa mengubah data. Beberapa model tidak mendukung Views.

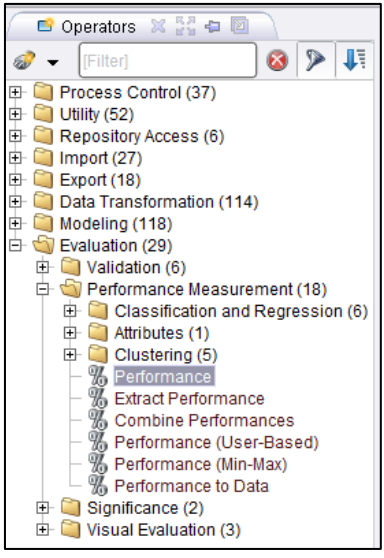


Gambar 4.22 Parameter Apply Model

Dalam membuat pohon keputusan untuk menentukan apakah seseorang berpotensi sakit Hipertensi, kita juga menggunakan operator Performance. Operator ini digunakan untuk evaluasi kinerja. Operator ini memberikan daftar nilai kriteria

kinerja. Kriteria kinerja secara otomatis ditentukan agar sesuai dengan jenis tugas belajar. Berbeda dengan operator lain, operator ini dapat digunakan untuk semua jenis tugas belajar. Secara otomatis menentukan jenis tugas belajar dan menghitung kriteria yang paling umum untuk jenis tersebut.

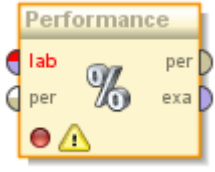
Untuk menemukan operator Performance, pilih Evaluation pada View Operator, lalu pilih Performance and Measurement, lalu pilih Performance. Setelah menemukan operator Performance, seret (drag) operator tersebut lalu letakkan (drop) ke dalam view Process.



Gambar 4.23 Hirarki Operator Performance

Operator Performance memiliki port input yaitu, **labelled data (lab),** Port ini mengharapkan mengharapkan ExampleSet berlabel. Apply Model merupakan contoh yang baik dari operator yang menyediakan data berlabel. Pastikan bahwa ExampleSet memiliki atribut label dan atribut prediksi. **performance (per)** Ini adalah parameter opsional yang membutuhkan Performance Vector.

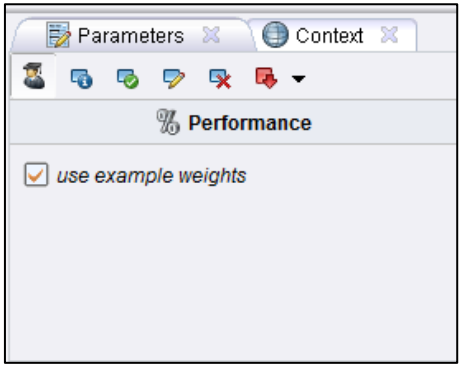
Selain itu, Operator ini juga memiliki port output yaitu, **performance (per),** port ini memberikan Peformance Vector (kita menyebutnya outputperformance-vektor untuk saat ini). Peformance Vector adalah daftar nilai kinerja kriteria. **example set (exa),** ExampleSet yang diberikan sebagai masukan dilewatkan tanpa mengubah ke output melalui port ini.



Gambar 4.24 Operator Performance

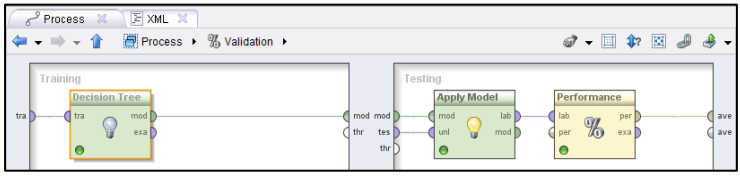
Operator ini hanya memiliki satu parameter yaitu, **use example weights** (boolean) Parameter ini memungkinkan contoh bobot contoh yang akan digunakan untuk perhitungan kinerja jika

memungkinkan. Parameter ini memiliki tidak memiliki efek jika atribut tidak memiliki peran bobot.



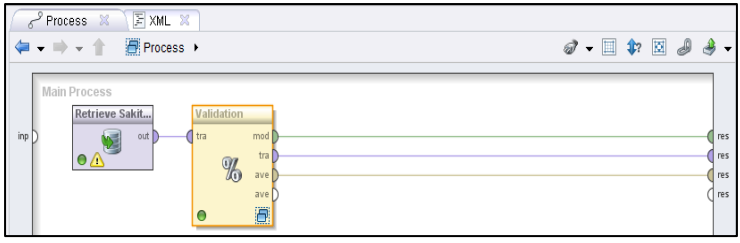
Gambar 4.25 Parameter Performance

Selanjutnya, susun dan hubungkan port-port dari operator decision tree, operator Apply Model dan operator Performance seperti yang terlihat pada Gambar 55.



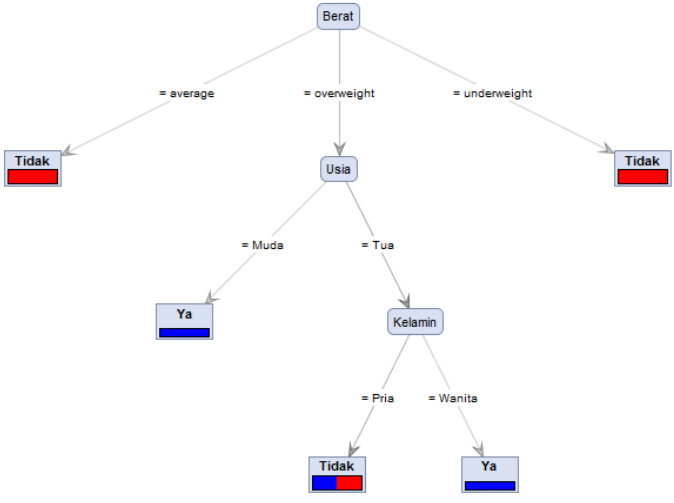
Gambar 4.26 Susunan Operator Decision Tree, Apply Model, Performance

Kemudian hubungkan operator retrieve (tabel SakitHipertensi) dengan operator validation dengan menarik garis pada port input dan output yang terdapat pada operator tersebut, seperti yang tampak pada Gambar 56.



Gambar 4.27 Susunan Operator Retrieve dengan Operator Validation

Setelah parameter dari masing-masing operator diatur, dan posisi operator disusun dengan benar, klik Run, lalu tunggu beberapa detik hingga RapidMiner akan menampilkan hasil Keputusan decision tree berupa graph pohon. seperti yang tampak pada Gambar 4.28.



Gambar 4.28 Tampilan Decision Tree

Chapter 5

Neural Network

Apa itu Neural Network?

Dapat dikatakan bahwa neural netwok dapat mempelajari pemetaan input data ke output data. Neural network merupakan model komputasi yang terinspirasi oleh prinsip-prinsip mengenai bagaimana cara otak manusia bekerja. Mereka dapat mempelajarinya dari data, mereka mampu men-generalisasi dengan baik, dan mereka tahan dengan kebisingan.

