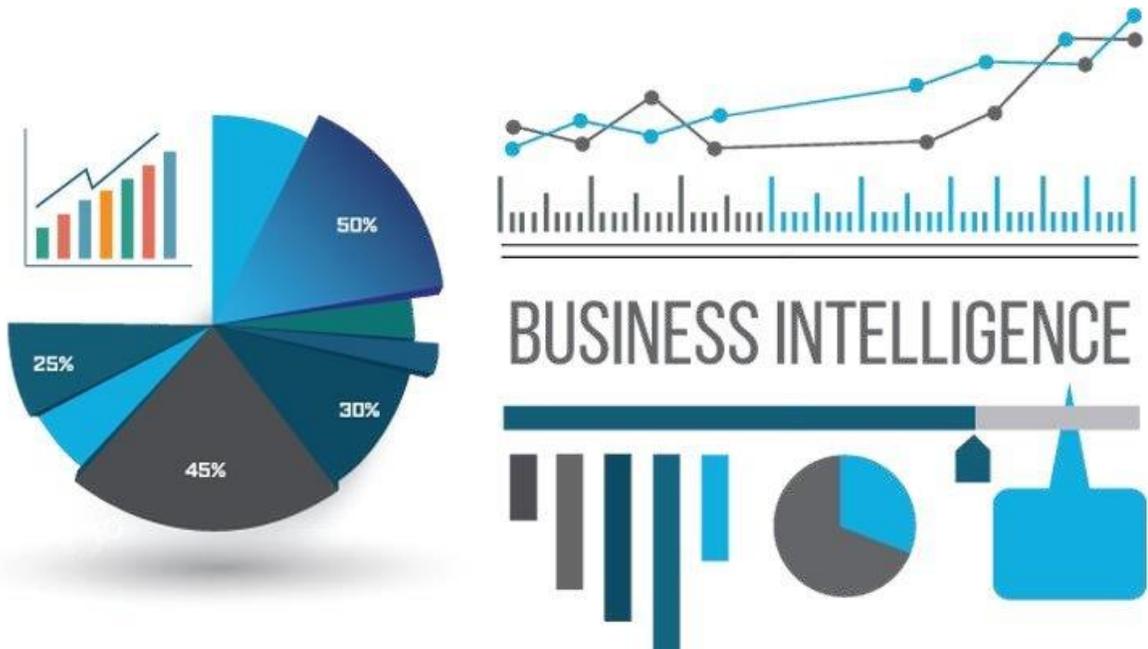


BUKU AJAR

Kecerdasan Bisnis



Pengarang:
Valentinus Roby Hananto

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

Heart & Mind Towards Excellence

KATA PENGANTAR

“Salah satu buku ajar kecerdasan bisnis terbaik yang pernah ada. Pokok bahasan dikupas secara tajam, sistematis, dan komprehensif. Bahasa yang digunakan mudah dipahami sehingga konsep kecerdasan bisnis cepat melekat di benak pembaca. Selain itu, contoh-contoh yang diberikan juga dapat dipraktikkan dengan mudah sehingga menuntun Anda menuju kesuksesan mata kuliah Kecerdasan Bisnis ini.” – **Julianto Lemantara, S.Kom., M.Eng., OCA., MCTS**

"Data is data, no more. Data would be useful when you turn it into information. Also, an analytical skill is required to create the reliable information and find the best solution. For me, that is a business intelligence. Learn and explore how Business Intelligence works through this book and you will receive the answer." - **Edo Yonatan Koentjoro, S.Kom., M.Sc.**

PRAKATA

Kecerdasan bisnis (*Business Intelligence*) adalah suatu ilmu yang mendukung pengambilan keputusan pada organisasi bisnis. Tujuan utamanya adalah menyediakan akses interaktif terhadap data, memungkinkan manipulasi data, dan memberi manajer kemampuan untuk melakukan analisis bisnis. Dengan kemampuan untuk melakukan analisis terhadap data, performa, dan situasi lingkungan bisnis, maka manajer dapat melakukan pengambilan keputusan yang lebih baik.

Dengan buku ajar kecerdasan bisnis ini, pembaca khususnya mahasiswa diharapkan dapat mengetahui dan memahami kecerdasan bisnis dengan lebih terkonsep. Buku ini membahas mengenai pengantar kecerdasan bisnis, *data warehousing*, *business analytics* dan OLAP, *data mining*, *business performance management*, serta *dashboard system*. Selain berisi tentang konsep dasar, di dalam buku ajar ini juga terdapat panduan praktik menggunakan beberapa aplikasi / *tools* serta latihan-latihan untuk memperkuat pemahaman konsep.

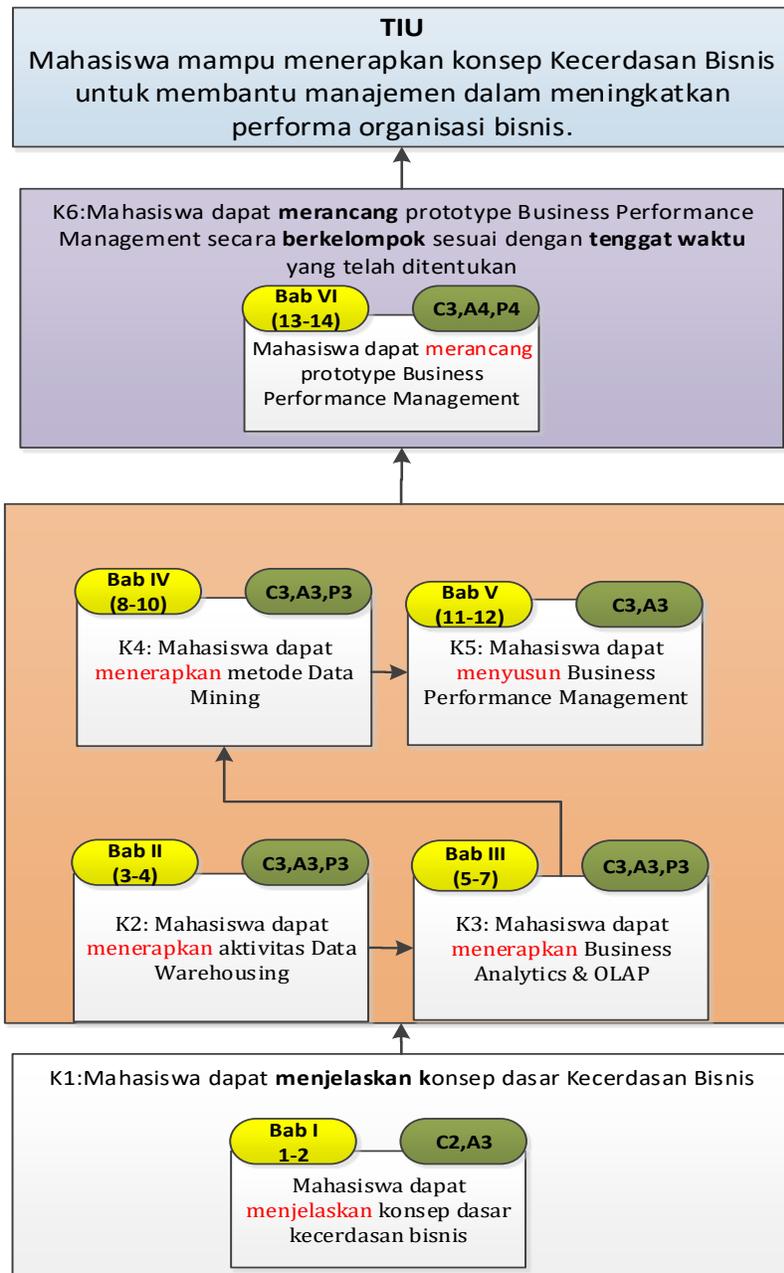
Penulis berharap buku ajar kecerdasan bisnis ini dapat menjadi salah satu sumber bacaan dan referensi baik bagi mahasiswa maupun dosen. Semoga para pembaca buku ini dapat menerapkan konsep, teknologi, metode, dan *tools* terkait kecerdasan bisnis dalam membantu organisasi bisnis menghadapi tantangan yang ada saat ini.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak atas dukungannya kepada penulis selama proses pembuatan buku ajar ini. Penulis terbuka apabila ada saran, masukan, maupun kritik yang membangun dari pembaca untuk perbaikan buku ajar ini di masa mendatang.

Surabaya, September 2017

Valentinus Roby Hananto
valentinus@stikom.edu

PETA CAPAIAN BELAJAR



Entry Behaviour:
Mahasiswa memahami desain basis data dan kualitas data

DAFTAR ISI

KATA PENGANTAR I

Prakata	ii
Peta Capaian Belajar	iii
Daftar Isi	iv
Daftar Gambar	vii
Daftar Tabel	xii
BAB I PENGANTAR KECERDASAN BISNIS	1
1.1. LINGKUNGAN BISNIS DAN ORGANISASI	2
1.2. KONSEP KECERDASAN BISNIS (BI)	4
1.3. KAPABILITAS BI	5
1.4. ARSITEKTUR DAN KOMPONEN BI	9
1.5. MANFAAT BI DALAM ORGANISASI BISNIS	13
1.6. PROSES TRANSAKSIONAL VS <i>ANALYTICAL</i>	13
1.7. IMPLEMENTASI BI	14
1.8. ISU-ISU TERKAIT KESUKSESAN BI	15
1.9. TEKNIK DAN APLIKASI BI	16
1.10. RANGKUMAN	17
1.11. LATIHAN SOAL	17
BAB II DATA WAREHOUSING	18
2.1. DEFINISI <i>DATA WAREHOUSE</i>	19
2.2. KARAKTERISTIK <i>DATA WAREHOUSE</i>	19
2.3. <i>FRAMEWORK DATA WAREHOUSE</i>	20
2.4. ARSITEKTUR <i>DATA WAREHOUSE</i>	22
2.5. INTEGRASI DATA	24
2.6. PROSES ETL (<i>EXTRACTION, TRANSFORMATION, LOAD</i>)	25
2.7. PENGEMBANGAN <i>DATA WAREHOUSE</i>	26
2.8. <i>MULTIDIMENSIONALITY</i>	27
2.9. SQL SERVER INTEGRATION SERVICES (SSIS)	28
2.10. RANGKUMAN	48
2.11. LATIHAN SOAL	49

BAB III BUSINESS ANALYTICS DAN OLAP	50
3.1. KONSEP <i>BUSINESS ANALYTICS</i>	51
3.2. KONSEP OLAP	52
3.3. OLAP VS OLTP	52
3.4. FITUR OLAP.....	53
3.5. <i>CUBE</i> DAN DIMENSI	55
3.6. VISUALISASI DATA.....	58
3.7. <i>CUSTOMER ANALYTICS</i>	60
3.8. <i>REAL TIME BI</i>	61
3.9. SQL SERVER ANALYSIS SERVICES (SSAS)	62
3.10. RANGKUMAN.....	93
3.11. LATIHAN SOAL	93
BAB IV DATA MINING.....	95
4.1. DEFINISI <i>DATA MINING</i>	96
4.2. KARAKTERISTIK <i>DATA MINING</i>	97
4.3. TIPE DATA	98
4.4. PENGELOMPOKAN <i>DATA MINING</i>	100
4.5. PENERAPAN <i>DATA MINING</i>	102
4.6. TAHAPAN METODOLOGI <i>DATA MINING</i>	103
4.7. METODE <i>DATA MINING</i>	109
4.8. <i>DATA MINING SOFTWARE</i>	116
4.9. PENGENALAN WEKA.....	117
4.10. WEKA <i>EXPLORER</i>	117
4.11. EVALUASI <i>DATA MINING</i>	120
4.13. RANGKUMAN.....	122
4.14. LATIHAN SOAL	123
BAB V BUSINESS PERFORMANCE MANAGEMENT	124
5.1. DEFINISI BPM.....	125
5.2. SIKLUS PROSES BPM	125
5.3. PENGUKURAN PERFORMA.....	127
5.4. METODOLOGI BPM	130
5.5. <i>BALANCED SCORECARDS</i>	130
5.6. <i>PERFORMANCE DASHBOARDS</i>	132
5.7. ARSITEKTUR BPM.....	133
5.8. RANGKUMAN.....	135
5.9. LATIHAN SOAL.....	135

BAB VI DASHBOARD SYSTEM	136
6.1. PENGERTIAN <i>DASHBOARD</i>	137
6.2. MANFAAT <i>DASHBOARD</i>	138
6.3. TIPE-TIPE <i>DASHBOARD</i>	138
6.4. PEMBUATAN <i>DASHBOARD</i> PADA EXCEL.....	139
6.5. RANGKUMAN.....	142
6.6. LATIHAN SOAL.....	143
DAFTAR PUSTAKA.....	144
INDEKS	145

DAFTAR GAMBAR

Gambar 1.1. Faktor pengubah lingkungan bisnis dan respon organisasi (Sumber : Turban, 2008).....	3
Gambar 1.2. Kapabilitas BI	5
Gambar 1.3. Tahapan Evolusi BI.....	9
Gambar 1.4. Arsitektur BI	10
Gambar 1.5. Arsitektur <i>Data Warehouse</i>	11
Gambar 1.6. <i>Business Analytics</i>	11
Gambar 1.7. <i>Business Performance Management</i>	12
Gambar 1.8. <i>Dashboard System</i>	12
Gambar 2.1. Karakteristik DW	19
Gambar 2.2. <i>Framework Data Warehouse</i>	21
Gambar 2.3. Arsitektur 3-tier.....	22
Gambar 2.4. Arsitektur 2-tier.....	22
Gambar 2.5. Arsitektur DW berbasis Web	23
Gambar 2.6. Proses ETL.....	26
Gambar 2.7. Skema <i>star</i> dan <i>snowflake</i>	28
Gambar 2.8. Membuka SQL Server Business Intelligence Development Studio.....	32
Gambar 2.9. Membuat proyek baru	33
Gambar 2.10. Tampilan Proyek SSIS Baru	33
Gambar 2.11. Mengganti nama SSIS Package	34
Gambar 2.12. Konfirmasi penggantian nama objek	34
Gambar 2.13. Menambahkan data source.....	35
Gambar 2.14. Tampilan awal Wizard Data Source	35
Gambar 2.15. Mendefinisikan koneksi	35
Gambar 2.16. Membuat koneksi dengan Connection Manager	36
Gambar 2.17. Memilih koneksi	36
Gambar 2.18. Mengisi nama Data Source	37

Gambar 2.19. Menambahkan <i>data source view</i>	37
Gambar 2.20. Tampilan awal wizard <i>data source view</i>	38
Gambar 2.21. Memilih <i>data source</i>	38
Gambar 2.22. Memilih objek yang akan dibuat pemandangan logik	39
Gambar 2.23. Mengeset nama <i>data source view</i>	39
Gambar 2.24. Menambahkan koneksi ke <i>data source</i>	40
Gambar 2.25. Memilih koneksi	40
Gambar 2.26. Kakas peralatan pada <i>control flow</i>	41
Gambar 2.27. Konfigurasi <i>Send Mail Task</i>	42
Gambar 2.28. Tab <i>Data Flow</i>	42
Gambar 2.29. <i>Data Flow Sources</i> dan <i>Data Destinations</i>	43
Gambar 2.30. Konfigurasi <i>Data Source</i> dan <i>Data Destination</i>	44
Gambar 2.31. Eksekusi <i>SSIS package</i>	44
Gambar 2.32. Pilihan <i>Data Flow Transformation</i>	45
Gambar 2.34. Mengatur transformasi <i>Derived Column</i>	46
Gambar 2.35. Menambahkan transformasi pada <i>Data Flow</i>	47
Gambar 2.36. <i>Progress</i> tab saat eksekusi <i>package</i>	48
Gambar 3.1. Klasifikasi BA.....	51
Gambar 3.2. Query dan hasilnya dari <i>sales by product</i>	52
Gambar 3.3. Fitur <i>drill down</i>	54
Gambar 3.4. <i>Slicing operations</i> pada sebuah <i>data cube</i> tiga dimensi.....	54
Gambar 3.5. Analisis <i>cube</i>	56
Gambar 3.6 <i>Cube Internet Sales AdventureWorksDW</i>	56
Gambar 3.7. Dimensi <i>Product, Balanced Hierarchy</i>	58
Gambar 3.8 Dimensi Lokasi, <i>Unbalanced Hierarchy</i>	58
Gambar 3.9. Visualisasi Data	59
Gambar 3.10, Sistem Informasi Geografis	60
Gambar 3.11. Membuat <i>Analysis Services Project</i> baru	63

Gambar 3.12 Memilih menu untuk membuat datasource baru	64
Gambar 3.13 Mendefenisikan Datasource	64
Gambar 3.14. Membuat Koneksi Datasource baru	65
Gambar 3.15. memilih service account impersonation	66
Gambar 3.16. Menu untuk membuat datasource view	67
Gambar 3.17. Memilih table dan view dari DSV	68
Gambar 3.18. Datasource View Designer	68
Gambar 3.19. Memilih metode pembuatan cube	69
Gambar 3.20. Memilih Fact table	70
Gambar 3.21. Memilih Measure	71
Gambar 3.22. Memilih dimensi, Attribute dan Hierarchy	72
Gambar 3.23. View Solution Explorer dengan cube dan dimension	72
Gambar 3.24. Dimension Designer	73
Gambar 3.25. Hasil penambahan <i>attribute</i>	74
Gambar 3.26. Membuat hierarchy	75
Gambar 3.27. mendefenisikan attribute relationship	76
Gambar 3.28. Perubahan attribute relationship	76
Gambar 3.29. Struktur attribute relationship	77
Gambar 3.30. Best practice warning	78
Gambar 3.31. Memilih key column pada dimensi	79
Gambar 3.32. Memilih name column	80
Gambar 3.33. Hasil perubahan hierarcy	81
Gambar 3.34. Penambahan attribute MonthNumberofYear	81
Gambar 3.35. Menentukan attribute relationship untuk attribute month	82
Gambar 3.36. Merubah properties untuk fungsi sortin	82
Gambar 3.37. Dimensi Product Designer	82
Gambar 3.38. Deploy analysis services project	83
Gambar 3.39. Struktur Cube	84

Gambar 3.40. Cube Browser	85
Gambar 3.41. Tampilan data cube	86
Gambar 3.42. Sales Amount by Product Color.....	86
Gambar 3.43. Edit DimCustomer	87
Gambar 3.44. Hierachy Customer Hierarchy	88
Gambar 3.45. Attribute relationship dimensi customer	88
Gambar 3.46. Tampilan data dengan hierarchy customer geography.....	88
Gambar 3.47. Filtering by customer birthdate	89
Gambar 3.48. Tampilan data customer dengan filter birthdate.....	89
Gambar 3.49. Tampilan data customer dengan filter birthdate product.....	90
Gambar 3.50. Tampilan filter column	91
Gambar 3.51. Menu Koneksi ke cube dari excel	91
Gambar 3.52. Excel Cube Browsing	92
Gambar 3.53. Tampilan Data Cuba pada Excel.....	93
Gambar 4.1. Disiplin ilmu yang menunjang Data Mining.....	97
Gambar 4.2. Taksonomi tipe data.....	99
Gambar 4.3. Klasifikasi <i>Data Mining</i>	100
Gambar 4.4. Metodologi CRISP-DM.....	104
Gambar 4.5. Persiapan data	106
Gambar 4.6. Metodologi SEMMA	109
Gambar 4.7. <i>Decision tree</i> untuk <i>dataset</i> cuaca	113
Gambar 4.8. Probabilitas untuk dataset cuaca	115
Gambar 4.9. Menu awal pada Weka.....	117
Gambar 4.10. Weka Explorer	118
Gambar 4.11. Dataset cuaca untuk Weka	119
Gambar 4.12. Membuka dataset pada Weka	119
Gambar 4.13. Nilai atribut <i>outlook</i>	120
Gambar 4.14. <i>Confusion Matrix</i>	121

Gambar 4.15. Pengukuran akurasi klasifikasi.....	122
Gambar 5.1. Siklus proses BPM.....	126
Gambar 5.2. Perspektif Balanced Scorecard.....	131
Gambar 5.3. <i>Strategy map</i>	132
Gambar 5.4. <i>Performance Dashboard</i>	133
Gambar 5.5. Arsitektur BPM.....	134
Gambar 6.1. Pembuatan <i>sparkline</i>	140
Gambar 6.2. Pembuatan <i>data bars</i>	141

DAFTAR TABEL

Tabel 1.1. Tools BI.....	16
Tabel 4.1. Dataset cuaca	112
Tabel 5.1. Hasil survei NPS.....	129

BAB I

PENGANTAR KECERDASAN BISNIS



Kompetensi

Mahasiswa dapat menjelaskan konsep dasar Kecerdasan Bisnis (Minggu 1 dan 2)

Bahan Kajian

Bab ini membahas materi-materi sebagai berikut:

- Lingkungan Bisnis dan Organisasi
- Konsep Kecerdasan Bisnis / *Business Intelligence* (BI)
- Kapabilitas BI
- Arsitektur dan Komponen BI
- Manfaat BI dalam organisasi bisnis
- Proses Transaksional vs *Analytical*
- Implementasi BI
- Isu-isu terkait kesuksesan BI
- Teknik dan aplikasi BI

1.1. Lingkungan Bisnis dan Organisasi

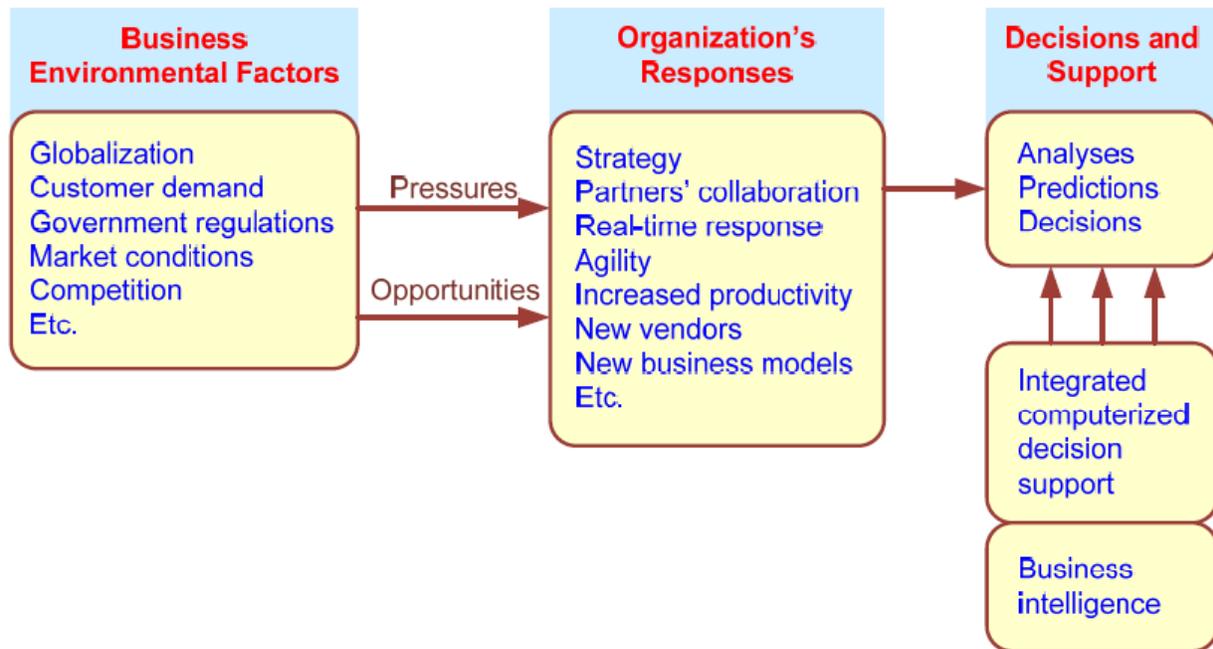
Lingkungan dari organisasi bisnis sudah mengalami banyak perubahan di abad ke-21. Dinamika perubahan ini mau tidak mau memaksa organisasi bisnis untuk beradaptasi agar bisa mempertahankan kelangsungan organisasinya. Ada banyak faktor yang mempengaruhi perubahan lingkungan bisnis organisasi. Globalisasi merupakan salah satu faktor penyebab perubahan ini. Globalisasi terjadi ketika perbedaan jarak dan waktu di dunia ini menjadi semakin kabur, sehingga hampir tidak ada batas yang membedakan antar negara. Perkembangan teknologi informasi dan telekomunikasi menjadi pemicunya. Dengan dukungan internet, kita dapat berkomunikasi dan bertukar informasi secara cepat, mudah, dan murah.

Dinamika inilah yang juga ikut merubah kondisi pasar saat ini. Kompetisi antar perusahaan dalam suatu industri menjadi semakin ketat. Kemudahan dalam mengakses informasi menuntut setiap perusahaan agar lebih kreatif dan dinamis, sehingga model bisnis mereka tidak mudah ditiru oleh pesaingnya. Setiap perusahaan juga berupaya agar memiliki keunggulan kompetitif, sehingga mereka bisa memenangkan persaingan untuk meraih pangsa pasar. Salah satunya dengan penerapan *e-business* ataupun *e-commerce*. Dengan dukungan teknologi informasi, transaksi jual beli dapat dilakukan secara *real-time* dan perusahaan bisa mendapatkan pasar yang lebih luas.

Selain itu perubahan juga terjadi pada sisi pelanggan. Pelanggan saat ini memiliki posisi tawar yang lebih kuat dibandingkan dulu. Dengan kemudahan untuk mengakses informasi dan semakin banyaknya opsi yang ada di pasar, loyalitas yang dimiliki terhadap satu produk menjadi semakin berkurang. Pelanggan bisa beralih dengan mudah ke produk lain yang menawarkan nilai lebih, baik dari segi kualitas, harga, maupun pelayanan. Pelanggan juga cenderung menjadi lebih “cerewet” saat ini. Mereka selalu menuntut adanya kustomisasi dari produk/jasa yang ditawarkan dan selalu membanding-bandingkan dengan produk lain.

Berbagai faktor perubahan lingkungan bisnis tadi, seperti dapat dilihat pada gambar 1.1, mau tidak mau menjadi perhatian bagi perusahaan bisnis. Perubahan ini bisa dilihat dari dua perspektif, sebagai tekanan (*pressures*) ataupun sebagai kesempatan (*opportunities*). Perubahan pada lingkungan bisnis dapat menjadi tekanan, karena perusahaan dituntut untuk beradaptasi agar bisa tetap bertahan dalam industri bisnis. Apabila mereka kurang tanggap atau terlambat dalam merespon perubahan yang ada, kelangsungan bisnis mereka menjadi terancam. Dalam industri telepon seluler/*smartphone*, kita bisa melihat bagaimana Nokia yang dulunya merupakan *market*

leader sekarang kehilangan dominasinya. Nokia terlambat dalam menyikapi perubahan yang terjadi di pasar, di mana orang sudah mulai beralih ke sistem operasi Symbian dan iOS yang sangat cepat perkembangannya. Pada masa transisi itu, Nokia justru masih tetap bertahan dengan sistem operasi Symbian-nya, yang pada akhirnya membuat mereka perlahan-lahan kalah bersaing dengan kompetitornya.



Gambar 1.1. Faktor pengubah lingkungan bisnis dan respon organisasi
(Sumber : Turban, 2008)

Di sisi lain, perubahan lingkungan bisnis juga bisa menjadi kesempatan (*opportunities*) untuk berinovasi dan mendapatkan pasar. Dalam industri *smartphone*, kita bisa melihat sekarang ini banyak “pemain” baru yang berhasil merusak tatanan pasar, sehingga tidak lagi didominasi oleh pemain-pemain lama. Peluang untuk meraih pasar akan terbuka lebar bagi perusahaan yang bisa merespon dan menyesuaikan strateginya dengan perubahan-perubahan yang ada pada lingkungan bisnis.

Oleh sebab itu, perusahaan sekarang ini harus bisa cepat merespon setiap perubahan yang terjadi pada lingkungan bisnis. Manajemen sebagai pengambil keputusan harus bisa merespon perubahan tersebut dengan reaktif, antisipatif, adaptif, dan proaktif. Mereka tidak boleh hanya bersikap pasif dan pada akhirnya telat untuk bereaksi atau mengambil keputusan. Kita bisa menganalogikan suatu perusahaan layaknya tubuh manusia. Sebagai manusia, apabila kita sudah

mulai merasa tidak enak badan dan terdapat gejala-gejala awal sakit, maka kita segera melakukan tindakan preventif supaya tidak sampai jatuh sakit. Contohnya yaitu dengan mengonsumsi makanan yang bergizi, multivitamin, serta dengan istirahat yang cukup. Kita selalu berusaha agar tidak sampai jatuh sakit, karena dampaknya lebih berat dan biaya pengobatannya juga lebih tinggi. Demikian pula pada suatu organisasi, mereka harus bisa melihat gejala-gejala atau perubahan yang terjadi pada lingkungan bisnisnya. Dengan demikian, manajemen bisa segera memberikan respon dengan lebih baik. Beberapa respon/tindakan yang bisa diambil oleh manajemen di antaranya:

- Membuat perencanaan strategis.
- Menggunakan bisnis model terbaru dan inovatif.
- Memperbaiki proses bisnis.
- Berpartisipasi dalam kerjasama bisnis.
- Memperbaiki sistem informasi perusahaan.
- Memperbaiki relasi kerjasama.
- Mendorong inovasi dan kreativitas.
- Meningkatkan layanan dan relasi dengan pelanggan
- Beralih ke *e-commerce*
- Beralih ke skema produksi *make-to-order*
- Mengimplementasikan TI untuk meningkatkan komunikasi, akses data (pemanfaatan informasi), dan kolaborasi
- Respon cepat terhadap kompetitor (dalam harga, promosi, produk dan jasa baru)
- Mengotomasi banyak aktivitas
- Mengotomasi pengambilan keputusan tertentu
- Meningkatkan pengambilan keputusan dengan analitik

1.2. Konsep Kecerdasan Bisnis (BI)

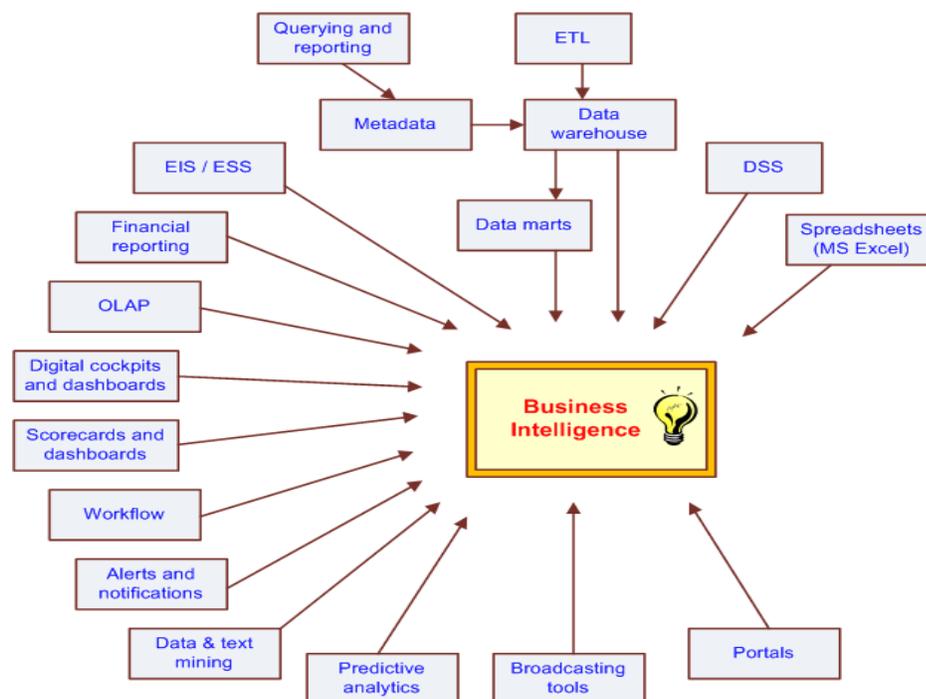
Kemampuan perusahaan untuk dapat bersaing dalam kompetisi tentunya sangat dipengaruhi oleh respon mereka dalam pengambilan keputusan. Pengambilan keputusan yang baik memerlukan informasi pendukung terkait kondisi internal organisasi maupun kondisi eksternal

lingkungan bisnis. Mengingat kondisi lingkungan bisnis yang semakin dinamis dan sangat cepat berubah, sistem pendukung ini haruslah bisa dengan cepat menyampaikan perubahan yang terjadi.

Kecerdasan Bisnis, atau dikenal secara luas dengan sebutan *Business Intelligence* (BI), pada dasarnya merupakan evolusi dari sistem pendukung keputusan. BI memiliki tujuan utama untuk menyediakan akses interaktif terhadap data, memungkinkan manipulasi data, dan memberi manajer kemampuan untuk melakukan analisis bisnis. Dengan kemampuan untuk melakukan analisis terhadap data, performa, dan situasi lingkungan bisnis, maka manajer dapat melakukan pengambilan keputusan yang lebih baik. Secara umum, BI adalah sebuah payung istilah yang mengkombinasikan berbagai arsitektur, teknik, *analytical tools*, aplikasi, dan metodologi yang memungkinkan kemudahan akses terhadap data untuk membantu manajer melakukan analisis bisnis. BI membantu dalam transformasi data, menjadi informasi (serta pengetahuan), kemudian menjadi keputusan dan akhirnya menjadi tindakan.

1.3. Kapabilitas BI

BI merupakan sebuah payung istilah yang membawahi berbagai kapabilitas, seperti dapat dilihat pada gambar 1.2.



Gambar 1.2. Kapabilitas BI

Spreadsheet merupakan alat yang dapat digunakan dengan mudah dan sederhana untuk menyusun laporan. Dengan menggunakan *spreadsheet*, kita dapat mengurutkan data maupun membuat grafik dengan mudah. Mengurutkan data dan membuat grafik merupakan fitur yang dapat membantu manajer dalam pembuatan keputusan. Namun, di balik kemudahan maupun kesedarhanaannya, *spreadsheet* memiliki beberapa kelemahan yaitu tidak efisien (terlalu lama dalam penyusunan laporan) dan memiliki resiko kesalahan yang cukup tinggi. Hal ini dikarenakan pengguna secara langsung dapat menginputkan data ke dalam lembar kerja.

Decision Support System (DSS) bisa diterjemahkan sebagai Sistem Pendukung Keputusan atau yang lebih dikenal dengan SPK. SPK merupakan sistem yang terkomputerisasi dalam mengelola dan menganalisis data dengan menggunakan model, tujuannya untuk meyakinkan (memperkuat) pemilihan keputusan.

Extract, Transform, Load (ETL) merupakan proses mengambil (*extract*) data dari berbagai sumber dan mengubah (*transform*) data tersebut menjadi format data yang dibutuhkan. Data yang telah diubah tersebut kemudian dimasukkan (*load*) ke dalam pusat data (*Data Warehouse*). *Extract* merupakan proses menghimpun data dari *database*. *Transform* merupakan proses mengkonversi data tersebut dengan cara menggabungkan beberapa data dengan dimensi yang sama, diukur menjadi data yang dibutuhkan (format) yang sesuai. Sebagai contoh, untuk mendapatkan data yang berkualitas perlu dilakukan *cleaning* (misalnya menyeragamkan format jenis kelamin dari “pria/laki-laki/*male*” menjadi “P”, dan dari “wanita/perempuan/*female*” menjadi “W”). *Load* merupakan proses menyimpan data yang telah di-konversi ke dalam database.

Data Warehouse (gudang data) yaitu tempat utama (pusat) yang digunakan untuk menyimpan semua data perusahaan yang berguna untuk mendukung keputusan. Data yang disimpan dalam *Data Warehouse* merupakan data penting dan konfidensial bagi perusahaan. Data tersebut dihimpun dari berbagai sumber internal dan eksternal dengan berbagai format. *Data Warehouse* digunakan untuk mendapatkan informasi yang membantu manajer dalam mengambil keputusan (menyelesaikan masalah).

Data Mart dan *Data Warehouse* sebenarnya memiliki fungsi yang serupa yaitu menyimpan data, namun *Data Mart* merupakan bagian kecil dari *Data Warehouse*. Sekumpulan data yang dibutuhkan oleh pengguna disimpan dalam *Data Mart*. Jika *Data Warehouse* menyimpan seluruh data perusahaan, maka *Data Mart* hanya menyimpan sekumpulan data pada suatu fungsi/divisi. Tujuan dari *Data Mart* adalah untuk memenuhi kebutuhan informasi dari sekelompok pengguna

tertentu. Sebagai contoh, untuk manajer *Human Resource Development* (HRD), data yang diperlukan adalah data yang berhubungan dengan HRD, misalnya absensi pegawai, performa pegawai, dan lain sebagainya.

Metadata merupakan sekumpulan data (informasi) mengenai data tersebut. Metadata dapat dikatakan sebagai gambaran dari data itu sendiri. Sebagai contoh, pada *file* (dokumen) MS Word jika diperhatikan lebih dapat diketahui jenis *file*, format yang digunakan, kapan dibuat, kapan *file* tersebut diubah, siapa pemilik *file* tersebut, dan lain-lain.

Query and Reporting merupakan suatu formula (kode) yang digunakan untuk memproses data dari *database* dan menampilkan informasi yang dibutuhkan kepada pengguna (manajer). Dengan bantuan perangkat lunak, memproses data menjadi informasi dapat dilakukan dengan lebih cepat dan akurat.

Executive Information System (EIS) merupakan sistem pendukung keputusan (SPK) yang digunakan untuk manajer dalam memilih keputusan. EIS menampilkan informasi penting yang mencakup keseluruhan aktivitas (bisnis) di perusahaan, namun EIS menyediakan informasi yang mudah dipahami oleh pengguna (manajer). Biasanya EIS menampilkan informasi tersebut dalam bentuk yang lebih ringkas seperti grafik.

Financial Reporting (Pelaporan Keuangan) merupakan proses menyampaikan (presentasi) laporan keuangan (*Financial Statement*). *Financial reporting* tidak hanya berupa *cash flow*, laporan laba-rugi, tetapi juga berisi rencana perkiraan manajemen di masa depan. Dengan mengkomunikasikan isi laporan keuangan, manfaat yang didapat menjadi lebih banyak.

Online Analytical Processing (OLAP) adalah proses untuk menganalisis data dengan sudut pandang yang berbeda (multidimensi). Dengan menggunakan OLAP informasi yang didapatkan menjadi lebih luas dan beragam. Misalnya dengan membandingkan data penjualan per bulan (dimensi waktu), manajer mendapatkan informasi tentang bulan dengan penjualan tertinggi. Dengan demikian manajer dapat mendapatkan pemahaman (informasi) yang lebih luas.

Dashboard merupakan suatu alat yang digunakan untuk menampilkan seluruh informasi singkat mengenai kegiatan (performa) perusahaan dengan menggunakan visualisasi atau grafik yang menyerupai *dashboard* pada kendaraan (*digital cockpit*). Informasi yang ditampilkan pada *dashboard* merupakan data yang telah diperoleh dan disimpan pada *data warehouse*, dan data tersebut telah diproses sedemikian rupa. Informasi yang ditampilkan disesuaikan dengan

kebutuhan dari pengguna, misalnya bagian/divisi penjualan bisa mendapatkan informasi tentang penjualan.

Scorecard Dashboard memiliki cara kerja dan fungsi yang hampir sama dengan *Digital Cockpit Dashboard*. Namun, *Scorecard Dashboard* memiliki fungsi lain untuk menampilkan informasi mengenai perkembangan terhadap tujuan strategis dari perusahaan (target). *Scorecard Dashboard* menggunakan teori pendekatan *Balanced Scorecard*. Sama halnya dengan *Digital Cockpit Dashboard*, *Scorecard Dashboard* menampilkan informasi secara *real time* dengan menggunakan grafik.

Workflow (alur kerja) merupakan serangkaian langkah-langkah yang diperlukan yang melibatkan beberapa orang dalam menyelesaikan tugas perusahaan. *Workflow* menggambarkan alur kerja yang berurutan, dan setiap langkah juga saling berinteraksi (berkesinambungan/berhubungan) dengan langkah lainnya. *Workflow* digambarkan dalam bentuk *flowchart* yang berguna untuk menggambarkan langkah langkah dalam alur kerja.

Alert and Notification merupakan pesan peringatan (pemberitahuan) yang dikirim kepada manajer, yang menunjukkan adanya kondisi atau kejadian yang penting (salah), sehingga manajer dapat mengambil keputusan dengan cepat. Sebagian besar manajer tidak selalu memantau informasi kondisi perusahaan terus menerus, akibatnya para manajer mungkin melewatkan sebuah informasi penting (masalah), sehingga keputusan (tindakan) untuk menyelesaikan masalah tersebut tidak dapat segera diambil dengan cepat. Dengan menggunakan fasilitas *alert*, para manajer tidak akan melewatkan informasi penting, dengan demikian tindakan dapat segera diputuskan.

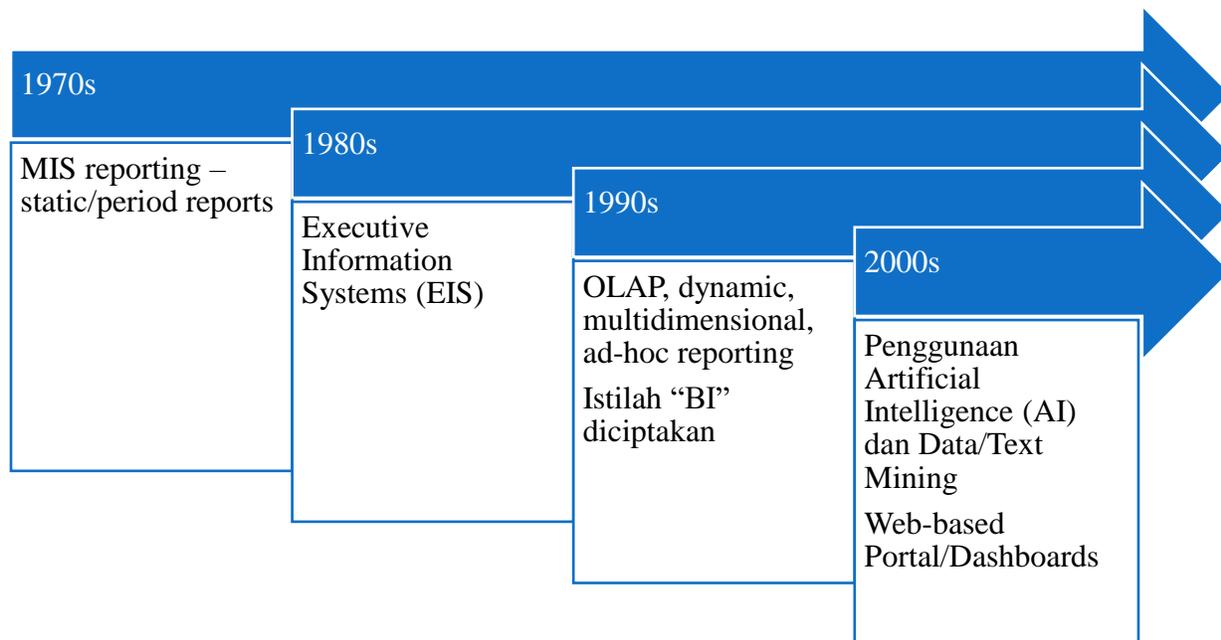
Data Mining merupakan proses mendapatkan informasi baru yang didapatkan dari pengumpulan data, kemudian data tersebut diproses dengan cara meringkas dan menggabungkannya dengan data lainnya. *Data Mining* dapat pula mencari suatu tren maupun pola dalam data. Contohnya, sebuah *supermarket* bisa mendapatkan informasi baru bahwa penjualan item X sering diikuti oleh item Y.

Predictive Analytics merupakan proses menganalisis pola pada data yang telah didapatkan. Dengan menganalisis data dengan membandingkan atau mencari hubungan antara berbagai jenis data, kita dapat memperoleh informasi mengenai resiko tindakan yang akan diambil dan peluang tindakan tersebut. *Predictive analytics* dapat juga meramalkan kejadian di masa yang akan datang, berdasarkan data yang diperoleh.

Broadcasting Tools merupakan alat yang digunakan untuk menyebarkan (membagikan) informasi kepada pengguna. **Broadcasting Tools** juga berfungsi untuk meningkatkan kinerja dalam pelaporan.

Portal merupakan wadah yang diperuntukkan menampung seluruh laporan di perusahaan dengan menggunakan sistem. **Portal** menampilkan seluruh laporan, dan informasinya disajikan dalam bentuk tabel maupun grafik.

Kapabilitas BI yang beragam sebenarnya dilatarbelakangi sejarah dari BI yang mengalami evolusi dari waktu ke waktu. Istilah BI sendiri baru muncul pada tahun 1990-an, sebelumnya masih berupa Management Information System (MIS) dan Executive Information System (EIS). Tahapan evolusi BI ini dapat dilihat pada gambar 1.3.



Gambar 1.3. Tahapan Evolusi BI

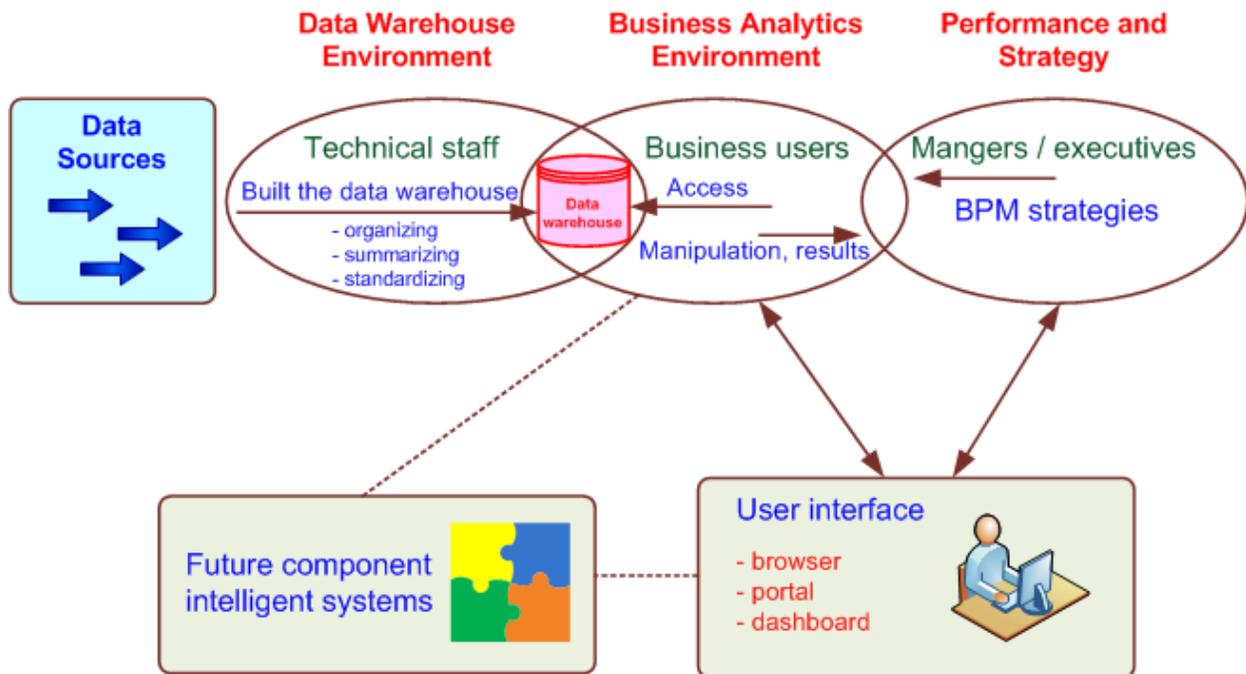
1.4. Arsitektur dan Komponen BI

Sistem BI terdiri dari 4 komponen utama, yaitu:

- **Data Warehouse**, yang merupakan sumber data dari BI
- **Business Analytics**, dengan kumpulan *tools* untuk manipulasi, *mining*, dan menganalisis data dari *data warehouse*;

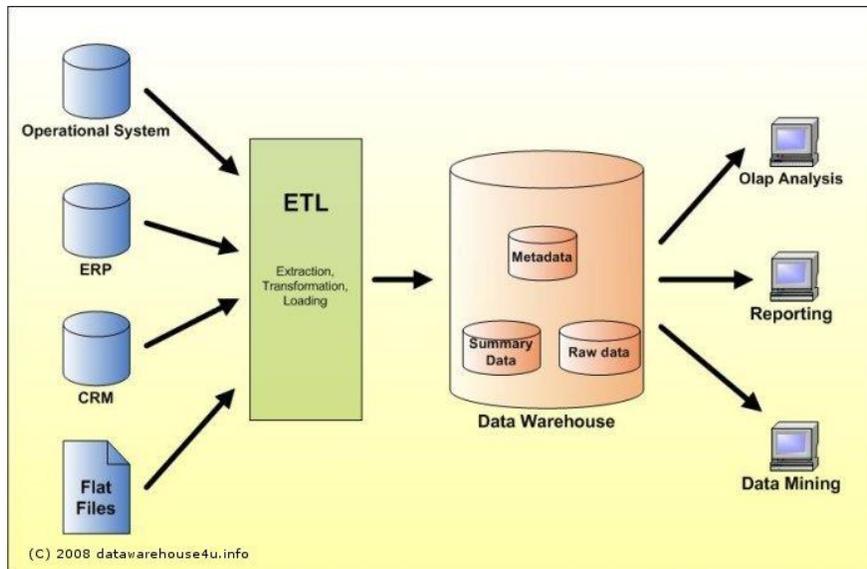
- **Business Performance Management (BPM)**, digunakan untuk monitoring dan menganalisis performa bisnis
- **User interface**, untuk menyampaikan informasi pada pengguna (contoh: *dashboard*)

Arsitektur dari sistem BI dapat dilihat pada gambar 1.4.