Biasanya jaringan saraf digunakan untuk masalah-masalah seperti klasifikasi (classification), prediksi (prediction), pengenalan pola (pattern recognition), pendekatan (approximation), dan asosiasi

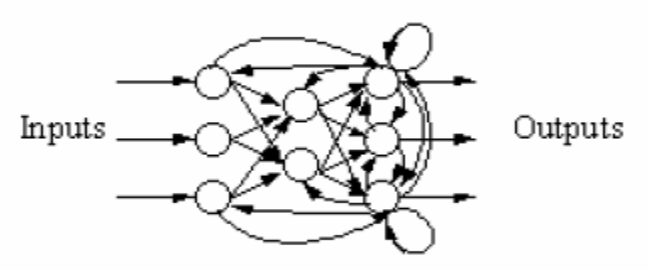
(association). Mereka hanya perlu belajar dari beberapa data sampel, dan setelah mereka telah mempelajarinya, mereka dapat bekerja dengan input data yang tidak diketahui, atau bahkan input data yang bising maupun tidak lengkap.

Secara umum Neural Network (NN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. NN ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut.

Secara sederhana NN adalah sebuah alat pemodelan data statistik non-linear. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada NN yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. Neuron adalah bagian dasar dari pemrosesan suatu Neural Network. Dibawah ini merupakan bentuk dasar dari suatu neuron.

Bentuk Neural Network

Setiap neural network terdiri dari unit pengolahan dasar yang saling berhubungan, yang disebut Neuron. Network belajar dengan memodifikasi bobot hubungan antara neuron selama proses pelatihan. Bentuk dasar arsitektur suatu Neural Network adalah sebagai berikut:



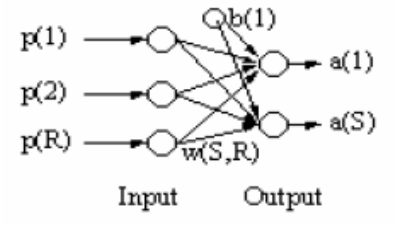
Gambar 5.1 Arsitektur Dasar Neural Network

Secara umum, terdapat tiga jenis Neural Network yang sering digunakanberdasarkan jenis network-nya, yaitu:

1. Single-Layer Neural Network
2. Multilayer Perceptron Neural Network
3. Recurrent Neural Networks

Single-Layer Neural Network

Neural Network jenis ini memiliki koneksi pada inputnya secara langsung ke jaringan output.

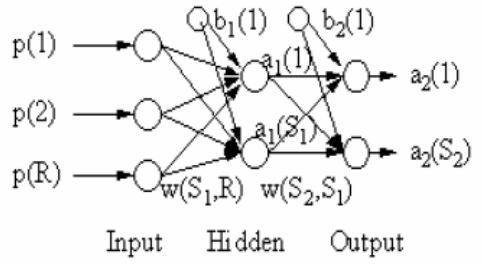


Gambar 5.2 Single-layer Neural Network

Jenis Neural Network ini sangatlah terbatas, hanya digunakan pada kasus-kasus yang sederhana.

Multilayer Perceptron Neural Network

Jenis Neural Network ini memiliki layer yang dinamakan “hidden”, ditengah layer input dan output. Hidden ini bersifat variable, dapat digunakan lebih dari satu hidden layer.

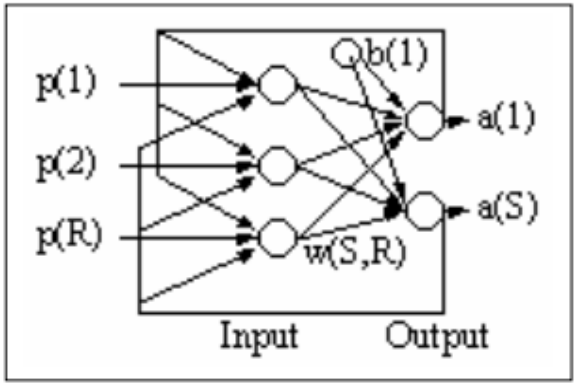


Gambar 5.3 Multilayer Perceptron Neural Network

Gambar di atas menunjukkan sebuah jaringan saraf sederhana yang dibuat dengan easyNeurons. Jenis jaringan ini disebut Multi Layer Perception dan itu merupakan salah satu jaringan yang paling umum digunakan.

Recurrent Neural Network

Neural network jenis ini memiliki ciri, yaitu adanya koneksi umpan balik dari output ke input.



Gambar 5.4 Recurrent Network

Kelemahan dari jenis ini adalah Time Delay akibat proses umpan balik dari output ke titik input.

Proses Pembelajaran pada Neural Network

Proses pembelajaran merupakan suatu metoda untuk proses pengenalan suatu objek yang sifatnya kontinuitas yang selalu direspon secara berbeda dari setiap proses pembelajaran tersebut. Tujuan dari pembelajaran ini sebenarnya untuk memperkecil tingkat suatu error dalam pengenalan suatu objek.

Secara mendasar, neural network memiliki sistem pembelajaran yang terdiri atas beberapa jenis berikut:

1. Supervised Learning
2. Unsupervised Learning

Supervised Learning

Sistem pembelajaran pada metoda Supervised learning adalah system pembelajaran yang mana, setiap pengetahuan yang akan diberikan kepada sistem, pada awalnya diberikan suatu acuan untuk memetakan suatu masukan menjadi suatu keluaran yang diinginkan. Proses pembelajaran ini akan terus dilakukan selama

kondisi error atau kondisi yang diinginkan belum tercapai. Adapun setiap perolehan error akan dikalkulasikan untuk setiap pemrosesan hingga data atau nilai yang diinginkan telah tercapai.

Unsupervised Learning

Sistem pembelajaran pada neural network, yang mana sistem ini memberikan sepenuhnya pada hasil komputasi dari setiap pemrosesan, sehingga pada sistem ini tidak membutuhkan adanya acuan awal agar perolehan nilai dapat dicapai. Meskipun secara mendasar, proses ini tetap mengkalkulasikan setiap langkah pada setiap kesalahannya dengan mengkalkulasikan setiap nilai weight yang didapat.

Siapa yang menggunakan Neural Network?

Beberapa aplikasi yang khas adalah gambar (image), sidik jari dan pengenalan wajah (fingerprint and face recognition), prediksi saham (stock prediction), prediksi untuk taruhan (sport bets prediction), klasifikasi pola dan pengakuan (pattern classification and recognition), pengawasan dan pengendalian (monitoring and control). Mereka digunakan dalam industri, kedokteran (diagnosa), aplikasi militer (seperti radar pada pengenalan citra),

keuangan dan robotika. Akhir-akhir ini mereka sangat popular di industri game karena berkat mekanisme belajar yang dilakukan, mereka dapat memberikan kontrol adaptif dan pembelajaran untuk karakter yang dikendalikan computer.

Kegunaan Neural Networks

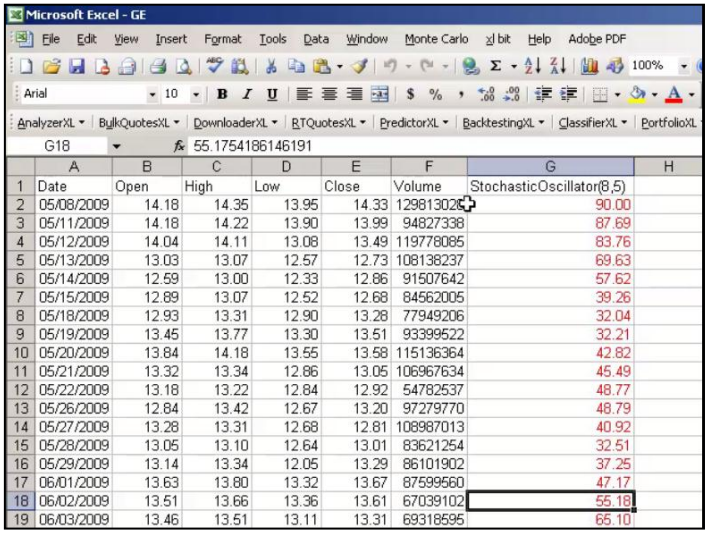
1. Pengenalan karakter optikal (Optical character recognition)
2. Pengenalan citra (Image recognition)
3. Pengenalan sidik jari (Fingerprint recognition)
4. Prediksi saham (Stock prediction)
5. Prediksi taruhan (Sport bets prediction)
6. Kontrol computer untuk karakter game (Computer controlled game characters)
7. Model statistical (Statistical modeling)
8. Data mining

Neural Network pada RapidMiner

Kita mulai dengan menggunakan data sederhana dalam tabel GE.xls. Data tersebut juga bisa kita dapatkan dengan melakukan pengunduhan melalui salah satu

add-ins Microsoft Excel yang bernama *DownloaderXL*, dimana data mengenai harga saham yang terjadi dalam rentang waktu tertentu telah dicatat pada sebuah *web hosting*.

Contoh Kasus:
Perkiraan harga saham dengan menggunakan metoda Neural Network.



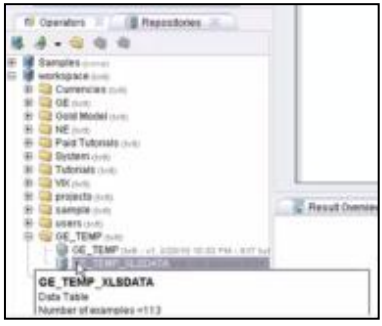
	A	B	C	D	E	F	G	H
1	Date	Open	High	Low	Close	Volume	StochasticOscillator(8,5)	
2	05/08/2009	14.18	14.35	13.95	14.33	129813020	90.00	
3	05/11/2009	14.18	14.22	13.90	13.99	94827338	87.69	
4	05/12/2009	14.04	14.11	13.08	13.49	119778085	83.76	
5	05/13/2009	13.03	13.07	12.57	12.73	108138237	69.63	
6	05/14/2009	12.59	13.00	12.33	12.86	91507642	57.62	
7	05/15/2009	12.89	13.07	12.52	12.68	84562005	39.26	
8	05/18/2009	12.93	13.31	12.90	13.28	77949206	32.04	
9	05/19/2009	13.45	13.77	13.30	13.51	93399522	32.21	
10	05/20/2009	13.84	14.18	13.55	13.58	115136364	42.82	
11	05/21/2009	13.32	13.34	12.86	13.05	106967634	45.49	
12	05/22/2009	13.18	13.22	12.84	12.92	54782537	48.77	
13	05/26/2009	12.84	13.42	12.67	13.20	97279770	48.79	
14	05/27/2009	13.28	13.31	12.68	12.81	108987013	40.92	
15	05/28/2009	13.05	13.10	12.64	13.01	83621254	32.51	
16	05/29/2009	13.14	13.34	12.05	13.29	86101902	37.25	
17	06/01/2009	13.63	13.80	13.32	13.67	87599560	47.17	
18	06/02/2009	13.51	13.66	13.36	13.61	67039102	55.18	
19	06/03/2009	13.46	13.51	13.11	13.31	69318595	65.10	

Gambar 5.5 Tabel GE.xls dalam Microsoft Excel

Buatlah *file* baru pada Microsoft Excel berdasarkan tabel harga saham. Berikan nama Header: Date, Open,

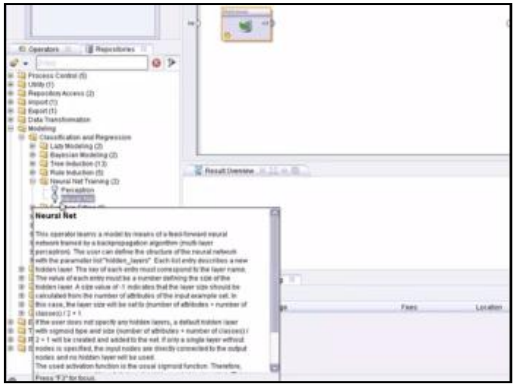
High, Low, Close, Volume, Stochastic Oscilator. Isilah sel seperti gambar [berapa]. Simpan dengan nama GE.xls

Lakukan pemilihan *repository* GE_TEMP_XLSDATA dengan melakukan *drag and drop* yang ditempatkan pada *panel main process* seperti gambar 5.6.



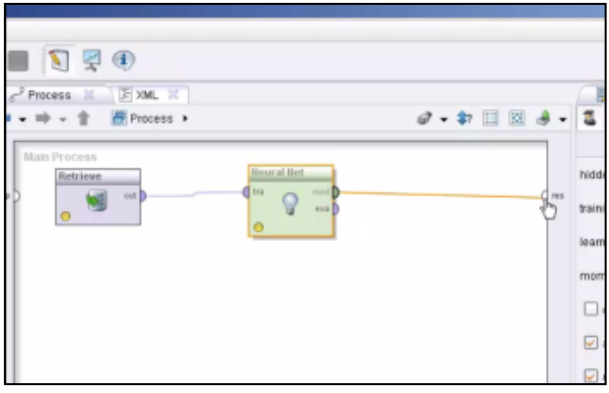
Gambar 5.6 Import Repository

Lakukan pemilihan operator *Neural Network* seperti gambar 5.7. Kemudian *drag and drop ke Main Process* seperti sebelumnya



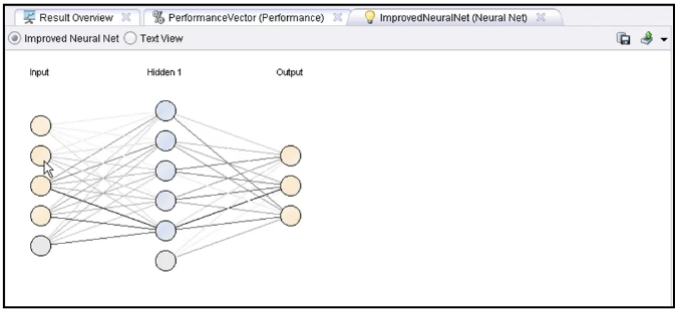
Gambar 5.7 Operator Neural network

Lakukan pembuatan hubungan antara *repository* dan *operator*, kemudian antara *operator* dengan hasil *output.*



Gambar 5.8 Menghubungkan Seluruh Operator ke Result

klik ikon Play . Tunggu beberapa saat, komputer membutuhkan waktu untuk menyelesaikan perhitungan.



Gambar 5.9 Ouput Neural Network

Gambar 5.9 merupakan grafik berbentuk *node* yang saling terhubung seperti layaknya sebuah jaringan syaraf dari hasil rules yang telah kita dapatkan

Chapter 6

Market Basket Analysis

Memahami Market Basket Analysis

Retail atau Eceran salah satu cara pemasaran produk meliputi semua aktivitas yang melibatkan penjualan barang secara langsung ke konsumen akhir, konsumen akhir membeli kumpulan produk dengan jumlah yang berbeda di waktu yang berbeda. Namun penjualan secara ritel hari ini bukanlah apa-apa jika insdustrinya tidak mampu berkompetisi dengan baik.