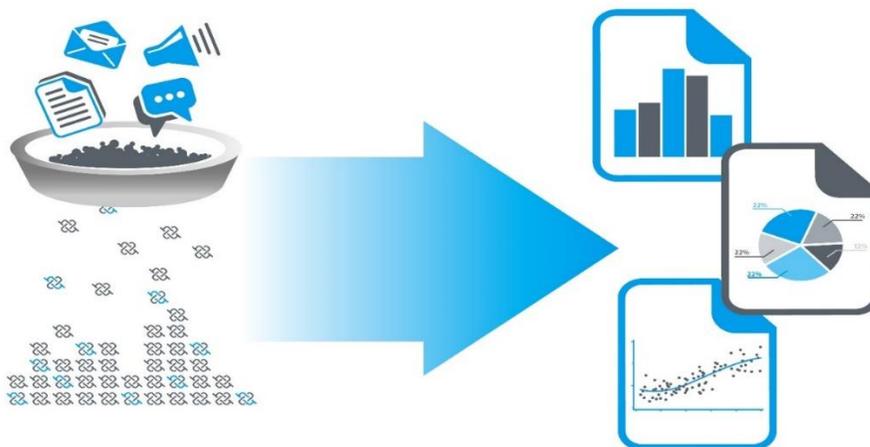
Gambar 1.4. Arsitektur BI

Data warehouse adalah landasan dari setiap sistem BI berukuran sedang maupun besar. Awalnya, *data warehouse* hanya berisi data historis yang sudah terorganisasi dan diringkas, sehingga pengguna dengan mudah dapat melihat dan memanfaatkannya. Sekarang beberapa data warehouse menyertakan akses ke data saat ini juga, sehingga dapat menjadi pendukung keputusan yang *real-time*.



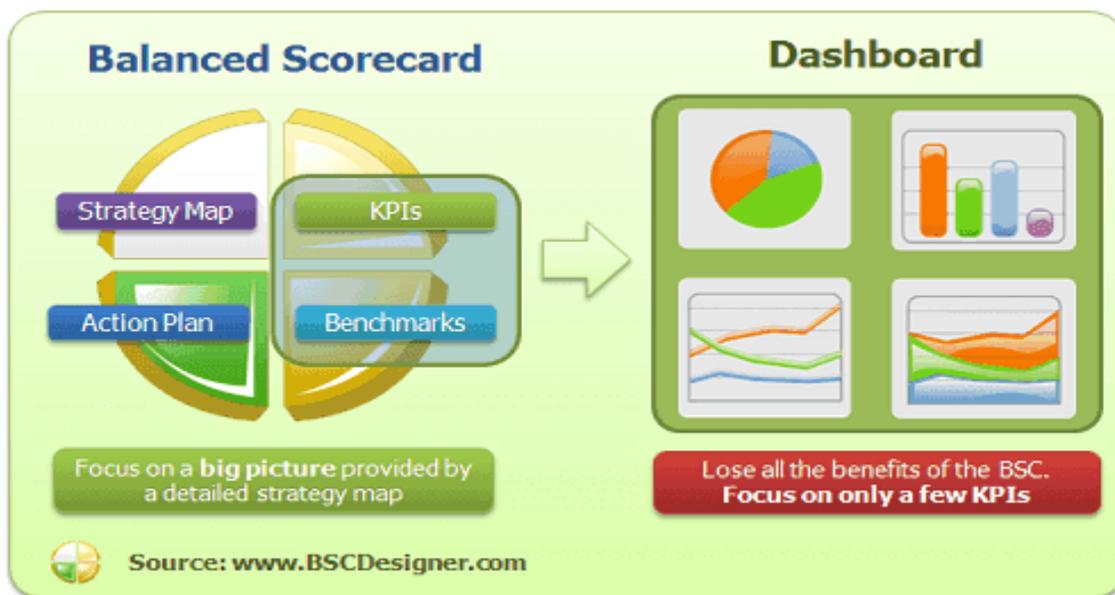
Gambar 1.5. Arsitektur *Data Warehouse*

Business analytics adalah *tools* yang dapat membantu *user* dalam transformasi data menjadi pengetahuan (contoh: *queries*, *data/text mining*, *statistic tools*, dan lain-lain).



Gambar 1.6. *Business Analytics*

Business Performance Management (BPM), atau bisa disebut juga *Corporate Performance Management* (CPM), merupakan portofolio aplikasi dan metodologi dalam framework BI yang menyediakan tools bagi enterprise untuk mengelola operasinya dengan lebih baik



Gambar 1.7. Business Performance Management

User Interface dapat berupa *dashboard* atau *tools* lain untuk visualisasi informasi. *Dashboard* merupakan sebuah tampilan komprehensif dari pengukuran performa organisasi, atau disebut *Key Performance Indicators (KPI)*.



Gambar 1.8. Dashboard System

1.5. Manfaat BI dalam Organisasi Bisnis

MicroStrategy mengelompokkan 5 ragam BI dan menawarkan *tools* untuk masing-masing kelompok, yaitu:

1. Penyampaian laporan dan *alert*
2. Laporan *enterprise* (dengan *dashboards* dan *scorecards*)
3. Analisis *cube* (atau disebut analisis *slice-and-dice*)
4. *ad-hoc queries*
5. Statistik dan *data mining*

Manfaat BI menurut survey oleh Thompson (2004) adalah:

1. Laporan yang lebih cepat dan akurat (81%)
2. Pengambilan keputusan yang lebih baik (78%)
3. Meningkatkan customer service (56%)
4. Meningkatkan pemasukan (49%)

Automation Decision System (ADS) adalah sebuah *rule-based system* yang menyediakan solusi untuk problem managerial yang bersifat repetitif pada satu area yang spesifik. Contohnya adalah sebuah *loan approval system*. ADS awalnya muncul pada industri penerbangan, yaitu atau *revenue optimization systems*. Contohnya yaitu harga tiket yang dinamis sesuai dengan permintaan aktual. Sekarang ini banyak industri jasa menggunakan model harga yang serupa.

1.6. Proses Transaksional vs *Analytical*

BI secara umum berbeda dengan pemrosesan transaksi rutin. *Transaction Processing Systems* (TPS) secara konstan menangani pencatatan transaksi ke *operational database* (*insert/update/delete*). Sebagai contoh adalah transaksi penarikan tunai pada ATM, atau pencatatan transaksi penjualan pada suatu website *e-commerce*. Setiap transaksi tersebut langsung melakukan *update* ke database. Sistem ini disebut *Online Transaction Processing* (OLTP) yang menangani *on-going business*. Aplikasi ERP, SCM, dan CRM melakukan penyimpanan data di *OLTP system*. Karena digunakan untuk transaksi operasional, tujuan utama dari OLTP adalah efisiensi yang tinggi.

Di samping OLTP, terdapat juga *Online Analytical Processing* (OLAP) yang berkaitan dengan pengambilan informasi dari data yang disimpan oleh OLTP. Contoh informasi yang dihasilkan yaitu laporan penjualan rutin berdasarkan produk, region, atau *sales person*. OLAP sering dibangun pada *data warehouse*, dimana data yang digunakan tidak lagi transaksional sehingga tidak mengganggu transaksi operasional. Tujuan utama OLAP adalah efektivitas (baru kemudian efisiensi), sehingga dapat menyediakan informasi yang akurat dalam rentang waktu tertentu.

1.7. Implementasi BI

Kesuksesan implementasi sistem BI dapat diukur dari luasnya penggunaannya untuk pengambilan keputusan yang lebih baik. Secara tipikal, pengguna BI terdiri dari semua level hirarki manajemen (tidak hanya level eksekutif seperti EIS). Sebuah sistem BI yang sukses bisa memberi manfaat kepada organisasi secara menyeluruh.

Agar dapat sukses, sistem BI harus selaras dengan strategi bisnis. BI dapat merubah cara organisasi menjalankan bisnis nya dengan memperbaiki proses bisnis, atau merubah pengambilan keputusan menjadi *data/fact/information driven activity*. BI mestinya mendukung dalam eksekusi strategi bisnis, bukan menjadi penghalang. Untuk mendukung implementasi BI, dapat dibentuk suatu tim yang disebut *BI Competency Center* (BICC) dengan fungsi sebagai berikut:

- BICC dapat menunjukkan bagaimana BI berkaitan dengan strategi dan pelaksanaan strategi.
- BICC dapat mendorong interaksi antara komunitas pengguna bisnis potensial dan SI pada organisasi.
- BICC dapat berfungsi sebagai repositori dan penyebarluasan praktik BI terbaik di antara berbagai lini bisnis.
- Standar keunggulan dalam praktik BI dapat dianjurkan dan didorong di seluruh perusahaan.
- Organisasi dapat belajar banyak melalui interaksi dengan komunitas pengguna, seperti pengetahuan tentang beragam jenis alat analisis yang dibutuhkan.
- Komunitas pengguna bisnis dan organisasi dapat lebih memahami mengapa platform *data warehouse* harus cukup fleksibel untuk menyediakan kebutuhan bisnis yang berubah.

- Membantu *stakeholder* seperti eksekutif tingkat tinggi melihat bagaimana BI dapat memainkan peran penting dalam organisasi.

1.8. Isu-isu terkait kesuksesan BI

Dewasa ini, permintaan untuk “*real-time*” BI semakin besar. Teknologi saat ini dapat mendukung “*real-time*”, di antaranya:

- Pencatatan data lebih cepat (RFID, *sensors*)
- Perkembangan *database* dan berbagai teknologi *software*
- Infrastruktur telekomunikasi
- Kemampuan komputasi meningkat dan harga semakin murah

Terdapat beberapa opsi dalam implementasi BI, yaitu: mengembangkan semuanya dari awal, membeli sebuah sistem komplet, menggunakan *shell BI system* dan meng-*customize*, atau menggunakan konsultan dari luar. Hampir semua aplikasi BI dibangun dengan *shell* yang disediakan oleh vendor yang mungkin membuat solusi khusus untuk klien atau bekerja sama dengan penyedia jasa *outsourcing* lain. Masalah yang dihadapi perusahaan adalah memilih dari alternatif berikut: membeli, sewa, atau mengembangkan. Masing-masing alternatif ini memiliki beberapa opsi. Salah satu kriteria utama untuk membuat keputusan adalah justifikasi dan analisis biaya-manfaat (*cost-benefit analysis*).

Melakukan justifikasi dengan *cost-benefit analysis* lebih mudah untuk menghitung biaya yang dikeluarkan. Akan tetapi, biasanya lebih susah untuk menghitung manfaat, karena kebanyakan bersifat *intangibles* (tidak terukur). Parameter lain yang perlu diperhatikan adalah terkait keamanan (*security and privacy*). Ini adalah isu yang sangat penting dalam pengembangan sistem komputerisasi, terutama BI yang berisi data yang memiliki nilai strategis. Selain itu, privasi karyawan dan pelanggan perlu dilindungi.

Integrasi BI dengan sistem dan aplikasi yang ada pada organisasi juga harus diperhatikan, biasanya dengan sistem ERP, SCM, CRM. Selain itu, integrasi dengan *stakeholder* luar juga dilakukakn dengan media internet, di antaranya dengan pelanggan, *vendor*, pemerintah, dan sebagainya.

1.9. Teknik dan aplikasi BI

Sejumlah besar alat dan teknik telah dikembangkan selama bertahun-tahun untuk mendukung pengambilan keputusan manajerial. Beberapa dari mereka muncul dengan nama dan definisi yang berbeda. Kategori alat terkomputerisasi utama dirangkum dalam tabel 1.1.

Tabel 1.1. *Tools BI*

Kategori tools	Nama tools
<i>Data Management</i>	<i>Database, database management system (DBMS)</i> <i>ETL system</i> <i>Data Warehouse (DW)</i>
<i>Reporting status tracking</i>	<i>Online Analytical Processing (OLAP)</i> <i>Executive Information System (EIS)</i>
<i>Visualization</i>	<i>Geographical Information System (GIS)</i> <i>Dashboard</i>
<i>Strategy and performance management</i>	<i>Business Performance Management (BPM)</i> <i>Balanced Scorecard (BSC)</i>
<i>Business Analytics</i>	<i>Data Mining</i> <i>Web Mining & Text Mining</i>

Baru-baru ini telah terjadi lonjakan besar perangkat lunak dan penyedia aplikasi BI. Beberapa nama perusahaan ini yang sangat populer adalah: Teradata, MicroStrategy, Microsoft, IBM + Cognos + SPSS, SAP + Business Objects, Oracle + Hyperion, SAS, dan banyak lainnya. Telah terjadi banyak konsolidasi karena perusahaan perangkat lunak besar mengakuisisi perusahaan lain untuk membangun portofolio penawaran penuh. Sebagai contoh, SAP mengakuisisi Business Objects, IBM mengakuisisi Cognos pada tahun 2008 dan SPSS di tahun 2009, dan Oracle mengakuisisi Hyperion. Perusahaan baru sekarang muncul dalam teks, web, dan analisis data. Selain itu, perusahaan berkolaborasi dengan perusahaan lain untuk membangun kemitraan. Sebagai contoh, SAS dan Teradata telah mengadakan kemitraan untuk menawarkan data warehouse dan kemampuan analitis prediktif secara bersama-sama.

1.10. Rangkuman

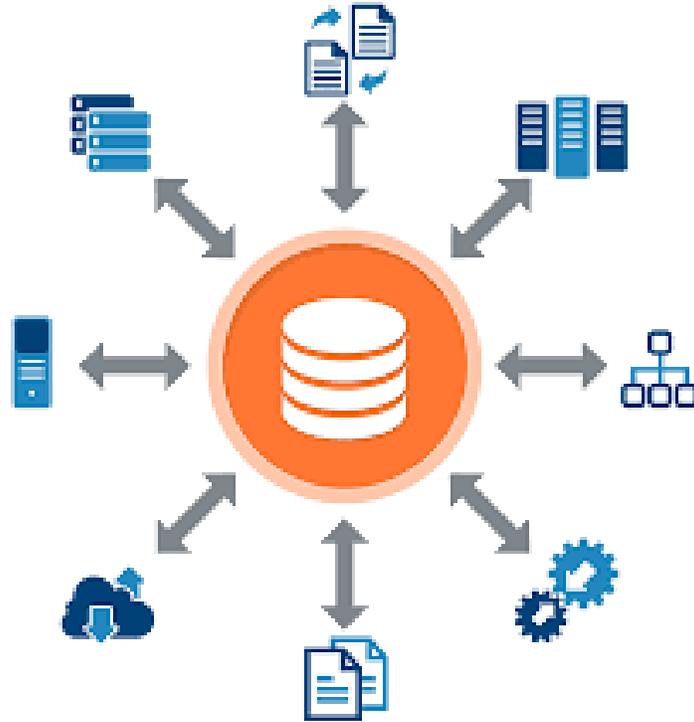
- Lingkungan bisnis sudah menjadi semakin kompleks dan berubah dengan cepat, sehingga pengambilan keputusan menjadi lebih sulit.
- Organisasi bisnis harus dapat merespon dan mengadaptasi perubahan dengan pengambilan keputusan yang cepat dan baik.
- Jangka waktu pengambilan keputusan menjadi lebih pendek, akan tetapi dampak dari pengambilan keputusan bisa membesar. Oleh sebab itu dibutuhkan penggunaan Sistem Pendukung Keputusan (SPK) yang terkomputerisasi.
- SPK menggunakan data, model, dan juga pengetahuan manajemen untuk menemukan solusi dari masalah terstruktur maupun tak terstruktur.
- Metode *Business Intelligence* (BI) memanfaatkan sebuah penyimpanan terpusat yang disebut *data warehouse* (DW) yang dapat dimanfaatkan untuk *Data Mining*, *Online Analytical Processing* (OLAP), *Business Performance Management* (BPM), dan visualisasi data.
- Arsitektur BI terdiri dari DW, *Business Analytics*, BPM, dan *user interface* (misalnya *dashboard*).

1.11. Latihan Soal

1. Carilah literatur terbaru (satu tahun terakhir) tentang pemanfaatan dari *Business Intelligence* (BI). Buatlah rangkuman dalam satu lembar dan sertakan sumber dari literature tersebut.
2. Jelaskan kaitan antara BI dengan SPK.
3. Apakah isu-isu yang dapat muncul dalam implementasi BI?

BAB II

DATA WAREHOUSING



Kompetensi

Mahasiswa dapat menerapkan aktivitas *Data Warehousing* (Minggu 3 dan 4)

Bahan Kajian

Bab ini membahas materi-materi sebagai berikut:

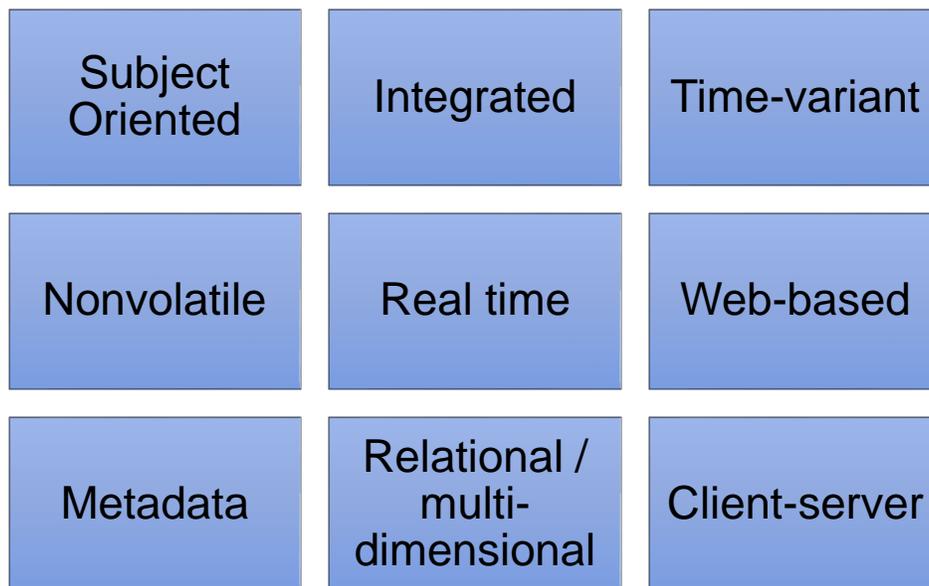
- Definisi *Data Warehouse* (DW)
- Karakteristik DW
- *Framework* DW
- Arsitektur DW
- SQL Server Integration Services (SSIS)
- Integrasi Data
- Proses ETL (*Extraction, Transformation, Load*)
- Pengembangan DW
- *Multidimensionality*

2.1. Definisi *Data Warehouse*

Data Warehouse (DW) adalah sebuah repository fisik dimana data relasional dikelola untuk menyediakan data yang berkualitas dengan standar format untuk skala enterprise. *Data warehouse* merupakan kumpulan dari *database* yang terintegrasi dan berorientasi subjek untuk menunjang fungsi DSS, dimana setiap unit data bersifat *nonvolatile* dan relevan pada suatu momen waktu.

2.2. Karakteristik *Data Warehouse*

Data Warehouse memiliki 9 karakteristik utama seperti dapat dilihat pada gambar 2.1.



Gambar 2.1. Karakteristik DW

Subject oriented (Berorientasi subjek), artinya data disusun menurut subjek rinci, seperti penjualan, produk, atau pelanggan, dan hanya berisi informasi yang relevan untuk dukungan keputusan. Orientasi subjek memungkinkan pengguna untuk menentukan tidak hanya bagaimana kinerja bisnis mereka, namun juga alasan di baliknya. *Data warehouse* berbeda dari database operasional sehingga sebagian besar database operasional memiliki orientasi produk dan disesuaikan untuk menangani transaksi yang memperbarui basis data. Orientasi subjek memberikan pandangan yang lebih komprehensif tentang organisasi.

Integrated. Integrasi sangat erat kaitannya dengan orientasi subjek. *Data warehouse* harus menempatkan data dari sumber yang berbeda ke dalam format yang konsisten. Untuk melakukannya, terdapat konflik yang bisa muncul seperti format penamaan dan perbedaan satuan ukuran yang digunakan.

Time-variant (*time series*). Sebuah *data warehouse* menyimpan data historis. Data tidak selalu memberikan status terkini (kecuali dalam sistem *real-time*). DW mendeteksi tren, penyimpangan, dan hubungan jangka panjang untuk peramalan dan perbandingan, yang mengarah pada pengambilan keputusan. Setiap *data warehouse* memiliki kualitas pada waktu tertentu. Waktu adalah satu dimensi penting yang harus dimiliki semua *data warehouse*. Data untuk analisis dari berbagai sumber berisi banyak titik waktu (misalnya harian, mingguan, dan bulanan).

Nonvolatile. Setelah data dimasukkan ke dalam *data warehouse*, pengguna tidak dapat mengubah atau memperbarui data. Data usang akan dibuang, dan perubahan dicatat sebagai data baru.

Web-based. *Data warehouse* biasanya dirancang sesuai dengan lingkungan komputasi yang efisien untuk aplikasi berbasis Web.

Relational / multidimensional. Sebuah gudang data menggunakan struktur relasional atau struktur multidimensi.

Client-server. Sebuah *data warehouse* menggunakan arsitektur *client-server* untuk menyediakan akses mudah bagi pengguna akhir. •

Real time. *Data warehouse* yang lebih baru memberikan kemampuan akses data dan analisis real-time

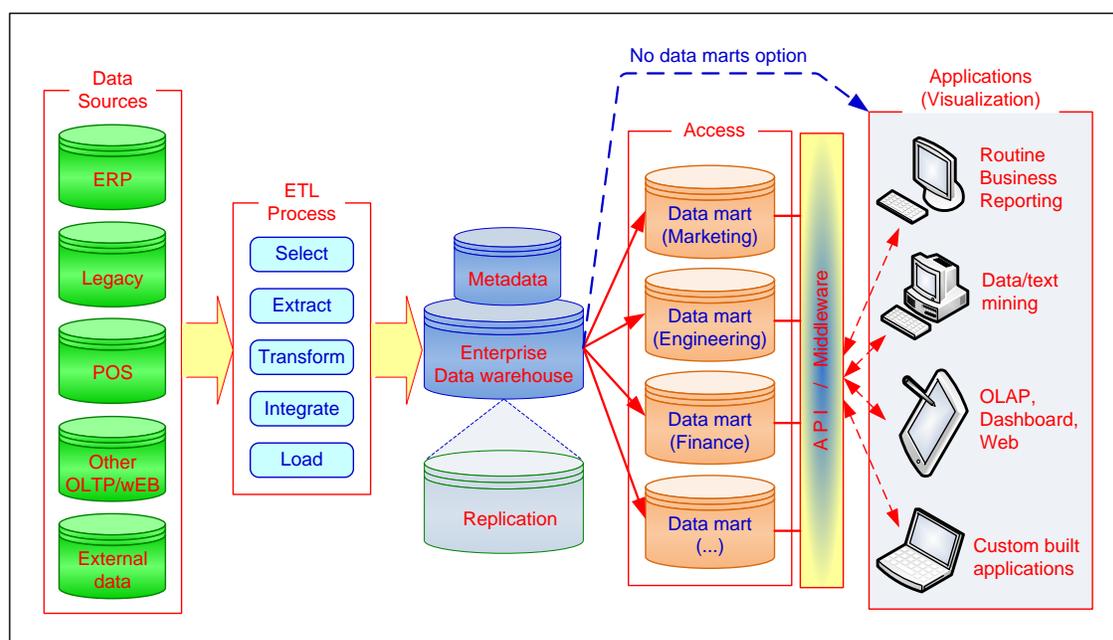
Metadata. Sebuah gudang data berisi metadata (data tentang data), yaitu bagaimana data disusun dan bagaimana menggunakannya secara efektif.

2.3. Framework Data Warehouse

Agar dapat bekerja dengan baik, terdapat konsep *framework data warehouse* seperti pada Gambar 2.2. Berikut ini adalah komponen utama dari framework DW:

- **Data source.** Data bersumber dari beberapa sistem operasional independen dan mungkin dari penyedia data eksternal (seperti data sensus dari pemerintah). Data mungkin juga berasal dari proses transaksi online (OLTP) atau sistem ERP. Data web dalam bentuk log Web mungkin juga *input* sebuah *data warehouse*.

- *Data extraction and transformation*. Data diekstraksi dan ditransformasikan dengan benar menggunakan perangkat lunak yang ditulis khusus atau komersial yang disebut ETL.
- *Data loading*. Data dimasukkan ke area pementasan, di mana mereka ditransformasikan dan dibersihkan. Data tersebut kemudian siap dimuat ke *data warehouse* dan / atau *data mart*.
- *Enterprise Data Warehouse (EDW)*. Sebuah data warehouse dengan skala enterprise yang digunakan untuk mendukung pengambilan keputusan. Pada intinya, EDW mendukung semua analisis keputusan dengan memberikan informasi terperinci yang relevan yang berasal dari berbagai sumber.
- *Data mart* adalah sebuah *data warehouse* yang lebih kecil (per departemen) yang hanya menyimpan data yang relevan dengan area tertentu. Ada dua jenis data mart:
 - 1) *Dependent data mart*, sebuah *subset* yang diturunkan langsung dari *data warehouse*;
 - 2) *Independent data mart*, sebuah *data warehouse* kecil yang khusus didesain untuk suatu unit bisnis atau departemen
- *Operational data stores (ODS)*. Sebuah tipe *database* yang biasanya digunakan sebagai penyimpanan sementara untuk *data warehouse* (*short-term memory*)
- *Metadata*, data tentang data. Dalam *data warehouse*, metadata mendeskripsikan konten dari *data warehouse*, membantu dalam mengkonversi data menjadi informasi/pengetahuan

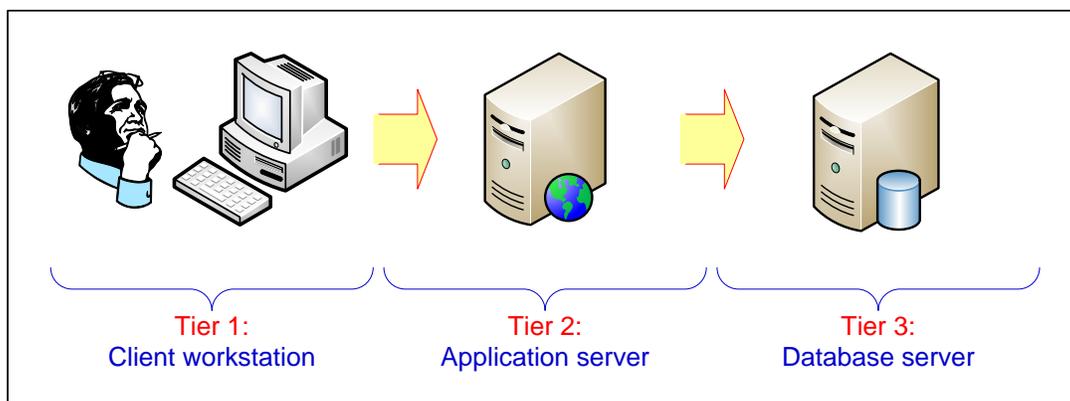


Gambar 2.2. Framework Data Warehouse

2.4. Arsitektur *Data Warehouse*

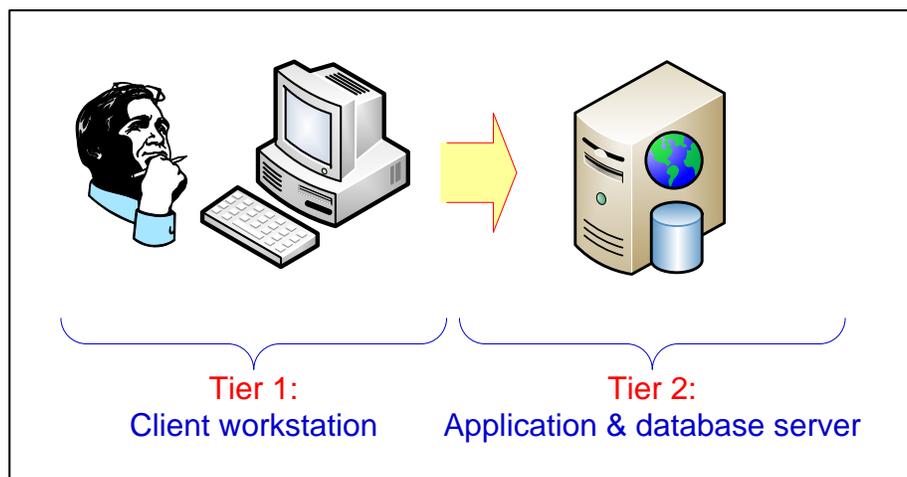
Ada beberapa arsitektur sistem informasi dasar yang bisa digunakan untuk *data warehouse*. Secara umum, arsitektur ini biasanya disebut arsitektur *client-server* atau *n-tier*, dengan yang paling umum digunakan adalah arsitektur 2-tier dan 3-tier. Arsitektur 3-tier (seperti pada gambar 2.3) terdiri dari tiga komponen berikut:

1. *Data acquisition software (back-end)*
2. *Data warehouse* yang berisi data & *software*
3. *Client (front-end) software* yang memungkinkan *user* untuk melakukan akses dan analisis data dari DW



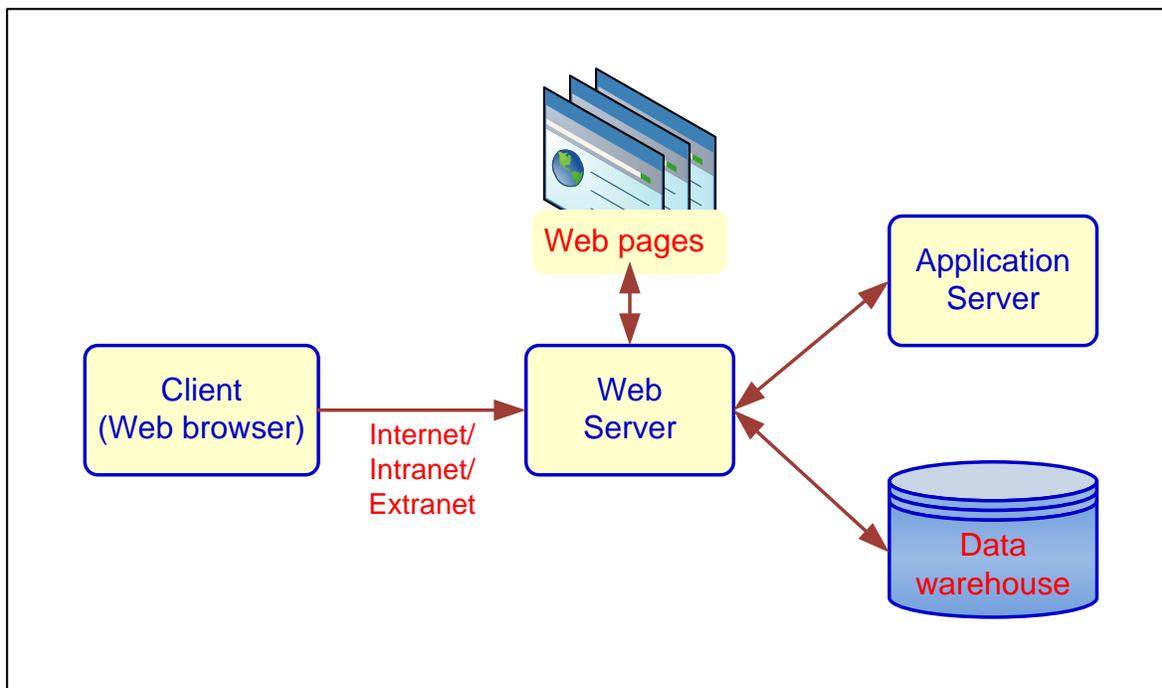
Gambar 2.3. Arsitektur 3-tier

Sedangkan, pada arsitektur 2-tier, dua tier awal pada arsitektur three-tier dikombinasikan menjadi satu (seperti pada gambar 2.4).



Gambar 2.4. Arsitektur 2-tier

Data warehouse dan Internet adalah dua teknologi utama yang menawarkan solusi penting untuk mengelola data perusahaan. Integrasi kedua teknologi ini menghasilkan *data warehouse* berbasis Web. Pada Gambar 2.5, kita dapat melihat arsitektur penyimpanan data berbasis Web. Arsitekturnya 3-tier yang mencakup *PC client*, *Web server*, dan *application server*. Di sisi klien, pengguna memerlukan koneksi internet dan *web browser* melalui antarmuka pengguna grafis yang familiar (GUI). Internet / intranet / extranet adalah media komunikasi antara *client* dan *server*. Di sisi *server*, *web server* digunakan untuk mengelola arus masuk dan keluar informasi antara klien dan server. Hal ini didukung oleh *data warehouse* dan *application server*. *Data warehouse* berbasis web menawarkan beberapa keuntungan yang menarik, termasuk kemudahan akses, independensi platform, dan biaya yang lebih rendah.



Gambar 2.5. Arsitektur DW berbasis Web

Berikut ini adalah hal-hal yang perlu diperhatikan dalam memilih arsitektur *Data Warehouse*:

- Apa database management system (DBMS) yang akan digunakan?
- Apa *parallel processing* dan/atau *partitioning* akan digunakan?
- Apa *data migration tools* yang digunakan untuk mengisi *data warehouse*?
- Apa *tools* yang digunakan untuk mendukung *data retrieval* dan analisis?

2.5. Integrasi Data

Integrasi data terdiri dari tiga proses utama yang, ketika diterapkan dengan benar, mengizinkan data diakses dari berbagai alat analisis ETL dan *data warehouse*: akses data (yaitu kemampuan untuk mengakses dan mengekstrak data dari sumber data), federasi data (yaitu integrasi pandangan bisnis di beberapa penyimpanan data), dan perubahan (berdasarkan identifikasi, pengambilan, dan penyampaian dari perubahan yang dilakukan pada sumber data perusahaan). Beberapa vendor, seperti SAS Institute, Inc., telah mengembangkan alat integrasi data yang kuat. Integrasi data perusahaan SAS mencakup perangkat integrasi data pelanggan yang meningkatkan kualitas data dalam proses integrasi. Oracle Business Intelligence Suite membantu mengintegrasikan data juga.

Tujuan utama sebuah *data warehouse* adalah mengintegrasikan data dari banyak sistem. Berbagai teknologi integrasi yang memungkinkan integrasi data dan metadata yaitu:

- *Enterprise application integration* (EAI)
- *Service-oriented architecture* (SOA)
- *Enterprise information integration* (EII)
- *Extraction, transformation, and load* (ETL)

Integrasi aplikasi enterprise (EAI) menyediakan media untuk mendorong data dari sistem sumber ke dalam *data warehouse*. Ini melibatkan pengintegrasian aplikasi fungsional dan difokuskan untuk berbagi fungsionalitas (bukan data) di seluruh sistem, sehingga memungkinkan fleksibilitas dan penggunaan kembali. Secara tradisional, solusi EAI berfokus pada penggunaan kembali aplikasi pada tingkat pemrograman antarmuka aplikasi. Baru-baru ini, EAI dilengkapi dengan menggunakan layanan SOA (kumpulan proses bisnis atau fungsi) yang didefinisikan dan didokumentasikan dengan baik. Menggunakan *Web-services* adalah cara khusus untuk menerapkan SOA. EAI dapat digunakan untuk memfasilitasi akuisisi data secara langsung ke *data warehouse* secara *real time*.

Enterprise information integration (EII) adalah alat yang berkembang yang menjanjikan integrasi data real-time dari berbagai sumber seperti database relasional, *Web-services*, dan database multidimensional. Ini adalah mekanisme untuk menarik data dari sistem sumber untuk memenuhi permintaan informasi. Alat EII menggunakan metadata yang telah ditetapkan untuk menghasilkan *view* yang membuat data terintegrasi muncul secara relasional ke pengguna akhir.

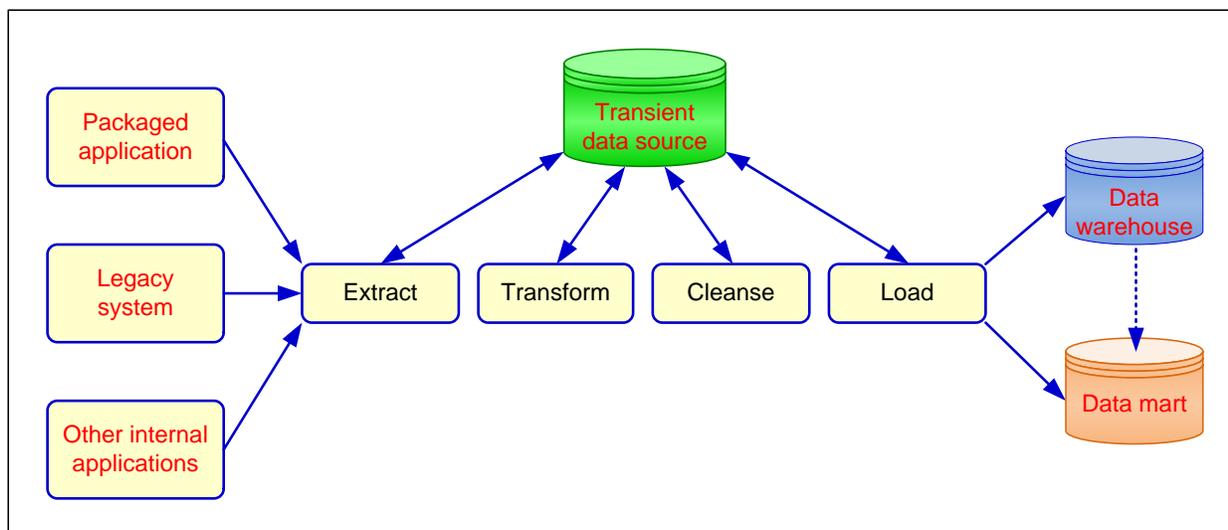
XML mungkin merupakan aspek terpenting dari EII karena XML memungkinkan data diberi tag baik pada saat pembuatan maupun selanjutnya. Tag ini dapat diperluas dan dimodifikasi untuk mengakomodasi hampir semua area pengetahuan.

2.6. Proses ETL (*Extraction, Transformation, Load*)

Inti sisi teknis dari proses *data warehouse* adalah *Extraction, Transformation, and Load* (ETL). Teknologi ETL, yang telah ada selama beberapa waktu, berperan penting dalam proses dan penggunaan *data warehouse*. Proses ETL adalah komponen integral dalam proyek data-sentris. Manajer TI sering dihadapkan pada tantangan karena proses ETL biasanya menghabiskan 70 persen waktu dalam proyek data-sentris.

Proses ETL terdiri dari *extraction* (yaitu membaca data dari satu atau lebih database), *transformation* (yaitu mengubah data yang diekstraksi dari bentuk sebelumnya menjadi bentuk yang diperlukan sehingga dapat ditempatkan ke dalam *data warehouse*), dan *load* (yaitu memasukkan data ke dalam *data warehouse*). Transformasi terjadi dengan menggunakan aturan atau tabel *lookup* atau dengan menggabungkan data dengan data lainnya. Alat ETL juga memindahkan data dari sumber ke target, mendokumentasikan bagaimana elemen data (misalnya *metadata*) berubah saat mereka berpindah antara sumber dan target, bertukar *metadata* dengan aplikasi lain sesuai kebutuhan, dan mengelola semua proses dan operasi yang mendukung (misalnya penjadwalan, manajemen kesalahan, log audit, statistik). Tujuan dari proses ETL adalah mengisi *data warehouse* dengan data yang terintegrasi dan bersih. Data yang digunakan dalam proses ETL dapat berasal dari sumber manapun: aplikasi *mainframe*, aplikasi ERP, CRM, *file flat*, atau *spreadsheet* Excel. Runtutan dari proses ETL dapat dilihat pada gambar 2.6.

Proses migrasi data ke *data warehouse* melibatkan ekstraksi data dari semua sumber yang relevan. Sumber data dapat terdiri dari *file* yang diambil dari *database* OLTP, *spreadsheet*, *database* pribadi (misalnya Microsoft Access), atau file eksternal. Biasanya, semua *file input* ditulis ke satu set tabel *staging*, yang dirancang untuk memudahkan proses pemuatan. Sebuah *data warehouse* berisi banyak aturan bisnis yang mendefinisikan hal-hal seperti bagaimana data akan digunakan, aturan *summarization*, standardisasi atribut yang dikodekan, dan aturan perhitungan. Setiap masalah kualitas data yang berkaitan dengan *file* sumber perlu dikoreksi sebelum data dimasukkan ke dalam *data warehouse*.



Gambar 2.6. Proses ETL

2.7. Pengembangan *Data Warehouse*

Banyak organisasi perlu membuat *Data Warehouse* yang digunakan untuk mendukung keputusan. Ada dua pendekatan yang dapat digunakan dalam pengembangan DW. Pendekatan pertama adalah dari Bill Inmon, yang sering disebut "bapak *Data Warehouse*." Inmon mendukung pendekatan pengembangan top-down yang menyesuaikan database relasional tradisional dengan kebutuhan pengembangan DW perusahaan secara luas, yang juga dikenal sebagai pendekatan EDW (*Enterprise Data Warehouse*). Pendekatan kedua adalah dari Ralph Kimball, yang mengusulkan pendekatan bottom-up menggunakan pemodelan dimensi, juga dikenal sebagai pendekatan *data mart*. Mengetahui bagaimana kedua konsep data warehouse ini baik kesamaan maupun perbedaannya membantu kita lebih memahami konsep pengembangan DW.

Model Inmon: pendekatan EDW. Pendekatan Inmon menekankan pengembangan *top-down*, menggunakan metodologi dan alat pengembangan database yang mapan, seperti diagram hubungan entitas (*entity-relationship diagram / ERD*), dan penyesuaian pendekatan pengembangan spiral. Pendekatan EDW tidak menghalangi pembuatan *data mart*. EDW adalah ideal dalam pendekatan ini karena memberikan pandangan perusahaan yang konsisten dan komprehensif.

Model Kimball: pendekatan *data mart*. Strategi *data mart* Kimball adalah pendekatan "rencana besar, pembangunan kecil". *Data mart* adalah gudang data berorientasi objek atau berorientasi departemen. Ini adalah versi skala kecil dari *data warehouse* yang berfokus pada

permintaan departemen tertentu seperti pemasaran atau penjualan. Model ini menggunakan pemodelan data dimensional, yang dimulai dengan tabel. Kimball menganjurkan sebuah metodologi pengembangan yang memerlukan pendekatan *bottom-up*, yang dalam kasus *data warehouse* berarti membangun satu *data mart* setiap waktu.

Data warehouse memberikan keunggulan kompetitif yang berbeda bagi perusahaan yang secara efektif membuat dan menggunakannya. Karena ukurannya yang sangat besar dan sifat intrinsiknya, *data warehouse* memerlukan pemantauan yang sangat kuat untuk mempertahankan efisiensi dan produktivitas yang memuaskan. Administrasi dan pengelolaan gudang data yang sukses memerlukan keterampilan dan kemampuan yang melampaui apa yang dibutuhkan oleh *database administrator* (DBA) tradisional. *Data warehouse administrator* (DWA) harus terbiasa dengan perangkat lunak, perangkat keras, dan jaringan berkinerja tinggi. Dia juga harus memiliki wawasan bisnis yang solid. Karena *data warehouse* memberi *input* bagi sistem BI dan DSS yang membantu manajer dalam melakukan pengambilan keputusan, DWA harus terbiasa dengan proses pengambilan keputusan sehingga dapat merancang dan memelihara struktur *data warehouse* dengan baik. Pada akhirnya, DWA harus memiliki kemampuan komunikasi yang sangat baik.

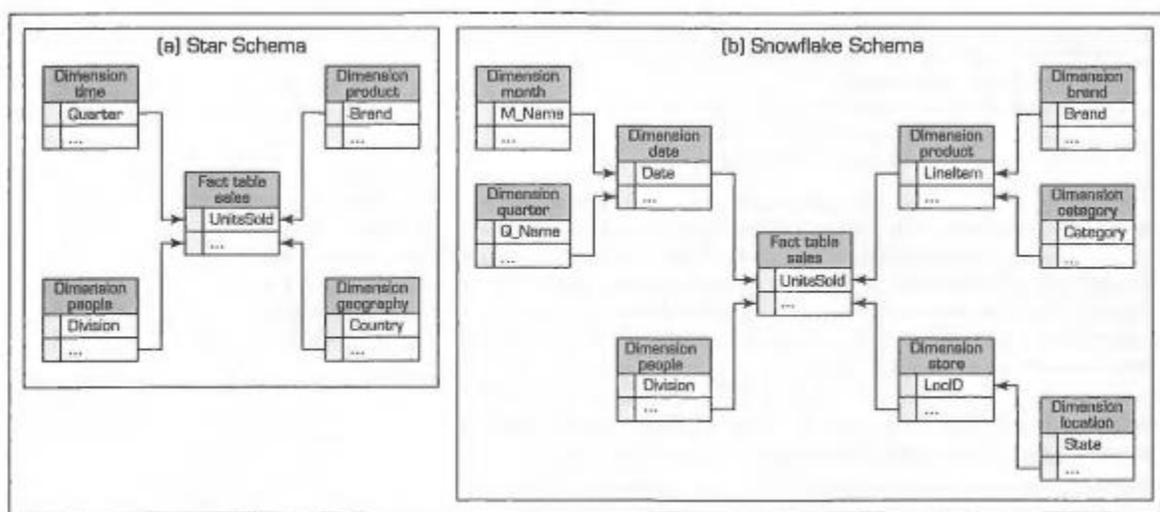
2.8. Multidimensionality

Banyak variasi arsitektur *data warehouse* yang dimungkinkan. Apapun arsitektur yang digunakan, desain representasi data di *data warehouse* selalu berbasiskan pemodelan dimensi. Pemodelan dimensi adalah sistem berbasis *retrieval* yang mendukung akses kueri dengan volume tinggi. Representasi dan penyimpanan data di *data warehouse* harus dirancang sedemikian rupa sehingga tidak hanya mengakomodasi, tetapi juga meningkatkan pemrosesan *query* multidimensi yang kompleks. Seringkali, *star schema* dan *snowflakes schema* diterapkan sebagai pemodelan dimensi di *data warehouse*.

Star schema adalah gaya pemodelan dimensi yang paling umum digunakan dan simpel. Skema bintang berisi tabel fakta sentral yang dikelilingi oleh beberapa tabel dimensi (Adamson, 2009). Tabel fakta berisi sejumlah besar baris yang sesuai dengan fakta dan tautan eksternal (*foreign key*). Sebuah tabel fakta berisi atribut deskriptif yang diperlukan untuk melakukan analisis keputusan dan pelaporan kueri, dan *foreign key* digunakan untuk menghubungkan ke tabel dimensi. Tabel fakta memfasilitasi *data warehouse* dalam mendukung analisis keputusan.

Di sekeliling tabel fakta sentral (dan dihubungkan melalui *foreign key*) adalah tabel dimensi. Tabel dimensi berisi informasi klasifikasi dan agregasi tentang baris fakta pusat. Tabel dimensi berisi atribut yang menggambarkan data yang terdapat dalam tabel fakta, mereka membahas bagaimana data akan dianalisis dan dirangkum. Tabel dimensi memiliki hubungan *one-to-many* dengan baris di tabel fakta pusat. Dalam *query*, dimensi digunakan untuk mengiris dan membagi nilai numerik di tabel fakta untuk memenuhi kebutuhan informasi tertentu. Skema bintang dirancang untuk memberikan waktu respons-respons yang cepat, kesederhanaan, dan kemudahan pemeliharaan untuk struktur basis data *read-only*. Skema bintang sederhana ditunjukkan pada Gambar 2.7(a).

Skema *snowflakes* adalah susunan tabel yang logis dalam database multidimensi sedemikian rupa sehingga diagram relasi entitas menyerupai kepingan salju. Terkait erat dengan skema bintang, skema *snowflakes* diwakili oleh tabel fakta terpusat (biasanya hanya satu), yang terhubung ke banyak dimensi. Namun, dalam skema *snowflakes*, dimensi di-normalisasi ke dalam beberapa tabel terkait, sedangkan dimensi skema *star* di-denormalisasi dengan masing-masing dimensi hanya direpresentasikan oleh satu tabel. Skema *snowflakes* sederhana ditunjukkan pada Gambar 2.7(b).