Lanskap yang kompleks dan cepat berubah, persaingan yang ketat, dan pelanggan yang semakin menuntut mendorong *retailer* harus memikirkan kembali bagaimana mereka beroperasi. Kemampuan untuk memahami pola pikir konsumen adalah hal yang sangat penting bagi *retailer*.

Teknologi telah membantu *retailer* dengan memungkinkan untuk menyimpan data konsumen dengan volume yang sangat besar dan biaya yang sangat wajar. *Retailer* kini dapat memiliki miliyaran informasi tentang informasi pelanggan mereka. Informasi ini dapat menjawab pertanyaan-pertanyaan penting termasuk: Kapan pelanggan akan membeli? Bagaimana pembayaran dilakukan? Berapa banyak dan apa item tertentu yang dibeli? Apa hubungan antara barang yang dibeli?

Tidak ada keraguan bahwa data point-of-sales (POS) ini yang (ketika digunakan secara efektif) diberdayakan pengecer untuk lebih memahami bisnis mereka dan meningkatkan pengambilan keputusan. Pengecer proaktif menggunakan informasi ini untuk memberikan penawaran yang ditargetkan yang sesuai dengan harapan konsumen dan kemudian memberikan dampak penghasilan positif.

Namun pada dasarnya, bagaimanakan *retailer* menggunakan miliyaran informasi ini? Jawabannya adalah menghubungkan produk-produk yang ada.

Sering kali, sebagai konsumen, kita cenderung mengabaikan bagaimana barang secara fisik diatur dalam sebuah toko *retail* atau supermarket. Apa yang mungkin terlihat (bagi kita) hanyalah seperti sebuah 'distribusi acak', namun sebenarnya hal tersebut merupakan pengaturan barang yang direncanakan secara cermat. Pada intinya, toko *retail* menilai pola pembelian pelanggan dan mengatur produk-produk yang akan dibeli secara sesuai. Sehingga menyebabkan pelanggan melakukan kegiatan pembelian beberapa produk sekaligus tanpa disadarinya.

Teknik untuk menemukan hubungan dari produk-produk yang dibeli secara bersamaan inilah yang dikenal sebagai *Market Basket Analysis* (MBA). Seperti namanya, *Market Basket Analysis* pada dasarnya melibatkan penggunaan data transaksional konsumen untuk mempelajari pola pembelian dan menjelajahi kemungkinan (probabilitas dan) *cross-selling*. Tujuan dari MBA adalah untuk memanfaatkan data penjualan efektif untuk meningkatkan taktik pemasaran dan penjualan di tingkat toko.

Contoh yang paling umum dari Market Basket Analysis adalah “Beer dan Diapers”. Contoh ini merupakan kasus dari salah satu toko retail besar yang ada di US, Wal-Mart. Seorang manajer toko menemukan hubungan yang kuat antara salah satu merek popok bayi (diapers) dan salah satu merek beer pada beberapa pembeli. Analisa pembelian mengungkapkan bahwa kegiatan pembelian dilakukan oleh laki-laki dewasa pada hari jumat malam terutama sekitar jam enam dan tujuh sore. Setelah beberapa observasi, supermarket mengetahui bahwa:

- Karena bungkus dari popok bayi sangat besar, para istri, dimana dalam banyak kasus adalah seorang ibu rumah tangga, akan menyuruh suaminya untuk membelinya.
- Pada akhir dari minggu, para suami dan ayah akan menghabiskan minggunya dengan membeli beberapa beer.

Jadi, apa yang akan dilakukan supermarket dari pengetahuan ini?

- Mereka menempatkan *premium beer* tepat disebelah *diapers*
- Hasilnya adalah para ayah akan membeli diapers dan yang biasanya membeli beer biasa sekarang

membeli *premium beer* seperti yang sudah diperkirakan.

- Secara signifikan, para pria yang biasanya tidak membeli bir sebelum mulai berbelanja akan membelinya karena itu begitu mudah dilihat dan diambil - hanya sebelah popok (cross-sell)

Istilah Market Basket Analysis sendiri datang dari kejadian yang sudah sangat umum terjadi di dalam pasar swalayan, yakni ketika para konsumen memasukkan semua barang yang merak beli ke dalam keranjang (basket) yang umumnya telah disediakan oleh pihak swalayan itu sendiri. Informasi mengenai produk-produk yang biasanya dibeli secara bersama-sama oleh para konsumen dapat memberikan "wawasan" tersendiri bagi para pengelola toko atau swalayan untuk menaikkan laba bisnisnya (Albion Research, 2007).

Metodologi Association Rules

Metodologi Association Rules, atau Analisis Asosiasi adalah sebuah metodologi untuk mencari relasi (asosiasi) istimewa/menarik yang tersembunyi dalam himpunan data (atau data set) yang besar. Salah satu penerapan Metode Association rules adalah pada Market Basket Analysis.

Association rule adalah sebuah ekspresi implikasi dari bentuk X → Y, dimana X dan Y adalah itemset yang saling terpisah (disjoint), dengan kata lain X ∩ Y = ∅. Dalam menentukan Association Rule, terdapat suatu interestingness measure (ukuran ketertarikan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Ada dua ukuran yaitu:

1. Support: Bagian transaksi yang mengandung kedua X dan Y.

   $$Support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi}$$

   Atau jika terdapat dua buah item dalam X, nilai support diperoleh dari rumus berikut:

   $$Support(A \cap B) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Total\ Transaksi}$$

2. Confidence: Seberapa sering item dalam Y muncul di transaksi yang mengandung X.

   $$Cofidence = P(B|A) = \frac{Jumlah\ transaksi\ mengandung\ A\ dan\ B}{Jumlah\ transaksi\ mengandung\ A}$$

Kudua ukuran ini nantinya berguna dalam menentukan interesting association rules, yaitu untuk dibandungkan dengan batasan (threshold) yang ditentukan oleh user. Batasan tersebut umumnya bernama *minimum support* dan *minimum confidence*.

Mengapa menggunakan Support dan Confidence? Support adalah ukuran yang penting karena jika aturan memiliki support yang kecil, maka kejadian bisa saja hanyalah sebuah kebetulan. Aturan Support yang rendah juga cenderung tidak menarik dari perspektif bisnis karena mungkin tidak akan memberikan keuntungan saat mempromosikan barang-barang yang jarang dibeli pelanggan bersamaan. Untuk alasan ini, dukungan sering digunakan untuk menghilangkan ketidak-menarikan ini. Confidence, adalah ukuran kehandalan dari kesimpulan yang dibuat oleh aturan. Semakin besar Confidence, semakin besar kemungkinan untuk Y hadir dalam transaksi yang mengandung X. Confidence juga memberikan probabilitas bersyarat dari Y yang diberikan ke X.

Contoh Association Rules

Untuk lebih memahami Association Rules, mari kita telusuri contoh berikut. Sebuah toko retail telah melakukan transaksi dengan pembeli seperti yang tertulis pada tabel.

Tabel 6.1 Tabel Transaksi

Kode Transaksi	Produk yang terjual
001	Pena, Roti, Mentega
002	Roti, Mentega, Telur
003	Buncis, Telur, Susu

004	Roti, Mentega
005	Roti, Mentega, Kecap, Telur, Susu

Tahap pertama adalah mencari nilai dari Support sesuai dengan rumus yang telah disebutkan sebelumnya. Misalnya, Untuk transaksi yang memuat {roti, mentega} ada 4, maka nilai supportnya adalah 80%. Lalu jumlah transaksi yang memuat {Roti, Mentega, Susu} ada 2, maka nilai supportnya adalah 40%. Sedangkan transaksi yang memuat {buncis} hanya 1, maka nilai supportnya adalah 20%. Jika kita tentukan bahwa *minimum support*nya adalah 30%, maka rule yang memenuhi adalah sebagai berikut:

Tabel 6.2 Kombinasi Produk dan Nilai Support

Kombinasi Produk	Nilai Support
{roti}	80%
{mentega}	80%
{telur}	60%
{susu}	60%
{roti, mentega}	80%
…	…
{mentega, telur, susu}	40%
{roti, mentega, telur, susu}	40%

Setelah semua pola kombinasi dan nilai dari Supportnya ditemukan, barulah dicari *Association Rules*

yang memenuhi syarat minimum untuk confidence. Bila ditentukan syarat minimum untuk confidence sebesar 50% maka Association Rules yang dapat dipakai adalah:

Tabel 6.3 Association Rules dan Nilai Confidence

Association Rules	Support	Confidence
{roti} → {mentega}	80%	100%
{roti} → {telur}	40%	50%
{roti} → {susu}	40%	50%
{roti} → {mentega, telur}	40%	50%
{roti} → {mentega, susu}	40%	50%
{roti} → {telur, susu}	40%	50%
{roti} → {mentega, telur, susu}	40%	50%
…	…	
{mentega, telur} → {roti}	40%	100%
…	…	
{roti, mentega, susu} → {telur}	40%	100%
{roti, telur, susu} → {mentega}	40%	100%
{mentega, telur, susu} → {roti}	40%	100%

Assosiation Rule akan dipilih sesuai kebijakan manajer toko, semakin tinggi support dan confidence semakin baik hasilnya. Misalkan kita ambil contoh yaitu {mentega, telur} → {roti} yang memiliki nilai Support 80% dan Confidence 100%, artinya adalah: “Seorang konsumen yang membeli mentega dan telur memiliki kemungkinan 100% untuk juga membeli roti. Aturan ini

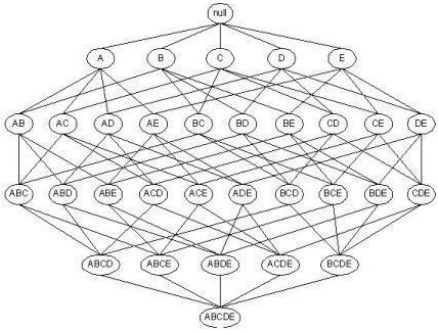
cukup signifikan karena mewakili 40% dari catatan selama ini."

Frequent Itemset Generation dan Rule Generation

Frequent Itemset Generation

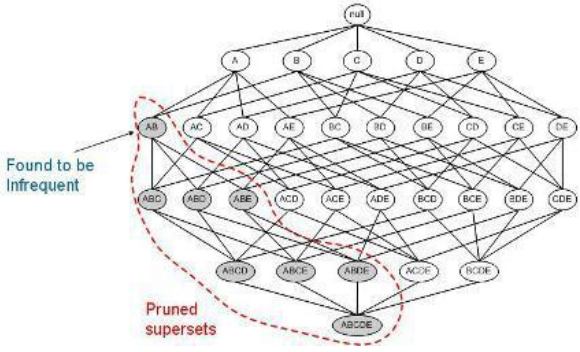
Tujuannya adalah untuk menemukan semua itemset yang memenuhi *minimum support*. Item set ini sering disebut dengan frequent. Namun Masalah utama pencarian Frequent Itemset adalah banyaknya jumlah kombinasi itemset yang harus diperiksa apakah memenuhi minimum support atau tidak. Salah satu cara untuk mengatasinya adalah dengan mengurangi jumlah kandidat itemset yang harus diperiksa.

Apriori adalah salah satu pendekatan yang sering digunakan pada Frequent Itemset Mining. Prinsip Apriori adalah jika sebuah itemset infrequent, maka itemset yang infrequent tidak perlu lagi diexplore supersetnya sehingga jumlah kandidat yang harus diperiksa menjadi berkurang. Kira kira ilustrasinya seperti ini:



Gambar 6.1 Frequent Item Set tanpa Apriori

Pada gambar 36, pencarian Frequent Itemset dilakukan tanpa menggunakan prinsip Apriori. Dengan menggunakan prinsip Apriori, pencarian Frequent Itemset akan menjadi seperti di bawah ini:



Gambar 6.2 Frequent Item Set dengan Apriori

Dapat dilihat bahwa dengan menggunakan Apriori, jumlah kandidat yang harus diperiksa cukup banyak berkurang.

Rule Generation

Tujuannya adalah untuk mengekstrak semua aturan yang memiliki high-confidence dari itemsets yang ditemukan dari langkah sebelumnya. Aturan ini disebut Strong Rules.

Market Basket Analysis pada RapidMiner

Sekali lagi, pencarian Rule pada Association Rules merupakan sebuah proses yang luar biasa panjang. Manusia tidak akan mampu untuk melakukan pengitungan dengan berates-ratus data (belum kombinasi dari seluruh item yang ada). Maka dari itu, untuk mencari seluruh Rules yang ada, RapidMiner telah menyediakan tools untuk mempermudah pengguna. Untuk memahami cara menggunakan tools ini, ikuti manual berikut secara seksama.

Contoh Kasus :

Transaksi Penjualan Sederhana.

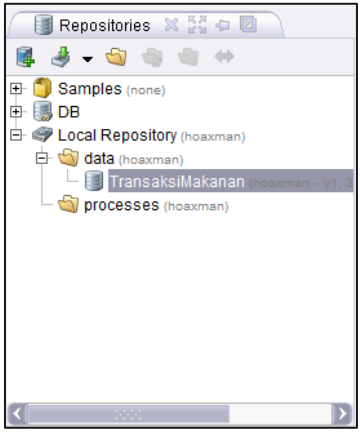
Kita mulai dengan menggunakan data sederhana yang kita miliki yang terdapat pada sub bab pengenalan Market Basket Analysis, Tabel 5.1.

	A	B	C	D	E	F	G	H
1	TID	PENA	ROTI	MENTEGA	TELUR	BUNCIS	SUSU	KECAP
2	001	1	1	1	0	0	0	0
3	002	0	1	1	1	0	0	0
4	003	0	0	0	1	1	1	0
5	004	0	1	1	0	0	0	0
6	005	0	1	1	1	0	1	1
7								

Gambar 6.3 Tabel Penjualan Sederhana

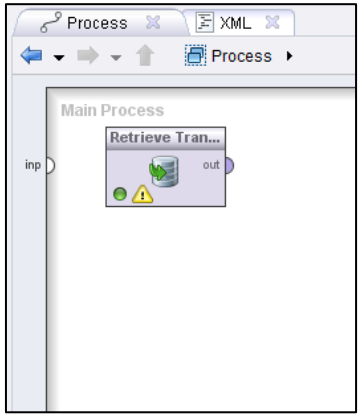
Buatlah Table baru pada Microsoft Excel berdasarkan tabel 5.1. Berikan nama Header: TID (Transaction ID), PENA, ROTI, MENTEGA, TELUR, BUNCIS, SUSU, KECAP. Isilah cell seperti gambar 5.3. Simpan dengan nama TransaksiMakanan.xls.

Lakukan *Importing Data* kedalam Repositori, seperti yang sudah dijelaskan pada **Bab 2**. Browse table Microsoft Excel yang telah dibuat, dan masukan kedalam Local Repository, seperti gambar disamping.