Gambar 2.7. Skema *star* dan *snowflake*

2.9. SQL Server Integration Services (SSIS)

SQL Server Integration Services atau biasa disingkat dengan SSIS adalah sebuah *tool* yang digunakan untuk melakukan proses *Extract, Transform, and Load* (ETL) dan diklasifikasikan

sebagai fitur *Business Intelligence* (BI). ETL adalah proses untuk mengumpulkan data dari berbagai sumber (*Extract*), melakukan transformasi (*Transform*), untuk kemudian menyimpannya ke dalam sistem yang lain (*Load*). Dalam kaitannya dengan BI, SSIS adalah fitur yang digunakan untuk menarik data dari ERP, *relational database*, atau *file* untuk kemudian hasilnya disimpan ke dalam *data warehouse*.

Package

Package berisi kumpulan tugas (yang disebut *task*) yang dieksekusi dengan urutan tertentu dan merupakan komponen utama SSIS. *Package* dapat disimpan di SQL Server pada *database msdb*, ataupun disimpan sebagai sebuah file *.dtsx*. Inovasi signifikan SSIS adalah desain arsitektur *package* untuk manajemen kontrol proses. Arsitektur kontrol proses SSIS terdiri dari komponen *control flow*, *data flow*, dan *event handler*. Setiap komponen tersebut memiliki kumpulan objek yang dapat digunakan pada saat proses pembuatan *package*.

Control Flow

Control flow adalah level tertinggi dari proses kontrol yang dapat digunakan untuk mengatur aktivitas proses *data flow* dan proses lainnya di dalam sebuah *package* yang didalamnya terdapat *task* dan *container*. *Task* adalah elemen *control flow* yang merupakan sebuah unit kerja. Sebuah *package* terdiri dari satu atau lebih *task*. *Task* dapat terhubung dan berjalan secara berurutan (*serial*) dengan menggunakan *precedence constraints*, ataupun tidak terhubung dan berjalan secara paralel sesuai dengan desain *package* yang kita inginkan.

Beberapa contoh *task* disediakan oleh SSIS dan sering digunakan diantaranya adalah:

- *Data Flow Task*: mengeksekusi *data flow* untuk ETL
- *Execute SQL Task*: mengeksekusi SQL Statement atau *stored procedure*
- *Bulk Insert Task*: melakukan *load* data ke dalam sebuah tabel menggunakan *BULK INSERT SQL*
- *Execute Package Task*: mengeksekusi *package* lain yang terdapat di SQL Server ataupun File System
- *Execute Process Task*: mengeksekusi program/aplikasi yang terdapat di komputer
- *File System Task*: Menangani operasi file seperti membuat atau menghapus sebuah directory, ataupun melakukan copy, perpindahan atau penghapusan file
- *Send Mail Task*: mengirimkan e-mail melalui SMTP

- *Analysis Services Processing Task*: memproses cube, dimensi, atau mining model SQL Server Analysis Services
- *Data Mining Query Task*: mengeksekusi *predictive query* terhadap data-mining model di Analysis Services
- *Data Profiling Task*: merupakan fitur baru di SSIS 2008 yang digunakan untuk mendapatkan profil data yang disimpan di SQL Server untuk mengidentifikasi masalah yang dapat timbul terkait kualitas data
- *Script Task*: digunakan untuk melakukan fungsi-fungsi yang tidak tersedia pada SSIS dengan bantuan *script* Visual Studio. Pada versi ini bahasa pemrograman yang dapat digunakan adalah Visual Basic dan C#

Data Flow

Data Flow digunakan untuk proses ETL yang didalamnya terdapat komponen kontrol proses. Komponen *data flow* terdiri dari *source*, *transformation*, dan *destination*. *Source* adalah dimana kita mendefinisikan lokasi sumber data. Terdapat enam komponen *source* bawaan SSIS:

- *OLE DB Source*: untuk mengambil data dari OLE DB seperti SQL Server, Access, Oracle, atau DB2
- *Excel Source*: mengambil data dari Excel. Komponen ini juga mempermudah SQL query terhadap Excel
- *Flat File Source*: mengambil data dari *file text*
- *Raw File Source*: mengambil data yang dihasilkan dari *Raw File Destination*. Format file-nya berbeda dengan *Flat File* dan biasanya digunakan untuk membaca data dengan cepat.
- *XML Source*: mengambil data dari dokumen XML
- *ADO NET Source*: komponen baru yang menggantikan *Data Reader Source* di SSIS 2005 dan digunakan untuk mengambil data dari koneksi ADO .NET.

Transformation adalah komponen kunci di dalam *data flow* yang mengubah data ke dalam format yang diinginkan atau digunakan untuk membersihkan dan melakukan standarisasi terhadap data. Beberapa komponen *transformation* yang biasa digunakan:

- *Lookup*: menghasilkan suatu nilai hasil dari referensi ke suatu tabel
- *Derived Column*: membuat sebuah kolom baru yang merupakan hasil penggabungan atau proses dari kolom lain
- *Sort*: mengurutkan data

- *Aggregate*: melakukan agregasi data
- *Data Conversion*: mengubah tipe data sebuah kolom menjadi tipe data yang lain
- *Conditional split*: membagi data berdasarkan kondisi tertentu yang sesuai
- *Merge Join*: menggabungkan dua set data menjadi satu set data menggunakan fungsi join
- *Fuzzy Grouping*: melakukan pembersihan data dengan mencari baris yang memiliki duplikasi
- *Fuzzy Lookup*: mencocokkan dan melakukan standarisasi data berdasarkan *fuzzy logic*
- *Union All*: menggabungkan banyak set data kedalam satu set data
- *Slowly Changing Dimension*: mengatur kondisi *insert* atau *update* data pada *dimension*
- *OLE DB Command*: mengeksekusi *OLE DB command* untuk setiap baris data
- *Script Component*: melakukan transformasi data dengan menggunakan *script*

Destination menerima data dari *source* atau *transformation* untuk kemudian menyimpannya ke dalam data source OLE DB atau file text. Komponen *destination* yang terdapat di SSIS:

- *Data Mining Model Training*: melatih *mining model Analysis Services* dengan cara mengirimkan data hasil transformasi
- *ADO NET Destination*: komponen baru yang menggantikan *Data Reader Destination* dan digunakan untuk menyimpan data ke koneksi ADO .NET
- *Dimension Processing*: melakukan *load* dan pemrosesan *dimension* yang terdapat pada *Analysis Services*
- *Excel Destination*: menyimpan data ke dalam *file Excel*
- *Flat File Destination*: Menyimpan data ke *file text*
- *OLE DB Destination*: menyimpan data ke koneksi OLE DB seperti SQL Server, Oracle, atau Access
- *Partition Processing*: Melakukan proses *update*, *incremental*, atau *full* partisi *Analysis Services*
- *Raw File Destination*: menyimpan data ke dalam *raw file* yang nantinya digunakan oleh *Raw File Source*.
- *Recordset Destination*: menyimpan data ke dalam sebuah kumpulan data ADO
- *SQL Server Destination*: menyimpan data ke SQL Server dengan efisien

- *SQL Server Compact Destination*: menyimpan data ke SQL Server yang terdapat di dalam *Pocket PC*

Business Intelligence Development Studio

Business Intelligence Development Studio (BIDS) adalah kakas yang digunakan untuk mendesain, membuat, dan menguji *package*. BIDS menyediakan sebuah lingkungan pengembangan yang berorientasi grafis yang membantu proses *copy*, pengaturan dan pembuatan *package* menggunakan menu dan kotak kakas dengan metode *drag-and-drop*.

Import and Export Wizard

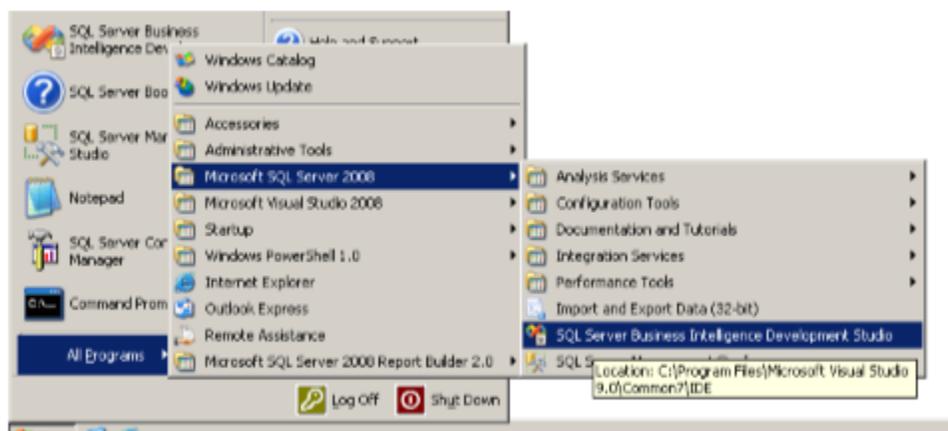
Import and Export Wizard adalah kakas yang digunakan untuk membuat sebuah *package* sederhana yang isinya memindahkan data dari sebuah *source* ke *destination*. *Package* hasil wizard ini dapat digunakan sebagai kerangka awal pembuatan *package* yang nantinya bisa dimodifikasi sehingga tidak perlu membuatnya dari nol.

Package Installation Wizard

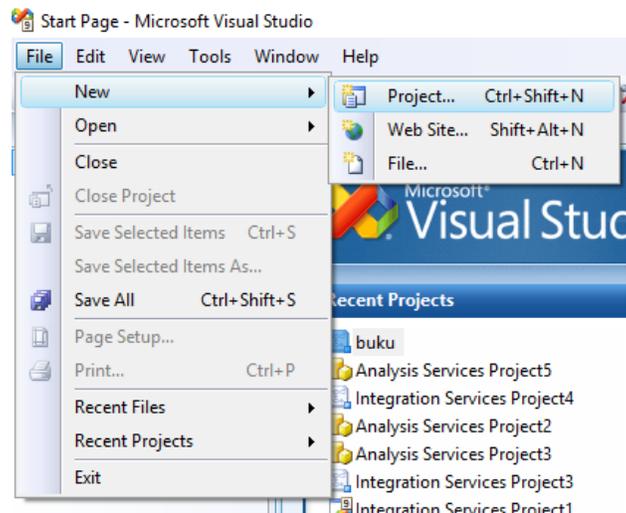
Kakas ini digunakan untuk menyiapkan *package* yang telah kita buat dengan menggunakan BIDS dan kemudian menginstal nya ke SQL Server.

Mengembangkan Solusi SSIS

Untuk mulai mengembangkan sebuah solusi SSIS, yang pertama harus dilakukan adalah membuka BIDS lewat Start > All Programs > Microsoft SQL Server 2008 > SQL Server Business Intelligence Development Studio.

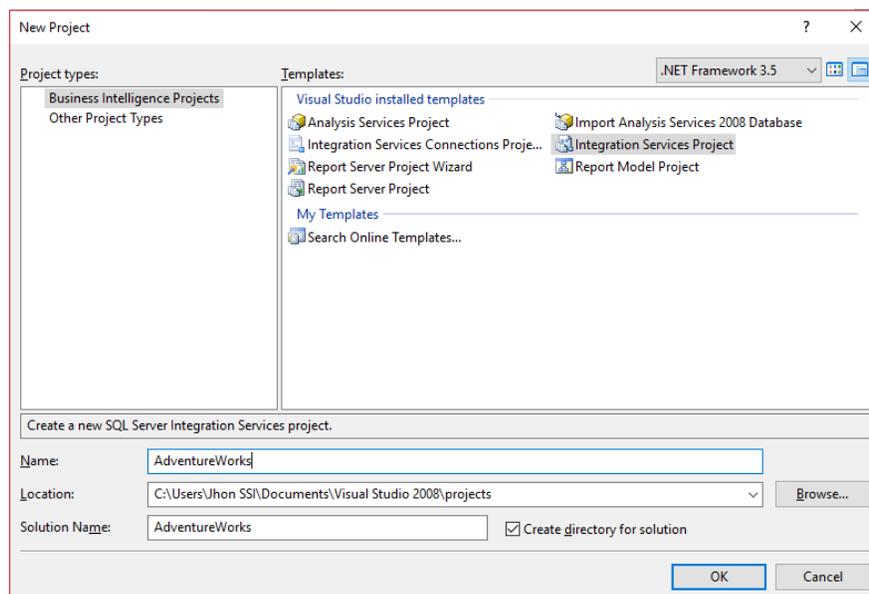


Gambar 2.8. Membuka SQL Server Business Intelligence Development Studio
Setelah terbuka, pilih menu File > New > Project...



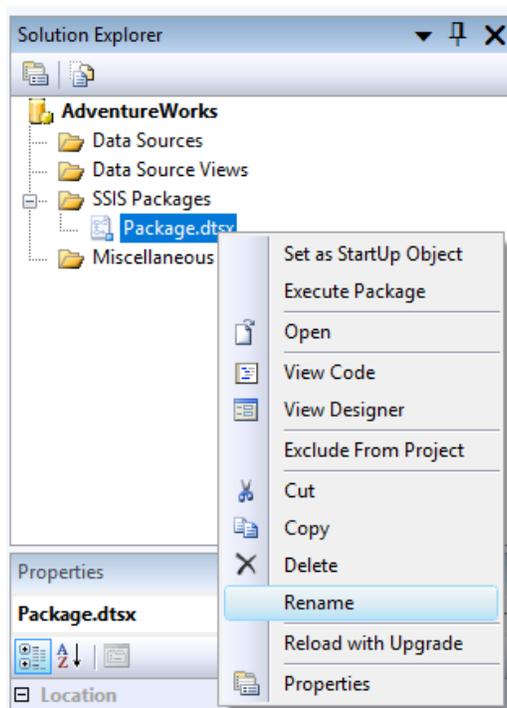
Gambar 2.9. Membuat proyek baru

Pilih *Business Intelligence Project* sebagai *Project Types*, *Integration Services Project* sebagai *Templates*, ketik nama proyek, lalu klik tombol OK.



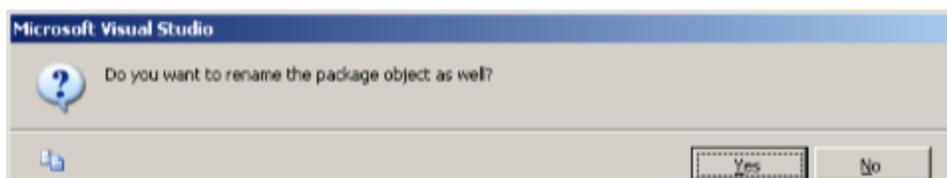
Gambar 2.10. Tampilan Proyek SSIS Baru

Setelah proyek dibuat, akan terdapat sebuah package dengan nama Package.dtsx di Solution Explorer. Klik kanan Package.dtsx lalu pilih Rename



Gambar 2.11. Mengganti nama SSIS Package

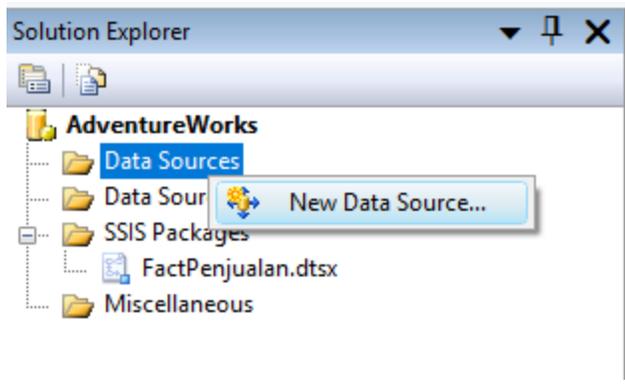
Ganti namanya dengan FactPenjualan.dtsx. Pilih Yes apabila muncul konfirmasi untuk mengganti nama objek package.



Gambar 2.12. Konfirmasi penggantian nama objek

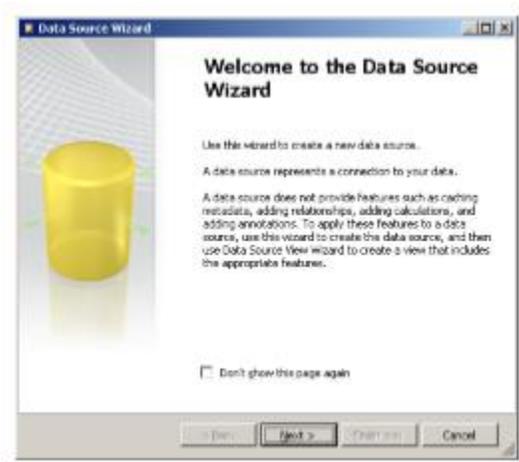
Menambahkan *Data Source*

Data Source adalah koneksi yang dapat digunakan untuk *source* atau *destination* pada *data flow*. Data source dapat berupa koneksi ke semua sumber data OLE-DB seperti SQL Server, Oracle, atau DB2. Data source dapat disimpan secara local di sebuah package atau digunakan secara bersama-sama dengan package yang lain di dalam BIDS. Untuk membuat data source yang digunakan secara berbagi dengan package yang lain, klik kanan pada folder Data Source yang terdapat di *Solution Explorer* lalu pilih *New Data Sources*



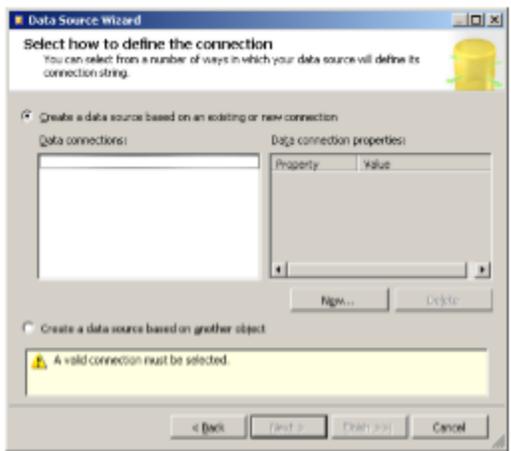
Gambar 2.13. Menambahkan data source

Data Source Wizard akan ditampilkan, klik *Next* pada layar *Welcome*



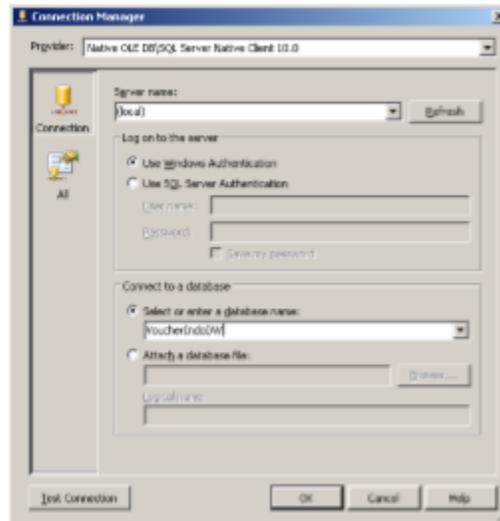
Gambar 2.14. Tampilan awal Wizard Data Source

Pada layar “*Select how to define the Connection*”, klik tombol *New*.



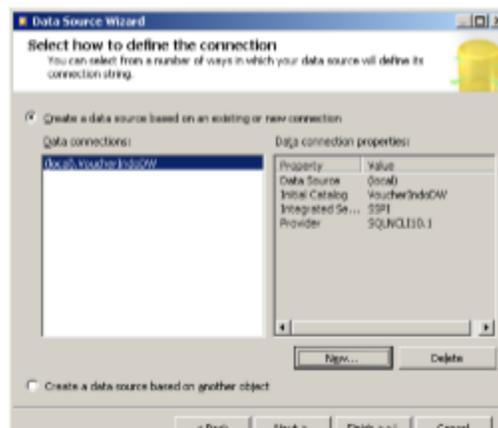
Gambar 2.15. Mendefinisikan koneksi

Pada layar *Connection Manager*, ketik (*local*) pada *Server Name*, lalu pilih database yang digunakan pada combo “*Select or enter a database name*”, lalu klik OK. Kita bisa menggunakan database *AdventureWorksDW* untuk latihan.



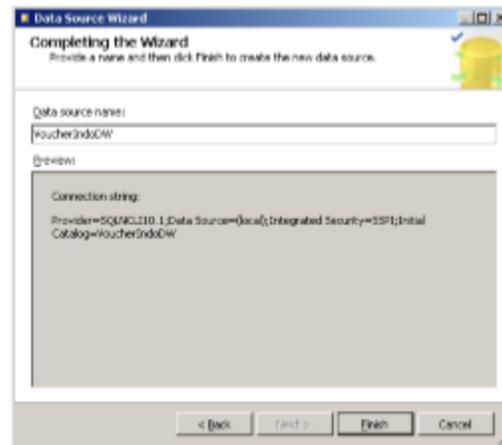
Gambar 2.16. Membuat koneksi dengan Connection Manager

Setelah muncul database yang dipilih pada layar *Select how to define the connection*, klik tombol *Next*



Gambar 2.17. Memilih koneksi

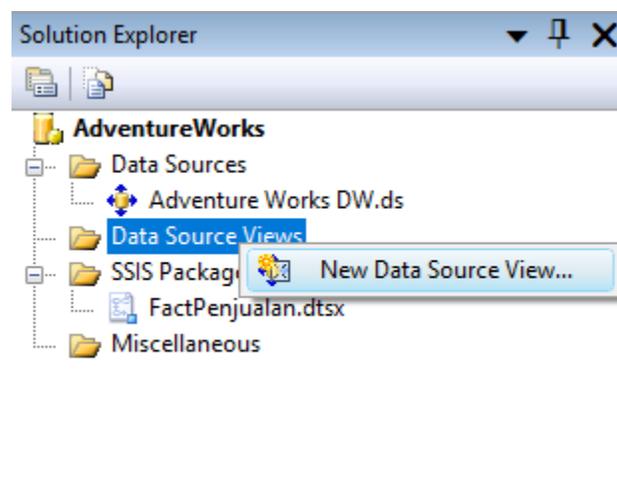
Isikan nama *data source*, lalu klik *Finish*



Gambar 2.18. Mengisi nama Data Source

Menambahkan *Data Source View*

Data Source View (DSV) adalah tampilan logik dari data yang merupakan kumpulan *table*, *view*, *stored procedure*, dan *query* yang dapat digunakan secara bersama-sama dalam project. DSV berguna terutama dalam data model yang besar dan kompleks. Untuk membuat DSV, klik kanan pada *Data Source Views* yang terdapat di Solution Explorer lalu klik *New Data Source View*.



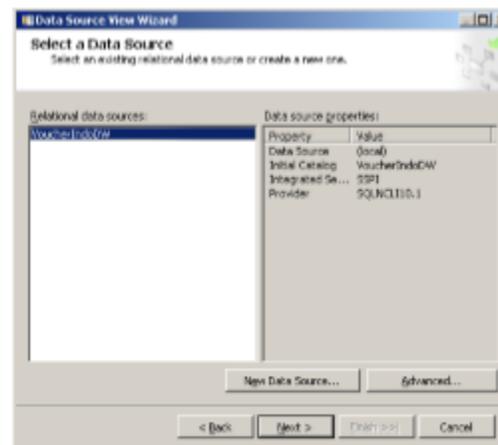
Gambar 2.19. Menambahkan *data source view*

Wizard New Data Source View akan ditampilkan, klik *Next*.



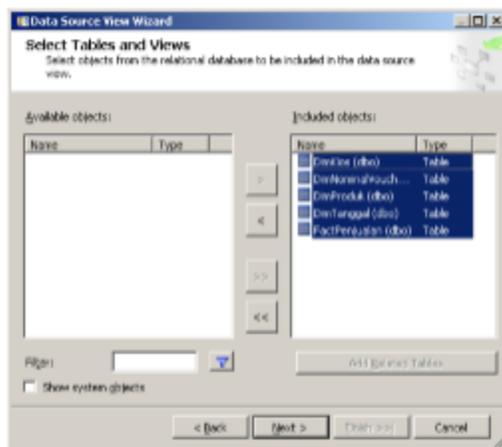
Gambar 2.20. Tampilan awal wizard data source view

Pada layar “*Select a Data Source*”, pilih *AdventureWorksDW* dari Relational data source lalu klik Next



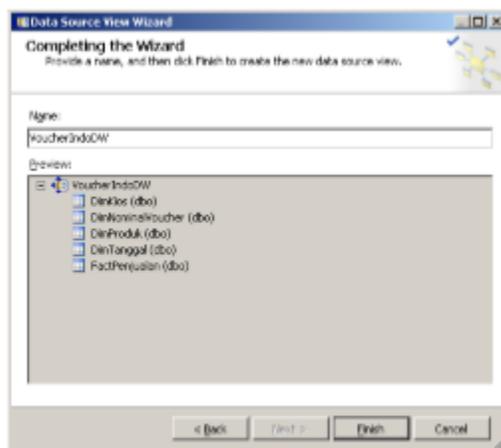
Gambar 2.21. Memilih data source

Pilih tabel dan view yang akan dipakai dalam DSV pada daftar Available objects, lalu klik tombol > untuk menyimpannya ke dalam daftar Included objects, atau klik tombol >> untuk menyimpan semua objek ke dalam daftar Included objects. Klik tombol *Next* setelah selesai memilih.



Gambar 2.22. Memilih objek yang akan dibuat pemandangan logik

Pada layar *Completing the Wizard*, isikan nama DSV lalu klik Finish

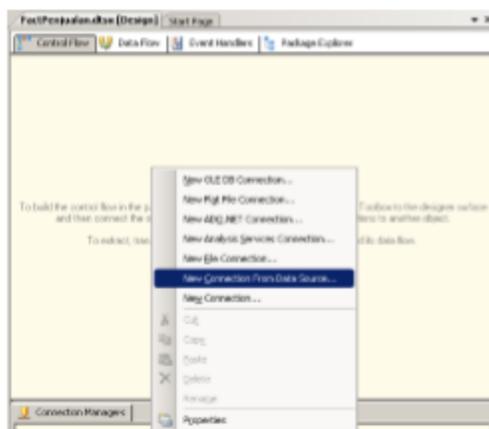


Gambar 2.23. Mengeset nama data source view

Menambahkan Koneksi

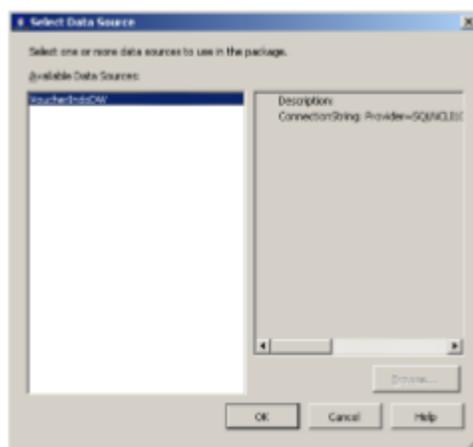
Dengan menggunakan contoh *AdventureWorks*, kita akan membuat dua buah koneksi yaitu ke file Excel dan ke database *AdventureWorks* dalam package *FactPenjualan.dtsx*. Untuk koneksi ke database *AdventureWorks* kita akan mempelajari cara membuatnya dari *Connection Manager*, sedangkan koneksi ke file Excel akan ditunjukkan pada bagian *Membuat Data Flow*.

Buka file *FactPenjualan.dtsx*, klik kanan pada *Connection Manager* lalu pilih *New Connection From Data Source*.



Gambar 2.24. Menambahkan koneksi ke data source

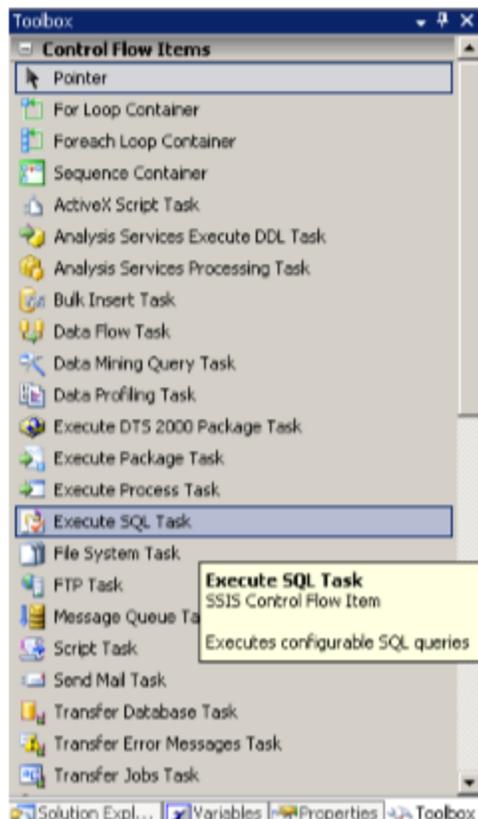
Pilih *AdventureWorks* dari daftar Available Data Source lalu klik OK



Gambar 2.25. Memilih koneksi

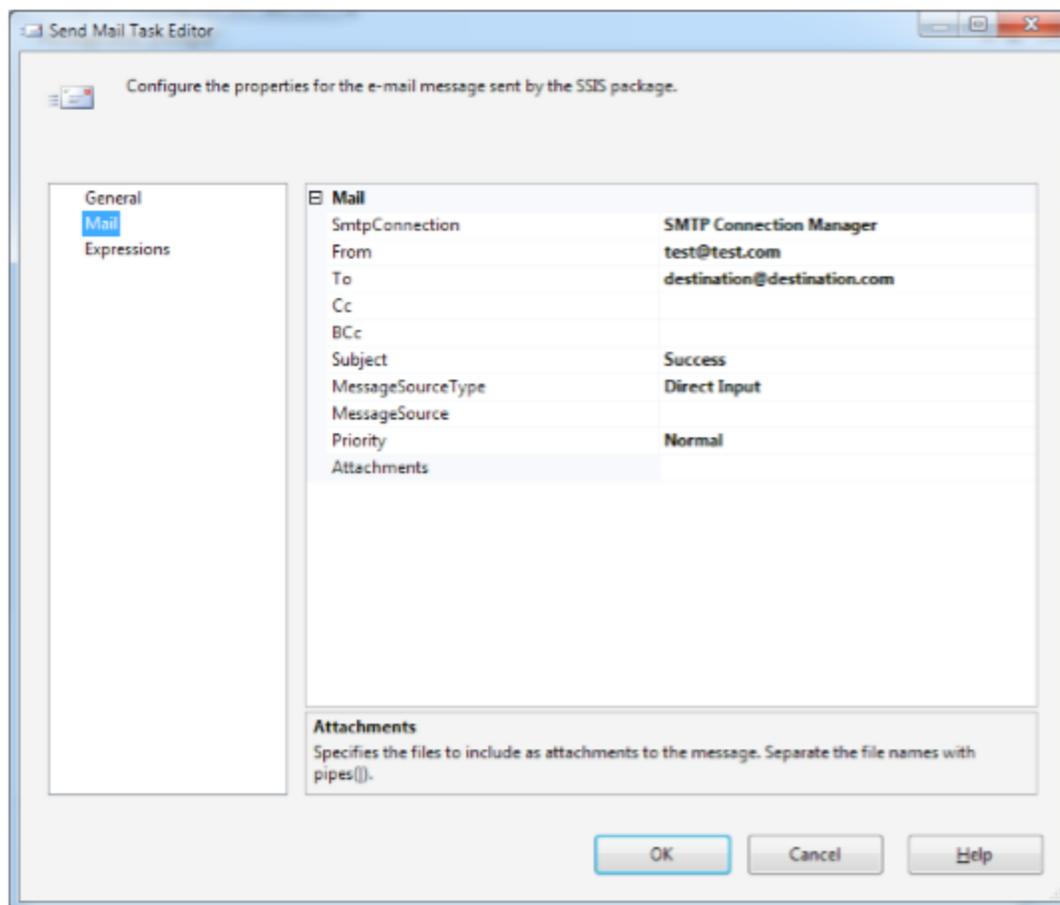
Menambahkan *Control Flow Task*

Langkah selanjutnya dalam *Package design* adalah memilih satu atau lebih item *control flow* dari *toolbox* dan menyeretnya ke bagian desain *Control Flow*. Tab *Control Flow* adalah untuk mengatur alur kerja untuk paket SSIS Anda. Tab desain *Control Flow* memungkinkan Anda untuk memilah-milah sebuah paket menjadi berbagai jenis *tasks* dan *data flow*. Setiap *data flow* dapat memisahkan pergerakan data dan transformasi oleh unit kerja atau lini bisnis. Gambar 2.26 menunjukkan kotak peralatan untuk item *Control Flow*. Anda dapat melihat dengan melihat nama item pada Gambar 2.26 bahwa *Control Flow* mengacu pada item yang melakukan semacam tindakan (kebanyakan semua nama tugas sudah cukup jelas).



Gambar 2.26. Kakas peralatan pada *control flow*

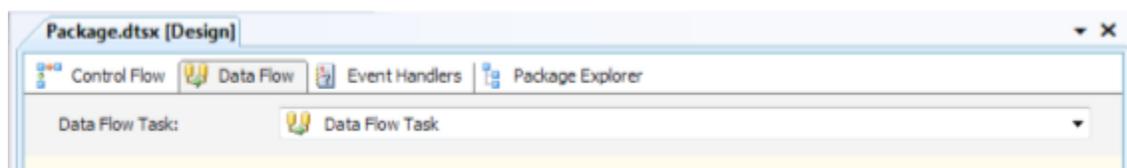
Untuk *package* pertama Anda, Anda hanya akan menggunakan dua *task* dari kakas peralatan ini: *Data Flow Task* dan *Send Mail Task*. *Drag* masing-masing item ke area kerja sentral di BIDS, letakkan *Data Flow Task* tepat di atas *Send Mail Task*. Klik *Data Flow Task*, dan tarik garis hijau dari *Data Flow Task* ke *Send Mail Task* sampai keduanya terhubung. Anda akan melihat bahwa *Send Mail Task* memiliki ikon peringatan merah di atasnya. Jika Anda mengarahkan *mouse* Anda di atas ikon kesalahan, *tooltip* akan memberi tahu Anda bahwa "*SMTP server is not specified*," sehingga tugas tidak dapat dijalankan. Untuk meninjau penyebab kesalahan ini, klik kanan pada *Send Mail Task* dan pilih *Edit*. Kemudian, pilih *Mail* dari panel sebelah kiri dan Anda akan diberi dialog konfigurasi *Send Mail Task Editor*, sebuah contoh yang ditunjukkan pada Gambar 2.27. Sejauh ini, Anda telah membuat sebuah *package* yang akan melengkapi semacam pergerakan data atau arus dan akan mengirim *email* melalui SMTP agar berhasil menyelesaikan tugas aliran data tersebut (setelah Anda menambahkan informasi lokasi *server* SMTP). Langkah selanjutnya dalam perancangan *pseudo-paket* ini adalah dengan mengkonfigurasi *Data Flow* itu sendiri.



Gambar 2.27. Konfigurasi *Send Mail Task*

Menambahkan *Data Flow*

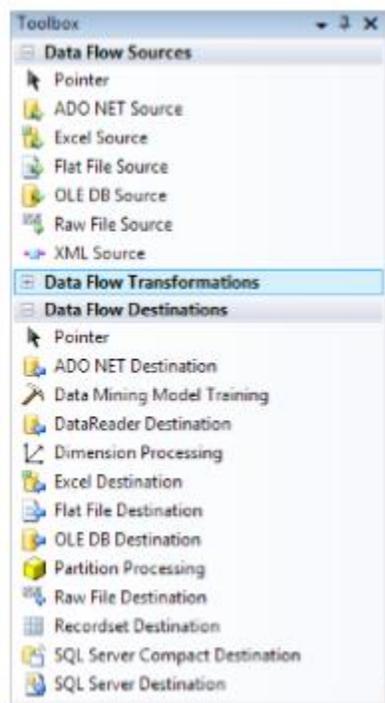
Untuk mulai mengkonfigurasi *Data Flow Task* Anda, klik dua kali *Data Flow Task* pada permukaan desain Flow Control. Ini akan membawa Anda ke tampilan Desain *Data Flow* untuk *Data Flow Task* tertentu (lihat Gambar 2.28).



Gambar 2.28. Tab *Data Flow*

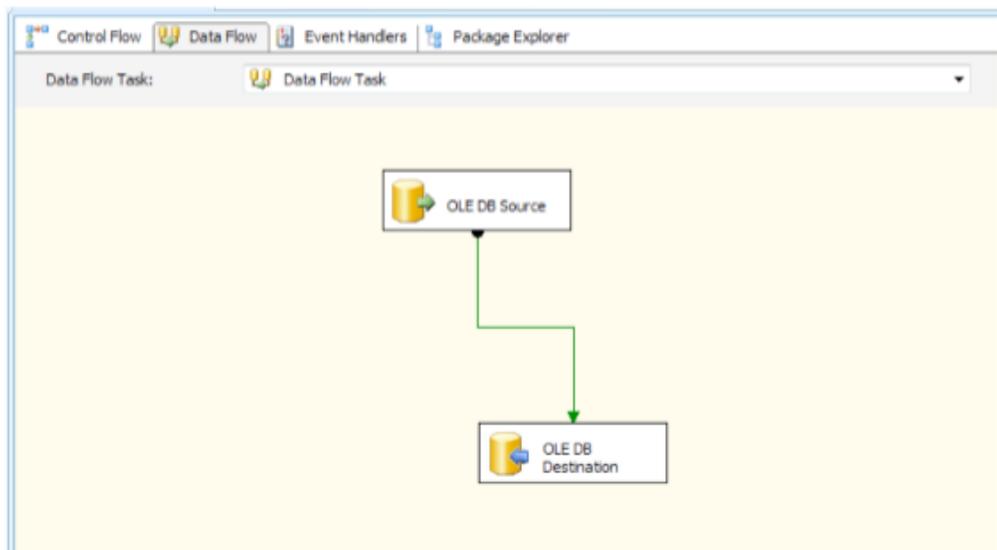
Anda biasanya akan memilih setidaknya satu sumber data, satu transformasi, dan satu tujuan dari kotak peralatan. Untuk saat ini, Anda tidak perlu menambahkan transformasi ke aliran

data. Anda cukup menambahkan *data source* dan *data destination*. Gambar 2.29 menunjukkan semua *data source* dan *data destination* dari kaskas peralatan BIDS.



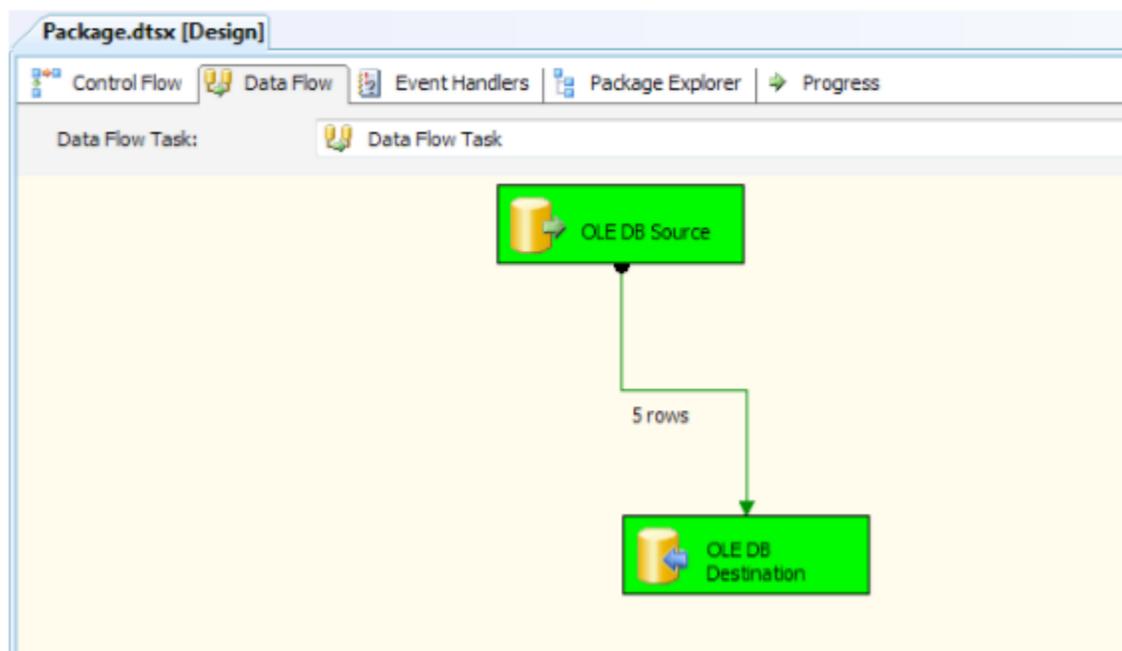
Gambar 2.29. *Data Flow Sources* dan *Data Destinations*

Untuk contoh khusus ini, Anda hanya perlu *drag-and-drop* sebuah *OLE DB Source* dan sebuah *OLE DB Destination*. Untuk mengkonfigurasi setiap *item* *OLE DB*, klik kanan dan pilih *Edit*. Selanjutnya, koneksikan *OLE DB Source* Anda dengan koneksi *AdventureWorks* yang ada, dan pilih *DimEmployee* sebagai tabel sumber Anda. Setelah menyelesaikan langkah ini, seret panah hijau dari *OLE DB Source* ke *OLE DB Destination*. Buka *OLE DB Destination* Anda, dan klik *New* untuk membuat tabel tujuan baru. Klik *OK* pada dialog *Create Table* untuk menginstruksikan *SSIS* untuk membuat tabel [*OLE DB Destination*], dan atur sebagai tujuan arus data. Akhirnya, klik *Mappings* di panel sebelah kiri agar *SSIS* secara otomatis memetakan sumber ke kolom tujuan berdasarkan nama, dan klik *OK*. Dalam kasus ini, tidak ada transformasi yang ditambahkan antara sumber dan tujuan. Meskipun tidak lazim di dunia nyata, contoh ini menunjukkan aliran data yang paling sederhana, dan ditunjukkan pada Gambar 2.30.



Gambar 2.30. Konfigurasi *Data Source* dan *Data Destination*

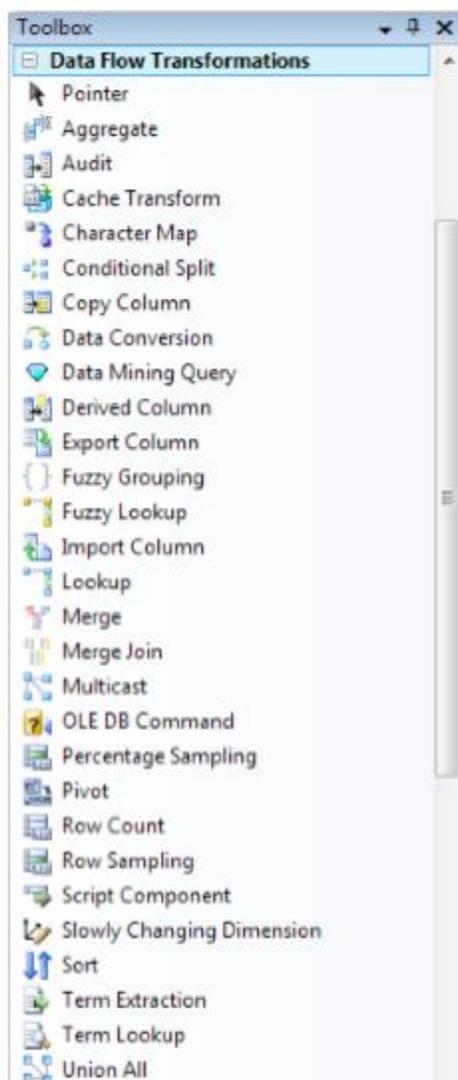
Untuk menguji *package* pertama Anda, klik kanan nama paket di *Solution Explorer*, dan pilih *Execute Package*. BIDS tidak hanya memungkinkan Anda untuk mengeksekusi paket tapi juga menunjukkan status eksekusi dengan mengubah warna sumber dan *item* tujuan: kuning untuk dieksekusi, hijau untuk kesuksesan, dan merah karena kegagalan. Jumlah baris juga ditampilkan di tampilan desain. Gambar 2.31 menunjukkan hasil dari keberhasilan eksekusi *package* ini.



Gambar 2.31. Eksekusi SSIS *package*

Menambahkan Transformasi pada *Data Flow*

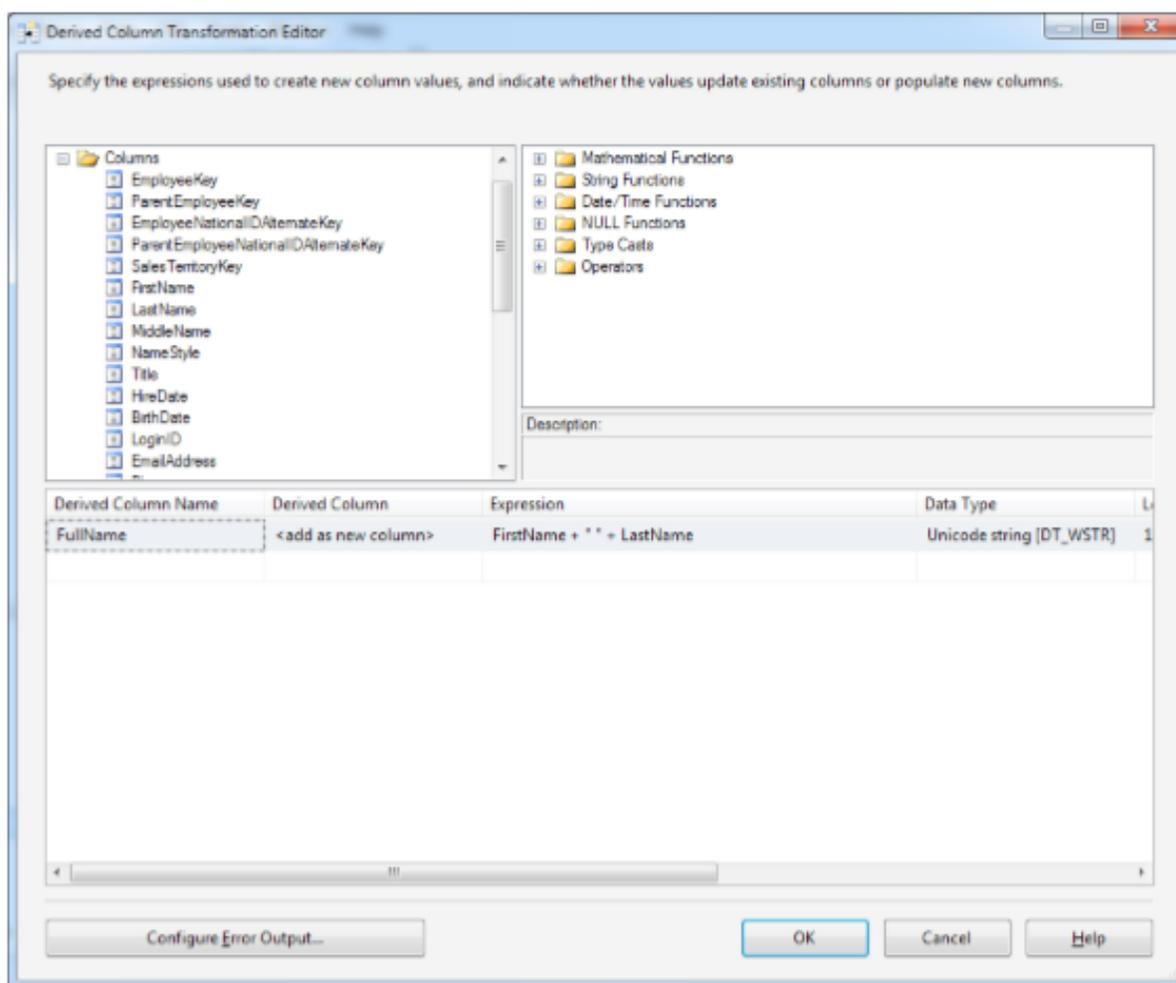
Kita tidak akan mencoba seluruh transformasi yang tersedia pada SSIS. Dengan menggunakan peta data Anda, Anda ingin menggunakan satu atau beberapa transformasi data yang disertakan dalam paket Anda. Gambar 2.32 menunjukkan pilihan yang mungkin.