Gambar 6.4 Repositori

Lakukan Drag dan Drop Tabel TransaksiMakanan tadi kedalam Process. Sehingga Operator Database muncul dalam Main Proses seperti gambar 5.5.



Gambar 6.5 Database dalam Main Process

Untuk melakukan Market Basket Analysis, kita membutuhkan setidaknya tiga buah operator, antara lain Association Rule, FP-Growth, dan Numerical to Binomial.

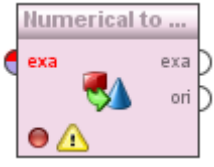
Gambar 6.6 Operator Create Association Rules

Assocuation rules dilakukan dengan menganalisis data pada *frequent if/then patterns*

menggunakan kriteria support dan confidence untuk mengidentifikasikan suatu relasi antar item. *Frequent if/then pattern* digali menggunakan operator FP-Growth. Operator Create Association Rules menggunakan frequent itemsets ini dan menghasilkan association rules.

Gambar 6.7 Operator FP-Growth

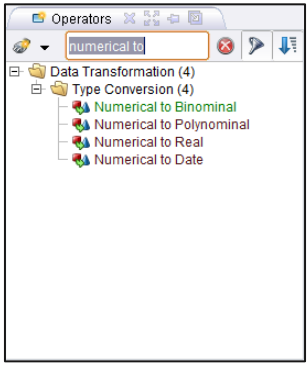
Frequent itemsets merupakan kelompok item yang sering muncul bersama-sama dalam data. Operator *FP-Growth* mengkalkulasikan semua frequent itemset dari input yang diberikan menggunakan struktur data FP-tree. Adalah wajib bahwa semua atribut dari masukan merupakan bilangan binominal (true/false).



Gambar 6.8 Operator Numerical to Binominal

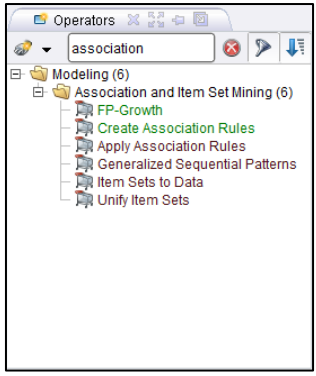
Operator *Numerical to Binominal* diperlukan untuk mengubah nilai atribut yang berada pada table TransaksiMakanan menjadi binominal.

Selanjutnya lakukan Pencarian Filter untuk memudahkan kita menemukan operator yang dibutuhkan, lakukan seperti pada gambar berikut.



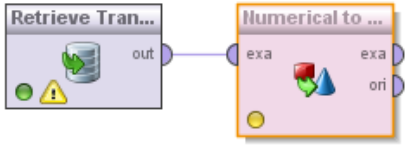
Gambar 6.9 Pencarian Operator Numerical to Binominal

Untuk Mencari Operator *Numerical to Binominal*, lakukan pencarian seperti gambar disamping. Operator ini terdapat pada hirarki: Data Transformation → Type Conversion



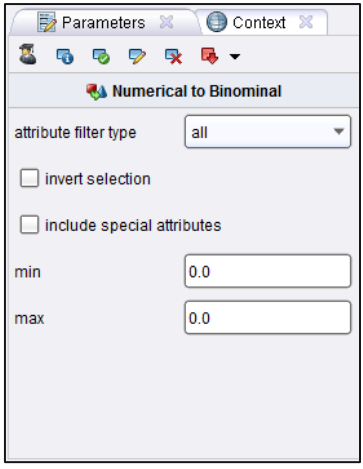
Gambar 6.10 Pencarian Association Rules

Susunlah ketiga operator tersebut menjadi seperti gambar 5.11.



Gambar 6.11 Menghubungan Database TransaksiMakanan pada Operator Numerical to Binomial

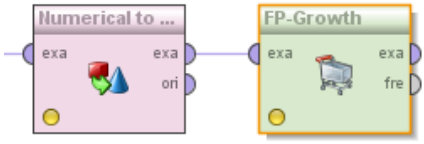
Hubungkan Tabel TransaksiMakanan yang kita miliki dengan operator Numerical to Binominal. Proses ini akan membuat nilai dari Tabel Transaksi makan mejadi *Binominal Attributes*.



Gambar 6.12 Parameter Numerical to Binomial

Data yang kita miliki merupakan data sederhana. Kita hanya memperhitungkan 1 buah penjualan produk pada setiap transaksinya. Maka nilai yang terbaik untuk menjadi *false* adalah ketika tidak ada produk tertentu yang terjual dalam suatu transaksi, jadi kita sini nilai *min* dan *max* menjadi 0, Sehingga yang bernilai *false* adalah ketika sebuah produk tidak terdapat pada sebuah transaksi.

Hubungkan operator *Numerical to Binominal* dengan operator FP-Growth pada example output.



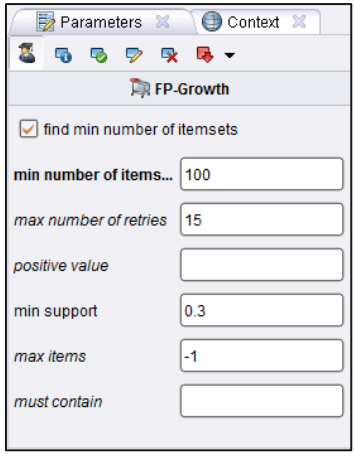
Gambar 6.13 Menghubungkan Operator Numerical to Binomial dengan Operator FP-Growth

Terdapat dua buah output untuk *Numerical to Binominal*, yaitu example dan original.

- Example, *numeric attributes* dikonversikan menjadi *binominal attributes* melalui output ini.
- Original, *numeric attributes* dilewatkan tanpa konversi. Biasanya digunakan untuk proses tertentu saat dibutuhkan.

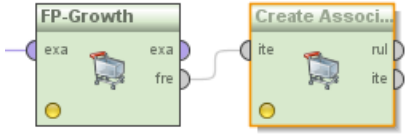
Lewatkan output pada *example*.

Isilah Parameter FP-Growth seperti gambar berikut. Sesuai dengan contoh pada sub bab seselumnya, isilah *minimum support* senilai 30% atau 0.3.



Gambar 6.14 Parameter FP-Growth

Kemudian hubungkan operator *FP-Growth* dengan operator *Association Rules*.



Gambar 6.15 Menghubungkan Operator FP-Growth dengan Operator Create Association Rules

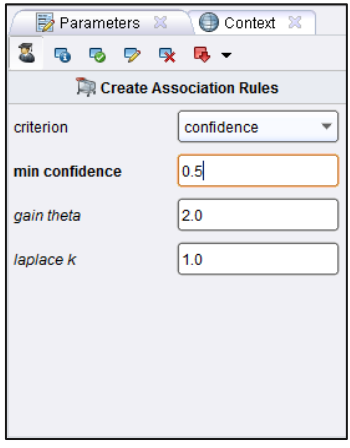
Terdapat dua buah output pada operator FP-Growth, yakni *example* dan *frequent*.

- *Example*, input yang diberikan dilewatkan tanpa adanya perubahan. Biasanya digunakan untuk proses tertentu saat dibutuhkan.

- *Frequent*, frequent itemset dikirimkan melalui output ini.

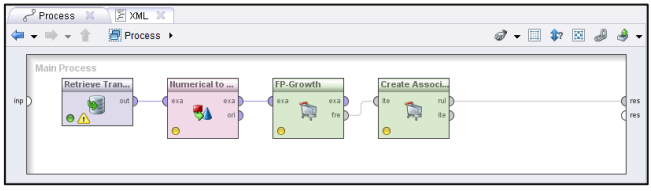
Lewatkan output pada *frequent.*

Kemudian isilah Parameter Association Rules seperti gambar berikut. Sesuai dengan contoh pada sub bab seselumnya, isilah *minimum confidence* senilai 50% atau 0.5.



Gambar 6.16 Parameter Association Rules

Setelah itu hubungkan Association Rules pada result. Sehingga seluruhnya membentuk seperti gambar 5.17. lalu klik ikon Play . Tunggu beberapa saat, komputer membutuhkan waktu untuk menyelesaikan perhitungan.



Gambar 6.17 Susunan Operator Association Rules

Setelah beberapa detik, akan muncul sebuah tab Association Rules yang baru, yang isinya adalah sebuah table berisi seluruh itemset yang memenuhi parameter FP-Growth dan Association Rules. Totalnya terdapat 152 rules yang ditemukan.

No.	Premises	Conclusion	Support	Confid...	LaPl...	Gain	p-s	Lift	Convi...
131	TELUR, SUSU, KECAP	MENTEGA	0.200	1	1	-0.200	0.040	1.250	∞
132	ROTI, SUSU	MENTEGA, TELUR, KECAP	0.200	1	1	-0.200	0.160	5	∞
133	MENTEGA, SUSU	ROTI, TELUR, KECAP	0.200	1	1	-0.200	0.160	5	∞
134	ROTI, MENTEGA, SUSU	TELUR, KECAP	0.200	1	1	-0.200	0.160	5	∞
135	ROTI, TELUR, SUSU	MENTEGA, KECAP	0.200	1	1	-0.200	0.160	5	∞
136	MENTEGA, TELUR, SUSU	ROTI, KECAP	0.200	1	1	-0.200	0.160	5	∞
137	ROTI, MENTEGA, TELUR, SUSU	KECAP	0.200	1	1	-0.200	0.160	5	∞
138	KECAP	ROTI, MENTEGA, TELUR, SUSU	0.200	1	1	-0.200	0.160	5	∞
139	ROTI, KECAP	MENTEGA, TELUR, SUSU	0.200	1	1	-0.200	0.160	5	∞
140	MENTEGA, KECAP	ROTI, TELUR, SUSU	0.200	1	1	-0.200	0.160	5	∞
141	ROTI, MENTEGA, KECAP	TELUR, SUSU	0.200	1	1	-0.200	0.120	2.500	∞
142	TELUR, KECAP	ROTI, MENTEGA, SUSU	0.200	1	1	-0.200	0.160	5	∞
143	ROTI, TELUR, KECAP	MENTEGA, SUSU	0.200	1	1	-0.200	0.160	5	∞
144	MENTEGA, TELUR, KECAP	ROTI, SUSU	0.200	1	1	-0.200	0.160	5	∞
145	ROTI, MENTEGA, TELUR, KECAP	SUSU	0.200	1	1	-0.200	0.120	2.500	∞
146	SUSU, KECAP	ROTI, MENTEGA, TELUR	0.200	1	1	-0.200	0.120	2.500	∞
147	ROTI, SUSU, KECAP	MENTEGA, TELUR	0.200	1	1	-0.200	0.120	2.500	∞
148	MENTEGA, SUSU, KECAP	ROTI, TELUR	0.200	1	1	-0.200	0.120	2.500	∞
149	ROTI, MENTEGA, SUSU, KECAP	TELUR	0.200	1	1	-0.200	0.080	1.667	∞
150	TELUR, SUSU, KECAP	ROTI, MENTEGA	0.200	1	1	-0.200	0.040	1.250	∞
151	ROTI, TELUR, SUSU, KECAP	MENTEGA	0.200	1	1	-0.200	0.040	1.250	∞
152	MENTEGA, TELUR, SUSU, KECAP	ROTI	0.200	1	1	-0.200	0.040	1.250	∞

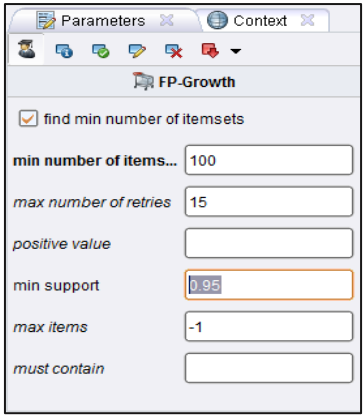
Gambar 6.18 Hasil Association Rules Pertama

Tentunya ini akan menyulitkan kita untuk mengambil kesimpulan karena jumlah rules yang terlalu banyak. Maka dari itu yang harus kita lakukan adalah mengubah nilai *minimum support* dan *minimum confidence*.

Klik ikon Edit  untuk kembali pada *model view.* Lalu klik Operator FP-Growth.

Gambar 6.19 Operator FP-Growth

Kemudian lihat bagian parameter. Ubah nilai minimum support menjadi 95%, seperti yang sudah dijelaskan pada sub bab Association Rules, semakin tinggi nilai support maka semakin dapat dipercaya rules yang dihasilkan. Namun perhitungkan juga hasilnya nanti. Terkadang jika nilai minimum supportnya terlalu tinggi, maka akan muncul kemungkinan tidak ditemukannya rules yang memenuhi.



Gambar 6.20 Mengubah Parameter FP-Growth

Sekarang kita beralih pada Operator Create Association Rules.

Gambar 6.21 Operator Create Association Rules

Ubah nilai minimum confidence menjadi 95% atau 0.95, semakin tinggi nilai confidence maka semakin dapat dipercaya rules yang dihasilkan. Namun perhitungkan juga hasilnya nanti. Terkadang jika nilai minimum confidence terlalu tinggi, maka akan muncul kemungkinan tidak ditemukannya rules yang memenuhi.

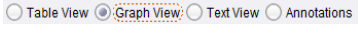
Gambar 6.22 Mengubah Parameter Association Rules

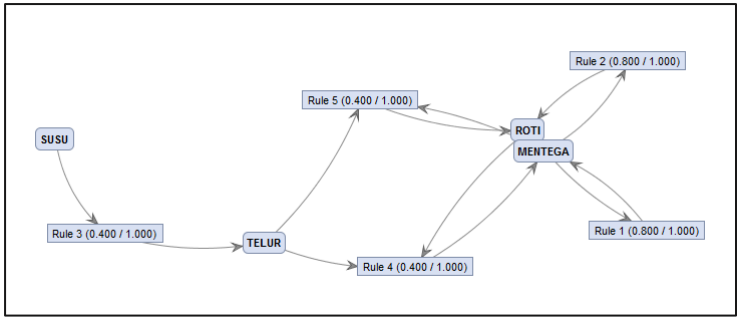
Klik ikon Play untuk menampilkan hasil yang baru.

No.	Premises	Conclusion	Support	Confid...	LaPla...	Gain	p-s	Lift	Convi...
1	ROTI	MENTEGA	0.800	1	1	-0.800	0.160	1.250	∞
2	MENTEGA	ROTI	0.800	1	1	-0.800	0.160	1.250	∞
3	SUSU	TELUR	0.400	1	1	-0.400	0.160	1.667	∞
4	ROTI, TELUR	MENTEGA	0.400	1	1	-0.400	0.080	1.250	∞
5	MENTEGA, TELUR	ROTI	0.400	1	1	-0.400	0.080	1.250	∞

Gambar 6.23 Hasil Association Rules Kedua

Maka sekarang yang dihasilkan menjadi lima buah rules. Kita bisa mengambil salah satu dari rules ini untuk dijadikan sebuah pegangan dalam strategi penjualan retail. Tentunya yang memiliki nilai support dan confidence yang tinggi.