Gambar 2.32. Pilihan *Data Flow Transformation*

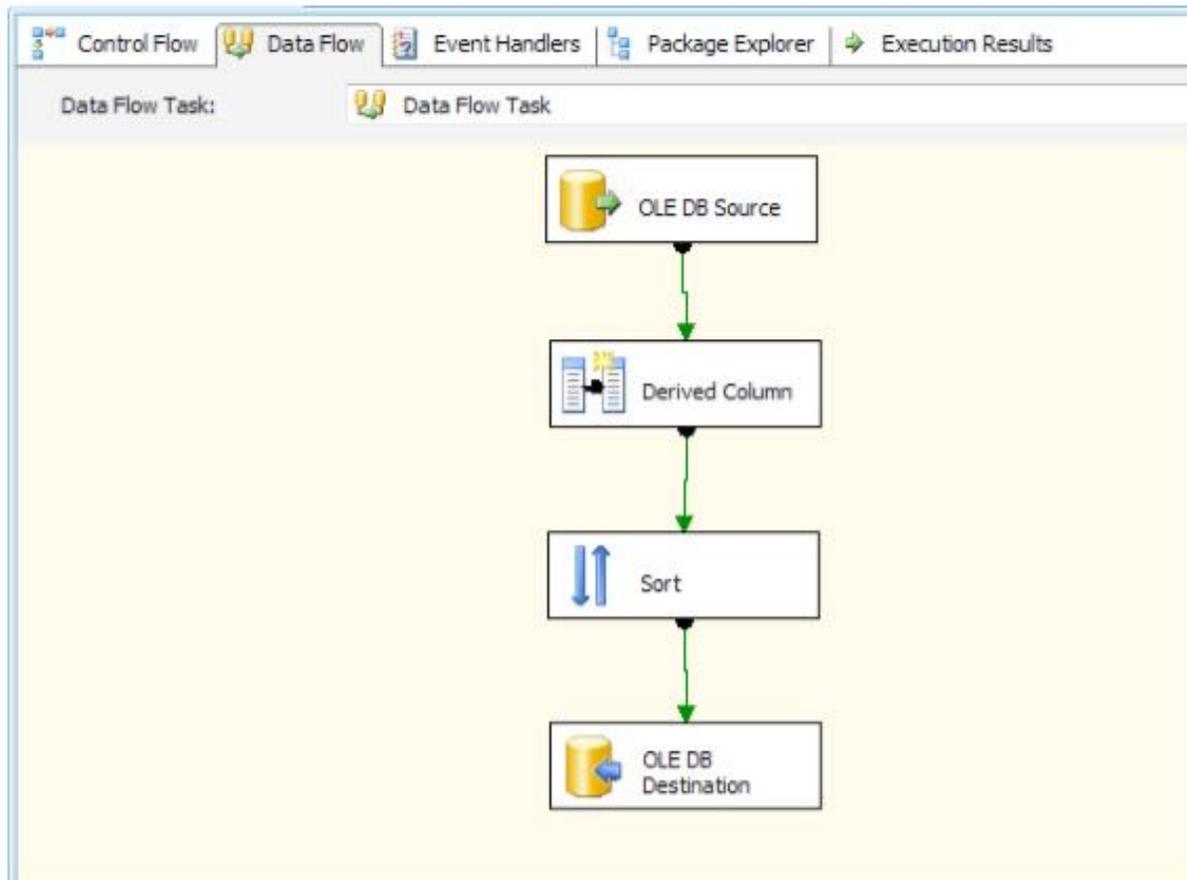
Untuk mengembangkan *package* dasar Anda, tambahkan dua transformasi ke *Data Flow*: *Derived Column* dan *Sort*. Lanjutkan dengan menghapus jalur aliran data (jalur hijau) antara OLE DB *Source* dan OLE DB *Destination*. Klik OLE DB *Source* Anda, dan seret jalur aliran data hijau (jalur sukses) ke transformasi *Derived Column*. Konfigurasi transformasi *Derived Column* dengan mengklik kanan transformasi pada permukaan desain, dan pilih Edit. Kotak dialog *Derived*

Column muncul di mana Anda dapat membuat kolom baru menggunakan sintaks ekspresi SSIS. Anda akan melihat bahwa semua kolom dan variabel yang tersedia dalam aliran data ditunjukkan di kotak dialog *Derived Column Transformation Editor*. Selain itu juga ada referensi fungsi. Untuk membuat kolom baru, lengkapi *expression field*. Untuk transformasi ini, masukkan FullName sebagai *Derived Column Name*, FirstName + "" + LastName sebagai *Expression*, dan klik OK. Entri ini hanya menggabungkan kolom FirstName dan LastName, seperti yang ditunjukkan pada Gambar 2.33.



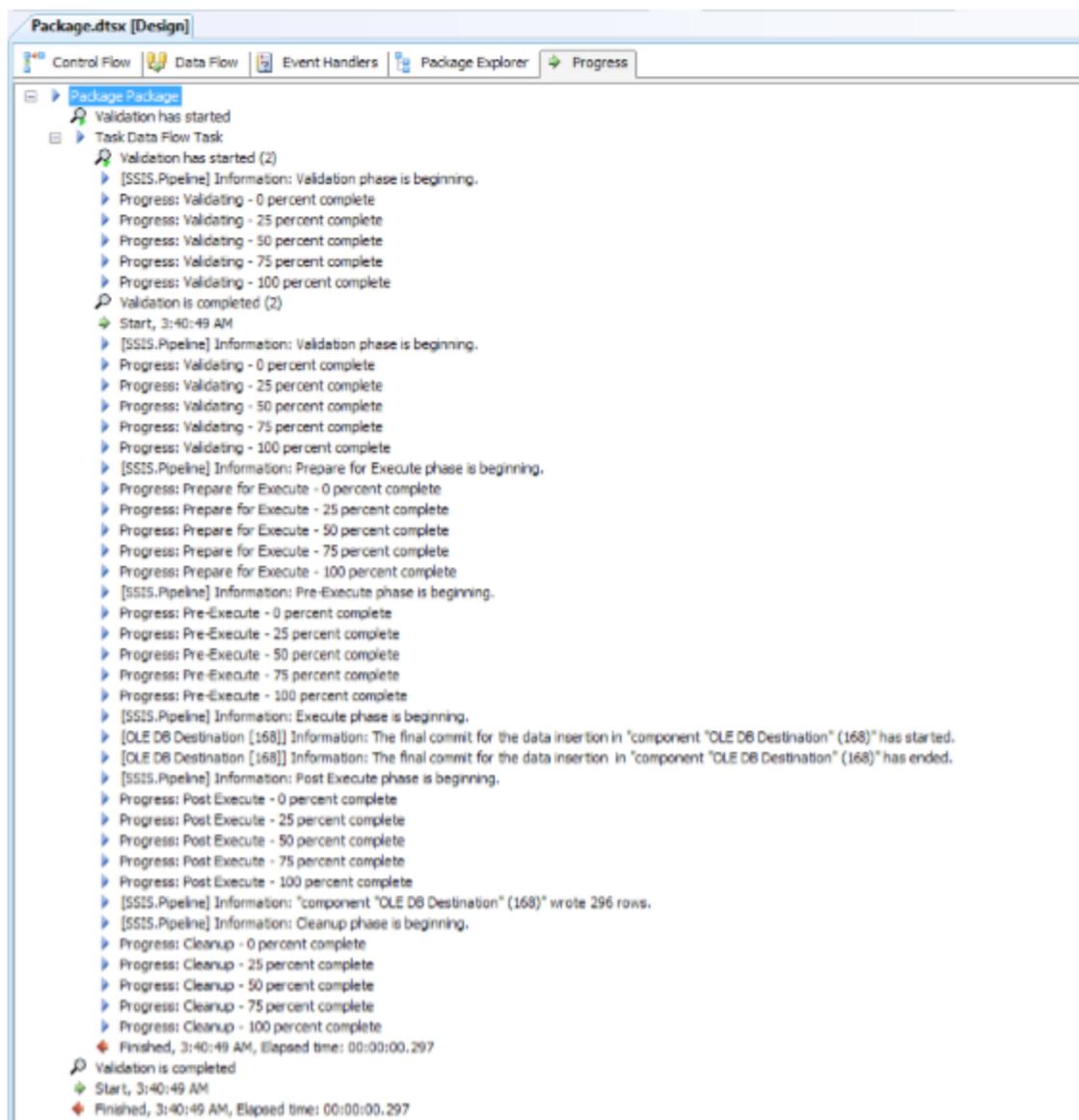
Gambar 2.34. Mengatur transformasi *Derived Column*

Selanjutnya, seret jalur alur data dari *Derived Column* ke *Sort*. Buka *Sort*, dan centang EmployeeKey dari *Available Input Columns*, lalu klik OK. Akhirnya, sambungkan *Sort* ke OLE DB Destination. Paket Anda sekarang akan seperti pada Gambar 2.35.



Gambar 2.35. Menambahkan transformasi pada *Data Flow*

Saat Anda menjalankan *package*, tab *Progress* tersedia di perancang SSIS. *Progress Pane* menunjukkan informasi yang sangat rinci tentang pelaksanaan setiap tugas dan langkah dalam paket. Seperti ditunjukkan pada Gambar 2.36, tingkat detailnya bisa sangat membantu dalam memahami kondisi yang terlibat dengan eksekusi *package*.



Gambar 2.36. *Progress* tab saat eksekusi *package*

Setelah eksekusi *package* berhasil, Anda dapat melihat bahwa transformasi data sudah berhasil dilakukan. Tabel yang baru dibuat pada OLE DB *Destination* akan berisi data yang merupakan hasil transformasi

2.10. Rangkuman

- Sebuah *data warehouse* adalah repositori data yang dibangun secara khusus dimana data diorganisasikan sehingga mudah diakses oleh *end user* untuk berbagai aplikasi.

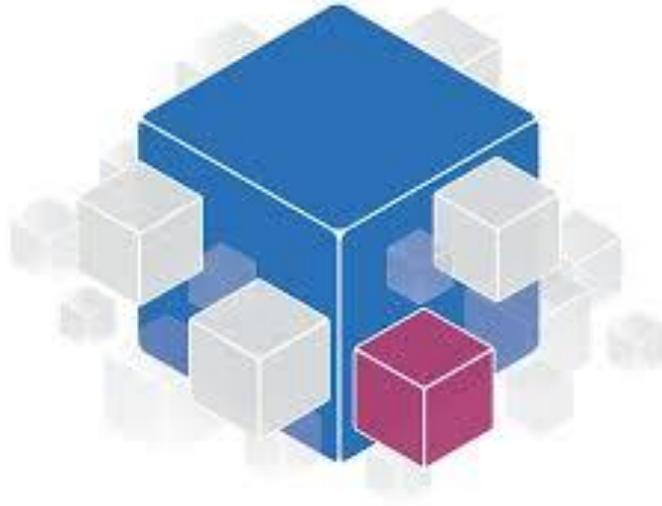
- *Data mart* berisi data pada satu topik (misalnya pemasaran). *Data mart* bisa berupa replikasi dari *subset* data di *data warehouse*. *Data mart* adalah solusi yang lebih murah yang bisa diganti dengan atau bisa melengkapi *data warehouse*. *Data mart* dapat bersifat independen atau dependen pada *data warehouse*.
- Integrasi data mencakup tiga proses utama dari ETL: *extraction, transformation, and load*. Ketika ketiga proses ini diterapkan dengan benar, data dapat diakses dan diakses oleh beragam perangkat ETL dan analisis serta lingkungan pergudangan data.
- Teknologi ETL menarik data dari banyak sumber, membersihkannya, dan memasukkannya ke dalam *data warehouse*.
- Keamanan dan privasi data dan informasi merupakan isu penting bagi profesional *data warehouse*.

2.11. Latihan Soal

1. Bandingkan integrasi data dan ETL. Bagaimana hubungan mereka?
2. Apa itu *data warehouse*, dan apa manfaatnya? Mengapa aksesibilitas Web penting untuk *data warehouse*?
3. *Data mart* dapat menggantikan atau melengkapi *data warehouse*. Bandingkan dan diskusikan pilihan ini.
4. Sebutkan perbedaan dan kesamaan antara peran *database administrator* dan *data warehouse administrator*.
5. Jelaskan bagaimana integrasi data dapat menghasilkan tingkat kualitas data yang lebih tinggi.
6. Ambil sebuah *database OLTP* yang Anda punya (bisa dari AdventureWorks, Northwind, ataupun *database* yang lain). Lakukan proses transformasi data (*sorting, derived column, lookup*) menggunakan SQL Server Integration Service.

BAB III

BUSINESS ANALYTICS DAN OLAP



Kompetensi

Mahasiswa dapat menerapkan *Business Analytics* dan OLAP (Minggu 5 – 7)

Bahan Kajian

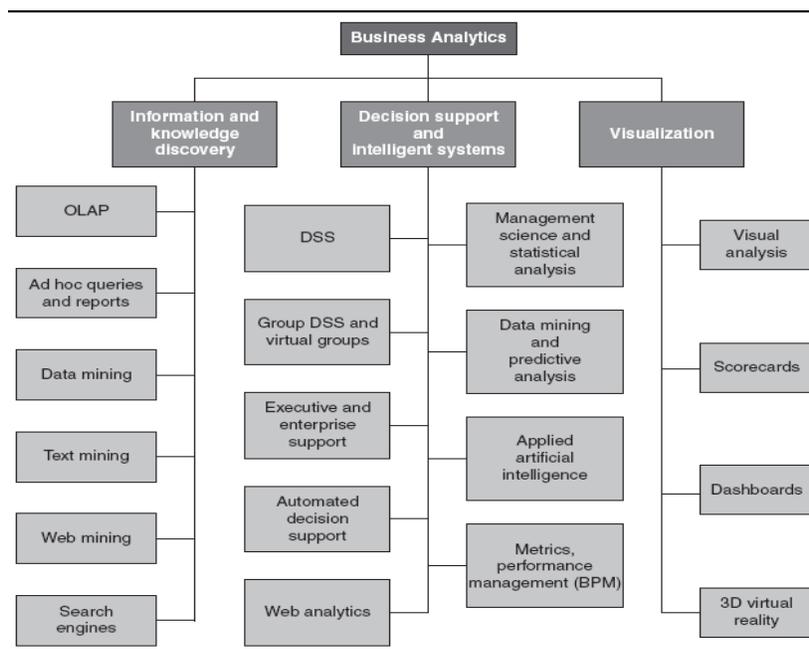
Bab ini akan membahas materi-materi sebagai berikut:

- Konsep *Business Analytics* (BA)
- Konsep OLAP
- OLAP vs OLTP
- Fitur OLAP
- *Cube* dan Dimensi
- Visualisasi Informasi
- *Customer Analytics*
- *Real Time BI*
- Penerapan BI/BA
- SQL Server Analysis Service (SSAS)

3.1. Konsep *Business Analytics*

Analytics merupakan sebuah ilmu dari analisis. Biasanya ini mengacu pada analisis data. Banyak metode dan *tools* yang dapat digunakan untuk melakukan analisis. Dalam konteks pemanfaatan di dunia bisnis, kita bisa menggunakan istilah *Business Analytics* (BA). BA adalah pemanfaatan aplikasi dan teknik untuk mengumpulkan, menganalisis, dan menyediakan akses terhadap data untuk membantu dalam pengambilan keputusan bisnis yang lebih baik. Penggunaan aplikasi BA secara langsung membutuhkan data-data bisnis. BA mencakup penggunaan sistem informasi manajemen dalam membantu pengambilan keputusan, terutama dalam bentuk dukungan OLAP.

Secara umum, BA dapat dibagi menjadi 3 kategori besar seperti dapat dilihat pada gambar 3.1. Kategori pertama adalah *information and knowledge discovery*. Beberapa aktivitas dari kategori ini, seperti OLAP, akan dibahas pada bab ini. Aktivitas lainnya seperti *Data Mining* akan dibahas di bab berikutnya. Kategori berikutnya adalah *decision support and intelligence systems*. Aktivitas pada kategori ini ada yang akan dibahas berikutnya, seperti *Business Performance Management* (BPM), dan sebagian akan dibahas pada mata kuliah lain, seperti Sistem Pendukung Keputusan dan Manajemen Sains. Kategori yang terakhir adalah *visualization*. Salah satu bentuk visualisasi, yaitu *Dashboard*, akan dibahas pada bab terakhir pada buku ini.



Gambar 3.1. Klasifikasi BA

3.2. Konsep OLAP

Online analytical processing (OLAP) merupakan aktivitas berbasis data yang dilakukan *end users* untuk menjalankan *query* data dan menganalisis hasilnya. Data analisis ini biasanya dilakukan pada *Data Warehouse*. Beberapa aktivitas yang dilakukan OLAP diantaranya:

- Melakukan *query* data
- Meminta laporan khusus
- Melakukan analisis statistikal
- Menghasilkan visualisasi informasi

3.3. OLAP vs OLTP

Bayangkan anda adalah seorang developer yang bekerja menggunakan database Adventure Work. Suatu ketika, seseorang dari divisi pemasaran menanyakan kepada anda hasil penjualan berdasarkan tipe produk, maka anda akan menuliskan query pada SQL Server Management Studio dengan melakukan query join terhadap table product ke salesorderdetail seperti terlihat pada gambar di bawah ini.

```

SQLQuery1.sql - ZERO\...\Admini...1))*
SELECT [Name], SUM([LineTotal]) AS [Sales per Product]
FROM [Sales].[SalesOrderDetail] AS SOD
INNER JOIN [Production].Product as P
ON SOD.ProductID = P.ProductID
GROUP BY [Name]

```

	Name	Sales per Product
1	All-Purpose Bike Stand	39591.000000
2	AWC Logo Cap	51229.445623
3	Bike Wash - Dissolver	18406.972080
4	Cable Lock	16240.220000
5	Chain	9377.710144
6	Classic Vest, L	12839.700000
7	Classic Vest, M	90250.600550
8	Classic Vest, S	156398.067950
9	Fender Set - Mountain	46619.580000
10	Front Brakes	50299.311000
11	Front Derailleur	44484.267800
12	Full Frame Classic	68842.214246

Gambar 3.2. Query dan hasilnya dari *sales by product*

Setelah tugas Anda selesai, tentu Anda akan melakukan pekerjaan anda kembali, tetapi kemudian orang dari bagian produksi datang untuk meminta data penjualan berdasarkan tanggal pembuatan produk, serta data penjualan berdasarkan warna produk. Belum selesai Anda menuliskan query, ada orang dari departemen lain ingin melihat data penjualan berdasarkan periode penjualan (tahun, kuartal, bulan), *sales person*, *product category* dan *customer*. Dari permasalahan yang anda hadapi sekarang, anda harus menulis *query* ulang sesuai data yang diinginkan oleh user.

Dari contoh kasus di atas kita membutuhkan sebuah OLAP. Sebuah database OLAP lebih optimal untuk kebutuhan analisis dan mendukung proses pengambilan keputusan. Sumber data yang dibutuhkan untuk analisis seringkali dari database transaksional (dengan banyak operasi *insert*, *update* dan *delete*) yang sering disebut sebagai *Online Transactional Processing (OLTP)*. Pada waktu tertentu, database OLTP akan diintegrasikan ke dalam Data Warehouse. *Business user* dapat melakukan *query* ke *Data Warehouse* untuk menganalisis penjualan dengan berbagai perspektif.

Dari contoh di atas, kita dapat membedakan OLTP dengan OLAP. OLTP adalah sebuah sistem yang memiliki peran utama dalam menerima dan menyimpan data berkaitan dengan fungsi bisnis harian seperti POS, CRM, SCM, ERP. Fokus utamanya adalah efisiensi dalam menjalankan aktivitas rutin. Sedangkan OLAP adalah sebuah sistem yang didesain untuk menghasilkan informasi dengan menyediakan analisis pada data organisasi sesuai dengan kebutuhan *user*. Fokus utama dari OLAP adalah efektivitas.

3.4. Fitur OLAP

Berikut ini adalah operasi-operasi yang dapat dilakukan pada OLAP:

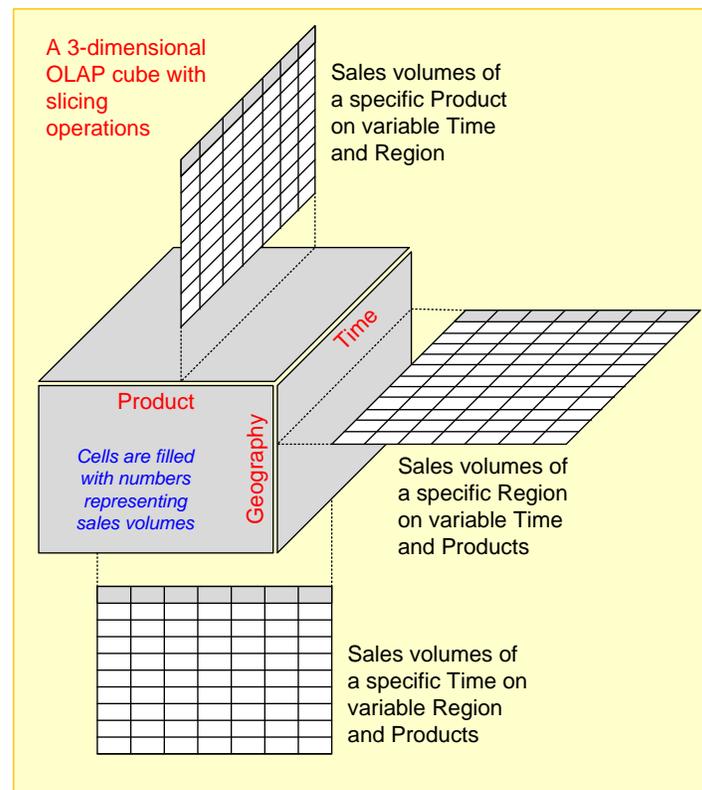
- *Slice* – sebuah *subset* 2 dimensi dari *multidimensional cube*
- *Dice* – sebuah *slice* yang terdiri lebih dari 2 dimensi
- *Drill Down/Up* – fitur untuk melakukakn navigasi antar level data antara data paling ringkas (up) hingga data paling detail (down). Dengan *drill down*, kita dapat melakukan investigasi dari suatu informasi secara detail (contoh: tidak hanya menemukan total penjualan, tetapi juga penjualan berdasarkan region, produk, atau salesperson).

Year	Rolls-Royce	Mercedes-Benz	Grand Total
2003	\$1,111,455.00	\$407,845.00	\$1,539,300.00
2004	\$740,970.00	\$653,780.00	\$1,434,750.00
2005	\$2,222,910.00	\$479,180.00	\$2,842,090.00
2006			
Grand Total			

Category	Model	Price	Units	Order Date	Trademark
Sport	CLK55 AMG Cabriole	\$79,645.00	1	2004	Mercedes-Benz
Sport	SL500 Roadster	\$83,800.00	1	2004	Mercedes-Benz
Sport	SL500 Roadster	\$83,800.00	1	2004	Mercedes-Benz
Sport	SL500 Roadster	\$83,800.00	2	2004	Mercedes-Benz
Sport	CLK55 AMG Cabriole	\$79,645.00	3	2004	Mercedes-Benz

Gambar 3.3. Fitur *drill down*

- Roll Up – komputasi semua relasi data untuk satu atau lebih dimensi
- Pivot – digunakan untuk merubah orientasi dimensi dari suatu laporan atau hasil query

Gambar 3.4. *Slicing operations* pada sebuah *data cube* tiga dimensi

Dengan adanya OLAP, kita dapat menyediakan fungsi pelaporan (*reporting*) yang dibutuhkan oleh manajemen. Laporan sendiri dapat dibagi menjadi dua, yaitu laporan rutin dan laporan *ad-hoc* (*on-demand*). Laporan rutin akan dibuat secara otomatis dan didistribusikan secara rutin pada periode tertentu. Sedangkan kebutuhan informasi yang tidak tersedia pada laporan rutin membuat kita membutuhkan *ad-hoc report*. Di sini diperlukan penggunaan *ad-hoc query*, yaitu kueri yang tidak dapat ditentukan sebelum kueri dibutuhkan (*on demand*).

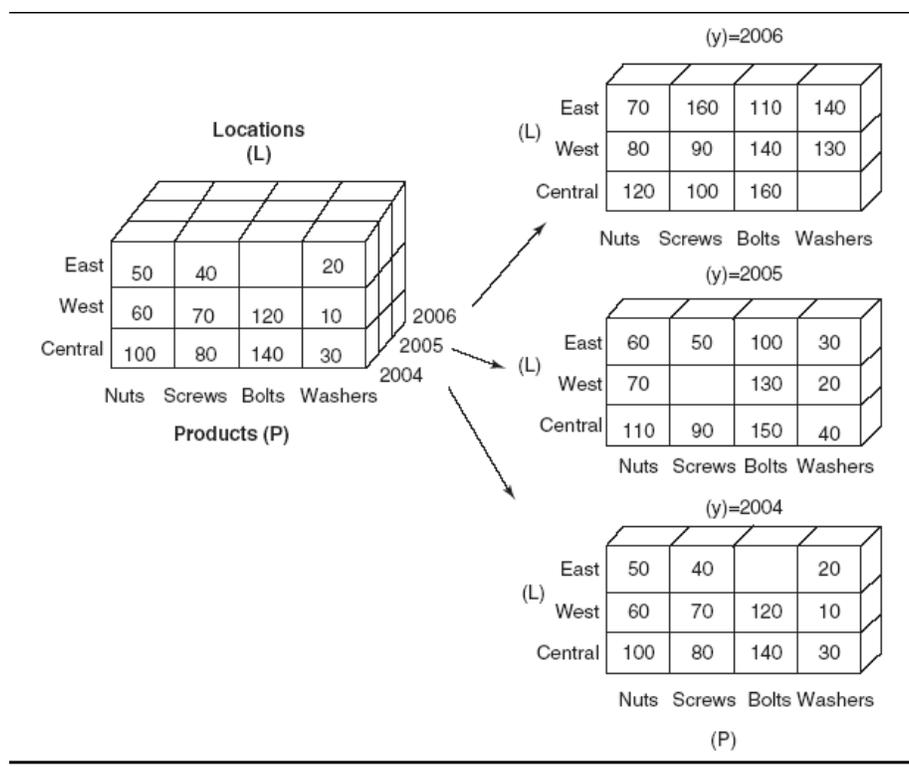
3.5. *Cube* dan Dimensi

Untuk memudahkan analisis dari data, kita bisa mengorganisir data tersebut dengan konsep multi dimensi. *Multidimensionality* adalah kemampuan untuk mengatur, menyajikan, dan menganalisis data dengan beberapa dimensi atau sudut pandang (perspektif). Sebagai contoh, penjualan dapat dianalisis berdasarkan 4 dimensi, yaitu wilayah, produk, tenaga penjual, dan waktu. Tiga faktor yang harus diperhatikan dalam konsep *multidimensionality* yaitu:

- *Dimension*. Menentukan dimensi yang tepat untuk melakukan analisis
- *Measures*. Satuan yang digunakan dalam pengukuran, misalnya jumlah penjualan, nilai uang, jumlah stok, dan lain-lain
- *Time*. Satuan periode waktu, contohnya adalah harian, mingguan, tahunan.

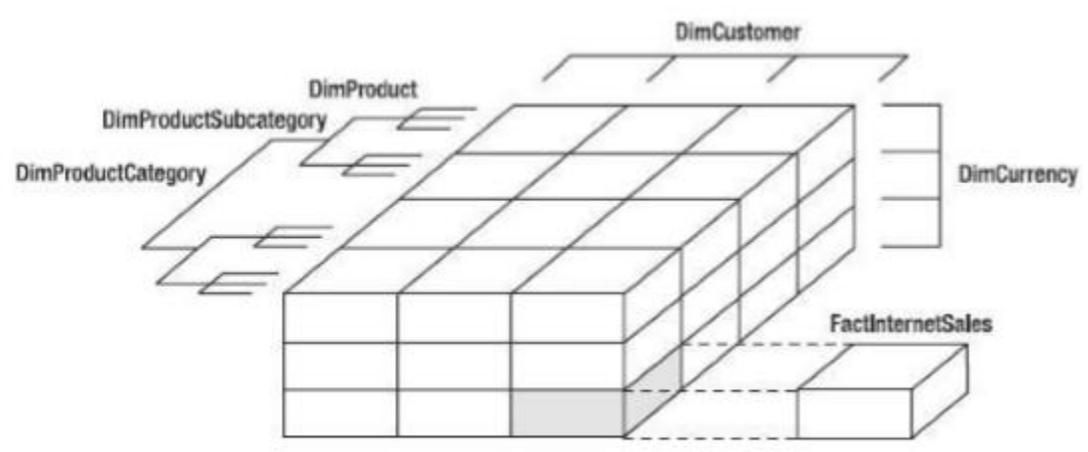
Data cube merupakan objek dua dimensi, tiga dimensi, atau lebih tinggi di mana setiap dimensi data mewakili pengukuran dari sebuah perspektif. *Cube* bisa menjadi sebuah *subset* dari data yang saling terkait yang diatur untuk memungkinkan pengguna menggabungkan atribut apa pun di dalam *cube* (misalnya, toko, produk, pelanggan, pemasok) dengan metrik di dalam kubus (misalnya, penjualan, keuntungan, unit, usia) untuk menciptakan berbagai tampilan dua dimensi, atau irisan, yang bisa ditampilkan di layar komputer.

Ilustrasi dari analisis *cube* dapat dilihat pada gambar 3.5. *Cube* pada gambar ini menggunakan jumlah penjualan sebagai *measure*, dengan 3 dimensi yang digunakan yaitu *Location*, *Product*, dan *Year*. Di sini kita melakukan operasi *slice* terhadap *cube* pada dimensi waktu (*year*), sehingga terbentuk 3 potongan *cube* masing-masing untuk tahun 2006, 2005, dan 2004. Kemudian, pada masing-masing tahun, kita bisa menganalisis penjualan berdasarkan lokasi dan jenis produk. Misalnya pada tahun 2006, kita bisa memperoleh informasi bahwa produk yang paling laris di bagian *West* adalah produk *Bolts* (140 unit).



Gambar 3.5. Analisis *cube*

Inti dari konsep OLAP dan *Analysis Services* adalah kubus (*Cube*). Gambar 3.6. di bawah ini adalah contoh struktur data dari database AdventureworksDW. Analisis data terdapat pada tabel factinternetsales. Elemen yang terdapat pada factinternetsales adalah OrderQuantity, SalesAmount, DiscountAmount dan TaxAmount. Mekanisme untuk melakukan administrasi dan akses nilai ini telah disediakan oleh *Analysis Services*.



Gambar 3.6 *Cube Internet Sales* AdventureWorksDW

Secara umum kita akan memandang *cube* yang terdiri dari 3 dimensi yaitu panjang, lebar, dan tinggi. Namun, *Analysis Services* tidak membatasi jumlah dimensi dari suatu *cube*. Sales dan quantity adalah *measure* yang dapat di baca oleh 5 dimensi secara bersamaan yaitu *dimproduct*, *dimproductsubcategory*, *dimproductcategory*, *dimcustomer* dan *dimcurrency*. Dari *cube* diagram di atas terdapat 3 dimensi (*DimProduct*, *DimCustomer*, dan *DimCurrency*) karena lebih mudah untuk di gambarkan, tetapi bukan berarti *Analysis Services* hanya dibatasi oleh 3 dimensi.

Cells

Tanpa memperhatikan banyaknya jumlah dimensi, *cube* terdiri dari beberapa *cell*. *Cell* merupakan *atomic unit* (unit terkecil) dari sebuah *cube* pada *Analysis Services*. Kita dapat memfokuskan analisis dengan mengidentifikasi *cell* dalam *cube* terhadap dimension. Contohnya, suatu *measure sales* dapat diidentifikasi secara unik dari berbagai perspektif: *customer*, *sales person*, *currency*, maupun *product*.

Measure dan Fact Table

Didalam sebuah *cube*, *measure group* seperti *Factinternetsales* terdapat *quantitative values* seperti *OrderQuantity* dan *SalesAmount*. Selama *value* tersebut bersifat numerik, *measure* dapat di-agregasikan. *Measure* terdapat pada *fact table*, yang mana dalam satu tabel pada database bisa terdiri dari banyak *measures*. *Measurement* tidak hanya terdapat pada *field* tabel dalam sebuah database, tetapi dapat berupa hasil perhitungan (*Sales* tanpa komisi, *sales plus tax*, dan lain-lain).

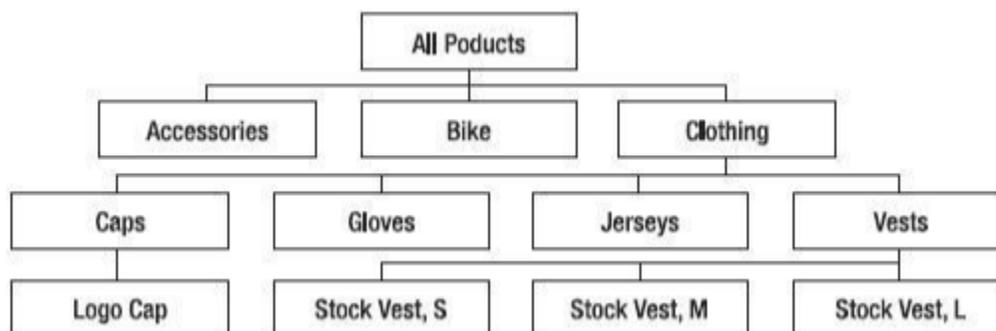
Dimension dan Attribute

Defenisi formal dari dimensi pada *Analysis Services* adalah, “Sebuah dimensi akan mengelompokkan sebuah *measure*”. Dimensi *DimCurrency* akan mengelompokkan *measure* pada *Factinternetsales* pada *Dollar*, *Rupiah* atau *Euro*. Dimensi *DimCustomer* akan mengelompokkan *measure* pada *customer* yang melakukan pembelian terhadap *product*. Setiap dimensi akan memiliki karakteristik tersendiri, misalnya pada *DimProduct*, *Product* memiliki warna merah atau biru. Warna adalah atribut dari sebuah dimensi.

Hierarchies

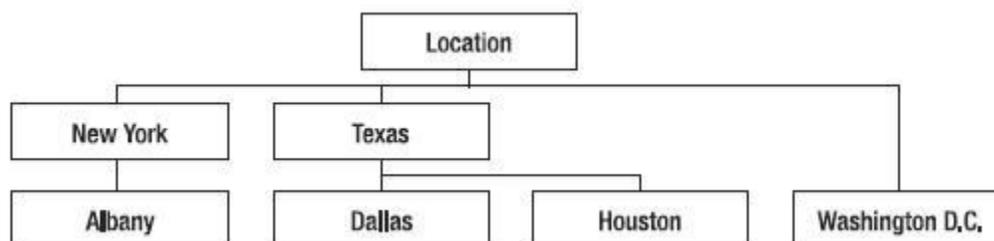
Kita dapat mengurutkan data dalam dimensi pada sebuah *hierarchies*. Pada dimensi *product*, *level* pada *product* dapat berupa *category*, *subcategory* dan *product*. Tabel tersebut merupakan gabungan dari *category*, *subcategory* dan *product* yang telah di jadikan satu dalam satu dimensi. *Hierarchies* dapat di *breakdown* ke dalam bentuk *balanced hierarchies* dan *unbalanced hierarchies*.

Untuk dapat lebih memahami dari *hierarchies*, perhatikan level pada *category*, *subcategory*, dan *product* pada gambar 3.7 berikut. Level paling dasar pada *hierarchies* adalah *product* (logo cap, stockVest S). Data ini menggambarkan level yang paling bawah pada sebuah *hierarchies*. Setiap level memiliki kedalaman data yang sama dari *root* pada sebuah *hierarchies*, sehingga *hierarchies* seperti ini dikatakan *balanced hierarchies*. Dengan demikian *value* dari *sales* dapat di lihat berdasarkan *product*, *subcategory* dan *category*-nya.



Gambar 3.7. Dimensi *Product*, *Balanced Hierarchy*

Untuk *unbalanced hierarchies* dapat di lihat pada *geographics hierarchy* di bawah ini. Dari gambar 3.8 di bawah dapat di lihat bahwa level paling bawah adalah *city*, dan Washington DC berbeda dari beberapa *state* lainnya. Kedalaman antara level ini dan level *root* di atasnya pada *hierarchy* berbeda dengan level lainnya. *Analysis Services 2008* telah mendukung kedua kondisi dari *hierarchy* di atas.

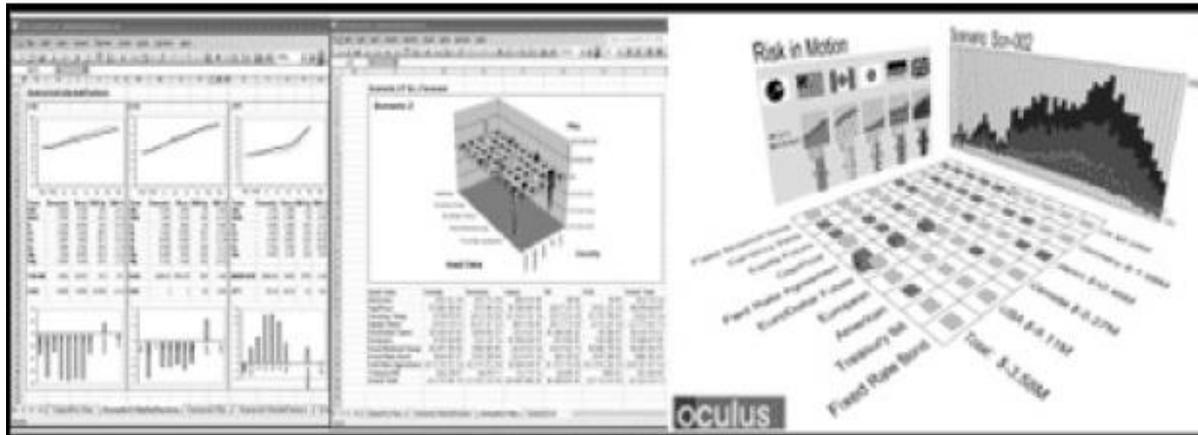


Gambar 3.8 Dimensi Lokasi, *Unbalanced Hierarchy*

3.6. Visualisasi Data

Visualisasi data merupakan suatu bentuk presentasi grafis, animasi, atau video dan hasil analisis data. Visualisasi ini menjadi penting karena kecepatan dalam mengidentifikasi tren

penting dalam data perusahaan dapat memberikan keunggulan kompetitif. Seberapa besar pengaruh dari tren tersebut dapat dianalisis dengan menggunakan model prediktif yang memberikan keuntungan bisnis yang signifikan. Gambar di bawah ini adalah suatu contoh visualisasi yang dibuat untuk analisis risiko.



Gambar 3.9. Visualisasi Data

Geographical information system (GIS)

GIS adalah satu contoh visualisasi informasi yang menggunakan data spasial, seperti peta digital. GIS adalah kombinasi teks, grafik, ikon, dan simbol pada peta. Karena GIS menjadi semakin canggih dan terjangkau, GIS dapat membantu lebih banyak perusahaan dan pemerintah dalam memahami:

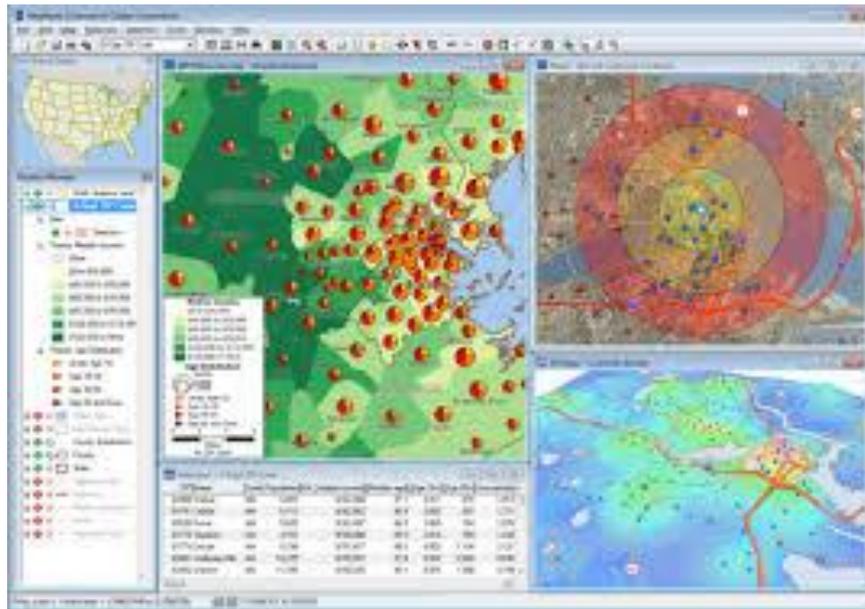
- Lokasi kendaraan, pekerja, dan sumber daya mereka berada
- Kemana mereka harus pergi untuk melayani pelanggan
- Jalur terbaik untuk pergi dari satu lokasi ke lokasi lain

Aplikasi GIS digunakan untuk memperbaiki pengambilan keputusan di sektor publik dan swasta, di antaranya:

- Pengiriman kendaraan darurat
- Manajemen transit
- Pemilihan lokasi fasilitas
- Manajemen risiko kekeringan
- Pengelolaan satwa liar

Pemerintah daerah menggunakan aplikasi GIS untuk pemetaan yang digunakan dan aplikasi pengambilan keputusan lainnya.

Sistem penentuan posisi global (GPS) merupakan perangkat nirkabel yang menggunakan satelit untuk memungkinkan pengguna mendeteksi posisi di bumi item (misalnya mobil atau orang) perangkat dilekatkan pada, dengan ketepatan yang wajar. Sebagian besar vendor perangkat lunak SIG menyediakan akses Web yang terhubung langsung ke perangkat lunak mereka. Beberapa perusahaan menerapkan GIS di Internet untuk penggunaan internal atau untuk digunakan oleh pelanggan mereka dalam menemukan lokasi toko terdekat).



Gambar 3.10, Sistem Informasi Geografis

3.7. Customer Analytics

“The purpose of business is to create and keep a customer”. Pernyataan ini dibuat oleh Peter Drucker, seorang konsultan manajemen di abad 20 (Sauro, 2015). Memang benar bahwa untuk menunjang keberlangsungan hidup perusahaan saat ini, bukan hanya tentang memaksimalkan pendapatan dan meminimalkan biaya. Bagaimana cara untuk menarik pelanggan dan mempertahankannya dengan memenuhi kebutuhan dan keinginan pelanggan merupakan faktor penting bagi perusahaan.

Customer analytics merupakan metode yang dapat digunakan untuk memahami dan memenuhi kebutuhan pelanggan. Dengan demikian, relasi dengan pelanggan dapat terjaga dengan baik. Secara umum, *customer analytics* mencakup aktivitas-aktivitas berikut:

- Pengumpulan data. Mengumpulkan data terkait transaksi yang dilakukan pelanggan, hasil survey, dan observasi terkait pelanggan
- Penggunaan model matematika untuk menemukan pola.
- Mendapatkan wawasan tentang perilaku pelanggan
- Mendukung pengambilan keputusan berdasarkan *history* transaksi
- Memetakan pengalaman pelanggan untuk identifikasi peluang pengembangan

3.8. Real Time BI

Dewasa ini, kecenderungan untuk menggunakan perangkat lunak BI yang menghasilkan update data *real-time* untuk analisis real-time sehingga mendukung pengambilan keputusan secara real-time berkembang pesat. Pemicunya adalah kebutuhan untuk mendapatkan informasi yang tepat dan cepat untuk personil operasional dan taktis sehingga mereka dapat memanfaatkan BA secara *real-time* untuk membuat keputusan. Akan tetapi, terdapat beberapa kekhawatiran tentang sistem *real-time*. Isu penting dalam komputasi *real-time* adalah tidak semua data harus terus diperbarui. Ketika laporan dibuat secara *real-time*, hasil dari seseorang mungkin tidak sesuai dengan orang lain sehingga bisa menimbulkan kebingungan.

Data *real-time* diperlukan dalam banyak kasus untuk pembuatan sistem *Automated Decision Support* (ADS). ADS adalah sistem berbasis aturan yang memberikan solusi untuk masalah manajerial yang berulang. Sistem ini mengotomasi proses pengambilan keputusan, yaitu dengan mengenkapsulasi kepakaran pengguna bisnis ke dalam seperangkat aturan bisnis yang diterapkan pada suatu sistem kerja berbasis aturan. ADS paling sesuai untuk keputusan yang harus sering dibuat dan / atau cepat, dengan menggunakan informasi yang tersedia secara elektronik. Beberapa aplikasi dari ADS yaitu:

- Konfigurasi produk atau layanan
- Pengoptimalan harga
- Keputusan *routing* atau segmentasi
- Kepatuhan perusahaan dan peraturan

- Deteksi penipuan (*fraud*)
- Peramalan dinamis
- Pengendalian operasional

3.9. SQL Server Analysis Services (SSAS)

Jika kita membahas Analysis Services pada SQL Server, maka kita membahas proses *Business Intelligence* (BI). BI dimulai dari proses integrasi data (ETL), dilanjutkan dengan penyajian data, kemudian interpretasi dari data tersebut yang berupa hasil analisis. BI merupakan *business tool* yang efektif untuk mendukung pengambilan keputusan, baik berupa keputusan strategik, taktikal ataupun operasional.

SQL Server 2008 adalah teknologi yang mendukung *development* dan *administration* dari BI *Application*. SQL Server Reporting dan Integration services adalah element dari BI, tapi inti dari BI tersebut adalah SQL Server 2008 Analysis Services (SSAS). *Analysis Services* adalah technology untuk OLAP (*Online Analytical Processing*) dan *Data Mining*. Untuk Development OLAP menggunakan Business Intelligence Development Studio (BIDS). Dengan menggunakan BIDS kita dapat membangun core component dari Analysis Services (Data source, Data Views, Dimension, Value, Cube, Role, dan sebagainya).

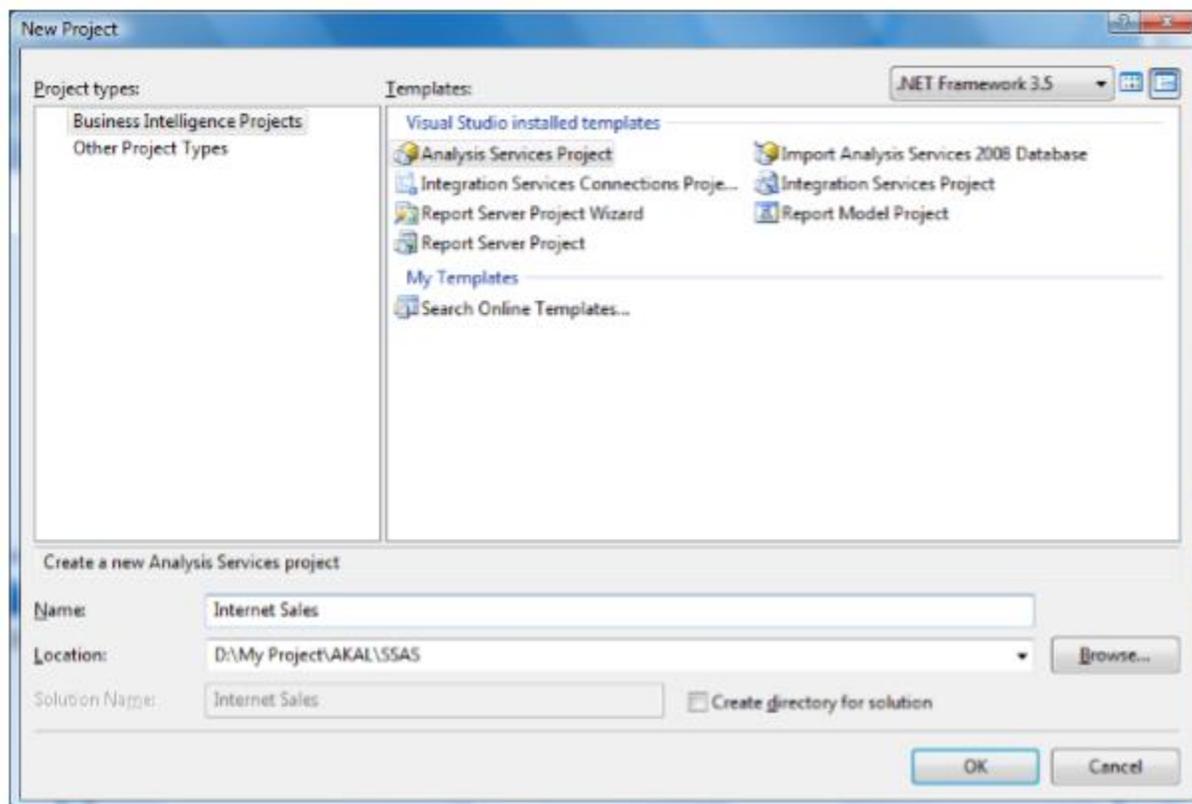
Pada SQL Server 2008, kita dapat menggunakan *datasource* Adventureworks2008 yang baik untuk di pelajari sebagai OLTP database dan AdventureWorksDW2008 sebagai Datawarehouse.

Bekerja dengan *Analysis Services*, Anda perlu memahami terminologi dari OLAP termasuk *Cube*, *Dimension* dan *Measures*. Konsep tersebut berhubungan dengan obyek yang digunakan di dalam *Analysis Services* dan teknologi yang ada hubungannya dengan *Analysis Services*.

Memulai Membuat *Analysis Services Project* baru

Business Intelligence Development Studio (BIDS) mendukung tool yang mendukung dalam pembuatan proyek Analysis Services. Cube diagram pada gambar 2.3 terdapat beberapa dimensi yaitu Dimproduct, DimCurrency dan dan DimCustomer. Cube juga terdapat fact table yaitu Factinternetsales. Kita dapat mengimplementasikan cube tersebut menggunakan Analysis Services project pada BIDS. Kita akan menambahkan dimensi keempat yaitu DimTime.

Untuk memulai Analysis Services project yang baru, buka BIDS (Microsoft SQL Server 2008 → Business Intelligent Development Studio) dan pilih File → New. Anda akan melihat Dialog box New Project, seperti yang di tunjukkan pada gambar 3.11 berikut.



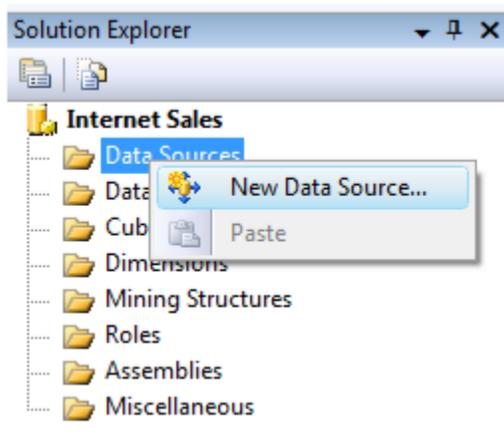
Gambar 3.11. Membuat Analysis Services Project baru

Secara default nama dari project dan solution akan mengandung nama yang generic seperti Analysis Services Project 1. Untuk memudahkan mengenali project kita dapat menuliskan nama project yang lebih spesifik seperti Internet sales pada name text box. Kemudian klik OK. Model yang di gunakan oleh BIDS dapat lebih dari satu project. Untuk skenario ini kita akan menggunakan satu solution dan satu analysis project.

DataSource

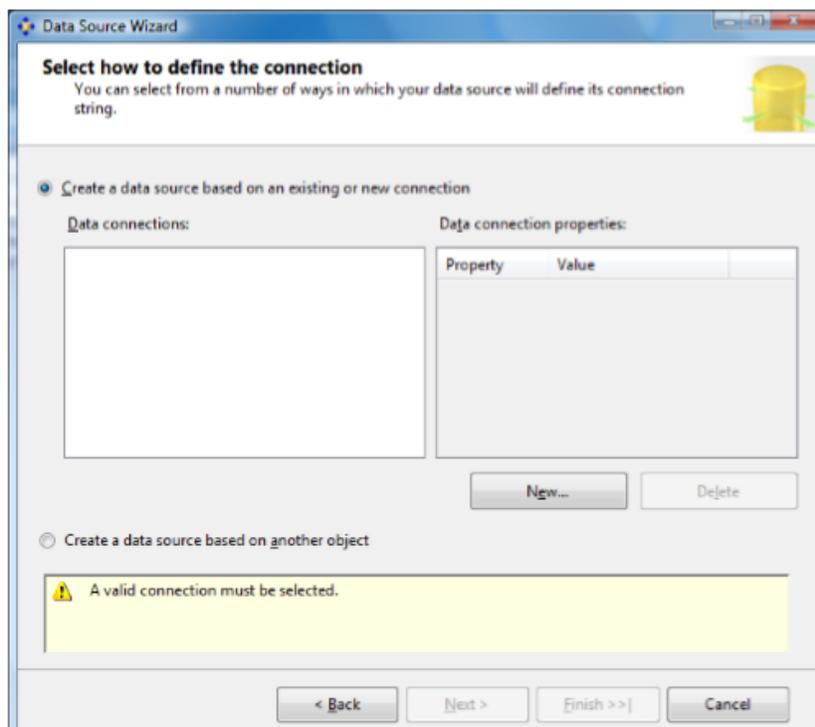
Pada saat Anda membuat analysis services project, anda harus membuat datasourse. Datasource ini akan menjadi acuan ke database mana project anda akan mengakses data. DataSource ini dapat mengakses ke SQL Server atau ke database lainnya dengan menggunakan provider yang sesuai seperti OLE DB atau ODBC. Untuk membuat datasource, pada solution

explorer (View → Solution Explorer), klik kanan datasource folder pada internet sales project dan pilih new datasource pada menu seperti pada gambar 3.12 berikut.



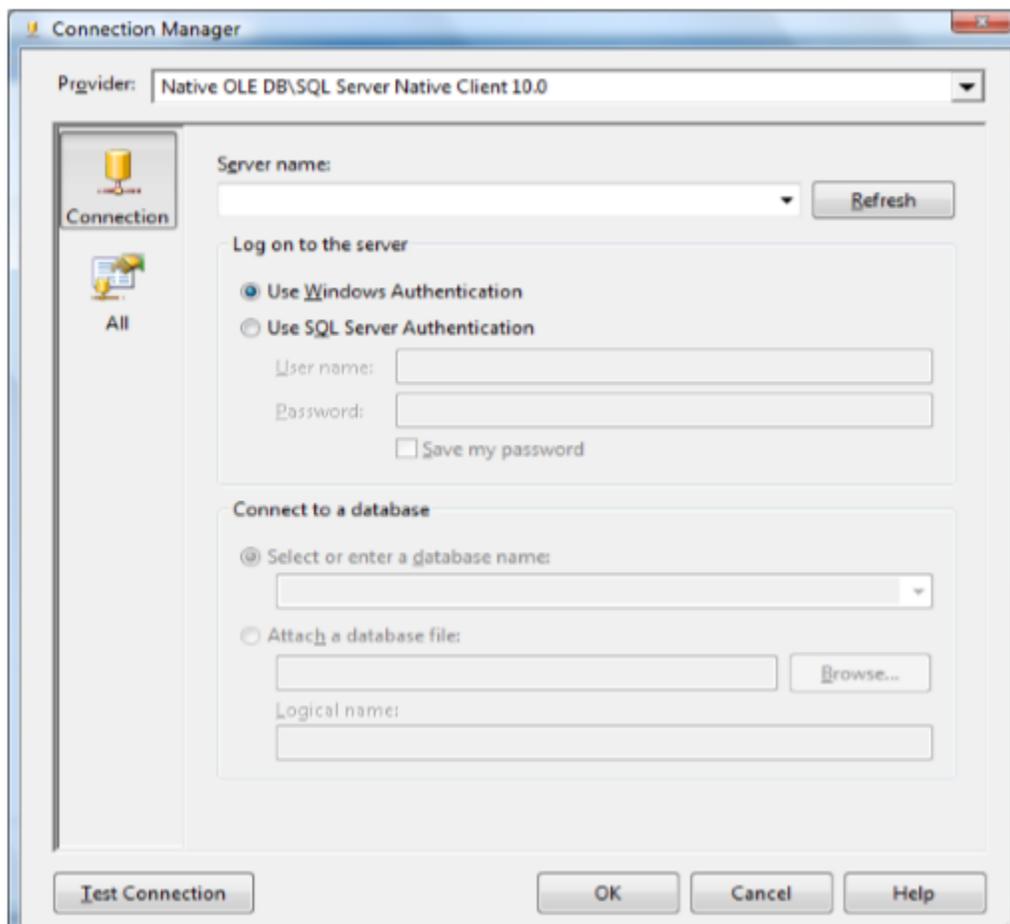
Gambar 3.12 Memilih menu untuk membuat datasource baru

Pilih New Datasource akan muncul Datasource wizard, klik Next untuk melewati welcome screen. Screen selanjutnya akan tampak pada gambar 3.13 berikut. Anda dapat membuat New Datasource di sini.



Gambar 3.13 Mendefenisikan Datasource

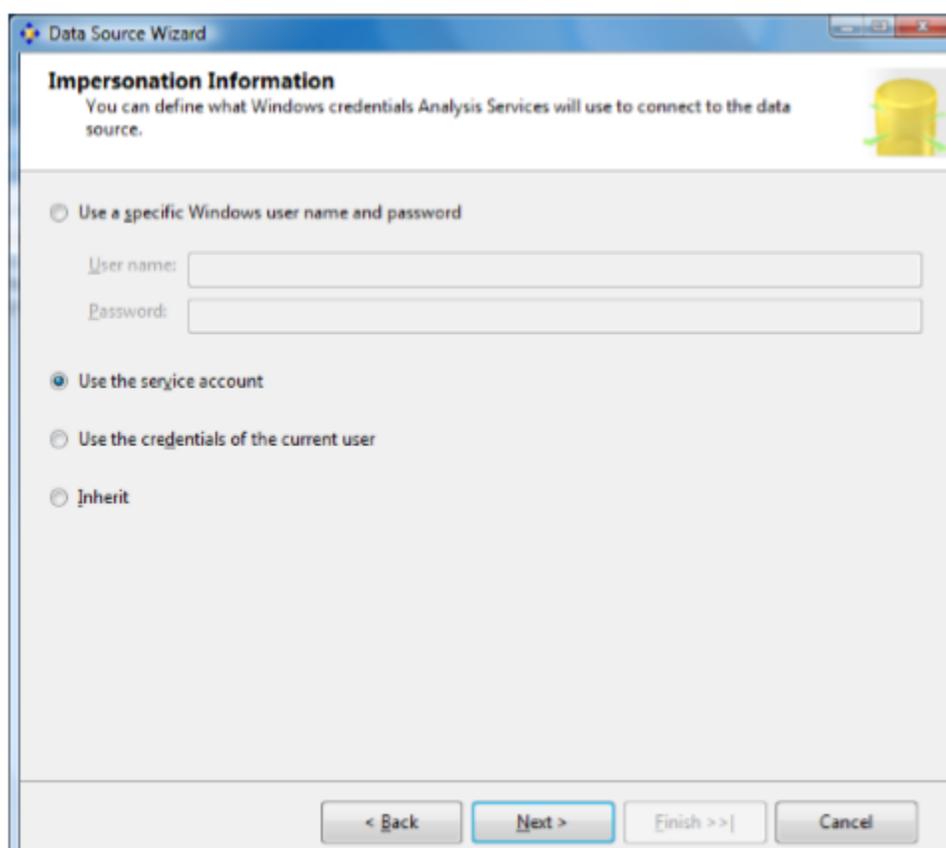
Sebuah datasource merupakan sekumpulan dari sebuah data connection (data provider, host, database, security credential dan sebagainya). Klik New untuk membuat datasource baru.



Gambar 3.14. Membuat Koneksi Datasource baru

Ketika connection Manger muncul seperti pada gambar 3.14 di atas, pilih Native OLE DB\SQL Server Native sebagai default. OLE DB adalah teknologi data access sebelum .NET dan provider yang masih di dukung oleh Microsoft. Pilih nama provider yang akan di gunakan untuk akses ke database dalam hal ini kita pilih Native OLE\DB SQL Server Native Client 10.0. Masukkan nama server database, jika database menggunakan instance maka nama instance harus di sebutkan (servername/instance), gunakan Use Windows Authentication untuk log on ke server database atau dengan menggunakan SQL Server Authentication. Pilih nama database yang akan di gunakan dalam hal ini kita menggunakan data dari database AdventureworksDW2008.

Setelah Konfigurasi connection manager selesai klik OK untuk kembali ke Datasource Wizard kemudian klik next, Anda akan melihat Impersonation menu seperti pada gambar 3.15 berikut. Pada form ini anda di minta untuk menentukan credential yang akan digunakan Analysis Services untuk melakukan koneksi ke datasource. Informasi yang di tampilkan pada menu impersonation sangat penting karena Analysis Server dan user yang tidak bekerja pada lingkungan BIDS pada akhirnya dapat terkoneksi dengan datasource. Bukan cara terbaik untuk memilih “*use the credential of the current user*”. Pada latihan ini kita akan menggunakan pilihan “*Use the services account*”. Dengan memilih opsi tersebut maka account yang berada pada Analysis Services dapat mengakses datasource. Kemudian klik next dan tentukan nama dari Datasource, setelah itu klik finish.

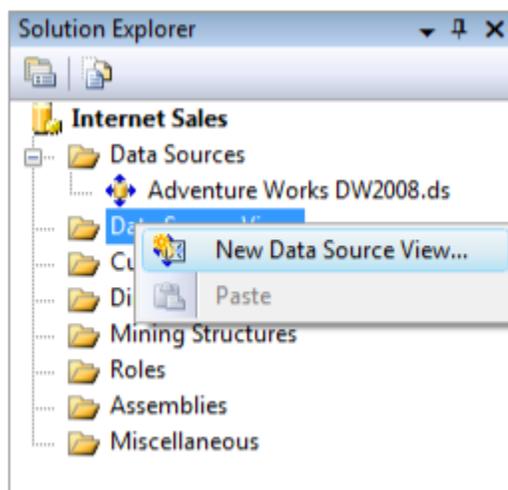


Gambar 3.15. memilih service account impersonation

Data Source View

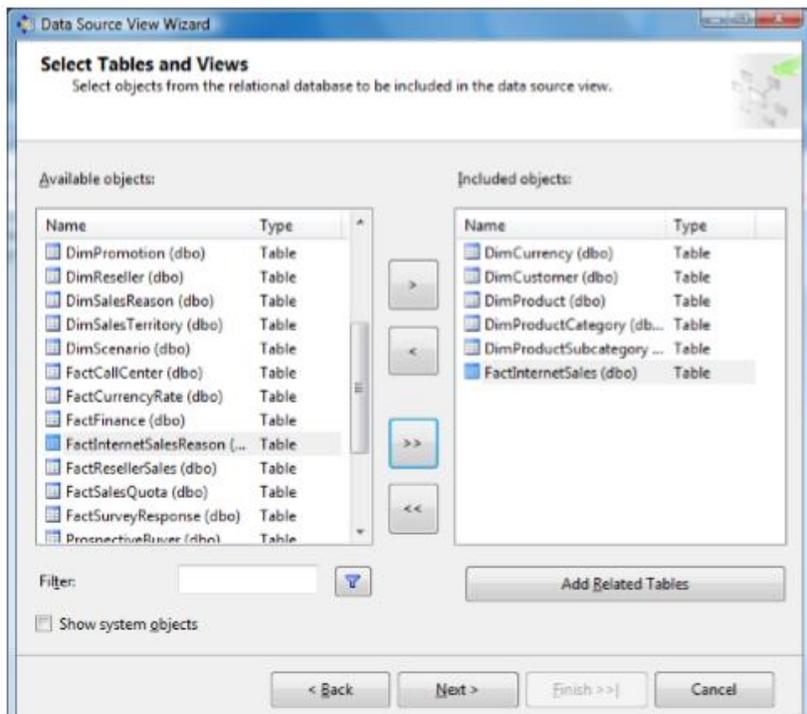
Data Source View (DSV) adalah salah satu bentuk metadata yang merupakan gabungan dari tabel dan view yang di gunakan pada Analysis Services project. Untuk membuat DSV, klik

kanan pada Datasource view folder pada Analysis Services project di dalam solution explorer dan pilih New Datasource View seperti pada gambar 3.16 berikut yang nantinya akan diteruskan pada datasource view wizard. Klik next untuk melewati welcome screen.



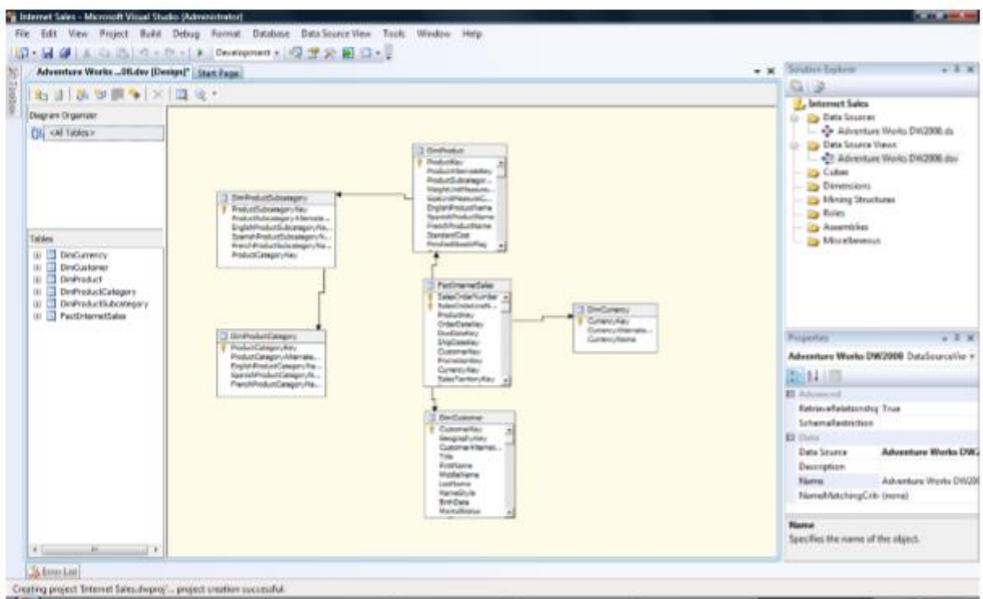
Gambar 3.16. Menu untuk membuat datasource view

Pada screen kedua pada wizard Anda dapat memilih datasource. Untuk contoh ini pilih datasource yang telah kita buat sebelumnya “AdventureWorksDW2008” kemudian klik next untuk memilih tabel dan view yang kita butuhkan. Untuk contoh kali ini pilih dengan cara klik ganda pada DimProductCategory, DimProductSubCategory, DimProduct, DimCustomer, DimCurrency, dan Factinternetsales tabel, untuk memindahkan object tersebut ke list included object. Atau dapat di lakukan dengan cara klik tombol “Add Related Table” akan memindahkan object dari available object yang memiliki foreign key/relasi dengan tabel yang sudah terdapat pada list included object



Gambar 3.17. Memilih table dan view dari DSV

Setelah tabel dipilih untuk ditampilkan pada DSV, kemudian klik Next. Anda dapat merubah nama dari DSV. Kemudian klik finish untuk membuat DSV. Diagram data yang akan di tampilkan pada DSV adalah sebagai berikut.

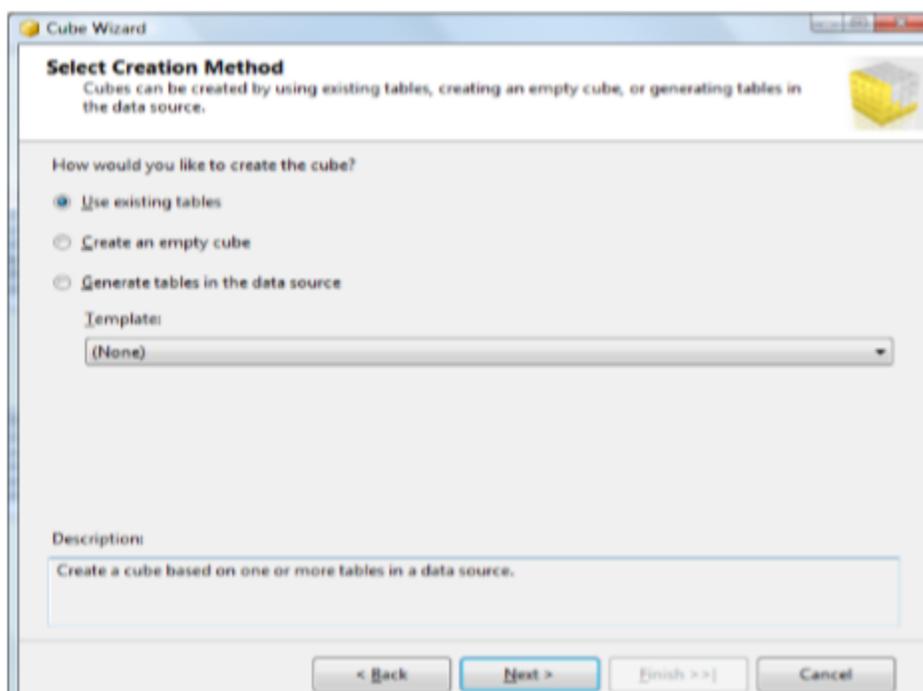


Gambar 3.18. Datasource View Designer

Pada Datasource view designer, Anda dapat memilih sebuah tabel dan berubah properties dari tabel tersebut pada window properties. Misalnya Anda ingin merubah tabel atau view dengan nama unik yang lebih mudah di kenali. Juga anda dapat menambahkan tabel atau view baru dan membuat relasi antara tabel dengan menarik cursor entity pada satu tabel ke tabel lainnya.

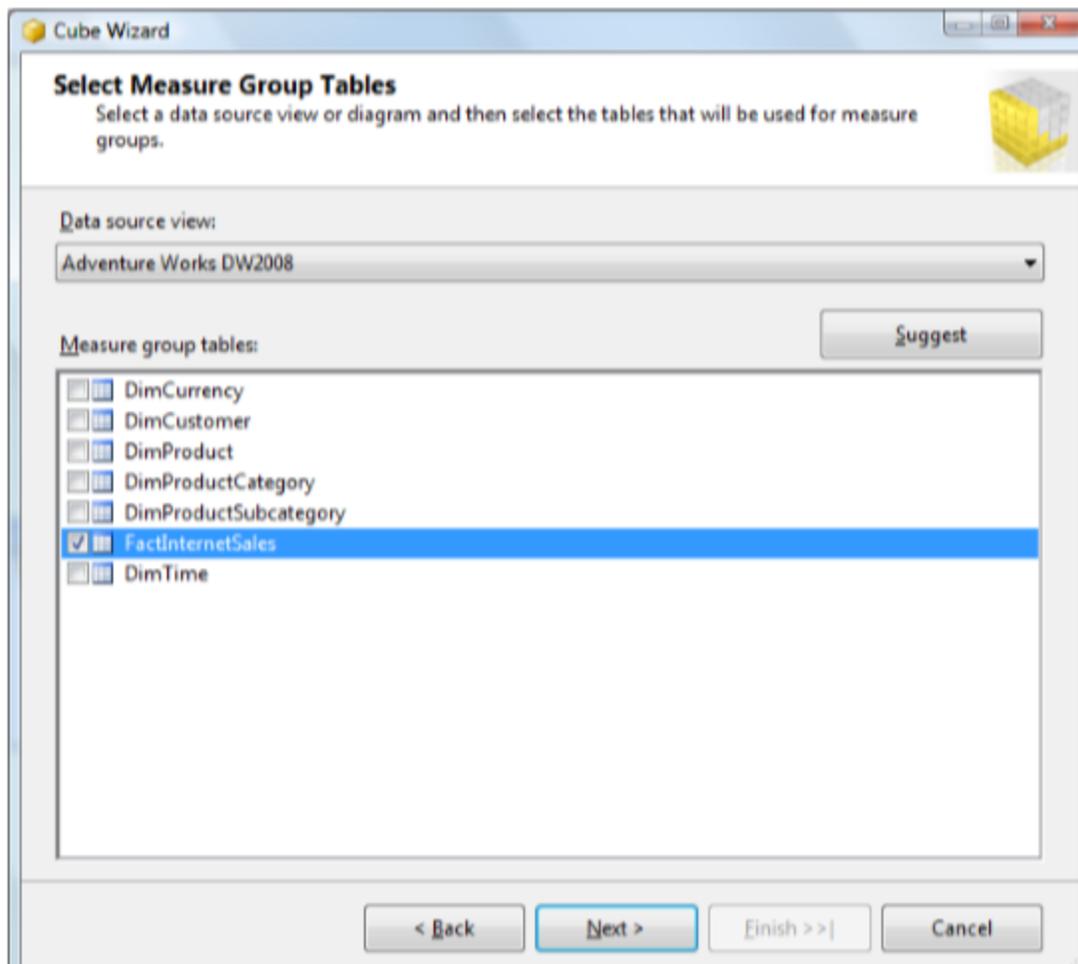
Mendefenisikan Cube

Membuat Cube baru dapat di lakukan dengan klik kanan pada folder cube pada solution Explorer, kemudian pilih New Cube pada menu.Cube memiliki banyak konfigurasi, untuk user baru kita akan menggunakan default, pilih Existing table seperti pada gambar 3.19 berikut.



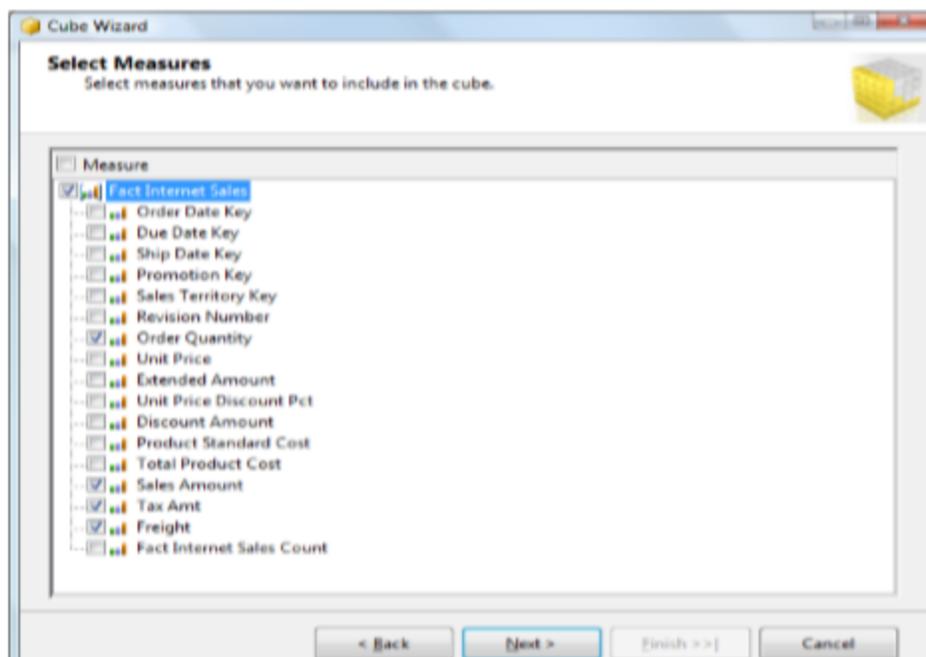
Gambar 3.19. Memilih metode pembuatan cube

Kemudian klik next untuk memilih measure group seperti terlihat pada gambar 3.20 berikut. Di sini Anda dapat memilih beberapa fact table berbeda atau hanya satu. Pada contoh ini pilih tabel FactInternetSales, yang memiliki field yang sangat potensial dijadikan measures (OrderQuantity, UnitPrice, SalesAmount, TaxAmount, Freight, dan Totalproductcost). Sebuah Fact table dapat terdiri dari satu measure atau lebih. Jika anda tidak yakin mana tabel yang dapat di jadikan fact table kamu dapat klik tombol suggest. Wizard akan memilih kandidat tabel yang dapat dijadikan fact table untuk anda review.



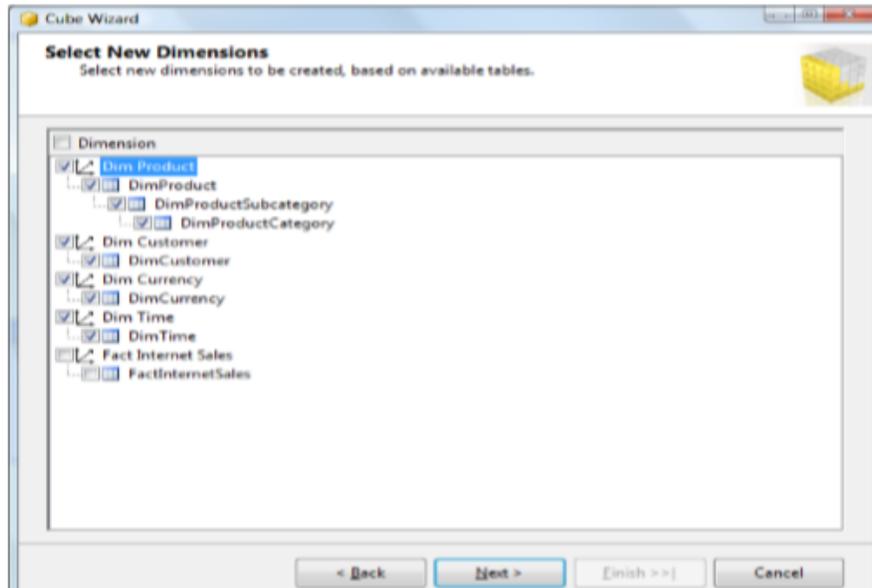
Gambar 3.20. Memilih Fact table

Setelah Anda memilih *fact table*, klik next. Pada screen selanjutnya anda dapat memilih field yang akan di jadikan measures dalam cube. Pada contoh kali ini pilih Order Quantity, Sales Amount, Tax Amount dan Freight (Sales Amount adalah total dari cost of goods sold/COGS yang diambil kedalam nilai Quantity, Price dan Discount). Kemudian *uncheck* kolom yang lainnya. Klik next untuk melanjutkan.



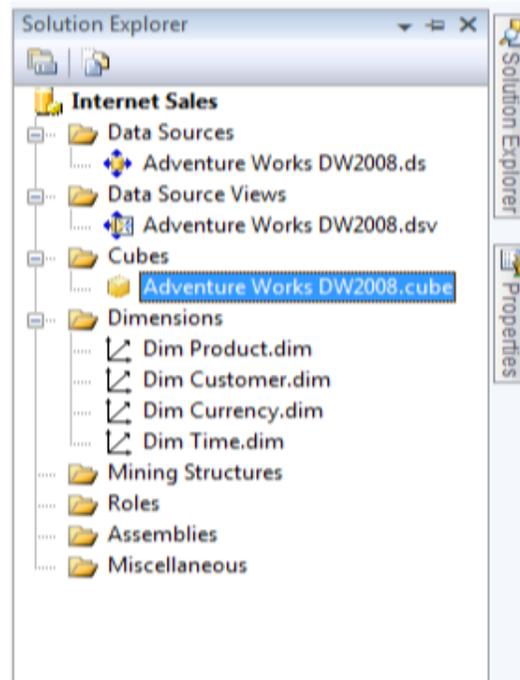
Gambar 3.21. Memilih Measure

Pada screen selanjutnya Anda dapat memilih dimensi yang digunakan. Pada gambar 3.22 pilih DimCurrency, Dim Customer, DimTime yang nantinya akan di tampilkan sebagai dimensi pada cube. Pada tabel tersebut tidak terdapat hierarchy, dimensi tersebut terdiri dari satu tabel. Dimensi ke-empat akan memiliki hierarchy secara automatic yang terdiri dari DimproductCategory, DimProductsbucategory dan Dimproduct. Cube wizard menyediakan hierarchy tersebut untuk kita, karena pada datasource view kita memiliki relasi dari factinternetsales ke Dimproduct kemudian ke DimProductSubcategori dan DimProductCategory. Untuk tabel Factinternetsales tidak dipilih.



Gambar 3.22. Memilih dimensi, Attribute dan Hierarchy

Untuk menyelesaikan cube wizard berikut klik Next kemudian klik Finish. Cube secara default akan mengambil nama dari DSV. Berikut cube dan dimensi yang terbentuk dari cube wizard pada gambar 3.23.

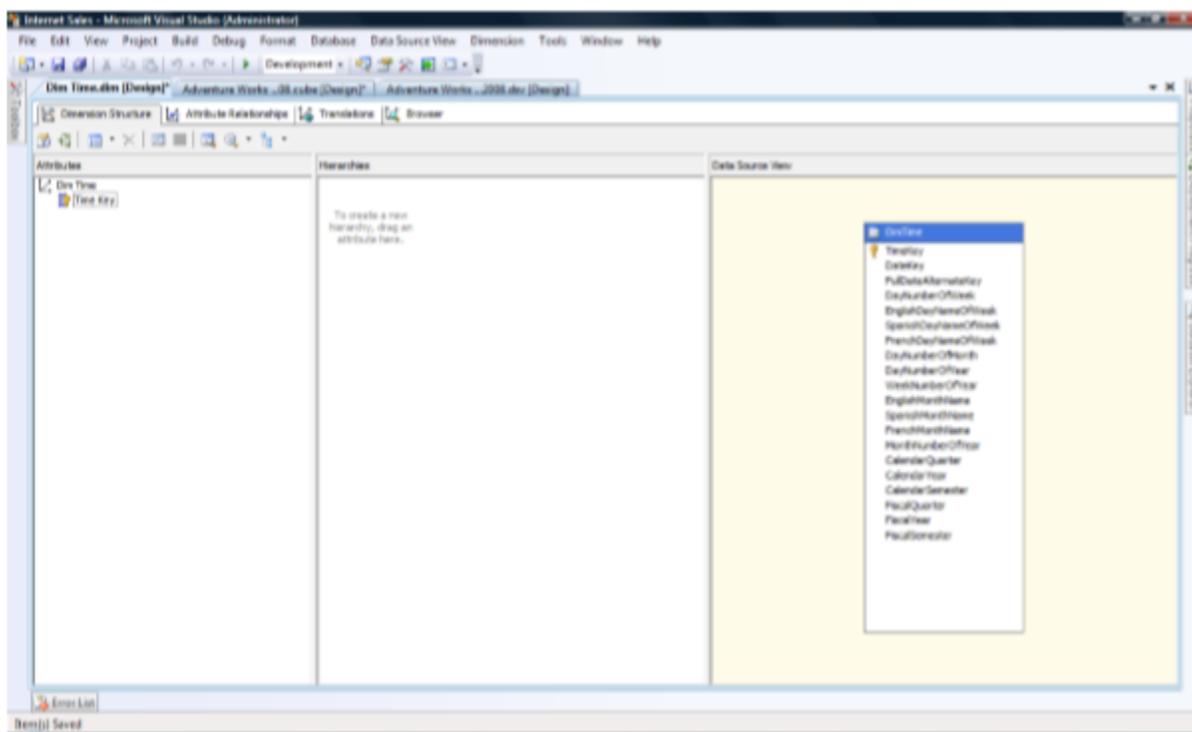


Gambar 3.23. View Solution Explorer dengan cube dan dimension

Konfigurasi Dimensi

Setelah Anda selesai dengan cube wizard, maka wizard akan membuat Dimensi. Dimensi yang dibentuk oleh wizard adalah dimensi dengan konfigurasi *default*. Anda harus melakukan konfigurasi ulang terhadap dimensi. Anda dapat melakukannya pada dimension designer. Pada dimension designer, Anda dapat mengubah properties, menambahkan attribute, mendefinisikan hierarchy baru, mendefinisikan attribute relationship, dan menambahkan local translation jika di butuhkan. Untuk advance user anda dapat membuat dimensi secara manual. Jika anda melakukan ini anda harus menambahkan dimensi ke cube secara manual.

Untuk melakukan konfigurasi dimensi, klik kanan pada dimensi di dalam solution explorer kemudian select open. Pilih tabel dimTime maka akan muncul dimensi designer seperti pada gambar 3.24 berikut.



Gambar 3.24. Dimension Designer

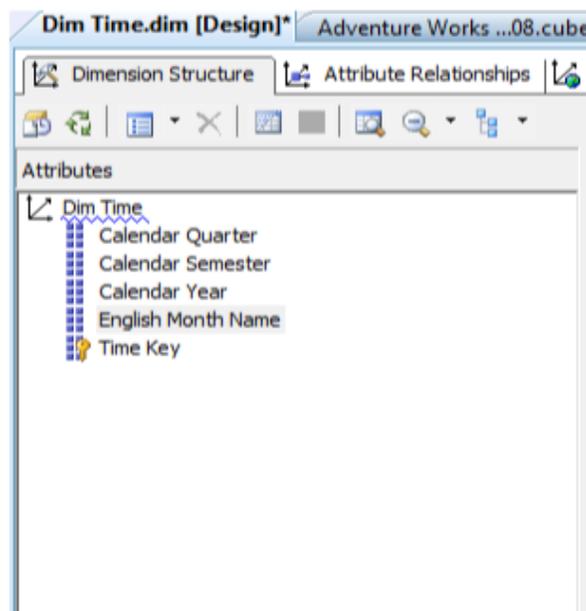
Pada Demension Designer terdapat 4 tab. Pada tab Dimension Structure, anda dapat memilih atribut dari dimensi dan menyusun atribut tersebut dalam hierarchy. Pada tab Attribute Relationship, Anda dapat mengatur relationship antara attribute. Relationship ini membantu anda pada saat melakukan penelusuran member dari sebuah hierarchy dari dimensi tersebut. Pada tab

Translation, Anda dapat menambahkan ataupun merubah metadata ke dalam bahasa yang berbeda. Tab Browser digunakan untuk melihat hasil dari dimensi yang telah kita process.

Menambahkan Attribute

Pada contoh ini, sebelum kita membuat hierarchy yang berguna, kita membutuhkan beberapa attribute lagi. Jika kita lihat pada gambar 3.27 hanya terdapat satu attribute yaitu datekey. Secara default hanya attribute pada wizard yang akan ditambahkan, dalam hal ini dimension key. Menambahkan attribute baru dapat dilakukan dengan cara drag attribute pada pane datasource view pada dimension designer yang berada paling kanan ke attribute pane di sebelah paling kiri. Drag attribute berikut: CalendarYear, CalendarSemester, CalendarQuarter, dan EnglishMonthName. Anda dapat memindahkannya tanpa sesuai urutan, atau pilih semua attribute dengan menahan tombol shift dan kemudian klik attribut yang diinginkan kemudian pindahkan ke attribute pane.

Anda dapat merubah nama dari atribut Englishmonthname yang lebih mudah di kenal dengan cara klik kanan pada attribute yang bersangkutan dan pilih rename menu. Misalnya anda merubah attribute EnglishMonthName menjadi Month. Anda dapat juga merubah key dari datekey ke simple date. Penggantian name attribute akan memudahkan user pada saat melakukan browse cube dan lebih memahami data yang disajikan. Hasil dari create attribute seperti pada gambar 3.25 berikut.

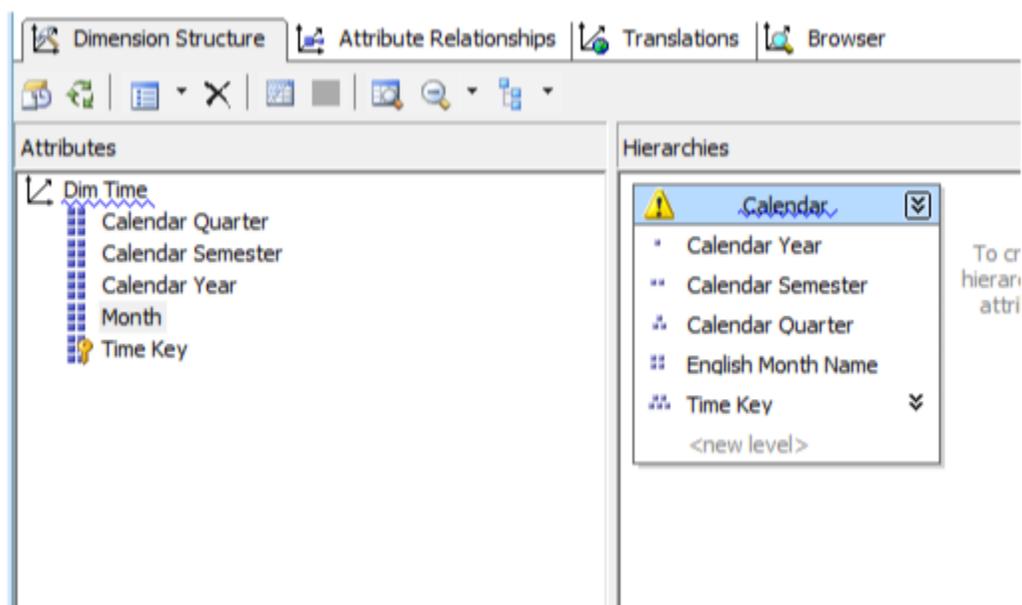


Gambar 3.25. Hasil penambahan *attribute*

Pada dimensi DimTime kita melihat terdapat garis biru yang berlekuk di bawah nama dari dimensi DimTime. Ini adalah peringatan *best practice*. Pindahkan kursor mouse Anda ke atas garis tersebut, maka akan muncul tooltip yang memberikan peringatan sebagai berikut “*Create Hierarchy in non-parent child dimensions*”. Pada umumnya Anda tidak harus menghiraukan dari peringatan tersebut, dan Anda dapat meneruskan design cube dan dimensi tanpa membuat hierarchy. Namun sebenarnya peringatan tersebut dapat menolong anda untuk desain yang lebih baik.

Membuat Hierarchy

Sekarang anda telah memiliki attribute pada dimensi DimTime. Sekarang saatnya anda membuat hierarchy untuk dimensi DimTime tersebut. Anda dapat memindahkan attribute dari atas atau dari bawah sesuai dengan urutan level dari sebuah hierarchy ke dalam hierarchy pane. Ada baiknya anda melakukannya sesuai dengan urutan level walaupun nantinya anda dapat merubah urutan dari levelnya. Pindahkan semua attribute satu demi satu untuk membentuk suatu hierarchy. Anda dapat merubah semua attribute maupun header space pada attribute yang lebih mudah untuk dikenali seperti calendar. Anda dapat melihatnya seperti pada Gambar 3.26 berikut.

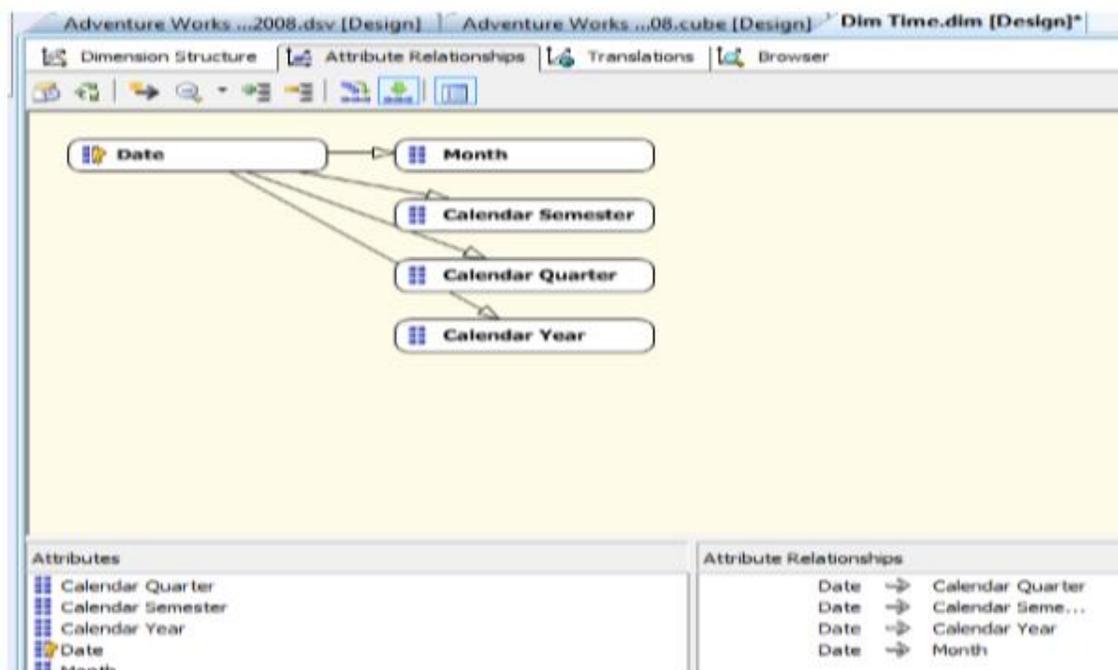


Gambar 3.26. Membuat hierarchy

Attribute Relationship

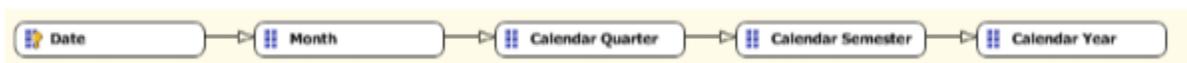
Pada Hierarchy calendar yang telah kita buat terdapat tanda <!> dan garis biru berlekuk di bawah nama hierarchy. Pindahkan mouse anda ke area tersebut akan muncul tooltip “*Attribute*

relationship does not exist between one or more levels of this hierarchy, this may result in decreased query performance". Pada tooltip ini Anda disarankan untuk membuat relationship dari hierarchy yang anda buat. Ini merupakan saran untuk desain yang baik. Klik attribute relationship tab untuk mendesain attribute relationship.



Gambar 3.27. mendefinisikan attribute relationship

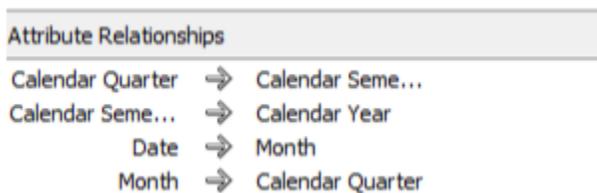
Pada gambar 3.27 di atas, anda dapat melihat seluruh attribute berelasi ke key dari dimensi DimTime yaitu date, tetapi tidak satupun dari atribut tersebut memiliki relasi ke lainnya. Hal ini mengakibatkan dimensi sulit untuk di navigate. Anda dapat melakukan drill down dari calendaryear to date, tetapi tidak ada relasi dari calendaryear ke calendarsemester. Untuk membuat relationship untuk memudahkan pada saat navigasinya, anda dapat melakukannya dengan cara yang sederhana dengan cara drag and drop atribut pada view ini ke dalam atribut yang akan menjadi parent-nya. Jadi, drag calendarsemester ke dalam calendaryear, calendarquarter ke dalam calendarsemester, dan month kedalam calendarquarter. Sekarang hierarchy anda akan tampak seperti gambar 3.28 berikut.



Gambar 3.28. Perubahan attribute relationship

Kita masih belum selesai sampai di sini. Kenapa? Coba anda kembali ke tab dimension structure, anda akan melihat garis biru yang melekok-lekok masih ada di bawah nama dimensi pada pane attribute. Peringatan yang di tampilkan adalah “*Define Attribute relationship as ‘Rigid’ where appropriate*”. Dari peringatan ini menerangkan bahwa attribut relationship yang telah kita buat adalah flexible, tetapi seharusnya adalah rigid. Rigid Attribute relationship adalah suatu relasi yang kita perkirakan tidak akan berubah. Seperti pada time dimension yang telah kita buat bulan Januari akan selalu berada pada bagian dari CalendarQuarter 1, tidak mungkin bulan Januari berada pada CalendarQuarter 2.

Kembali lagi ke tab attribute relationship dimana anda dapat melakukan perubahan attribute relationship. Pada bagian bawah dari tab attribute relationship anda dapat melihat attribute relation pane seperti pada gambar 3.29 berikut.



Gambar 3.29. Struktur attribute relationship

Anda dapat memilih semua dari relationship, klik kanan dan pilih properties, pada properties windows, ubah properties relationship type dari flexible ke rigid. Seluruh attribute relationship akan merubah menjadi rigid. Jika anda tidak merubah relationship ke dalam bentuk rigid, setiap kali Analysis Services di proses incremental update dari data, Analysis services akan menghapus aggregation. Anda harus melakukan proses lagi secara manual. Sehingga pada Analysis services yang menyimpan rigid aggregation, proses yang berulang-ulang tidak diperlukan.

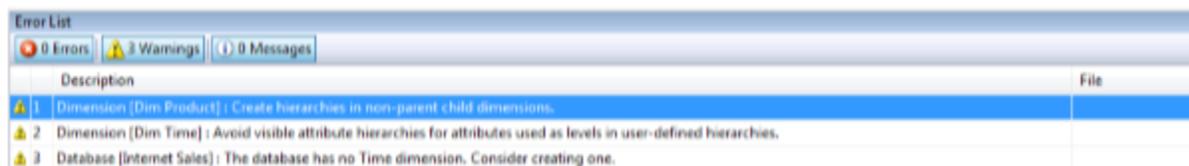
Membuat Attribut Hierarchy invisible

Sekarang kita sudah hampir menyelesaikan konfigurasi dari Dimensi Time. Pada Dimensi Designer klik tab dimension structure. Anda masih melihat garis biru melengkung dibawah nama dimensi bukan? Pindahkan cursor Anda ke nama dimensi DimTime. Peringatannya adalah “*Avoid visible attribute hierarchies for attribute used as level in user defined hierarchies*”. Peringatan ini muncul karena user mungkin akan merasa bingung jika attribute dari dimensi di tampilkan dengan cara pandang yang berbeda. Pada hierarchcy Calendar akan muncul attribute calendaryear dan di

dimensi DimTime juga akan muncul calendaryear. Mungkin dengan kondisi tertentu hal ini dapat diterima. Misalnya seorang user ingin menganalisa hasil penjualan di bulan November atau Desember pada semua tahun, untuk mengevaluasi penjualan pada session liburan. Dengan kondisi menggunakan attribute month pada hierarchy calendar hanya bisa menelusuri data December 2004 atau December 2005. Untuk kasus di atas kita bisa mendisable seluruh attribute pada Dimensi DimTime (bukan attribute pada hierarchy) kecuali attribute Month. Untuk melakukannya pilih attribute-attribute yang akan di invisible pada attribute pane, klik kanan dan pilih properties (klik F4), pada properties windows rubah nilai Attributevaluevisible menjadi false.