Untuk melihat dalam bentuk grafik. kita dapat memilih opsi Graph View. 



Gambar 6.24 Hasil dalam bentuk Graph View

Glossarium

Algoritma	Kumpulan perintah untuk menyelesaikan suatu masalah.
Apriori	Algoritma untuk frequent itemset mining dan association rule dalam database transaksional. DIhasilkan dengan menidentifikasi setiap buah item, dan memperluasnya menjadi kombinasi kumpulan item yang lebih besar asalkan himpunan item muncul cukup sering dalam database.
Association Rules	Sebuah metodologi untuk mencari relasi (asosiasi) istimewa/menarik yang tersembunyi dalam himpunan data (atau data set) yang besar.
Binominal Attributes	Atribut dengan tipe Binominal (true dan false).
Confidence	(Market Basket Analysis) Seberapa sering item dalam Y muncul di transaksi yang mengandung X.
Decision tree	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.
Disjoint	Himpunan terpisah, tidak ada elemen yang berhubungan diantara kedua himpunan yang bersangkutan
Flowchart	Sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma.
Frequent Itemset	Itemset yang memunyai support >= minimum support yang diberikan oleh user dalam Market Basket Analysis.
Market Basket Analysis	Teknik untuk menemukan hubungan dari produk-produk yang dibeli secara bersamaan.
MBA	Lihat Market Basket Analysis.
Minimum Support	Nilai Support Terkecil dalam Market Basket Analysis yang dapat di toleransi.
Minimum Confidence	Nilai Confidence terkecil dalam Market Basket Analysis yang dapat di toleransi.
Neural Network	Jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia.
Numeric Attributes	Atribut dengan tipe Numerical (1-9).
Operator	suatu tanda atau simbol yang dipakai untuk menyatakan suatu operasi atau manipulasi nilai.
Parameter	Nilai yang mengikuti acuan keterangan atau informasi yang dapat menjelaskan

	batas-batas tertentu dari suatu suatu sistem persamaan.
Pruning	Teknik dalam machine learning yang mengurangi ukuran pohon keputusan dengan menghapus bagian dari pohon yang memberikan sedikit kekuatan untuk mengklasifikasikan kasus.
RapidMiner	Sebuah tool yang digunakan untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi.
Repositori	Kumpulan paket yang siap untuk diambil dan digunakan sesuai dengan kebutuhan pengguna.
Simpul akar	Simpul tanpa ayah yang berada pada tingkat tertinggi.
Simpul daun	Semua simpul yang berada pada tingkat terendah.
Simpul internal	Semua simpul dari pohon yang memiliki anak tetapi bukan daun.
Support	(Market Basket Analysis) Bagian transaksi yang mengandung kedua X dan Y**.**
Teori graf	Cabang kajian yang mempelajari sifat-sifat graf.
Validasi	Tindakan yang membuktikan bahwa suatu proses/metode dapat memberikan hasil yang konsisten sesuai dengan spesifikasi yang telah ditetapkan.

Daftar Pustaka

Akhtar, Fareed dan Caroline Hahne. 2012. *RapidMiner 5 Operator Reference*, [online], (www.rapid-i.com, diakses tanggal 30 Januari 2013).

Amiruddin, dkk. *Penerapan Association Rule Mining Pada Data Nomor Unik Pendidik dan Tenaga Kependidikan Untuk Menemukan Pola Sertifikasi Guru*. Institut Teknologi Surabaya. Surabaya.

Basuki, Achmad dan Iwan Syarif. *Decision Tree*, [online], (http://lecturer.eepis-its.edu/~entin/Data%20Mining/Minggu%205%20Decision%20Tree.pdf, diakses tanggal 05 Februari 2013).

Khusnawi. 2007. *Pengantar Solusi Data Mining*. Yogyakarta.

Kusumadewi, Sri. 2003. *Artificial Intelligence: Teknik dan Aplikasinya*.

Mitchel, Tom M. 1997. *Machine Learning*. New York: McGraw-Hill.

Prasetyo, Bowo. 2011. *Mengenal RapidMiner*, [online], (www.slideshare.net/bowoprasetyo/RapidMiner, diakses tanggal 31 Januari 2013).

Prasetyo, Kokoh Philips. 2006. *APriori,* [online] (http://philips.wordpress.com/2006/06/07/apriori, diakses tanggal 03 Februari 2013)

----------. 2006. *Association Rule Mining,* [online]. (http://philips.wordpress.com/2006/05/10/association-rule-mining, diakses tanggal 03 Februari 2013).

Rafaida, Ropi. *Decision Tree (Pohon Keputusan)*, [online], (http://file.upi.edu/Direktori/FPEB/PRODI._MANAJEMEN_FPEB/197302052005012-ROFI_ROFAIDA/MATERI_KULIAH/DECISION_TREE.pdf, diakses tanggal 05 februari 2013).

Ross, Peter. 2000. Data Mining [online]. (http:// www.soc.napier.ac.uk/~peter/vldb/dm/dm.html, diakses tanggal 07 Februari 2013)

Wahono, Romi satria. *Data Mining:Proses Data Mining*, [online], (http://romisatriawahono.net/lecture/dm/ romi-dm-02-proses-june2012.pptx, diakses tanggal 31 Januari 2013).

2012. *RapidMiner 5.0 Manual English*, (online), (www.rapid-i.com, diakses tanggal 30 Januari 2013).

3 tips for Setting up Association Rules using RapidMiner, [online]. (http://www.simafore.com/blog /bid/110113/3-tips-for-setting-up-a-Market-Basket-Analysis-using-RapidMiner, diakses tanggal 08 Maret 2013).

Association Analysis: Basic Concepts and Algorithms, [online]. (http://www-users.cs.umn.edu/ ~kumar/dmbook/ch6.pdf, diakses tanggal 08 April 2013)

Decision Tree (Pohon Keputusan), [online], (http://www.google.co.id/url?sa=f&rct=j&url=http://novrina.staff.gunadarma.ac.id/Downloads/files/21783/Algoritma%2BC4.pdf&q=algoritma+c4&ei=6h9gUcbJFIqrrA

fT7IGQAw&usg=AFQjCNG7HbyNPOqa63Z-oPexX76TrIlJ7g, diakses tanggal 05 februari 2013).

Landasan Teori Market Basket Analysis, [online]. (http://library.binus.ac.id/eColls/eThesis/Bab2/2010-1-00498-MTIF%20Bab%202.pdf, diakses tanggal 08 April 2013)

Understanding the Concept of Market Basket Analysis, [online]. (http://www.thesmartcube.com/insights/blog/brand-strategy/understanding-the-concept-of-market-basket-analysis, diakses tanggal 08 Maret 2013)

RapidMiner Resources. (http://RapidMinerresources.com/uploads/videos/tomott/RapidMiner5-Vid1.flv, diakses tanggal 02 Februari 2013)

----------. (http://RapidMinerresources.com/uploads/videos/neural%20networks%201.flv, diakses tanggal 02 Februari 2013)

----------. (http://RapidMinerresources.com/uploads/videos/neural%20networks%202.flv, diakses tanggal 02 Februari 2013)