Menghilangkan Best Practice Warnings

Anda masih khawatir apabila masih ada pesan warning dibawah nama dimensi. Pada kasus ini kita akan menghilangkan warning pada Month attribute pada dimensi DimTime. Select Build → Build Solution pada BIDS, ketika proses build selesai Anda dapat melihat build succeeded pada status bar di bagian bawah screen. Buka error list window dengan cara klik view → error list, seperti yang anda lihat pada gambar 3.30 berikut. Beberapa warning yang berhubungan dengan dimensi dapat Anda ubah konfigurasinya, sama seperti pada dimensi DimTime.



Gambar 3.30. Best practice warning

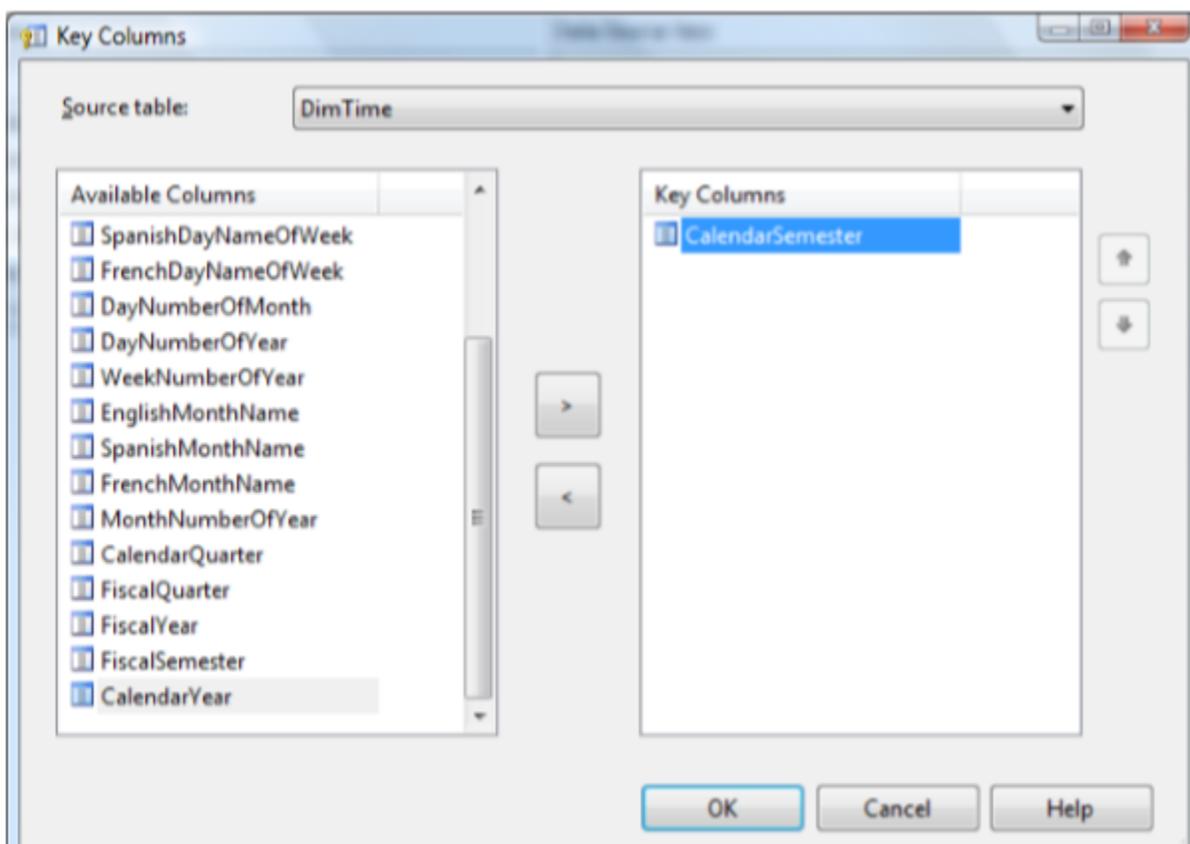
Klik kanan pada warning “Dimension [Dimtime] : *Avoid visible attribute hierarchy for attributes used as level in user defined hierarchy*”, dan kemudian pilih Dismiss. Pada menu peringatan Dismiss warning, Anda dapat menghilangkan hanya dengan menekan tombol OK, tetapi lebih baik anda memberikan pesan mengapa anda menghilangkan warning ini sebagai referensi anda nantinya. Tambahkan catatan kecil pada kotak pesan sebagai berikut “*allowing multiple views of month*” dan kemudian klik OK untuk menghilangkan pesan warning tersebut. Anda akan melihat garis biru melekung di bawah nama dimensi DimTime Menghilang.

Lihat kembali gambar 3.33 di atas, anda akan melihat pesan peringatan lainnya “*The database has no time dimension. Consider creating one*”. Apakah kita sudah membuat time dimensi? Ya, tetap Analysis Services tidak mengetahuinya. Untuk menandai bahwa dimensi yang

kita buat adalah sebagai time dimensi dengan cara sebagai berikut. Klik kanan nama dimensi (DimTime) pada attribute pane pada tab struktu dimensi dan kemudian pilih properties. Pada properties window temukan property Type dan ubah dari type regular menjadi Time. Kemudian ubah type dari calendaryear, calendar semester, calendarquarter, month, TimeKey secara berurutan dengan cara klik kanan pada attribut pilih properties → type → date → calendar → [year]/[Halyears]/[quarterofyear]/[monthofyear]/[date]. Untuk melihat apakah ini efektif, Anda dapat menjalankan “build →build solution” dan lihat pada jendela error list, warning tersebut telah menghilang.

Key Coloum Attribute

Kalau anda perhatikan data pada attribute calendarsemester akan memiliki nilai 1 dan 2, calendarquarter akan memiliki nilan 1, 2, 3, dan 4, dan attribute lainnya akan memiliki nilai yang sama setiap tahun. Klik attribute calendarsemester → klik kanan → properties → keycolumn, anda akan melihat gambar 3.31 seperti berikut.

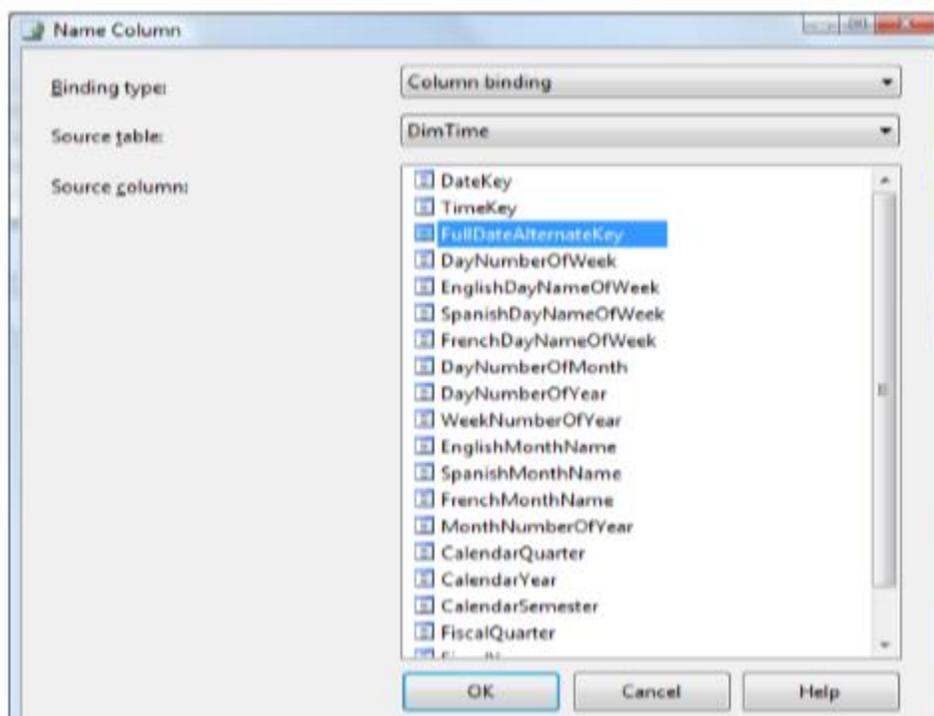


Gambar 3.31. Memilih key column pada dimensi

Anda dapat melihat bahwa Attribute calendarsemester memiliki keycolumns calendarsemester. Jika key column hanya calendarsemester maka keycolumn akan duplikat, misanya data 2004 memiliki calendarsemester 1 dan 2 begitu juga tahun 2005 dan seterusnya. Agar key columns unique maka tambahkan column calendaryear pada box availabel column ke box key columns. Lakukan hal yang sama untuk attribute calendarquarter dan month.

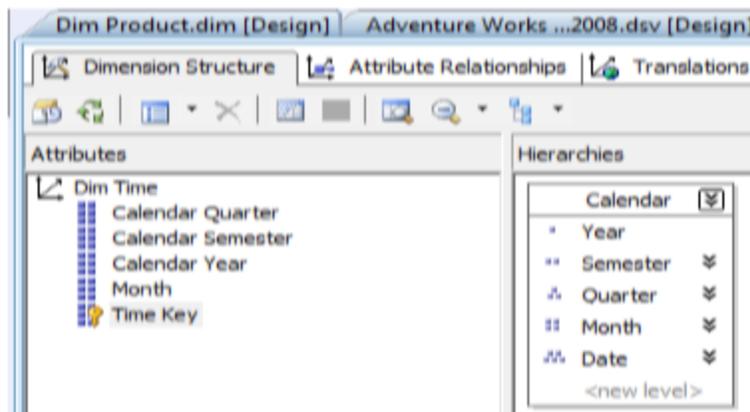
Configuring NameColumn

Kalau kita tidak melakukan konfigurasi nilai Namecolumn pada attribute maka secara default value dari Namecolumn akan sama dengan value dari keycolumns. Best practice untuk membuat hierarchy adalah gunakan keycolumn (Tipe data numeric) pada dimensi untuk attribute pada level paling akhir. Pada dimensi DimTime attribute keycolumn dari Timekey adalah timekey yang akan menggantikan nilai date. Untuk itu kita perlu menggantikan value dari namecolumn pada attribute timekey menjadi date dengan cara klik kanan pada attribute timekey pilih properties → namecolumn → Fulldatealternatekey seperti terlihat pada gambar 3.32 berikut.



Gambar 3.32. Memilih name column

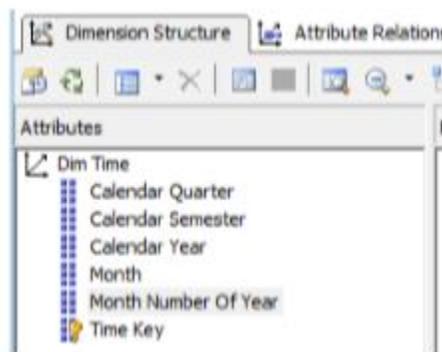
Klik OK, kemudian ubah name TimeKey pada hierarchy calendar menjadi date dengan cara klik kanan attribute timekey (F2) → rename → ubah menjadi date seperti pada gambar 3.33 berikut.



Gambar 3.33. Hasil perubahan hierarcy

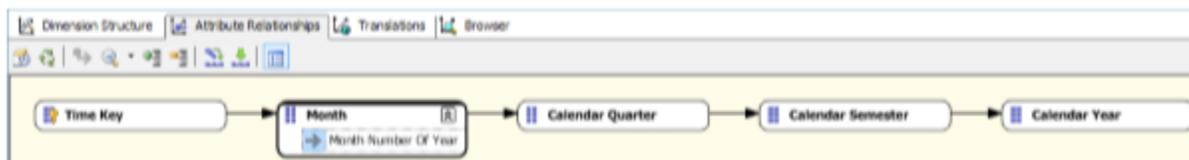
Sorting Attribute Member

Anda ingin mengurutkan nama bulan dari Januari sampai Desember, secara default Analysis Services akan mengurutkan member dari suatu attribute berdasarkan abjad. Apa jadinya hierarchy calendar Anda? Member Month dari hierarchy calendar adalah sebagai berikut Febuary, January dan March karena di-sort berdasarkan abjad. Lalu bagaimana anda dapat mengurutkan sesuai dengan urutan bulan yang tepat? Anda membutuhkan nama bulan berdasarkan nilai numerik untuk mengurutkannya. Klik kanan attribute month pada pane attribute pilih properties (F4), pada properties order by terdapat 4 pilihan yang dapat kita pilih (key, name, attribute key, attribute name). Kita membutuhkan attribute key yang bertipe numeric untuk attribute Month, tambakan column monthnumberofyear dari pane datasourceview ke pane attributes, ubah properties attributehierarchyvisible menjadi false pada attribute Month Number of Year.



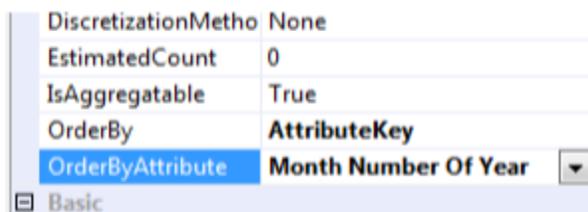
Gambar 3.34. Penambahan attribute MonthNumberofYear

Buka tab attribute relationship, pindahkan attribute MonthNumberofyear ke dalam attribute relationship Month dengan cara drag attribute relationship Month ke attribute MonthNumberofyear seperti terlihat pada gambar 3.35 berikut.



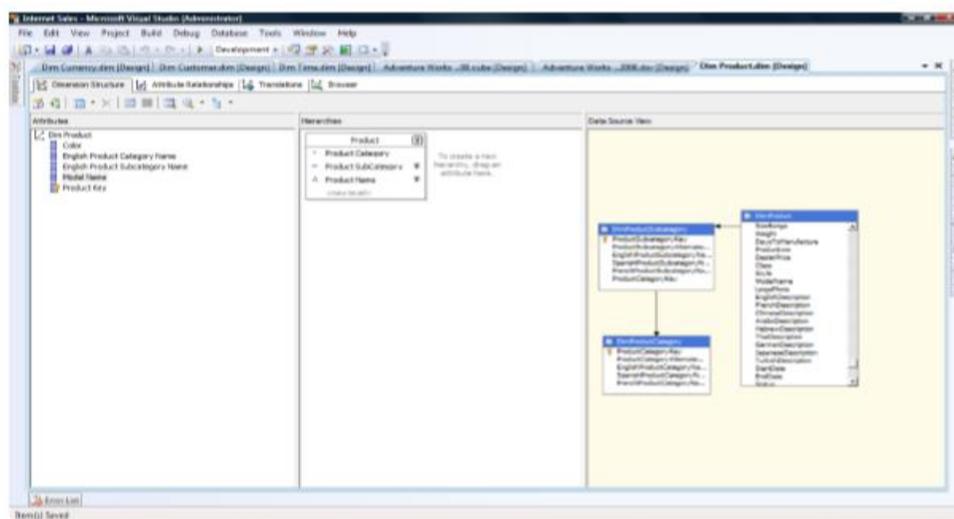
Gambar 3.35. Menentukan attribute relationship untuk attribute month

Kembali lagi ke tab dimension structure, klik kanan attribute month pilih properties, rubah properties order by ke AttributeKey dan rubah properties Order by attribute menjadi MonthNumberofyear seperti terlihat pada gambar 3.36 berikut.



Gambar 3.36. Merubah properties untuk fungsi sortin

Anda dapat melakukan hal yang sama untuk dimensi lainnya. Berikut konfigurasi dari Dimensi product.



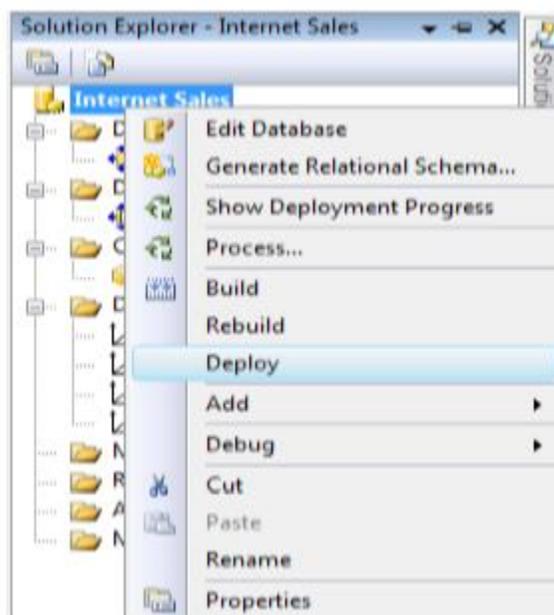
Gambar 3.37. Dimensi Product Designer

Deploying Project

Kita harus men-deploy sample project yang telah kita buat untuk menunjukkan beberapa advanced feature pada saat kita membangun project analysis services. Hal penting yang harus di sadari bahwa metadata project yang kita bangun bersifat local yang merepresentasikan object-object analysis services yang akan kita manage. Mendeploy sebuah project akan membangun cube structure pada Analysis Services server.

Deploy a Project

Anda dapat mendeploy project yang telah anda bangun dengan cara pilih “Build → Deploy Solution of Build project name”. Atau anda juga dapat melakukannya dengan klik kanan pada project di dalam solution explorer dan pilih deploy dari menu yang ada, seperti yang dapat anda lihat pada gambar 3.38 berikut.



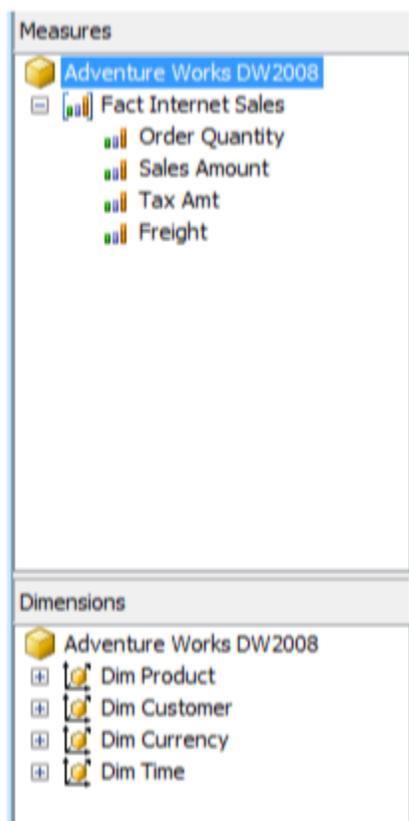
Gambar 3.38. Deploy analysis services project

Proses development tidak akan selesai seketika itu juga. Anda dapat melihat status dari process deployment menggunakan BIDS output window (view → other window → output). Ketika anda memilih Build → Deploy Solution, semua project yang akan di-deploy pada solution akan di-deploy. Proses deploy tidak akan membuat data yang ada akan terbentuk dalam cube yang kita bangun. Data yang ada tidak bisa kita tampilkan sampai cube diproses. Ketika object dari analysis

services diproses, maka Analysis Services akan menyalin data dari sumber data ke dalam objek kubus.

Menampilkan Struktur Kubus

Secara default, perancang kubus akan menampilkan tab struktur kubus, tab ini terdiri dari ukuran dan dimensi seperti yang terlihat pada gambar 3.39 berikut.



Gambar 3.39. Struktur Kubus

Pada jendela ukuran, Anda dapat mengklik pada fact internet sales dan menambahkan ukuran baru seperti unit cost dan total product cost. Dengan menggunakan menu yang ada pada tab struktur kubus Anda dapat menambahkan ukuran grup yang berupa sekumpulan dari beberapa bahkan ratusan ukuran yang telah dikategorikan sesuai dengan peran bisnisnya. Seperti cost measure group, tax measure group atau income measure group.

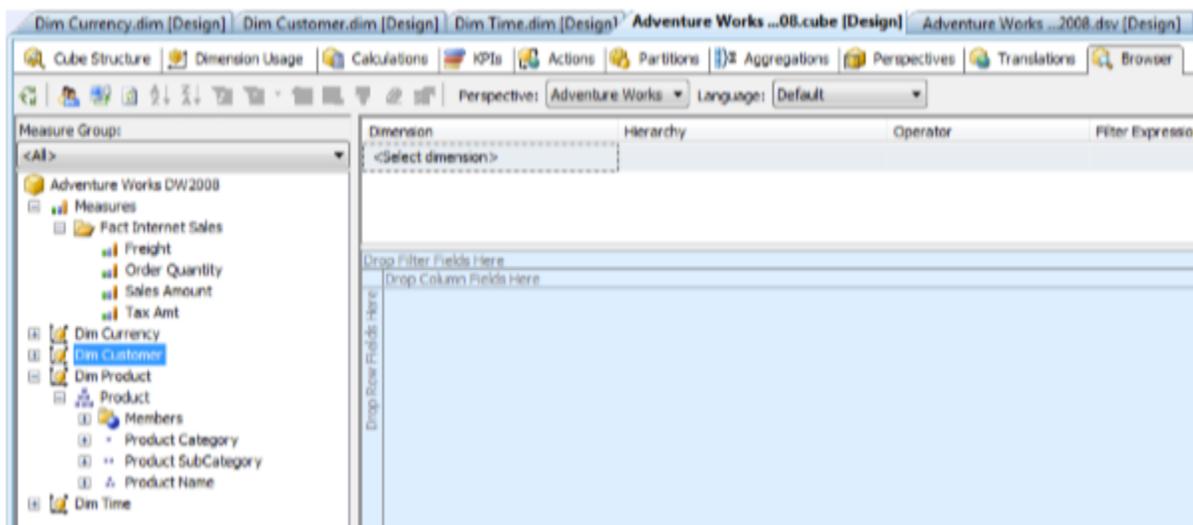
Pada jendela dimensi tab hierarki, Anda dapat mengakses DimCustomer, Anda dapat menggunakan GeographicKey pada DimCustomer untuk membuat layer hierarki pada dimensi DimCustomer dengan menambahkan tabel DimGeography ke tabel DimCustomer hierarki. Maka

agregasi baru bisa di buat oleh developer seperti aggregate data pada cube berdasarkan customer state, customer city dan postal code.

Tab Attribute pada window dimension dapat anda gunakan untuk melihat attribute-attribute dari dimensi termasuk setiap column pada setiap member dari dimensi. Pada dimensi Dimproduct, product memiliki relasi ke category dan subcategory.

Browsing Cube

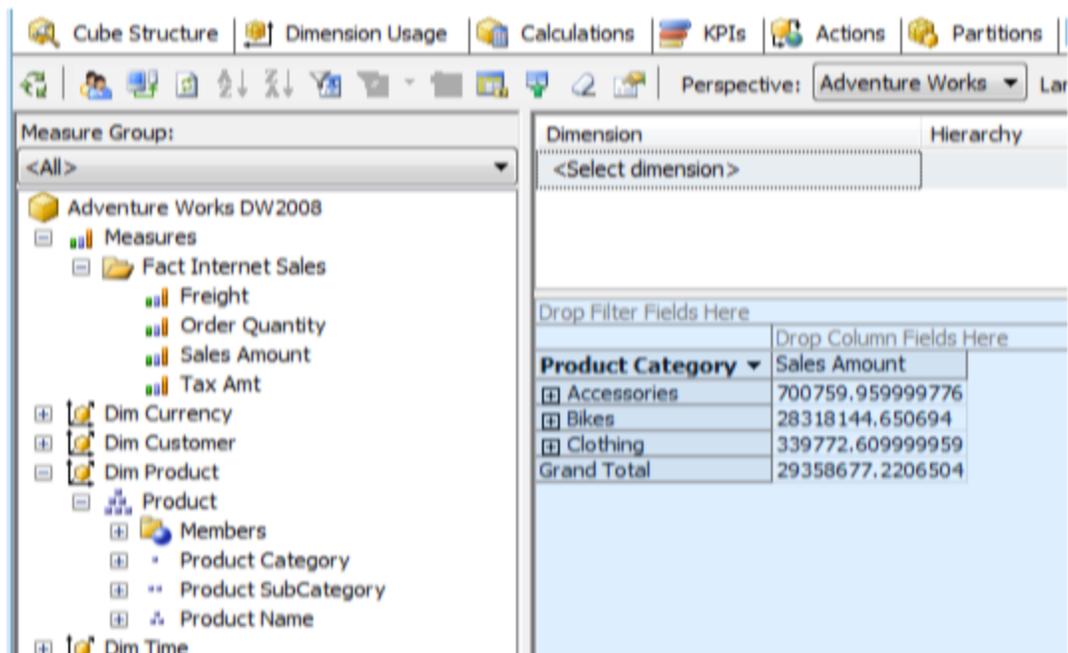
Untuk melihat data yang sudah terintegrasi dalam sebuah cube, cube harus pertama kali di process (pilih build → process atau klik kanan pada project di window solution explorer dan pilih process), setelah data selesai di process maka anda dapat melihat data dari cube. Dari cube designer pilih tab browser, anda akan melihat measures dan dimensions dari sebuah cube di tampilan pada pane navigation pada sebelah kiri dari window browser seperti anda lihat pada gambar 3.40 berikut.



Gambar 3.40. Cube Browser

Displaying Cube's Data

Untuk mendemonstrasikan kemampuan dari windows browser untuk menampilkan data pada sebuah cube, expand dimensi dimproduct, drag product hierarchies dari pane navigation di sebelah kiri dan letakkan hierarchy product tersebut ke daerah “Drop row fields here” pada windows browser. Kemudian drag dari measures → factinternetsales → sales amount pada daerah “Drop total or detail field here”. Seperti yang dapat anda lihat pada gambar 3.41 berikut.



Gambar 3.41. Tampilan data cube

Informasi yang di tampilkan pada window browser adalah informasi yang diminta oleh orang accounting. Bagaimana jika anda untuk melihat sales berdasarkan warna product dari departemen design. Dari dalam window browser pilih product catogory kemudian tekan yombol del. Drag product → color dan letakkan pada area drop row fields here pada browser window. Seperti terlihat pada gambar 3.42 berikut.

Drop Filter Fields Here	
	Drop Column Fields Here
Color ▼	Sales Amount
Black	8838411.9576015
Blue	2279096.28000007
Multi	106470.740000006
NA	435116.689999853
Red	7724330.52400018
Silver	5113389.08160051
White	5106.31999999993
Yellow	4856755.62750043
Grand Total	29358677.2206504

Gambar 3.42. Sales Amount by Product Color

Hanya dengan melakukan klik, drag dan drop anda sudah dapat menjawab pertanyaan dari departemen yang berbeda. Tidak perlu lagi menggunakan SQL Statement yang digunakan. Semua dapat dilakukan dengan memanipulasi tampilan bagaimana data seharusnya disajikan.

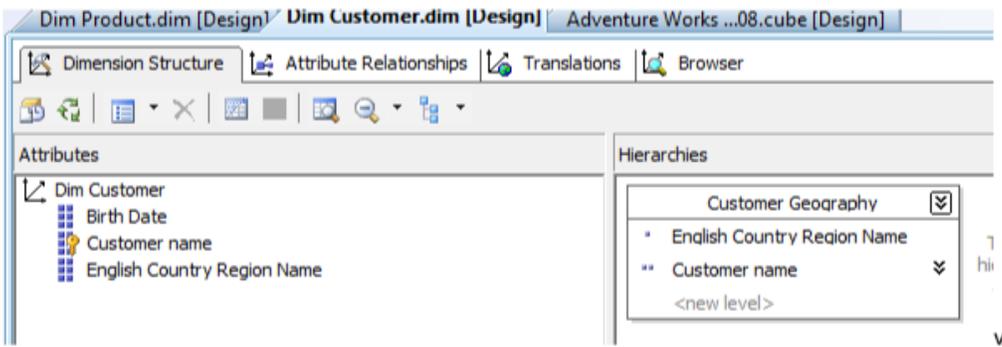
Browsing Cube with Hierarchy

Sekarang bagaimana untuk menampilkan data dengan struktur data yang rumit. Dari solution explorer, klik kanan adventureworksdw2008 DSV dan pilih opsi view designer dari menu. Anda akan melihat diagram relational dari semua tabel pada DSV. Klik kanan pada pada ruang kosong pada DSV designer, pilih add/remove tables, pada dialog box add/remove table, klik double Dimgeography untuk memindahkan table dari available object list ke included object list. Klik OK untuk menambahkan table DimGeography ke DSV. Pastikan secara visual table dimGeography telah terdapat foreign key pada geographykey beralasi one-of-many ke table DimCustomer. Relationship ini seharusnya sudah terbuat secara automatic. Jika tidak, kamu perlu membuatnya secara manual. Bukan berarti tabel DimGeography adalah bagian dari datasource bukan berarti secara automatic bagian dari DimCustomer dari cube. Untuk menambah tabel baru ke dimensi DimCustomer, klik double adventure work DW 2008 internet sales cube dan pilih tab cube structure. Pada pane navigation di sebelah kiri expand dimensi dimcustomer. Edit Dimcustomer seperti yang terlihat pada gambar 3.43 berikut.

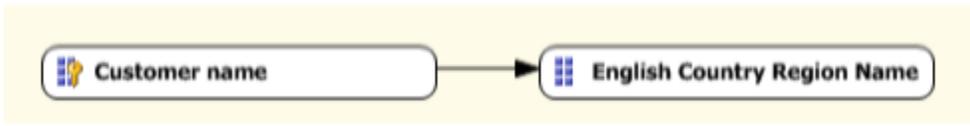


Gambar 3.43. Edit DimCustomer

Pada Dimension designer, klik kanan pada pane datasource view di sebelah kanan dan pilih Show table. Pilih kedua Dimcustomer dim DimGeography. Sekarang anda dapat melakukan drag dan drop pada kedua tabel untuk membangun dimensi customer dengan geography hierarchy. Setelah melakukan perubahan ini jangan lupa untuk mendeploy ulang solutionnya sehingga cube designer dapat mengakses data yang baru. Tambahkan attribute English country region name dan birthdate pada dimensi customer, kemudian konfigurasi attribute relationshipnya seperti terlihat pada gambar 3.44 dan 3.45 berikut.



Gambar 3.44. Hierachy Customer Hierarchy



Gambar 3.45. Attribute relationship dimensi customer

Sekarang anda dapat menambahkan English country region name dan customer name ke dalam area drop row fields here pada browser window. Drag measures sales amount dan order qty ke browser window pada area total or detail fields here. Seperti terlihat pada gambar 3.46 berikut.

Drop Filter Fields Here		Drop Column Fields Here	
English Country Region Name	Customer name	Order Quantity	Sales Amount
Australia		13345	9061000.58440184
Canada		7620	1977844.86209997
France		5558	2644017.71430033
Germany		5625	2894312.33820041
United Kingdom		6906	3391712.21090071
United States	Lauren	5	81.26
	Ian	5	113.96
	Sydney	6	138.45
	Chloe	3	2500.97
	Wyatt	3	2332.28
	Destiny	3	2371.96
	Ethan	2	2322.28
	Seth	6	122.24
	Russell	6	56.51
	Jennifer	2	2354.98
	Jesse	3	2350.96
	Amanda	3	2380.47
	Megan	3	3112.33
	Nathan	3	47.98
	Carol	5	81.26
	Ana	2	2329.98
	Noah	3	2351.96
	Angela	5	2372.95
	Chase	2	2304.98

Gambar 3.46. Tampilan data dengan hierarchy customer geography

Filtering Cube Data

Drag customer birthdate ke filter area, sekarang Hasil dari windows browser dapat di filter sesuai dengan range tanggal kelahiran dari customer. Dengan kondisi ini kamu dapat mengelompokkan customer yang memiliki lanjut untuk melakukan marketing campaign, seperti terlihat pada gambar 3.47 berikut.

Dimension	Hierarchy	Operator	Filter Expression
Dim Customer	Birth Date	Range (Inclusive)	1925-08-07 : 1935-08-14
<Select dimension>			

Gambar 3.47. Filtering by customer birthdate

Kemudian drag member dari dimensi customer ke area drop column field here pada browser window, drag measure order quantity dan sales amount ke area drop total detail fields here. Data yang di tampilkan adalah data yang telah di filter berdasarkan tanggal lahir (1925-08-07-1935-08-14), seperti terlihat pada gambar 3.48 berikut.

Dimension	Hierarchy	Operator	Filter Expression
Dim Customer	Birth Date	Range (Inclusive)	1925-08-07 : 1935-08-14
<Select dimension>			

Drop Filter Fields Here		Drop Column Fields Here	
English Country Region Name	Customer name	Order Quantity	Sales Amount
<input type="checkbox"/> Australia		68	30219,5132000001
<input type="checkbox"/> Canada		106	15435,98
<input type="checkbox"/> France		47	734,35
<input type="checkbox"/> Germany		57	1285,77
<input type="checkbox"/> United Kingdom	Alexandria	1	34,99
	Melvin	1	34,99
	Naomi	2	39,98
	Miguel	4	70,57
	Cedric	5	49,56
	Jennifer	3	61,96
	Michele	2	38,98
	Jennifer	3	92,48
	Michele	2	33,98
	Cynthia	1	29,99
	Melinda	3	48,97
	Holy	1	3,99
	Lee	2	53,98
	Kelvin	3	65,97
	Gabriella	2	38,98
	Erin	2	12,98
	Sebastian	2	38,98
	Dawn	2	25,48
	Luke	2	23,78
	Franklin	3	27,77
	Total	46	829,36
<input type="checkbox"/> United States		361	141656,1899
Grand Total		685	190161,163099999

Gambar 3.48. Tampilan data customer dengan filter birthdate

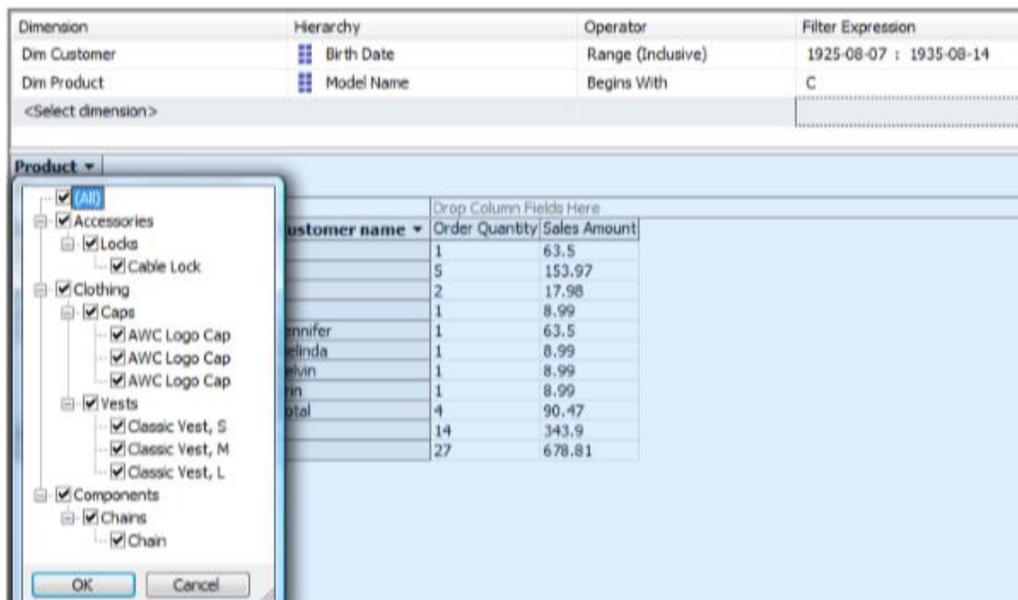
Anda dapat menambahkan beberapa filter dengan menambahkan atribut dimensi lainnya. Misalnya anda ingin memfilter data product name yang di mulai dengan awalan C, seperti terlihat pada gambar 3.49 berikut.

Dimension	Hierarchy	Operator	Filter Expression
Dim Customer	Birth Date	Range (Inclusive)	1925-08-07 : 1935-08-14
Dim Product	Model Name	Begins With	C
<Select dimension>			

Drop Filter Fields Here		Drop Column Fields Here	
English Country Region Name	Customer name	Order Quantity	Sales Amount
Australia		1	63.5
Canada		5	153.97
France		2	17.98
Germany		1	8.99
United Kingdom	Jennifer	1	63.5
	Melinda	1	8.99
	Kelvin	1	8.99
	Erin	1	8.99
	Total	4	90.47
United States		14	343.9
Grand Total		27	678.81

Gambar 3.49. Tampilan data customer dengan filter birthdate product

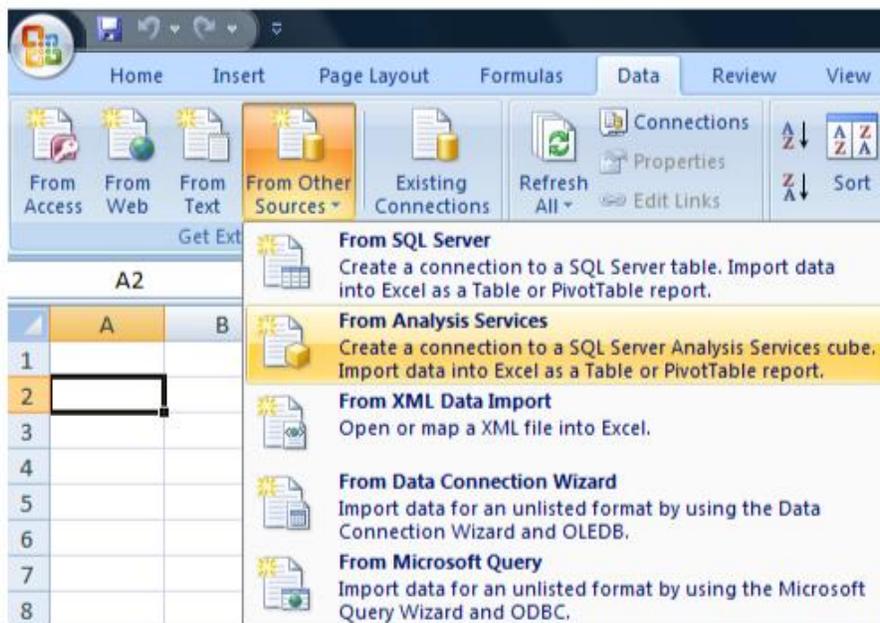
Pada area drop filter field here anda juga dapat memfilter data, tetapi pada area ini anda tidak bisa menambahkan filter expression. Filter ini hanya bersifat IN dan Equal berbeda dengan filter pada region paling atas window browser yang memiliki filter expression seperti Equal, Not Equal, IN, Not IN, Contents, Begin With, Range (Exclusive), Range (Inclusive) dan MDX. Drag hierarchy product pada dimproduct, anda akan melihat product yang di tampilkan pada hierarchy product hanya product yang berawalan C karena pada filter (region paling atas pada windows browser) kita pilih Model name product yang diawali dengan huruf C, seperti yang terlihat pada gambar 3.50 berikut.



Gambar 3.50. Tampilan filter column

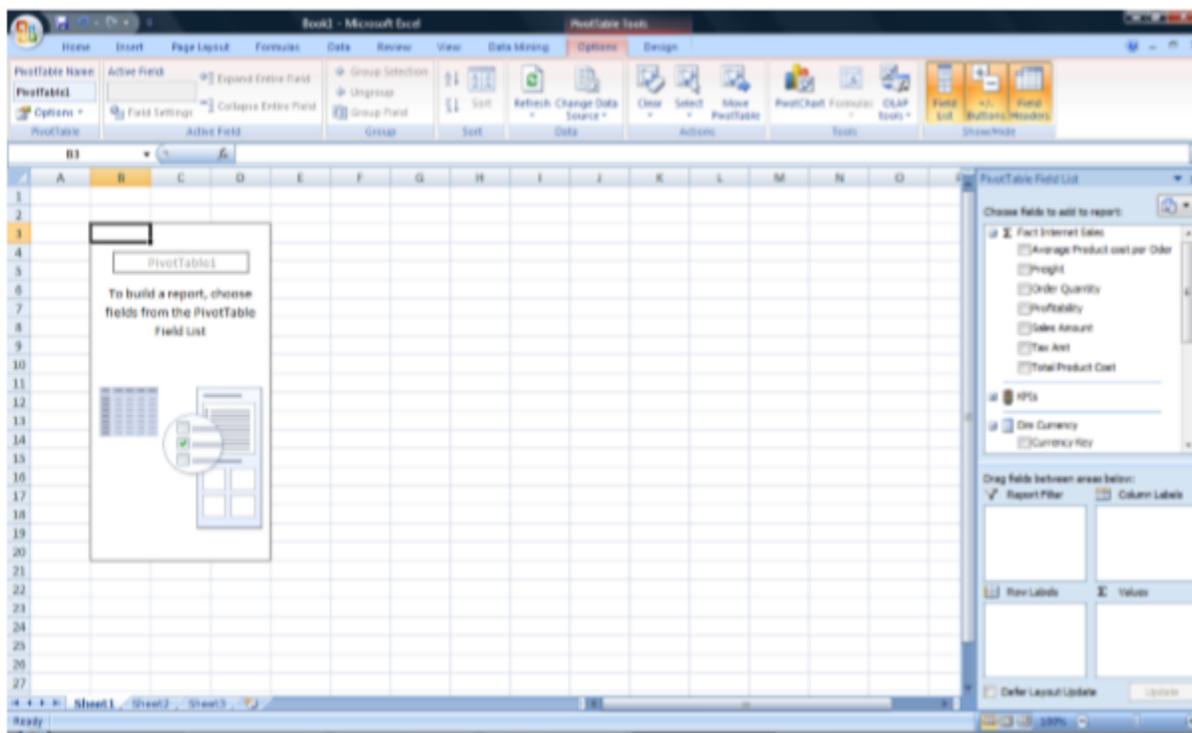
Mengakses Cube menggunakan Excel

MS Excel dapat digunakan sebagai frontend untuk mengakses data dari cube yang telah kita buat. Anda dapat melakukannya dengan mengikuti langkah berikut Buka aplikasi Excel, pada tab data pilih menu vests from other source dan pilih koneksi from Analysis Services.



Gambar 3.51. Menu Koneksi ke cube dari excel

Pilih Host Server dari Analysis Services kemudian klik next Pada menu selanjutnya anda akan di minta untuk memilih OLAP Database dan name cube. Klik next Pada menu selanjutnya isikan nama file kemudian finish. Anda akan diminta untuk memilih view untuk menampilkan data dari cube, kemudian OK. View akan tampak seperti pada Gambar 3.52 berikut.



Gambar 3.52. Excel Cube Browsing

Pada window PivotTable Field List anda dapat melakukan drag and drop object. Drag measure sales amount dan order quantity ke kolom value, hierarchy calendar pada dimensi time ke kolom report filter dan hierarchy product pada DimProduct pada kolom Row Labels, seperti terlihat pada gambar 3.53 berikut.

Row Labels	Order Quantity	Sales Amount
Accessories	38,092	\$700,759,96
Bike Racks	328	\$19,960,00
Hitch Rack - 4-Bike	328	\$19,960,00
Bike Stands	249	\$18,593,00
All-Purpose Bike Stand	249	\$18,593,00
Bottles and Cages	7,085	\$56,798,19
Water Bottle - 30 oz.	4,264	\$21,177,56
Mountain Bottle Cage	2,025	\$10,329,75
Road Bottle Cage	1,712	\$15,290,88
Cleaners	908	\$7,258,60
Fenders	2,121	\$46,619,58
Helmets	6,440	\$225,335,60
Hydration Packs	733	\$40,307,67
Tires and Tubes	17,332	\$245,529,32
Bikes	15,205	\$28,318,144,65
Clothing	4,101	\$138,772,61
Grand Total	60,304	\$70,318,677,22

Gambar 3.53. Tampilan Data Cuba pada Excel

Data tersebut kemudian dapat Anda visualisasikan menggunakan grafik atau diagram yang ada pada Excel.

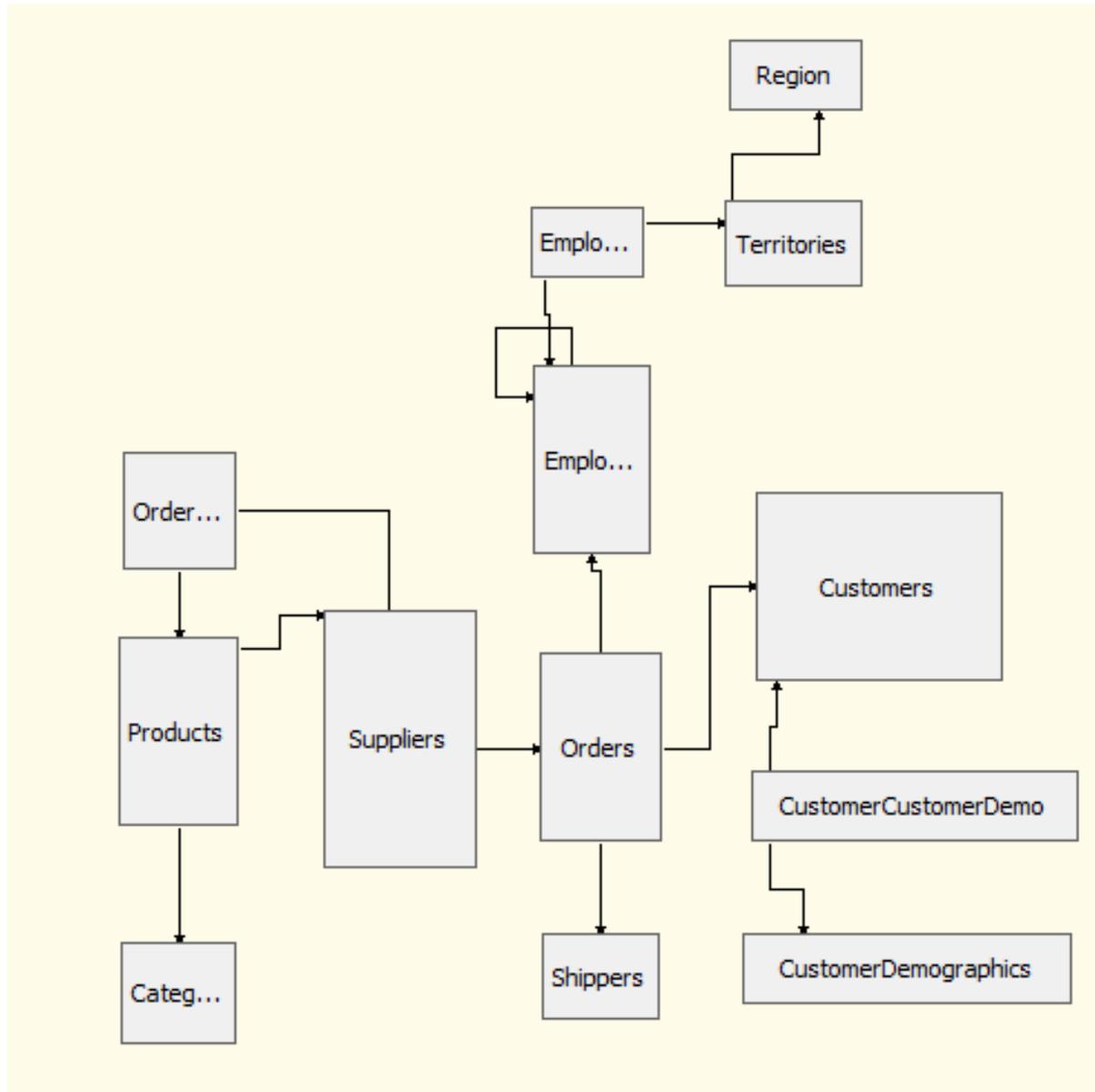
3.10. Rangkuman

- Business Analytics (BA) mencakup berbagai metode yang dapat dikategorikan menjadi tiga: *information discovery*, *decision support*, dan *visualization*.
- OLAP merupakan seperangkat *tools* yang digunakan dalam analisis data. OLAP mendukung dalam *reporting*, *querying*, analisis statistikan, dan visualisasi.
- Data multidimensionality membantu pengguna untuk melihat data secara cepat dari dimensi yang berbeda.
- Cube memudahkan pengguna untuk menghasilkan kueri dan laporan dengan cepat.

3.11. Latihan Soal

1. Hubungkan konsep Data Warehouse yang sudah Anda pelajari, dengan konsep OLAP dan visualisasi data.

2. Apakah yang membedakan antara OLTP dengan OLAP?
3. Bukalah database Northwind dengan struktur seperti berikut

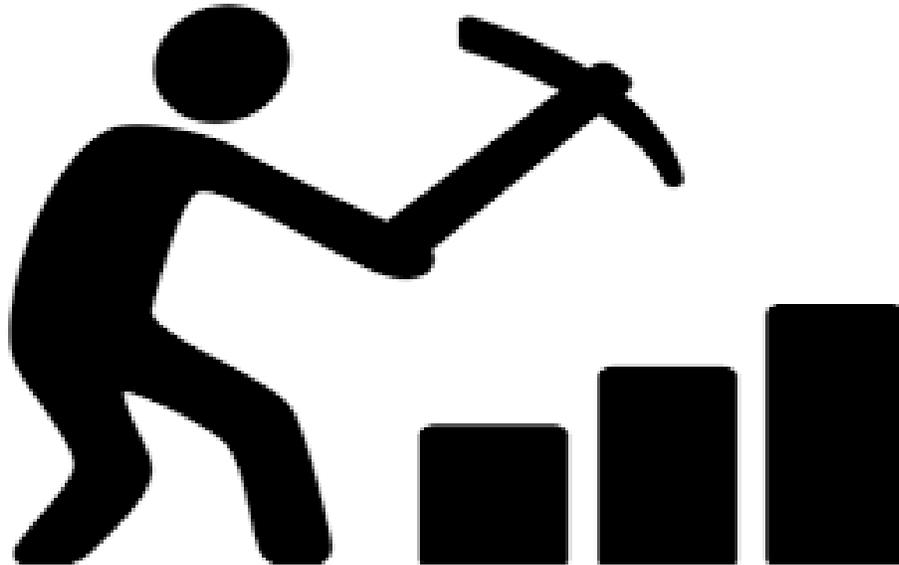


Seorang manajer penjualan berperan dalam menganalisis performa penjualan/order produk dari setiap supplier. Pada kesempatan ini, dia ingin memperoleh informasi tentang jumlah pesanan (order quantity) dari 3 supplier terbesar di negara USA. Manajer juga ingin mengetahui kategori produk terlaris dari tiap supplier tersebut. Analisis dilakukan untuk data order di tahun 1997.

- a) Rancanglah Cube pada SQL Server SSAS untuk menunjang kebutuhan manajer
- b) Sajikan informasi yang dibutuhkan manajer menggunakan Excel Pivot Table & Chart

BAB IV

DATA MINING



Kompetensi

Mahasiswa dapat menerapkan algoritma Data Mining (Minggu 8 – 10)

Bahan Kajian

Bab ini membahas materi-materi sebagai berikut:

- Definisi Data Mining
- Karakteristik Data Mining
- Tipe Data
- Pengelompokan Data Mining
- Penerapan Data Mining
- Tahapan Metodologi Data Mining
- Metode Data Mining (Klasifikasi, Clustering, Asosiasi)
- Data Mining Software
- Pengenalan Weka
- Weka Explorer
- Dataset, classifier, filter, visualisasi data
- Evaluasi data mining

4.1. Definisi *Data Mining*

Secara sederhana, *data mining* adalah istilah yang digunakan untuk menggambarkan penemuan atau "penambangan" pengetahuan dari sejumlah besar data. Sebenarnya, istilah data mining (penambangan data) adalah salah kaprah. Apabila ada orang menambang emas dari bebatuan atau tanah, istilah yang digunakan adalah "penambangan emas", bukan penambangan batu/tanah. Oleh karena itu, data mining mungkin seharusnya dinamai "*knowledge mining*" atau "*knowledge discovery*." Meskipun ada ketidakcocokan antara istilah dan maknanya, istilah *data mining* telah menjadi kesepakatan umum. Banyak nama lain yang terkait dengan data mining meliputi *knowledge extraction*, *pattern analysis*, *data archeology*, *information harvesting*, *pattern searching*, dan *data dredging*.

Secara teknis, *data mining* adalah proses yang menggunakan teknik statistik, matematis, dan kecerdasan buatan untuk mengekstrak dan mengidentifikasi informasi bermanfaat dan pengetahuan (atau pola) dari kumpulan data yang besar. Pola ini bisa dalam bentuk aturan bisnis, korelasi, tren, atau model prediksi (Nemati dan Barko, 2001). Sebagian besar literatur mendefinisikan data mining sebagai "proses nontrivial untuk mengidentifikasi pola yang valid, novel, berpotensi berguna, dan akhirnya dapat dipahami dalam data yang tersimpan dalam database terstruktur," dimana data disusun dalam catatan yang disusun oleh variabel kategoris, ordinal dan kontinu. Dalam definisi ini, arti dari istilah kunci adalah sebagai berikut:

- Proses menyiratkan bahwa data mining terdiri dari banyak langkah iteratif.
- Nontrivial berarti bahwa pencarian yang dilakukan bertipe eksperimen
- Valid berarti bahwa pola yang ditemukan harus benar pada data baru dengan tingkat keyakinan yang memadai.
- Novel berarti bahwa pola tersebut sebelumnya tidak diketahui pengguna dalam konteks sistem yang dianalisis.
- Berpotensi berguna berarti bahwa pola yang ditemukan harus mengarah pada beberapa keuntungan bagi pengguna atau tugas.
- Dapat dipahami berarti bahwa pola yang ditemukan pada akhirnya masuk akal secara naluri bisnis

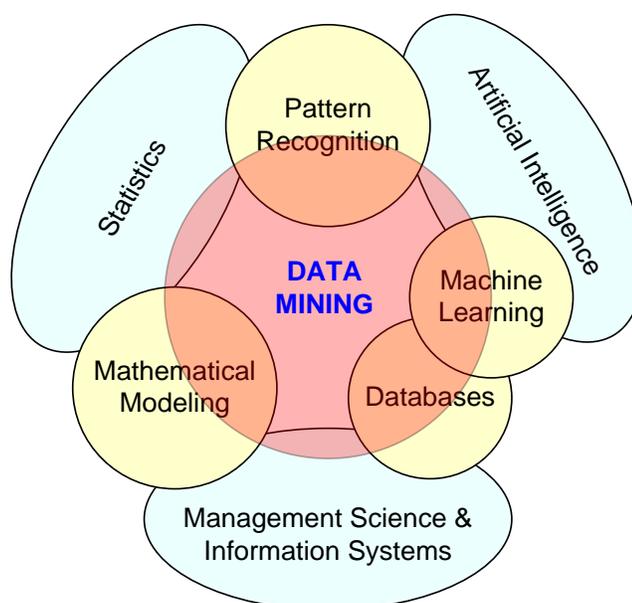
Sumber data dari sebuah *data mining* umumnya berasal dari *data warehouse*. Sepertinya yang sudah kita pelajari sebelumnya, dengan ketersediaan data warehouse kita memiliki kumpulan

data yang siap digunakan untuk proses analisis dengan tujuan memperoleh informasi atau pengetahuan. Karena itu, ketersediaan data menjadi kunci sukses dalam data mining.

Pengguna data mining biasanya adalah *end user*, yaitu orang yang membutuhkan informasi. Lingkungan data mining bisa berupa sistem client-server ataupun sebuah arsitektur berbasis web. Oleh sebab itu, sebuah data mining sebaiknya bersifat user-friendly dan mudah dipahami oleh pengguna. Yang menjadi perhatian pengguna adalah bagaimana informasi atau pengetahuan yang dihasilkan oleh data mining. Algoritma dari sistem data mining itu sendiri menjadi hal yang dipikirkan oleh pengembang sistem.

4.2. Karakteristik *Data Mining*

Data mining bukanlah disiplin baru namun merupakan sebuah definisi baru yang menggunakan banyak disiplin ilmu. Data mining diposisikan sebagai irisan dari banyak disiplin ilmu, termasuk statistik, kecerdasan buatan (artificial intelligence), machine learning, manajemen sains, sistem informasi, dan database (lihat Gambar 4.1). Dengan menggunakan kemajuan dalam semua disiplin ilmu ini, data mining berusaha untuk membuat kemajuan dalam mengekstrak informasi dan pengetahuan yang berguna dari database besar. Data Mining adalah bidang yang baru muncul yang telah menarik banyak perhatian dalam waktu singkat.



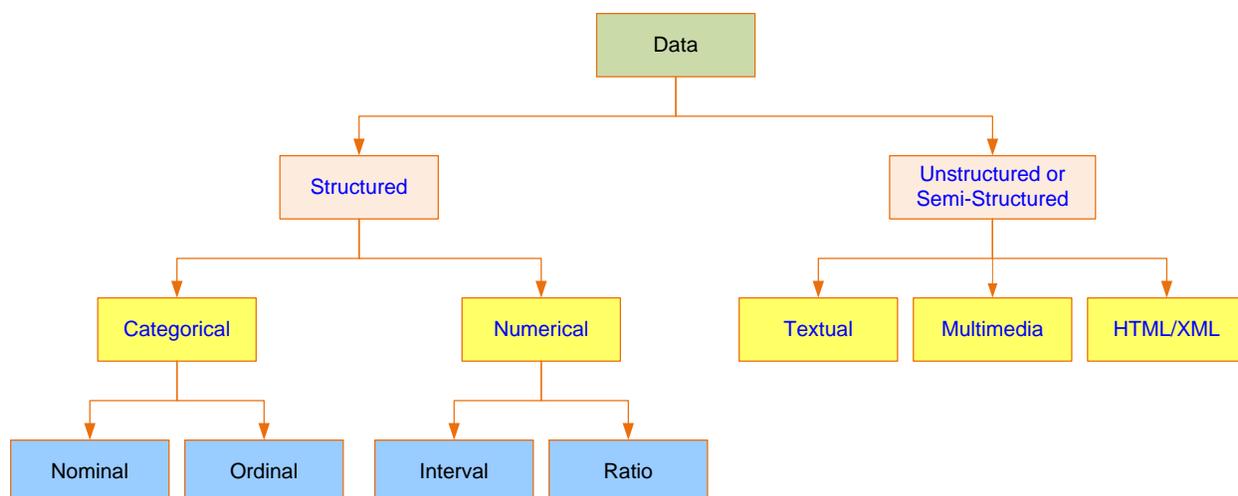
Gambar 4.1. Disiplin ilmu yang menunjang Data Mining

Berikut ini adalah beberapa karakteristik dari *Data Mining*:

- Data biasanya bersumber dari database yang sangat besar, yang terkadang berisi data dari beberapa tahun. Dalam banyak kasus, data dibersihkan dan diintegrasikan ke dalam data warehouse.
- Lingkungan data mining biasanya menggunakan arsitektur client / server ataupun arsitektur sistem informasi berbasis web.
- Miner (atau penambang informasi) sering merupakan *end user*, yang mengajukan pertanyaan khusus dan membutuhkan jawaban dengan cepat, dengan sedikit atau tanpa kemampuan pemrograman.
- Seringkali melibatkan penemuan hasil yang tidak terduga, sehingga mengharuskan pengguna berpikir kreatif termasuk termasuk menginterpretasikan temuan.
- *Tools data mining* mudah digabungkan dengan *spreadsheet* dan *software development tools* lainnya. Dengan demikian, data yang ditambang dapat dianalisis dan digunakan dengan cepat dan mudah.
- Karena banyaknya data dan proses pencarian yang besar, kadang kala perlu menggunakan pemrosesan paralel untuk data mining.
-

4.3. Tipe Data

Data adalah kumpulan fakta yang biasanya didapat sebagai hasil pengalaman, pengamatan, atau eksperimen. Data dapat terdiri dari angka, kata, gambar, dan sebagainya sebagai pengukuran dari satu set variabel. Pada data terstruktur, kita dapat mengklasifikasikan data sebagai kategoris atau numerik. Data kategoris dapat dibagi menjadi data nominal atau ordinal, sedangkan data numerik dapat dibagi menjadi interval atau rasio. Gambar 4.2 menunjukkan taksonomi sederhana data dalam data mining.



Gambar 4.2. Taksonomi tipe data

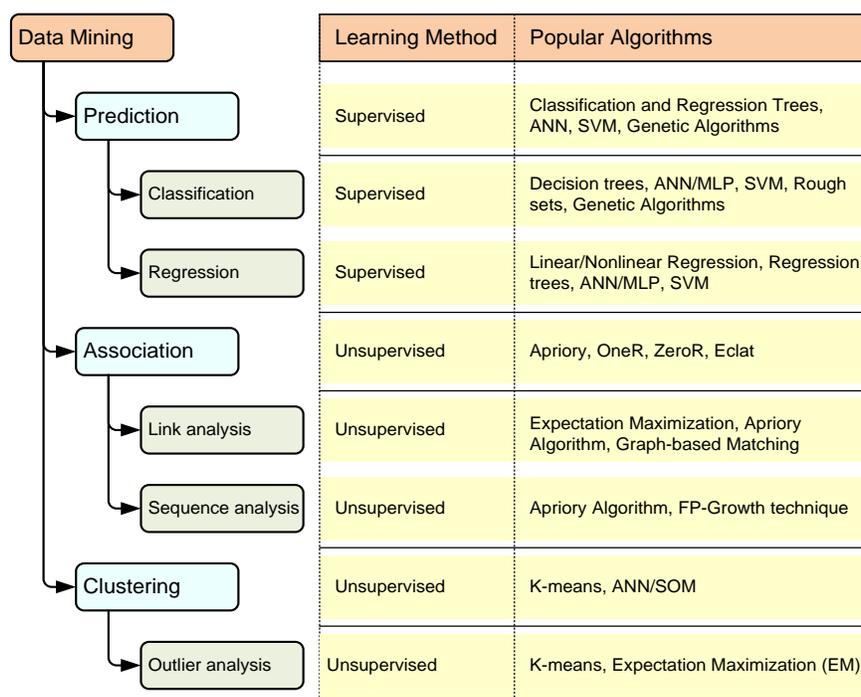
- **Data kategoris** mewakili label dari beberapa kelas yang digunakan untuk membagi variabel menjadi beberapa kelompok tertentu. Contoh data kategoris meliputi jenis kelamin, agama, kelompok umur, dan tingkat pendidikan.
- **Data nominal** berisi pemberian kode sederhana yang ditetapkan ke objek sebagai sebuah label, yang bukan hasil pengukuran. Misalnya, data status perkawinan dapat dikategorikan sebagai (1) lajang, (2) menikah, dan (3) bercerai.
- **Data ordinal** berisi kode yang diberikan ke objek sebagai label yang juga mewakili urutan peringkat di antara mereka. Misalnya, variabel skor kredit secara umum dapat dikategorikan sebagai (1) rendah, (2) sedang, dan (3) tinggi.
- **Data numerik** mewakili nilai numerik dari variabel tertentu. Contoh variabel bernilai numerik adalah umur, jumlah anak, dan total pendapatan rumah tangga.
- **Data interval** adalah variabel yang dapat diukur pada skala interval. Contoh umum pengukuran skala interval adalah suhu pada skala Celsius. Dalam skala khusus ini, unit pengukuran adalah 1/100 dari perbedaan antara suhu membeku dan suhu mendidih air dalam tekanan atmosfer; Artinya, tidak ada nilai nol absolut.
- **Data rasio** meliputi variabel pengukuran yang biasa ditemukan dalam ilmu fisika dan teknik. Massa, panjang, waktu, sudut, energi, dan muatan listrik adalah contoh pengukuran fisik yang merupakan skala rasio.

Pemahaman tentang tipe data yang akan digunakan menjadi penting, karena beberapa metode data mining bersifat khusus terhadap tipe data yang dapat ditangani. Tidak semua tipe data cocok

digunakan pada suatu metode data mining. Sebagai contoh, beberapa metode data mining memerlukan semua variabel (baik input maupun output) yang bernilai numerik, misalnya metode Neural Networks, Support Vector Machine, dan regresi. Oleh sebab itu, terkadang kita perlu merubah tipe data agar dapat mendukung kinerja suatu metode data mining (data numerik menjadi data kategoris, atau sebaliknya).

4.4. Pengelompokan *Data Mining*

Secara umum, tugas *data mining* dapat dikelompokkan menjadi tiga kategori utama: prediksi (prediction), asosiasi (association), dan pengelompokan (clustering). Berdasarkan cara di mana pola diekstraksi dari data historis, algoritma pembelajaran metode data mining dapat diklasifikasikan sebagai *supervised* atau *unsupervised*. Dengan algoritma pembelajaran yang diawasi (supervised), data pelatihan mencakup pemberian atribut/label kelas (yaitu variabel output atau variabel hasil). Sebaliknya, dengan pembelajaran tanpa pengawasan, data pelatihan tidak mencakup atribut deskriptif. Gambar 4.3 menunjukkan taksonomi sederhana untuk tugas-tugas data mining, bersama dengan metode pembelajaran, dan algoritma populer untuk setiap tugas data mining.



Gambar 4.3. Klasifikasi *Data Mining*

Prediction bisa disebut sebagai tindakan bercerita tentang masa depan. Istilah yang umumnya dikaitkan dengan prediksi adalah peramalan (*forecasting*). Bergantung pada sifat dari apa yang diprediksi, prediksi dapat disebutkan secara lebih spesifik sebagai klasifikasi (di mana perkiraan, seperti perkiraan cuaca besok, menggunakan data kategoris seperti "hujan" atau "cerah") atau regresi (di mana hal yang diprediksi, seperti perkiraan temperature udara besok, adalah data numerik seperti "30 ° C").

Classification, yang merupakan salah satu bentuk *supervised learning*, mungkin adalah tipe yang paling umum dari semua data mining. Tujuan klasifikasi adalah menganalisis data historis yang tersimpan dalam database dan secara otomatis menghasilkan model yang dapat memprediksi perilaku masa depan. Model induksi ini terdiri dari generalisasi terhadap *training dataset*, yang membantu membedakan kelas yang telah terklasifikasi. Harapannya adalah bahwa model tersebut kemudian dapat digunakan untuk memprediksi kelas-kelas dari data lain yang belum terklasifikasi dan, yang lebih penting, untuk memprediksi secara akurat kejadian masa depan yang sebenarnya. Metode klasifikasi yang umum digunakan yaitu Neural Networks, Decision Tree, Support Vector Machines, dan Genetic Algorithm.

Association adalah teknik yang populer dan digunakan untuk menemukan hubungan yang menarik antara variabel dalam database besar. Berkat teknologi pengumpulan data otomatis seperti *barcode scanner*, penggunaan teknik asosiasi untuk menemukan pola di antara produk dalam transaksi berskala besar yang dicatat oleh sistem penjualan point-of-sale (POS) di pasar swalayan dapat dilakukan. Dalam konteks industri ritel, aturan asosiasi sering disebut *market-basket analysis*.

Clustering mempartisi kumpulan dari objek ke dalam segmen tertentu yang anggotanya memiliki karakteristik serupa. Tidak seperti klasifikasi, dalam clustering, label kelas tidak diketahui sebelumnya. Ketika algoritma yang dipilih mempelajari suatu dataset, akan dilakukan identifikasi kesamaan hal berdasarkan karakteristik mereka, kemudian cluster ditetapkan. Karena cluster ditentukan menggunakan algoritma tipe heuristik, dan karena algoritma yang berbeda mungkin berakhir dengan rangkaian kelompok yang berbeda untuk kumpulan data yang sama, mungkin diperlukan pakar untuk menginterpretasikan, dan mungkin memodifikasi kelompok yang disarankan sebelum hasil pengelompokan digunakan secara aktual.

Setelah cluster yang masuk akal telah diidentifikasi, model dapat digunakan untuk mengklasifikasikan dan menafsirkan data baru. Tujuan pengelompokan adalah untuk menciptakan

kelompok sehingga anggota dalam setiap kelompok memiliki kesamaan maksimum dan anggota lintas kelompok memiliki kesamaan minimum. Teknik pengelompokan yang paling umum digunakan adalah k-means (dari statistik).

Penelitian Data mining dapat berupa *hypothesis-driven* atau *discovery-driven*. Data mining berbasis hipotesis (*hypothesis-driven*) dimulai dengan hipotesis oleh pengguna, yang kemudian berusaha untuk memvalidasi kebenaran hipotesis. Misalnya, seorang manajer pemasaran dapat memulai dengan proposisi berikut: "Apakah penjualan DVD player terkait dengan penjualan perangkat televisi?" Sementara itu, data mining yang berupa *discovery-driven* menemukan pola, asosiasi, dan hubungan lain yang tersembunyi dalam kumpulan data. Ini dapat mengungkap fakta yang sebelumnya tidak diketahui atau bahkan dipikirkan oleh organisasi.

4.5. Penerapan *Data Mining*

Data mining telah menjadi alat yang populer dalam menangani banyak masalah bisnis yang kompleks. Tujuan dari banyak aplikasi data mining di bidang bisnis ini adalah untuk memecahkan masalah yang mendesak atau untuk mengeksplorasi peluang bisnis yang sedang berkembang untuk menciptakan keunggulan kompetitif yang berkelanjutan.

Customer Relationship Management (CRM) adalah pengembangan baru yang muncul dari pemasaran tradisional. Tujuan CRM adalah menciptakan hubungan erat dengan pelanggan melalui pengembangan pemahaman yang intim tentang kebutuhan dan keinginan mereka. Dalam transaksi bisnis dengan pelanggan (misalnya, permintaan produk, penjualan, permintaan layanan, panggilan garansi), kita dapat mengumpulkan sejumlah besar data. Bila dikombinasikan dengan atribut lain (misalnya demografi dan sosioekonomi), data yang kaya informasi ini dapat digunakan untuk (1) mengidentifikasi calon pembeli produk / layanan baru yang paling mungkin terjadi; (2) memahami akar penyebab gesekan pelanggan untuk meningkatkan retensi pelanggan; (3) menemukan asosiasi varian waktu antara produk dan layanan untuk memaksimalkan penjualan dan nilai pelanggan; dan (4) mengidentifikasi pelanggan yang paling menguntungkan dan karakteristik kebutuhan mereka untuk memperkuat hubungan dan memaksimalkan penjualan.

Perbankan. Data mining dapat membantu bank dengan yang berikut: (1) mengotomatisasi proses aplikasi pinjaman dengan secara akurat memprediksi penunggak yang paling mungkin; (2) mendeteksi kartu kredit palsu dalam transaksi perbankan online; (3) mengidentifikasi cara untuk

memaksimalkan nilai pelanggan dengan menjual produk dan layanan yang paling mungkin mereka beli; dan (4) mengoptimalkan pengembalian tunai dengan meramalkan secara akurat arus kas entitas perbankan (contohnya jumlah uang pada mesin ATM dan kantor cabang bank).

Ritel dan logistik. Di industri ritel, *data mining* dapat digunakan untuk (1) memprediksi volume penjualan yang akurat di lokasi ritel tertentu untuk menentukan tingkat inventori yang benar; (2) mengidentifikasi hubungan penjualan antara produk yang berbeda (dengan *market-basket analysis*) untuk memperbaiki tata letak toko dan mengoptimalkan promosi penjualan; (3) memperkirakan tingkat konsumsi berbagai jenis produk (berdasarkan kondisi musiman dan lingkungan) untuk mengoptimalkan logistik sehingga memaksimalkan penjualan; dan (4) menemukan pola menarik dalam pergerakan produk (terutama untuk produk yang memiliki umur simpan terbatas karena rentan terhadap kadaluarsa, mudah rusak, dan kontaminasi) dalam rantai pasokan dengan menganalisis data sensorik dan RFID.

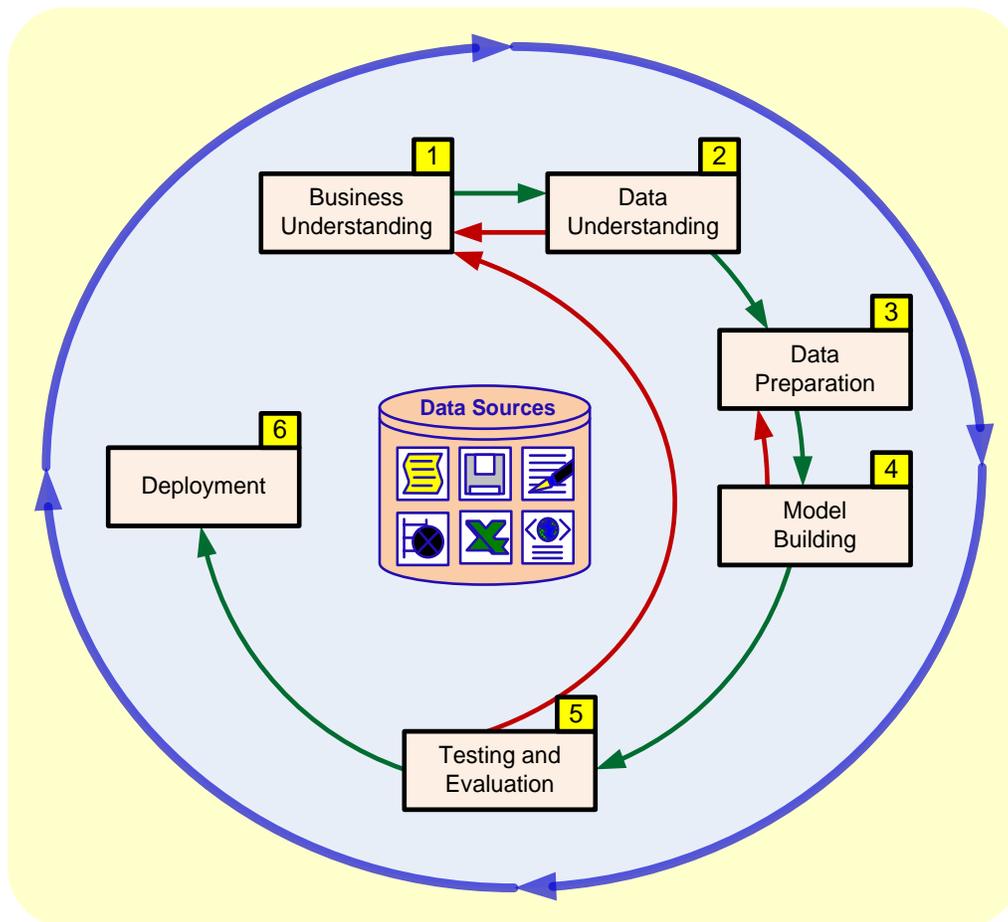
Perdagangan saham. Pialang dan pedagang menggunakan *data mining* untuk (1) memprediksi kapan dan berapa harga obligasi tertentu akan berubah; (2) meramalkan kisaran dan arah fluktuasi saham; (3) menilai dampak dari isu dan peristiwa tertentu terhadap keseluruhan pergerakan pasar; dan (4) mengidentifikasi dan mencegah kegiatan penipuan dalam perdagangan efek.

Kedokteran. Penggunaan data mining dalam bidang kedokteran harus dipandang sebagai pelengkap yang sangat berharga untuk penelitian medis tradisional, yang terutama bersifat klinis dan biologis. Analisis data mining dapat (1) mengidentifikasi pola baru untuk meningkatkan survivabilitas dari pasien kanker; (2) memprediksi tingkat keberhasilan pasien transplantasi organ untuk mengembangkan kebijakan pencocokan donor-organ yang lebih baik; (3) mengidentifikasi fungsi gen yang berbeda pada gen manusia; dan (4) menemukan hubungan antara gejala dan penyakit (serta penyakit dan perawatan yang berhasil) untuk membantu profesional medis membuat keputusan yang tepat dan tepat pada waktunya.

4.6. Tahapan Metodologi *Data Mining*

Untuk dapat melaksanakan proyek data mining secara sistematis, ada suatu metodologi proses yang dapat diikuti. Berdasarkan *best practices*, peneliti dan praktisi data mining telah mengajukan beberapa proses (alur kerja atau pendekatan langkah-demi-langkah sederhana) untuk

memaksimalkan peluang keberhasilan dalam melakukan proyek data mining. Upaya ini telah menghasilkan beberapa proses standar, dan salah satu yang bisa dibilang paling populer yaitu Cross-Industry Standard Process for Data Mining (**CRISP-DM**). Metodologi ini diusulkan pada pertengahan 1990an oleh sebuah konsorsium perusahaan Eropa untuk digunakan sebagai metodologi standar data mining (CRISP-DM, 2009). Gambar 4.4 mengilustrasikan proses yang diusulkan ini, yang merupakan urutan dari enam langkah, yang dimulai dengan pemahaman bisnis yang baik dan kebutuhan terhadap proyek data mining, dan diakhiri dengan penyebaran solusi (deployment) yang memenuhi spesifikasi kebutuhan bisnis.



Gambar 4.4. Metodologi CRISP-DM

Meskipun langkah-langkah ini bersifat berurutan, tetap dimungkinkan aktivitas yang berjalan mundur (*backtracking*). Karena data mining didorong oleh pengalaman dan eksperimen, tergantung pada situasi masalah dan pengetahuan / pengalaman analis, keseluruhan proses bisa sangat berulang (misalnya, seseorang harus terus maju bolak-balik melalui langkah-langkah

beberapa kali). Karena langkah terakhir dibangun berdasarkan hasil keputusan sebelumnya, orang harus memberi perhatian ekstra pada langkah-langkah sebelumnya agar tidak memasukkan keseluruhan penelitian ke jalur yang salah sejak awal.

Langkah 1: Pemahaman Bisnis (*Business Understanding*)

Elemen utama dari penelitian data mining adalah mengetahui apa tujuan dari penelitian ini. Menjawab pertanyaan semacam itu dimulai dengan pemahaman menyeluruh tentang kebutuhan manajerial akan pengetahuan baru dan spesifikasi eksplisit dari tujuan bisnis mengenai studi yang akan dilakukan. Contoh pertanyaan spesifik seperti "Apa profil khas pelanggan kita, dan berapa banyak nilai yang diberikan masing-masing pelanggan kepada kita?" Kemudian sebuah rencana proyek untuk menggali informasi semacam itu dikembangkan. Ini dapat dimulai dengan menentukan orang-orang yang bertanggung jawab untuk mengumpulkan data, menganalisis data, dan melaporkan temuannya.

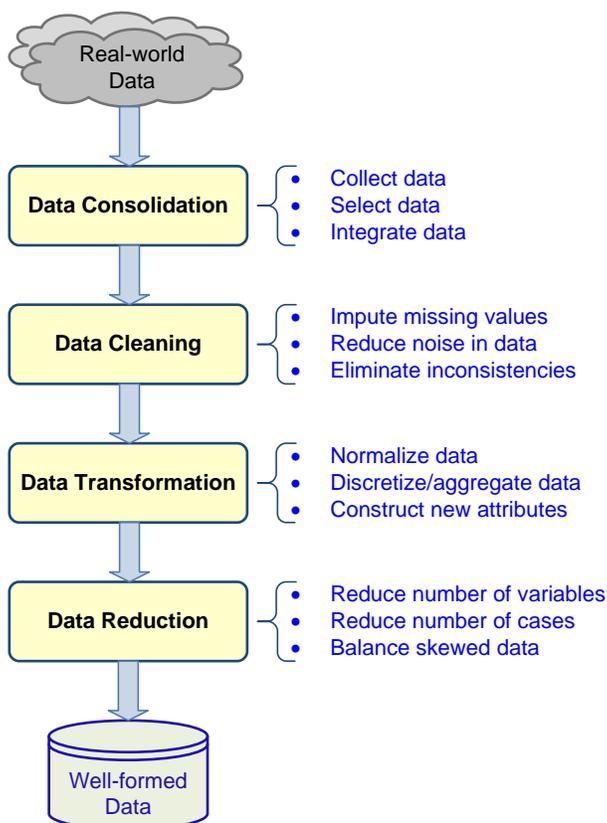
Langkah 2: Pemahaman Data (*Data Understanding*)

Setelah pemahaman bisnis, aktivitas utama proses data mining adalah mengidentifikasi data yang relevan dari banyak database yang ada. Beberapa poin kunci harus diperhatikan adalah tahap identifikasi dan seleksi data. Pertama dan terutama, analisis harus jelas dan ringkas tentang deskripsi tugas data mining sehingga data yang paling relevan dapat diidentifikasi. Misalnya, proyek data mining ritel mungkin berusaha untuk mengidentifikasi perilaku pengeluaran pembeli wanita, yang membeli pakaian musiman, berdasarkan demografi, transaksi kartu kredit, dan atribut sosioekonomi mereka. Selanjutnya, analisis harus membangun pemahaman yang mendalam tentang sumber data. Misalnya, dimana data yang relevan disimpan dan dalam bentuk apa; bagaimana proses mengumpulkan data - otomatis versus manual; siapa kolektornya, dan seberapa sering itu diperbarui. Setelah itu, tipe-tipe data yang akan digunakan juga harus diidentifikasi dengan jelas, karena berkaitan dengan penentuan metode data mining yang akan digunakan.

Langkah 3: Persiapan Data (*Data Preparation*)

Tujuan persiapan data (atau yang lebih umum disebut preprocessing data) adalah dengan mengambil data yang sudah diidentifikasi pada langkah sebelumnya dan mempersiapkannya untuk analisis dengan metode data mining. Dibandingkan dengan langkah-langkah lain dalam CRISP-DM, preprocessing data menghabiskan banyak waktu dan usaha, biasanya langkah ini mengambil sekitar 80 persen dari total waktu yang dihabiskan untuk proyek data mining. Upaya ini dilakukan dengan kenyataan bahwa data di dunia nyata umumnya tidak lengkap (ada atribut nilai yang

kurang, atau hanya berisi data agregat), *noisy* (mengandung kesalahan atau outlier), dan tidak konsisten (berisi ketidakseragaman format atau nama). Gambar 4.5 menunjukkan empat langkah utama yang diperlukan untuk mengubah data mentah di dunia nyata menjadi dataset yang dapat digunakan untuk data mining.



Gambar 4.5. Persiapan data

Pada tahap pertama preprocessing data, data yang relevan dikumpulkan dari sumber yang teridentifikasi (dicapai pada langkah sebelumnya, Pemahaman Data), *record* dan variabel yang diperlukan dipilih (berdasarkan pemahaman mendalam tentang data, bagian yang tidak relevan disaring), dan *record* yang berasal dari beberapa sumber data diintegrasikan. Pada tahap kedua preprocessing data, data dibersihkan (langkah ini juga dikenal dengan *data cleaning*). Pada langkah ini, nilai dalam dataset diidentifikasi dan ditangani. Dalam beberapa kasus, nilai yang hilang adalah anomali dalam dataset, dalam hal ini, nilai perlu diperkirakan (diisi dengan nilai yang paling cocok, bisa dengan mean/median/modus) atau diabaikan. Dalam kasus lain, nilai yang hilang merupakan hal yang alami dari kumpulan data (misal tingkat pendapatan sering tidak terjawab oleh orang-orang yang berada di tingkat pendapatan tertinggi). Pada langkah ini, analisis

juga harus mengidentifikasi *outlier* pada data dan kemudian menghilangkan atau menghaluskannya. Selain itu, inkonsistensi (nilai yang tidak biasa dalam suatu variabel) dalam data harus ditangani dengan menggunakan pengetahuan domain dan / atau pendapat dari pakar.

Pada tahap ketiga preprocessing data, data ditransformasikan untuk pemrosesan yang lebih baik. Misalnya, dalam banyak kasus, data dinormalisasi antara minimum dan maksimum tertentu untuk semua variabel untuk mengurangi potensi bias satu variabel (memiliki nilai numerik yang besar, seperti untuk pendapatan rumah tangga) yang mendominasi variabel lain (seperti jumlah tanggungan dalam keluarga, yang mungkin berpotensi lebih penting). Transformasi lain yang terjadi adalah diskritisasi dan / atau agregasi. Dalam beberapa kasus, variabel numerik dikonversi ke nilai kategoris (misalnya: rendah, sedang, dan tinggi).

Tahap akhir dari preprocessing data adalah reduksi data. Meskipun *data miner* ingin memiliki dataset yang besar, terlalu banyak data juga bisa menjadi masalah. Secara umum, data yang biasa digunakan dalam proyek data mining terdiri dari dua dimensi: variabel (jumlah kolom) dan *case/records* (jumlah baris). Dalam beberapa kasus seperti pemrosesan citra, jumlah variabel bisa sangat besar, sehingga analisis harus mengurangi jumlah variabel untuk menyederhanakan kompleksitas model. Karena variabel diperlakukan sebagai dimensi yang berbeda yang menggambarkan fenomena dari perspektif yang berbeda, dalam data mining, proses ini biasa disebut *dimensional reduction*.

Langkah 4: Pengembangan model (model building)

Pada tahap ini, berbagai metode pemodelan Data Mining dipilih dan diterapkan pada dataset yang sudah disiapkan untuk memenuhi kebutuhan bisnis yang spesifik. Langkah pembuatan model juga mencakup penilaian dan analisis komparatif dari berbagai model yang dibangun. Karena tidak ada metode atau algoritma terbaik yang diketahui secara universal untuk Data Mining, seseorang harus menggunakan berbagai jenis model yang layak secara bersamaan dengan eksperimen dan teknik evaluasi yang terdefinisi dengan baik untuk mengidentifikasi metode "terbaik" untuk tujuan tertentu. Bahkan untuk satu metode atau algoritma, sejumlah parameter perlu dikalibrasi untuk mendapatkan hasil yang optimal. Beberapa metode mungkin memiliki persyaratan khusus mengenai bagaimana data harus diformat, dengan demikian melangkah mundur ke tahap persiapan data seringkali diperlukan.

Bergantung pada kebutuhan bisnis, tugas data mining bisa berupa prediksi (klasifikasi atau regresi), asosiasi, atau clustering. Masing-masing tugas data mining ini dapat menggunakan

berbagai metode dan algoritma data mining. Beberapa metode data mining yang populer, termasuk pohon keputusan untuk klasifikasi, k-means untuk clustering, dan algoritma Apriori untuk aturan asosiasi, akan dijelaskan kemudian dalam bab ini.

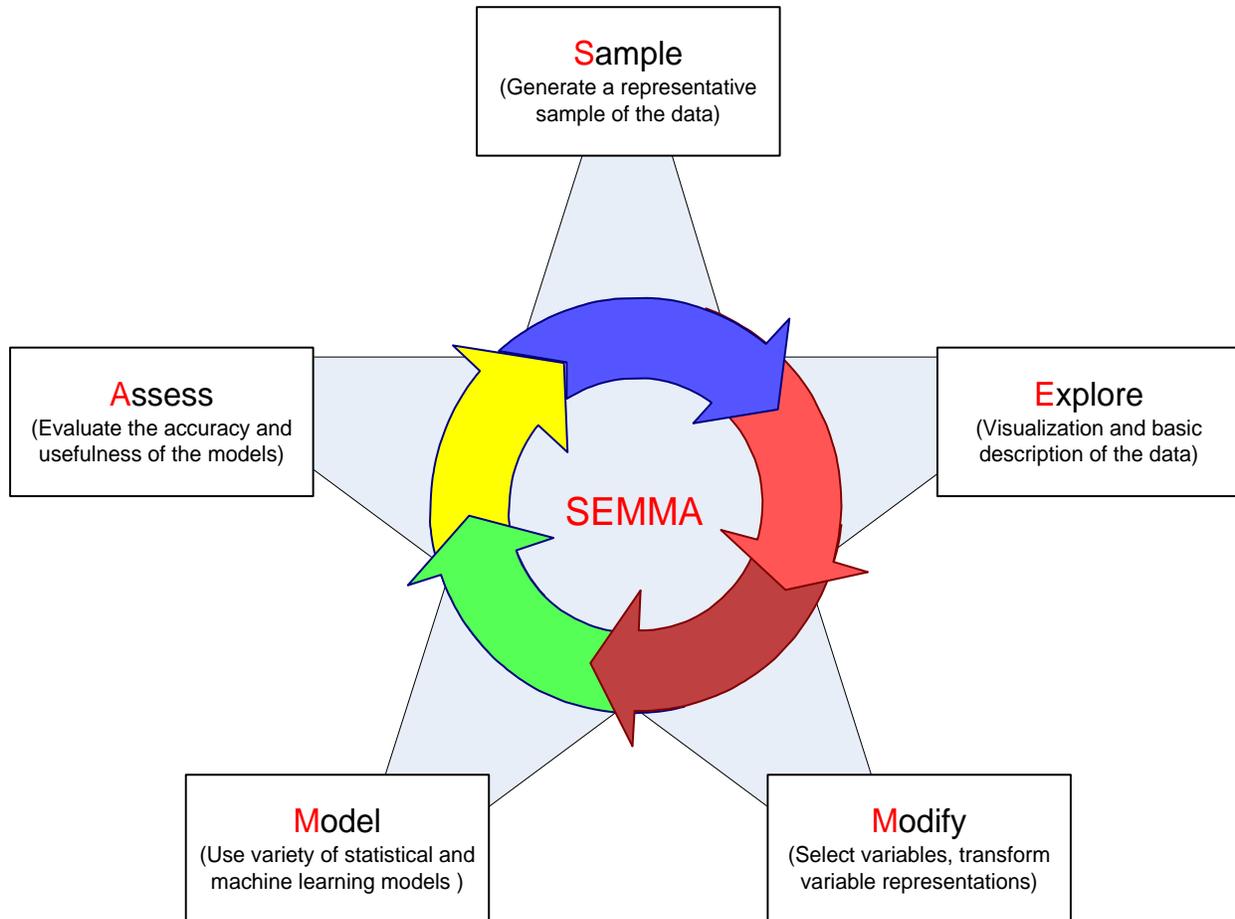
Langkah 5: *Testing and Evaluation*

Pada langkah 5, model yang sudah dikembangkan dinilai dan dievaluasi untuk akurasi. Langkah ini menilai sejauh mana model yang dipilih sesuai dengan tujuan bisnis, sehingga dapat diketahui apakah perlu untuk mengembangkan dan menilai lebih banyak model. Pilihan lainnya adalah menguji model yang dikembangkan dalam skenario dunia nyata jika waktu dan anggaran memungkinkan. Metode untuk evaluasi data mining akan dijelaskan kemudian dalam bab ini.

Langkah 6: Penerapan (*Deployment*)

Pengembangan dan penilaian model bukan merupakan akhir dari proyek data mining. Sekalipun tujuan dari model ini adalah untuk memiliki eksplorasi data yang sederhana, pengetahuan yang diperoleh dari eksplorasi data perlu disajikan sedemikian rupa sehingga *end user* dapat memahami dan mendapatkan keuntungan darinya. Bergantung pada kebutuhan, fase penerapan bisa sesederhana membuat laporan atau serumit penerapan proses data mining secara berulang di seluruh perusahaan. Penting untuk memahami terlebih dahulu tindakan apa yang perlu dilakukan agar model yang sudah dibuat bisa benar-benar dimanfaatkan.

Selain CRISP-DM, ada lagi metodologi terkenal yang dikembangkan oleh SAS Institute, yang disebut SEMMA (2009). Akronim SEMMA merupakan singkatan dari "Sample, Explore, Modify, Model, and Assess." Dimulai dengan sampel data yang representatif secara statistik, SEMMA mempermudah penerapan teknik statistik dan visualisasi eksploratif, memilih dan mengubah variabel prediktif yang paling signifikan, memodelkan variabel untuk memprediksi hasil, dan mengkonfirmasi ketepatan model. Representasi SEMMA dapat dilihat pada Gambar 4.6.



Gambar 4.6. Metodologi SEMMA

4.7. Metode *Data Mining*

Sejumlah teknik (atau algoritma) digunakan untuk pemodelan klasifikasi, termasuk yang berikut ini:

- **Decision tree analysis.** Analisis pohon keputusan (teknik *machine learning*) bisa dibilang merupakan teknik klasifikasi yang paling populer di arena data mining. Penjelasan tentang teknik ini diberikan pada bagian berikut.
- **Analisis statistik.** Teknik statistik adalah algoritma klasifikasi utama selama bertahun-tahun sampai munculnya teknik *machine learning*. Teknik klasifikasi statistik meliputi regresi logistik dan analisis diskriminan, yang keduanya membuat asumsi bahwa hubungan

antara variabel input dan output bersifat linier, data terdistribusi normal, dan variabel-variabelnya tidak berkorelasi dan saling independen satu sama lain.

- **Neural networks.** Ini adalah teknik *machine learning* yang paling populer yang dapat digunakan untuk masalah tipe klasifikasi.
- **Case-based reasoning.** Pendekatan ini menggunakan kasus historis untuk mengenali kesamaan dalam rangka menetapkan kasus baru ke dalam kategori yang paling mungkin.
- **Bayesian classifiers.** Pendekatan ini menggunakan teori probabilitas untuk membangun model klasifikasi berdasarkan kejadian masa lalu yang mampu menempatkan data baru ke dalam kelas yang paling mungkin. Naïve Bayes, salah satu teknik yang dapat digunakan, akan dijelaskan pada bagian berikut.
- **Algoritma genetika.** Penggunaan analogi evolusi alami untuk membangun mekanisme berbasis penelusuran untuk mengklasifikasikan sampel data.
- **Rough set.** Metode ini mempertimbangkan keanggotaan parsial label kelas ke kategori yang telah ditentukan sebelumnya dalam membangun model (kumpulan peraturan) untuk masalah klasifikasi.

Decision Tree

Sebelum menjelaskan rincian pohon keputusan, kita perlu membahas beberapa terminologi sederhana. Pertama, pohon keputusan mencakup banyak variabel input yang mungkin berdampak pada klasifikasi pola yang berbeda. Variabel input ini biasanya disebut atribut. Misalnya, jika kita membangun model untuk mengklasifikasikan risiko pinjaman berdasarkan dua karakteristik - pendapatan dan peringkat kredit - kedua karakter ini akan menjadi atribut dan keluaran yang dihasilkan adalah label kelas (misalnya, Rendah, medmm, atau berisiko tinggi). Kedua, pohon terdiri dari cabang dan simpul. Branch mewakili hasil tes untuk mengklasifikasikan suatu pola (berdasarkan uji) dengan menggunakan salah satu atribut. Simpul daun di sela mewakili pilihan kelas akhir untuk sebuah pola (rantai cabang dari simpul akar ke simpul daun yang dapat direpresentasikan sebagai pernyataan kompleks jika-kemudian). Gagasan dasar di balik pohon keputusan adalah pembagian secara rekursif membagi satu set pelatihan sampai masing-masing divisi seluruhnya terdiri dari contoh dari satu kelas. Setiap node nonleaf dari pohon berisi titik split, yang merupakan pengujian pada satu atau lebih atribut dan menentukan bagaimana data dibagi lebih jauh. Algoritma pohon keputusan, secara umum, membangun pohon awal dari data

pelatihan sehingga masing-masing simpul daun itu murni, dan kemudian memangkas pohon untuk meningkatkan generalisasinya, dan karenanya akurasi prediksi pada data uji.

Saat membangun pohon keputusan, tujuan di setiap simpul adalah menentukan atribut dan titik perpecahan dari atribut yang paling baik membagi catatan pelatihan untuk memurnikan representasi kelas pada simpul tersebut. Untuk mengevaluasi kebaikan perpecahan, beberapa indeks pemisahan telah diajukan. Dua yang paling umum adalah indeks Gini dan gain informasi. Indeks Gini digunakan dalam algoritma CART dan SPRINT (Scalable PaRalleizable Induction of Decision Trees). Versi keuntungan informasi digunakan di ID3 (dan versi yang lebih baru, C4.5 dan CS).

Induksi pohon keputusan adalah paradigma pembelajaran sederhana namun kuat. Dalam metode ini, seperangkat *training dataset* dipecah menjadi subset yang lebih kecil dan lebih kecil sementara pada saat bersamaan pohon keputusan berkembang secara bertahap. Pada akhir proses pembelajaran, sebuah pohon keputusan yang mencakup *training set* sudah terbentuk.

Secara umum, berikut ini adalah tahapan dalam melakukan pembelajaran Decision Tree:

1. Menguji semua atribut dan pilih pada yang akan berfungsi sebagai root terbaik
2. Membagi *training set* menjadi *subset* berdasarkan cabang dari node akar
3. Menguji atribut yang tersisa untuk melihat mana yang paling sesuai sebagai *branch* di bawah *root*
4. Melanjutkan proses ini untuk semua cabang lainnya sampai: a) Semua contoh dalam subset merupakan satu jenis; b) Tidak ada lagi contoh yang tersisa; c) Tidak ada lagi atribut yang tersisa

Untuk menentukan atribut mana yang terbaik di setiap cabang, kita menggunakan Entropy & Information Gain. Entropy (E) adalah jumlah bit minimum yang diperlukan untuk mengklasifikasikan contoh sebagai ya atau tidak. Entropy dapat dihitung dengan:

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Dimana S adalah training set, c adalah jumlah kelas, dan p_i adalah proporsi dari training set yang merupakan kelas i . Perlu diperhatikan bahwa operasi log yang digunakan adalah dengan basis 2. Untuk persamaan entropi, hasil dari $0 \log_2 0 = 0$. Sedangkan Information Gain $G(S,A)$ dimana A adalah atribut, dapat dihitung sebagai berikut:

$$G(S,A) \equiv E(S) - \sum_{v \text{ in Values}(A)} (|S_v| / |S|) * E(S_v)$$

Tabel 4.1. *Dataset cuaca*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Mari kita coba dengan contoh dataset prediksi cuaca seperti pada tabel 4.1. Diketahui bahwa $E([X +, Y-])$ menunjukkan bahwa ada X elemen pelatihan positif dan Y elemen pelatihan negatif. Oleh karena itu, entropi untuk data pelatihan $E(S)$, dapat digambarkan sebagai $E([9 +, 5-])$ karena dari 14 data pelatihan 9 di antaranya adalah iya dan 5 di antaranya adalah tidak. Sehingga, hasil perhitungan entropy adalah

$$\begin{aligned}
 E(S) &= E([9+,5-]) = (-9/14 \log_2 9/14) + (-5/14 \log_2 5/14) \\
 &= 0.94
 \end{aligned}$$

Selanjutnya, kita perlu menghitung informasi gain $G(S,A)$ untuk setiap atribut A dimana A diambil dari himpunan {Outlook, Temperature, Humidity, Wind}. Information gain untuk atribut {Outlook} adalah:

- $G(S, \text{Outlook}) = E(S) - [5/14 * E(\text{Outlook}=\text{sunny}) + 4/14 * E(\text{Outlook} = \text{overcast}) + 5/14 * E(\text{Outlook}=\text{rain})]$
- $G(S, \text{Outlook}) = E([9+,5-]) - [5/14 * E(2+,3-) + 4/14 * E([4+,0-]) + 5/14 * E([3+,2-])]$
- $G(S, \text{Outlook}) = 0.94 - [5/14 * 0.971 + 4/14 * 0.0 + 5/14 * 0.971]$
- $G(S, \text{Outlook}) = 0.246$

Information gain untuk atribut {Temperature} adalah:

- $G(S, \text{Temperature}) = E(S) - [4/14 * E(\text{Temperature}=\text{hot}) + 6/14 * E(\text{Temperature}=\text{mild}) + 4/14 * E(\text{Temperature}=\text{cool})]$
- $G(S, \text{Temperature}) = 0.94 - [4/14 * E([2+, 2-]) + 6/14 * E([4+, 2-]) + 4/14 * E([3+, 1-])]$
- $G(S, \text{Temperature}) = 0.94 - [4/14 + 6/14 * 0.918 + 4/14 * 0.811]$
- $G(S, \text{Temperature}) = 0.029$

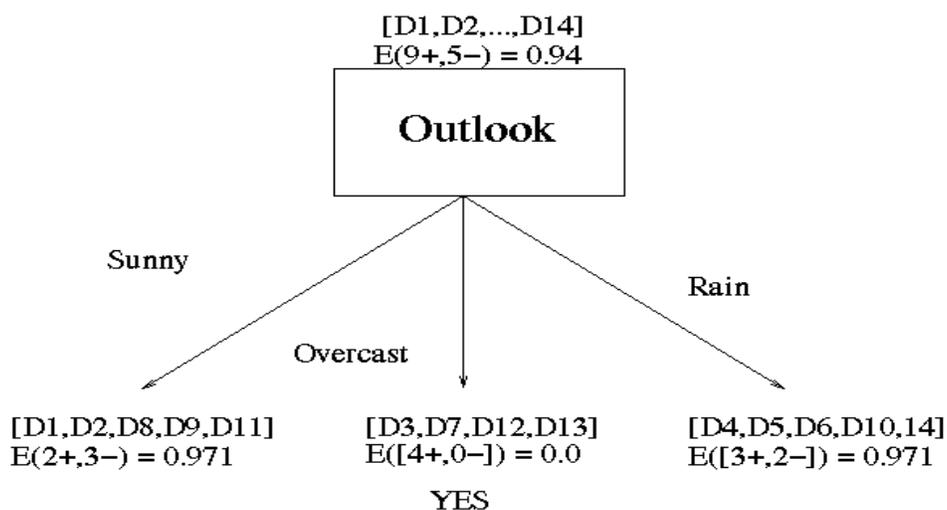
Information gain untuk atribut {Humidity} adalah:

- $G(S, \text{Humidity}) = E(S) - [7/14 * E(\text{Humidity}=\text{high}) + 7/14 * E(\text{Humidity}=\text{normal})]$
- $G(S, \text{Humidity}) = 0.94 - [7/14 * E([3+, 4-]) + 7/14 * E([6+, 1-])]$
- $G(S, \text{Humidity}) = 0.94 - [7/14 * 0.985 + 7/14 * 0.592]$
- $G(S, \text{Humidity}) = 0.1515$

Information gain untuk atribut {Wind} adalah:

- $G(S, \text{Wind}) = E(S) - [8/14 * E(\text{Wind}=\text{weak}) + 6/14 * E(\text{Wind}=\text{strong})]$
- $G(S, \text{Wind}) = 0.94 - [8/14 * E([6+, 2-]) + 6/14 * E([3+, 3-])]$
- $G(S, \text{Wind}) = 0.94 - [8/14 * 0.811 + 6/14 * 1.00]$
- $G(S, \text{Wind}) = 0.048$

Dari hasil perhitungan information gain keempat atribut tersebut, dapat diketahui bahwa atribut outlook adalah yang paling besar. Dengan demikian, atribut outlook dipilih sebagai root dari Decision Tree yang akan dibangun.



Gambar 4.7. Decision tree untuk dataset cuaca

Setelah kita mendapatkan root untuk Decision Tree, selanjutnya langkah-langkah tersebut akan dilakukan kembali untuk menentukan node dari masing-masing atribut Sunny, Overcast, dan Rain. Untuk branch (Outlook = overcast), karena semua data yang masuk dalam subset ini memiliki class output yang sama (Yes) dan memiliki nilai $E=0$, maka tidak perlu dipecah lagi.

Naïve Bayes

Algoritma Naïve Bayes berbasiskan perhitungan probabilistik dengan asumsi bahwa setiap fitur yang digunakan saling lepas. Naïve Bayes merupakan metode klasifikasi yang paling populer digunakan. Algoritma ini memiliki kelebihan dari sisi kecepatan pembelajaran dan toleransinya terhadap nilai yang hilang (*missing value*) dari fitur. Untuk menangani data numerik, algoritma ini menggunakan *probability density function*, artinya data dianggap mengikuti distribusi normal untuk kemudian dihitung nilai rata-rata dan simpangan bakunya. Untuk merepresentasikan sebuah kelas, terdapat karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi yang berguna untuk menjelaskan bahwa peluang masuknya sampel karakteristik tertentu kedalam kelas posterior. Peluang munculnya suatu kelas C (sebelum masuknya sampel tersebut, seringkali disebut prior), dikali dengan peluang kemunculan karakteristik sampel secara global disebut juga *evidence* (E). Nilai evidence selalu tetap untuk setiap kelas pada satu sampel. Nilai posterior tersebut dibandingkan dengan nilai posterior kelas lainnya untuk menentukan ke dalam kelas apa suatu sampel.

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

The diagram illustrates the components of the Naïve Bayes formula. It shows the equation $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$ with arrows pointing from descriptive labels to the corresponding parts of the formula:

- 'Likelihood' points to $P(x | c)$
- 'Class Prior Probability' points to $P(c)$
- 'Posterior Probability' points to $P(c | x)$
- 'Predictor Prior Probability' points to $P(x)$

Untuk Naïve Bayes rule, probabilitas apriori kelas C adalah probabilitas kejadian sebelum bukti terlihat, sedangkan probabilitas posteriori C adalah probabilitas kejadian setelah bukti terlihat. Untuk menghitung probabilitas dari kelas C jika ada evidence E yaitu:

$$P(C | E) = \frac{P(E | C)P(C)}{P(E)}$$

Pada Naïve Bayes untuk klasifikasi, kita menggunakan Naïve *assumption* yaitu *evidence* terbagi menjadi beberapa bagian (yaitu atribut) yang independen. Sehingga, untuk s atribut dapat dihitung sebagai berikut:

$$P[C | E] = \frac{P[E_1 | C]P[E_2 | C] \dots P[E_s | C]P(C)}{P[E]}$$

Sebagai latihan, kita menggunakan kembali *dataset* cuaca. Langkah pertama adalah menghitung *likelihood* dari masing-masing atribut. Di sini ada 4 atribut (*outlook*, *temperature*, *humidity*, *windy*), kemudian kita hitung berapa kali atribut tersebut muncul untuk tiap kelas (Yes dan No). Perhitungan tersebut dapat dilihat pada gambar 4.8 di bawah ini.

Outlook			Temperature			Humidity		Windy		Play			
	Yes	No		Yes	No		Yes	No		Yes	No		
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Gambar 4.8. Probabilitas untuk dataset cuaca

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Sekarang, kita akan mencoba untuk memprediksi kelas output dari suatu *instance*, jika diketahui *evidence* seperti di atas. Caranya adalah mengalikan *likelihood* dari setiap atribut (dibagi dengan berapa kali kelas tersebut muncul pada *dataset training*), kemudian dikalikan dengan probabilitas kelas prior (untuk kelas Yes adalah 9/14 dan kelas No adalah 5/14). Sehingga kita dapat menghasilkan perhitungan sebagai berikut:

Likelihood of the two classes

$$\text{For "yes"} = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$$

$$\text{For "no"} = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$$

Conversion into a probability by normalization:

$$P(\text{"yes"}) = 0.0053 / (0.0053 + 0.0206) = 0.205$$

$$P(\text{"no"}) = 0.0206 / (0.0053 + 0.0206) = 0.795$$

Dari hasil perhitungan di atas, dapat diketahui bahwa perobabilitas untuk terjadi kelas “Yes” adalah 0,205 (20,5%) sedangkan untuk kelas “No” adalah 0,795 (79,5). Kita dapat menyimpulkan bahwa untuk data dengan atribut outlook = “sunny”, temperature = “cool”, humidity = “high”, windy = “true”, kelas yang muncul adalah “No” dengan probabilitas 79,5%.

4.8. Data Mining Software

Banyak vendor perangkat lunak menyediakan alat data mining yang *powerful*. Contoh vendor tersebut di antaranya SPSS (PASW Modeler), SAS (Enterprise Miner), StatSoft (Statistica Data Miner), Salford (CART, MARS, TreeNet, RandomForest), Angoss (KnowledgestudIO, KnowledgeSeeker), dan Megaputer (PolyAnalyst). Seperti dapat dilihat, sebagian besar alat yang lebih populer dikembangkan oleh perusahaan perangkat lunak statistik terbesar (SPSS, SAS, dan StatSoft). Sebagian besar vendor *Business Intelligence tools* (misalnya, IBM Cognos, Oracle Hyperion, SAP Business Objects, MicroStrategy, Teradata, dan Microsoft) juga memiliki kapabilitas data mining yang terintegrasi ke dalam perangkat lunak yang mereka tawarkan. Alat BI ini masih lebih fokus pada pemodelan multidimensional dan visualisasi data sehingga tidak dianggap sebagai pesaing langsung dari vendor *software* data mining. Selain *software* komersial ini, beberapa *software open source* untuk data mining sudah tersedia secara online.

Salah satu tool data mining terpopuler (dan open source) saat ini adalah Weka, yang dikembangkan oleh sejumlah peneliti dari Universitas Waikato di Selandia Baru (bisa diunduh dari cs.waikato.ac.nz/ml/weka/). Weka mencakup sejumlah besar algoritma untuk berbagai tugas data mining dan memiliki antarmuka pengguna yang intuitif. Alat data mining gratis (untuk penggunaan nonkomersial) lainnya yang baru dirilis adalah RapidMiner (dikembangkan oleh

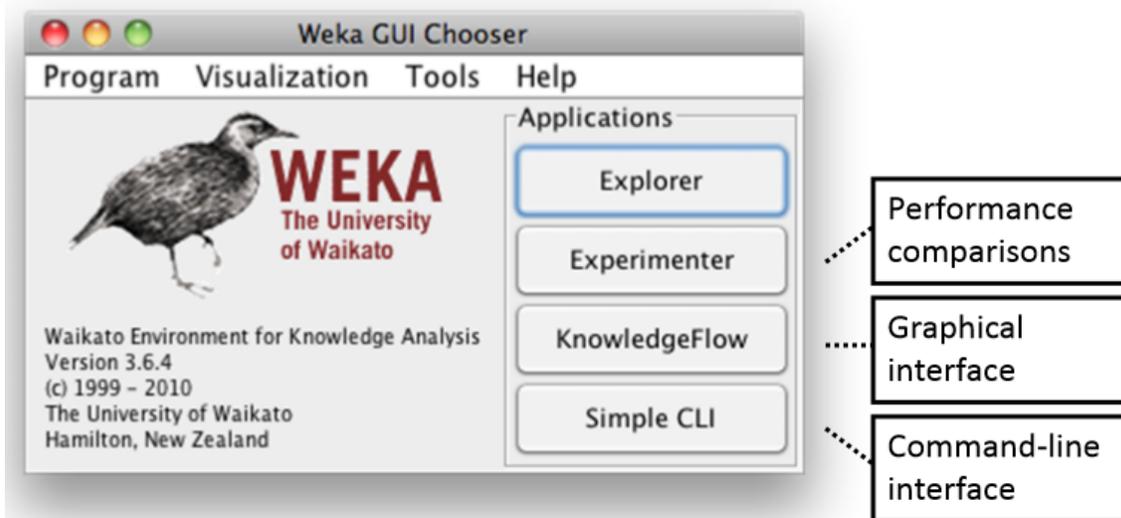
Rapid-I; dapat diunduh dari rapid-i.com). Antarmuka pengguna yang ditingkatkan secara grafis, penggunaan sejumlah algoritma yang cukup besar, dan penggabungan berbagai fitur visualisasi data membedakannya dari perangkat gratis lainnya.

4.9. Pengenalan Weka

WEKA adalah sistem data mining yang dikembangkan oleh University of Waikato di Selandia Baru yang menerapkan algoritma data mining. WEKA adalah fasilitas mutakhir untuk pengembangan teknik *data mining* dan aplikasinya terhadap masalah data dunia nyata. Terdapat berbagai algoritma machine learning untuk tugas data mining. Algoritma diterapkan langsung ke dataset. WEKA menerapkan algoritma untuk preprocessing data, klasifikasi, regresi, clustering, maupun asosiasi; Ini juga mencakup alat visualisasi. WEKA adalah perangkat lunak open source yang diterbitkan di bawah GNU General Public License. Weka dapat diunduh pada <http://www.cs.waikato.ac.nz/ml/weka> (untuk Windows, Mac, atau Linux).

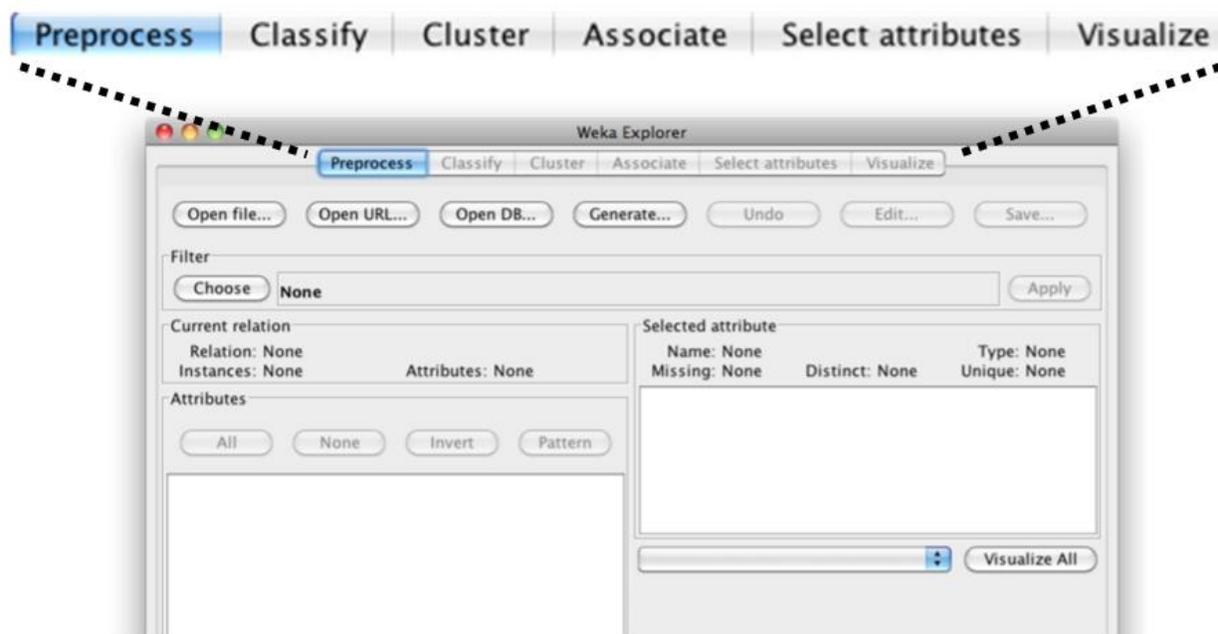
4.10. Weka Explorer

Kita akan memulai dengan mengenal bagian dari Weka Explorer. Pertama kali Adna membuka Weka, maka muncul tampilan seperti gambar di bawah ini. Klik pada Weka Explorer.



Gambar 4.9. Menu awal pada Weka

Sekarang Anda sudah masuk ke halaman Explorer seperti pada gambar 4.10. Di bagian atas, ada lima panel: panel Preprocess; Panel Classify, tempat Anda membangun *classifiers* untuk dataset; Clustering, prosedur clustering dari Weka; Association rules; Attribute selection; dan Visualization. Dalam latihan ini, kita akan menggunakan panel Preprocess untuk membuka file dan menyiapkan data, panel Classify untuk bereksperimen dengan classifiers, dan panel Visualisasi untuk memvisualisasikan dataset.



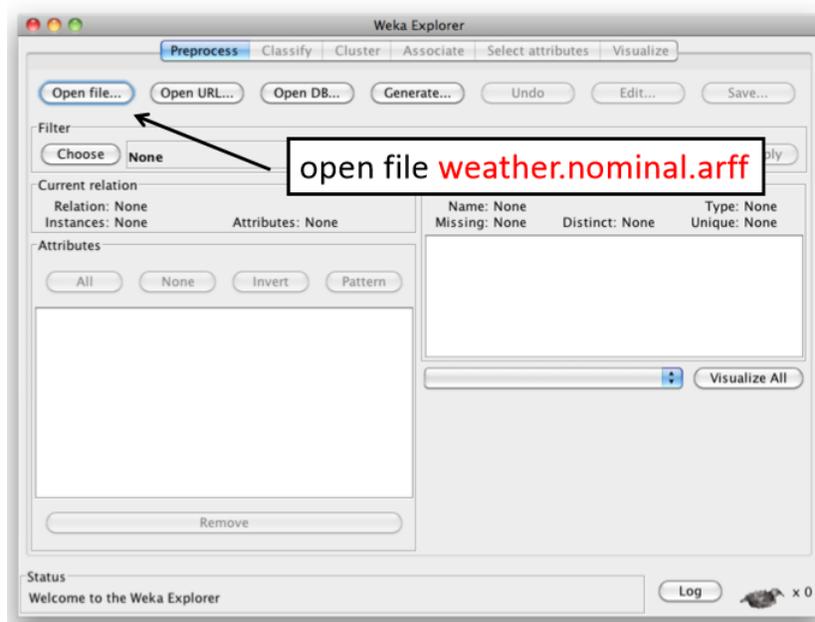
Gambar 4.10. Weka Explorer

Kita akan mulai dengan membuka sebuah dataset. Dataset yang akan kita buka adalah dataset cuaca; Ini adalah dataset kecil yang sudah kita lihat sebelumnya. Ada 14 *instances*, yaitu 14 sampel hari, dan untuk masing-masing hari ini, kita mempunyai nilai lima atribut. Empat berhubungan dengan cuaca: Outlook, Temperature, Humidity, dan Windy. Yang kelima, Play, adalah untuk menentukan apakah kita akan bermain atau tidak. Sebenarnya, apa yang akan kita lakukan adalah meramalkan atribut Play berdasarkan atribut lainnya.

		attributes				
		Outlook	Temp	Humidity	Windy	Play
instances	1	Sunny	Hot	High	False	No
	2	Sunny	Hot	High	True	No
	3	Overcast	Hot	High	False	Yes
	4	Rainy	Mild	High	False	Yes
	5	Rainy	Cool	Normal	False	Yes
	6	Rainy	Cool	Normal	True	No
	7	Overcast	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	Overcast	Mild	High	True	Yes
	13	Overcast	Hot	Normal	False	Yes
	14	Rainy	Mild	High	True	No

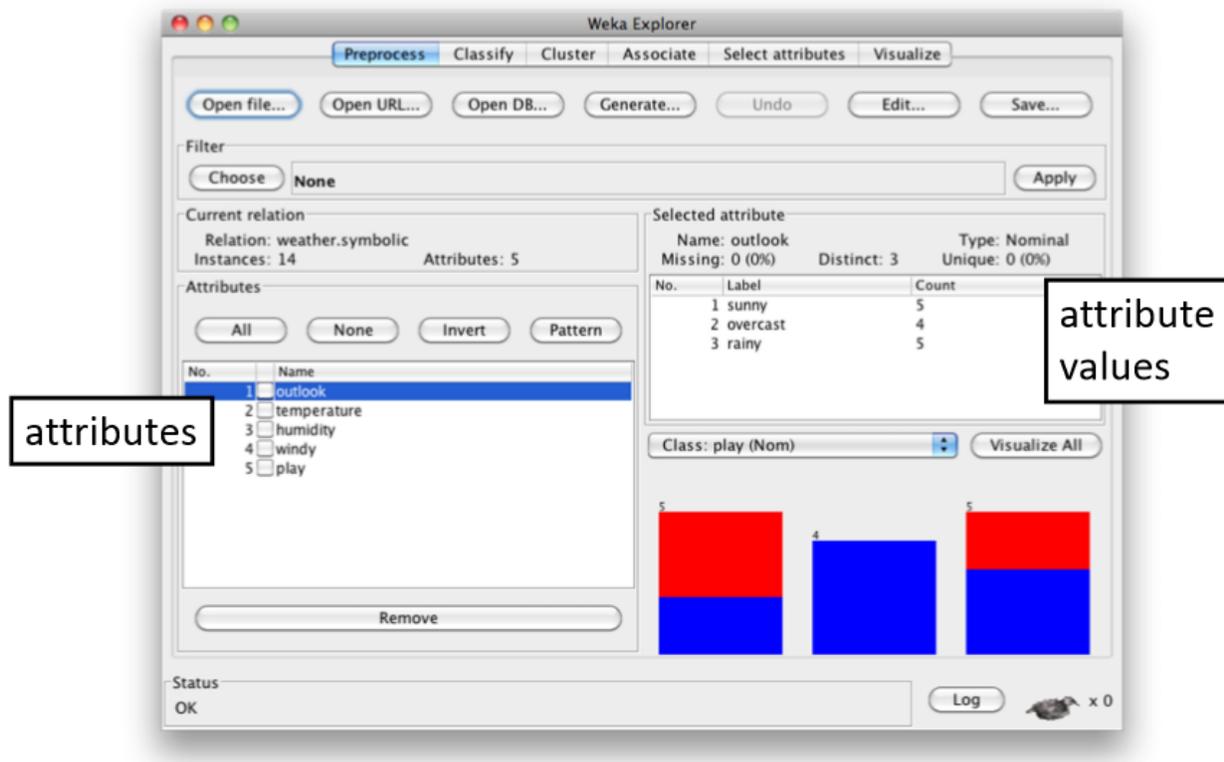
Gambar 4.11. Dataset cuaca untuk Weka

Mari kita buka dataset dan lihat di Weka. Kita akan membuka `weather.nominal.arff`. Semua file data Weka disebut file ARFF. Abaikan bar warna-warni ini saat ini. Ada 14 contoh; Ini sesuai dengan 14 hari yang kita lihat di dataset pada slide. Untuk setiap hari kita memiliki lima atribut: Outlook, Temperature, Humidity, Windy, dan Play. Jika Anda memilih salah satu atribut ini, kita dapat melihat nilainya.



Gambar 4.12. Membuka dataset pada Weka

Nilai atribut Outlook yaitu cerah, mendung, dan hujan. Tampilan pada gambar 4.13 adalah berapa kali mereka muncul di dataset: 5 hari cerah, 4 hari mendung, dan 3 hari hujan, dengan total 14 kejadian. Jika kita melihat atribut temperature, panas, ringan, dan sejuk adalah nilai yang mungkin, dan terdapat jumlah berapa kali jumlah tersebut muncul di dataset.



Gambar 4.13. Nilai atribut *outlook*

4.11. Evaluasi *Data Mining*

Dua tahapan metodologi yang paling umum dalam data mining mencakup pengembangan model dan pengujian / penerapan model. Pada tahap pengembangan model, kumpulan data input, termasuk label aktual dari kelas aktual digunakan. Setelah model telah dilatih, model diuji terhadap data sampel untuk penilaian akurasi dan akhirnya digunakan untuk penggunaan aktual yaitu memprediksi kelas dari instance data baru (di mana label kelas tidak diketahui). Beberapa faktor dipertimbangkan dalam menilai model, diantaranya sebagai berikut:

- Prediktif akurasi. Kemampuan model untuk memprediksi dengan benar label kelas dari data baru atau yang sebelumnya tidak terlihat. Prediksi akurasi adalah faktor penilaian yang paling umum digunakan untuk model klasifikasi. Untuk menghitung ukuran ini, label kelas

aktual dari kumpulan data uji dicocokkan dengan label kelas yang diprediksi oleh model. Keakuratan kemudian dapat dihitung sebagai tingkat akurasi, yaitu persentase sampel dataset uji yang diklasifikasikan dengan benar oleh model.

- Kecepatan. Biaya komputasi yang terlibat dalam menghasilkan dan menggunakan model, dimana lebih cepat dianggap lebih baik.
- *Robustness*. Kemampuan model untuk membuat prediksi yang cukup akurat, walaupun diberi data dengan *noise* atau dengan nilai yang hilang dan salah.
- Skalabilitas. Kemampuan untuk membuat model prediksi secara efisien walaupun dengan jumlah data yang besar. •
- *Interpretability*. Tingkat pemahaman dan wawasan yang diberikan oleh model

Dalam masalah klasifikasi, sumber utama untuk estimasi akurasi adalah *confusion matrix* (juga disebut matriks klasifikasi atau tabel kontingensi). Gambar 4.14 menunjukkan *confusion matrix* untuk masalah klasifikasi dua kelas. Angka di sepanjang diagonal dari atas kiri ke kanan bawah mewakili keputusan yang benar, dan angka di luar diagonal ini mewakili kesalahannya. Gambar 4.15 menunjukkan perhitungan untuk metrik akurasi umum untuk model klasifikasi. Untuk masalah yang tidak biner (lebih dari dua kelas output), *confusion matrix* semakin besar (ukuran yang sama dengan ukuran jumlah label kelas yang unik), dan metrik akurasi menjadi terikat pada akurasi ketepatan per kelas dan akurasi klasifikasi keseluruhan.

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

Gambar 4.14. *Confusion Matrix*

$\text{True Positive Rate} = \frac{TP}{TP + FN}$	The ratio of correctly classified positives divided by the total positive count (i.e., hit rate or recall)
$\text{True Negative Rate} = \frac{TN}{TN + FP}$	The ratio of correctly classified negatives divided by the total negative count (i.e., false alarm rate)
$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$	The ratio of correctly classified instances (positives and negatives) divided by the total number of instances
$\text{Precision} = \frac{TP}{TP + FP}$	The ratio of correctly classified positives divided by the sum of correctly classified positives and incorrectly classified positives
$\text{Recall} = \frac{TP}{TP + FN}$	Ratio of correctly classified positives divided by the sum of correctly classified positives and incorrectly classified negatives

Gambar 4.15. Pengukuran akurasi klasifikasi

4.13. Rangkuman

- Data Mining adalah proses menggali pengetahuan baru dari database
- Data Mining dapat menggunakan format file sederhana sebagai sumber data, dapat juga berjalan dengan data pada Data Warehouse
- Data Mining merupakan irisan dari banyak disiplin ilmu, termasuk statistik, *artificial intelligence*, sistem informasi, dan pemodelan matematika
- Organisasi bisnis menggunakan Data Mining untuk lebih memahami pelanggan mereka dan mengoptimalkan operasi bisnis yang ada
- Penerapan Data Mining dapat ditemukan di hampir semua bidang bisnis.
- Data Mining dapat dikelompokkan menjadi tiga tugas: prediction (klasifikasi atau regresi), clustering, atau association.
- Seperti halnya proyek sistem informasi, sebuah Data Mining harus mengikuti proses manajemen proyek agar dapat berhasil.
- CRISP-DM dan SEMMA merupakan contoh metodologi proses Data Mining yang sudah banyak digunakan.
- Tahap persiapan data merupakan tahap paling krusial dalam kesuksesan Data Mining. Data yang bagus menghasilkan informasi yang baik; Informasi yang baik mengarah pada keputusan yang baik.

- Metode klasifikasi belajar dari sampel sebelumnya yang berisi atribut input dan kelas output yang dihasilkan, dan sekali dilatih dengan benar akan dapat mengklasifikasikan kasus di masa depan.
- Metode clustering mempartisi sampel menjadi beberapa segmen atau kelompok. Setiap anggota segmen memiliki karakteristik yang sama.
- Data mining dapat berupa hypothesis-driven atau discovery-driven. Data mining hypothesis-driven dimulai dengan proposisi atau hipotesis oleh pengguna. Discovery-driven adalah penelitian yang lebih terbuka.
- Metode asosiasi digunakan untuk menemukan dua atau lebih item (atau kejadian atau konsep) yang berkaitan.
- Aturan asosiasi biasanya disebut sebagai *market-basket analysis*.
- Banyak *software data mining* yang tersedia, baik yang komersial maupun *open-source*. Alat data mining komersial yang paling populer adalah SPSS PASW dan SAS Enterprise Miner. Alat data mining gratis yang paling populer adalah Weka dan RapidMiner.

4.14. Latihan Soal

1. Carilah referensi jurnal tentang penerapan Data Mining dengan CRISP-DM. Buatlah resume berisi ringkasan jurnal tersebut.
2. Lakukan komparasi dari beberapa metode klasifikasi data mining. Sebutkan kelebihan dan kekurangan masing-masing metode tersebut.
3. Jika diketahui data cuaca dengan atribut outlook = “overcast”, temperature = “hot”, humidity = “high”, windy = “false”, tentukan class output menggunakan metode Naïve Bayes.

BAB V

BUSINESS PERFORMANCE MANAGEMENT



Kompetensi

Mahasiswa dapat menyusun Business Performance Management (BPM) (minggu 11-12)

Bahan Kajian

- Konsep BPM (*Business Performance Management*)
- BPM Proses: Strategi, Plan, Monitor, Act & Adjust
- *Critical Success Factors* (CSF)
- *Key Performance Indicators* (KPI)
- Pengukuran performa bisnis
- Metodologi BPM
- *Balanced Scorecards* (BSC)
- Arsitektur BPM

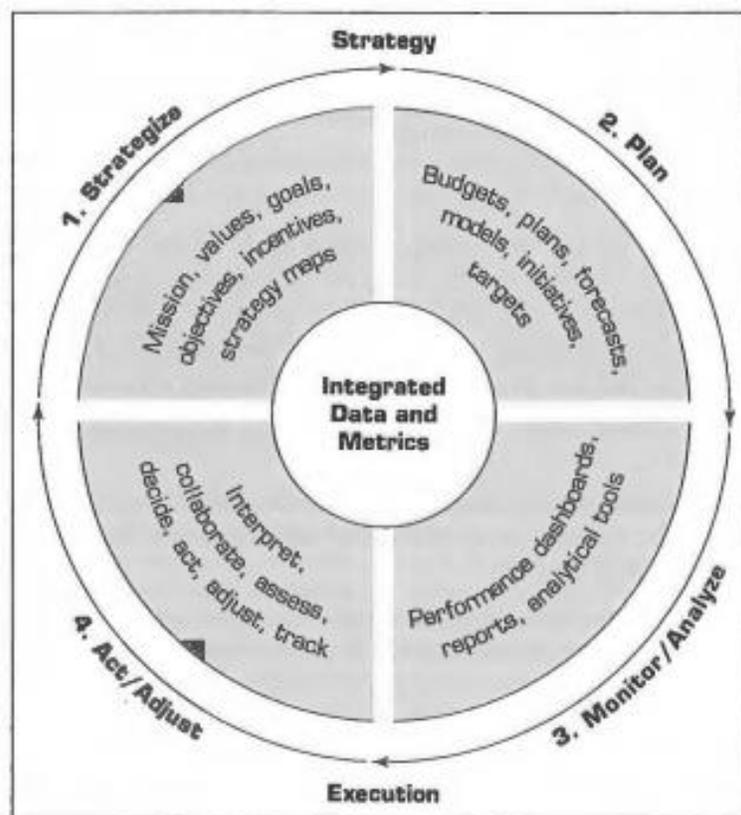
5.1. Definisi BPM

Business Performance Management (BPM) adalah sebuah sistem yang digunakan oleh organisasi untuk mengukur, memonitor, dan mengontrol performa bisnis. BPM adalah pengembangan dari BI dan melibatkan banyak teknologi, aplikasi, dan teknik BI. Menurut Hossain (2016), Business performance management (BPM) membantu organisasi mencapai peningkatan efektivitas dan daya saing dengan menjembatani kesenjangan antara strategi dan eksekusi.

5.2. Siklus Proses BPM

BPM mencakup siklus proses yang menghubungkan strategi hingga eksekusi untuk optimalisasi performa bisnis, yang dicapai melalui:

- Menentukan target dan tujuan
- Mengembangkan rencana untuk mencapai target
- Memonitor performa actual dibandingkan dengan target dan tujuan
- Mengambil tindakan perbaikan



Gambar 5.1. Siklus proses BPM

1. **Penentuan Strategi.** Istilah strategi memiliki banyak definisi. Untuk menghindari kebingungan, hal itu juga sering digunakan dalam kombinasi dengan berbagai istilah lain seperti visi strategis dan fokus strategis. Terlepas dari perbedaan makna, mereka semua menjawab pertanyaan "Ke mana kita ingin pergi di masa depan?" Bagi kebanyakan perusahaan, jawaban atas pertanyaan ini diberikan dalam rencana strategis. Anda dapat memikirkan rencana strategis sebagai peta, yang merinci tindakan untuk memindahkan organisasi dari keadaan saat ini ke visi masa depannya.
2. **Perencanaan.** Ketika manajer operasional mengetahui dan memahami apa (yaitu tujuan dan sasaran organisasi), mereka akan dapat menjelaskan bagaimana (yaitu, rencana operasional dan keuangan terperinci). Rencana operasional dan keuangan menjawab dua pertanyaan: Taktik dan inisiatif apa yang akan dicapai untuk memenuhi target kinerja yang ditetapkan oleh rencana strategis? Apa hasil keuangan yang diharapkan dari pelaksanaan taktik?
3. **Monitoring.** Ketika rencana operasional dan keuangan sedang berjalan, sangat penting agar kinerja organisasi dipantau. Kerangka kerja komprehensif untuk memantau kinerja harus membahas dua isu utama: apa yang harus dipantau dan bagaimana cara memonitor. Karena tidak mungkin melihat segala sesuatu, sebuah organisasi perlu berfokus pada pemantauan isu-isu spesifik. Setelah organisasi mengidentifikasi indikator atau langkah untuk dilihat, perlu dikembangkan strategi untuk memantau faktor-faktor tersebut dan merespons secara efektif.
4. **Aksi/perbaikan.** Sebuah proyek dapat gagal dalam beberapa cara yang berbeda, mulai dari mempertimbangkan terlalu sedikit pilihan atau skenario, gagal mengantisipasi pergerakan pesaing, mengabaikan perubahan dalam lingkungan ekonomi atau sosial, meramalkan permintaan secara tidak akurat, atau meremehkan investasi yang dibutuhkan untuk sukses, hanya untuk Nama beberapa kemungkinan. Inilah sebabnya mengapa penting bagi perusahaan untuk terus memantau hasilnya, menganalisis apa yang telah terjadi, menentukan mengapa hal itu terjadi, dan menyesuaikan tindakannya sesuai dengan itu.

5.3. Pengukuran Performa

Pengukuran performa merupakan hal yang vital dalam penerapan **Business Performance Management**. Tanpa pengukuran performa yang baik, kita akan kesulitan untuk memonitor performa organisasi sehingga pengambilan keputusan untuk meningkatkan performa organisasi menjadi lebih sulit dilakukan. Oleh karena itu, diperlakukan suatu sistem untuk mengatur pengukuran performa yang biasa disebut **Performance Measurement System**. Sistem ini bertujuan untuk membantu manajer mengontrol implementasi dari strategi bisnis, yaitu dengan membandingkan hasil aktual dengan tujuan dan target strategis yang sudah dicanangkan organisasi.

Dalam pengukuran performa ini, ada banyak aspek yang dapat dijadikan acuan atau perspektif. Suatu organisasi bisnis, ataupun stakeholder dari organisasi tersebut, umumnya akan melihat data keuangan/finansial sebagai aspek pengukuran performa. Walaupun ini bukan hal yang salah, akan tetapi performa organisasi tidak melulu hanya bergantung pada aspek finansial. Pengukuran finansial merupakan indikator yang menunjukkan apa kondisi yang terjadi saat ini, namun tidak menunjukkan alasan kenapa itu terjadi dan apa yang mungkin terjadi di masa depan. Bagaimana kondisi proses-proses yang ada di dalam organisasi untuk menghasilkan suatu nilai juga tidak bisa diukur melalui pengukuran finansial. Selain itu, pengukuran finansial lebih berfokus pada jangka pendek, sehingga menghasilkan sedikit informasi untuk jangka panjang.

Bagaimana melakukan pengukuran yang efektif merupakan tantangan yang harus dijawab setiap organisasi bisnis. Ada banyak sekali parameter yang dapat digunakan dalam melakukan pengukuran, namun tidak semuanya menjadi poin krusial dalam pengukuran performa. Sebagai contoh, untuk industri restoran makanan cepat saji, kecepatan pelayanan makanan tentu menjadi standar ukuran untuk mencapai kesuksesan. Oleh karena itu, mereka akan fokus untuk melakukan pengukuran pada aspek-aspek yang dapat menjadi faktor penentu kesuksesan yaitu **Critical Succes Factors (CSF)**. Dengan berfokus pada CSF, kita dapat mencegah sumber daya yang terbuang sia-sia karena pengukuran aspek-aspek yang tidak terlalu krusial bagi suatu organisasi. Pengukuran yang baik juga harus mempertimbangkan kebutuhan semua stakeholder (shareholder, karyawan, partner, supplier, dan sebagainya) dengan seimbang.

Key Performance Indicators (KPI) mewakili tujuan strategis dan mengukur kinerja terhadap suatu tujuan. Menurut Eckerson (2009), KPI bersifat multidimensi. ini berarti bahwa KPI memiliki beragam fitur yang membedakan, diantaranya:

- Strategi. KPI mewujudkan tujuan strategis.
- Target. KPI mengukur kinerja terhadap target tertentu. Sasaran ditentukan dalam sesi strategi, perencanaan, atau anggaran dan dapat mengambil bentuk yang berbeda (mis., Target pencapaian, target pengurangan, target absolut).
- Rentang. Target memiliki rentang kinerja (mis., di atas, tepat, atau di bawah target).
- *Encodings*. Rentang dikodekan dalam perangkat lunak, memungkinkan tampilan kinerja visual (misalnya: Hijau, kuning, merah). Pengkodean dapat didasarkan pada persentase atau aturan yang lebih kompleks.
- Time frame. Sasaran diberi kerangka waktu yang harus diselesaikannya. Kerangka waktu sering dibagi menjadi beberapa bagian yang lebih kecil untuk menghasilkan kinerja *milepost*.
- *Benchmark*. Target diukur berdasarkan *baseline* atau *benchmark*. Hasil tahun sebelumnya sering dijadikan patokan, namun angka arbitrase atau tolok ukur eksternal juga dapat digunakan.

Menurut Marr (2016), terdapat beberapa KPI yang dapat digunakan sebagai pengukuran pada beberapa aspek, di antaranya yaitu pelanggan, finansial, proses internal, dan karyawan. Salah satu KPI yang banyak digunakan untuk mengukur dan memahami pelanggan adalah *Net Promotor Score* (NPS). NPS digunakan untuk mengetahui seberapa besar kemungkinan pelanggan Anda merekomendasikan perusahaan atau produk Anda kepada orang lain. NPS dilandasi oleh premis mendasar bahwa pelanggan dapat dibagi dalam tiga kelompok: promotor, pasif, atau pencela.

Kita bisa menggunakan contoh perhitungan sederhana sebagai berikut. Sebuah perusahaan melakukan survei terhadap 1000 pelanggannya dengan pertanyaan: “Seberapa besar Anda merekomendasikan produk perusahaan ini kepada teman Anda?” Para responden dapat menjawab dengan menggunakan skala 1-10 (0 = tidak sama sekali, 5 = netral, 10 = kemungkinan besar). Hasil dari survei yang dilakukan dapat dilihat pada tabel 5.1.

Tabel 5.1. Hasil survei NPS

Nilai	Jumlah
0	1
1	2
2	0
3	0
4	2
5	5
6	10
7	110
8	170
9	400
10	300
TOTAL	1000

Dengan menggunakan skala 1 hingga 10, sebuah perusahaan dapat menghitung NPS-nya dengan mencatat presentase Promotor dan mengurangnya dengan presentasi Pencela. Promotor (nilai 9-10) adalah orang-orang yang setia untuk terus membeli dan memberi rujukan pada orang lain. Pasif (nilai 7-8) adalah pelanggan yang puas tapi tidak antusias, sehingga rentan terhadap penawaran kompetitor. Pencela (nilai 0-6) adalah pelanggan yang tidak senang, yang bisa merusak merek Anda dan merusak pertumbuhan lewat pemberitaan negatif. NPS dapat dihitung sebagai berikut:

$$\text{NPS} = (\% \text{ pelanggan Promotor}) - (\% \text{ pelanggan Pencela})$$

$$\text{Promotor} = (300+400)/1000 = 0,7 = 70\%$$

$$\text{Pencela} = (1+2+0+0+2+5+10)/1000 = 0,02 = 2\%$$

$$\text{NPS} = 70\% - 2\% = 68\%$$

Indikator ini menunjukkan bahwa pelanggan akan merekomendasikan produk dari perusahaan tersebut dengan kemungkinan sebesar 68%.

5.4. Metodologi BPM

Dengan suatu *Performance Measurement System* yang baik, organisasi akan memperoleh banyak manfaat di antaranya:

1. Menyelaraskan target strategis (*top-level*) dengan rencana operasional (*bottom-level*)
2. Mengidentifikasi peluang dan problem yang muncul setiap waktu
3. Menentukan prioritas dan mengalokasikan sumber daya berdasarkan prioritas
4. Fleksibel dalam mengatur pengukuran sesuai dengan kebutuhan dan strategi

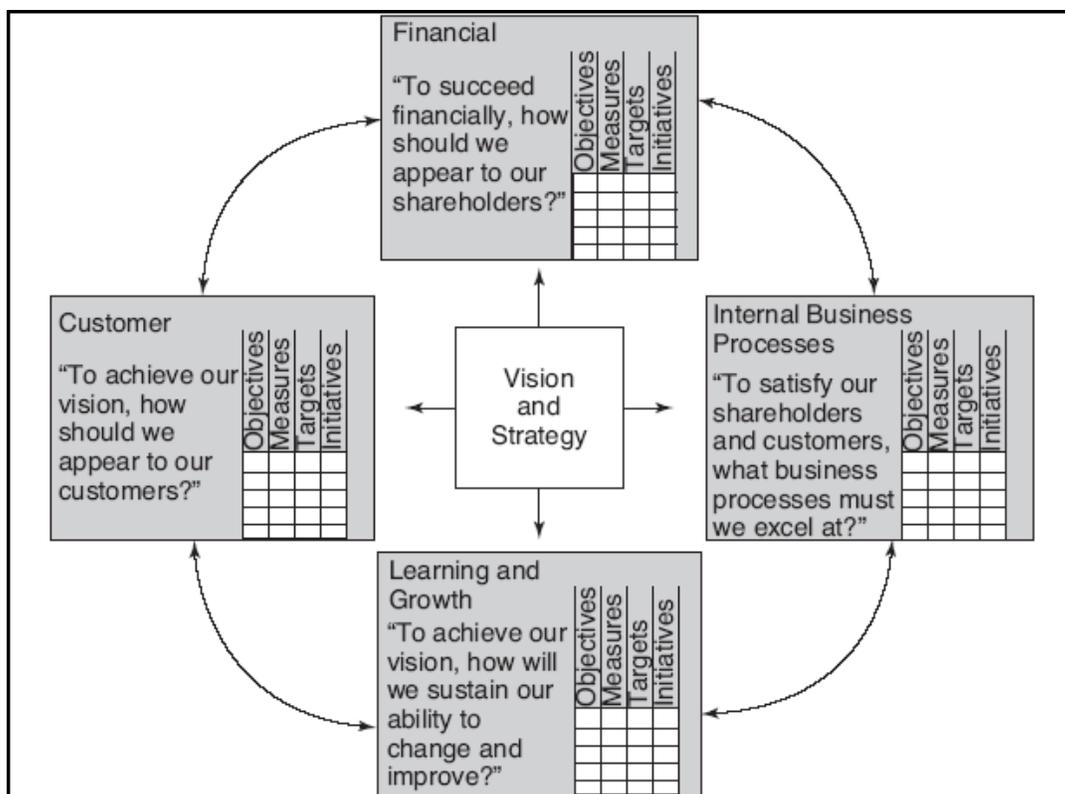
Agar pengukuran ini dapat memberikan manfaat secara optimal, ada beberapa metodologi yang dapat diadopsi. Sistem pengukuran kinerja yang efektif harus:

- a) Menggambarkan tanggung jawab, memahami kinerja aktual relatif terhadap tanggung jawab, dan memberi penghargaan dan mengenali prestasi.
- b) Identifikasi peluang untuk mengambil tindakan untuk memperbaiki proses dan prosedur berdasarkan data.
- c) Rencanakan dan ramalkan dengan cara yang andal dan tepat waktu

Salah satu metodologi BPM yang banyak digunakan adalah *Balanced Scorecards (BSC)*.

5.5. *Balanced Scorecards*

Balanced Scorecards (BSC) merupakan suatu metodologi pengukuran performa yang membantu menerjemahkan target dan tujuan di bidang keuangan, pelanggan, proses internal, dan pembelajaran dan pertumbuhan, menjadi sebuah aksi/tindakan. Kata *balanced* pada BSC berarti seimbang, artinya BSC didesain untuk mengatasi keterbatasan sistem pengukuran yang hanya berfokus pada aspek finansial. Dengan adanya tambahan 3 non-finansial aspek pada BSC (*customer, internal business process, dan learning and growth*), pengukuran performa organisasi diharapkan menjadi lebih berimbang dan mewakili seluruh komponen dalam organisasi bisnis.



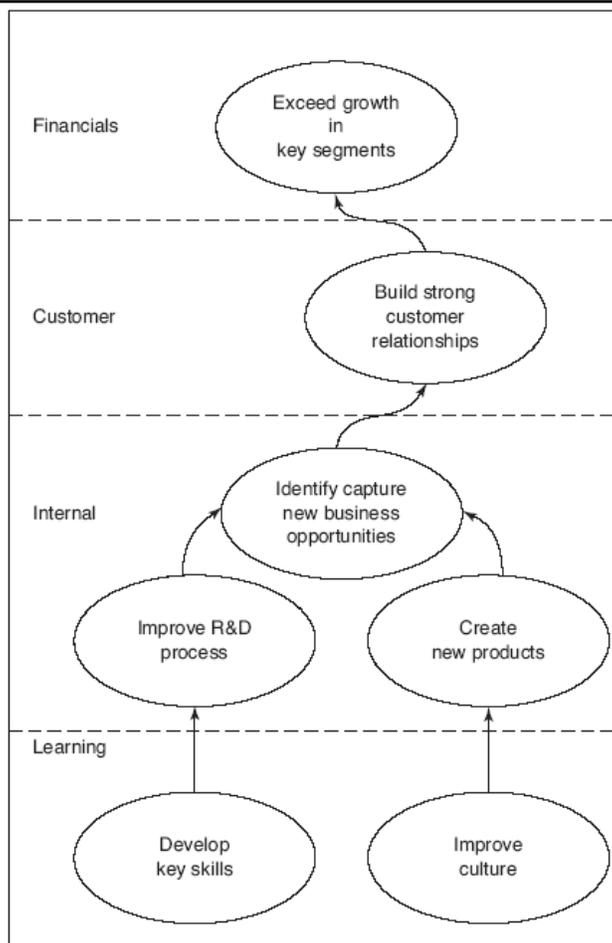
Gambar 5.2. Perspektif Balanced Scorecard

BSC bertujuan untuk menyelaraskan strategi dan tindakan. BSC memungkinkan sebuah organisasi untuk menyelaraskan tindakannya dengan keseluruhan strategi melalui serangkaian langkah yang saling terkait:

1. Identifikasi tujuan strategis untuk masing-masing perspektif
2. Mengaitkan langkah-langkah dengan masing-masing tujuan strategis; Campuran kuantitatif dan kualitatif harus digunakan.
3. Tetapkan target tindakan.
4. Buat daftar inisiatif strategis untuk mencapai setiap tujuan (yaitu, tanggung jawab).
5. Tautkan berbagai tujuan strategis melalui diagram sebab-akibat yang disebut peta strategi

Selanjutnya kita dapat menyusun sebuah *strategy map*, yaitu Tampilan visual yang menggambarkan hubungan antara tujuan organisasi utama untuk keempat perspektif BSC.

FIGURE 5.5 Sample Strategy Map



Gambar 5.3. Strategy map

5.6. Performance Dashboards

Dashboard merupakan komponen penting dalam BPM. *Dashboard* menyediakan informasi performa perusahaan dalam bentuk visual yang disusun dalam satu layar komputer, sehingga pengguna dapat melihat dan memahaminya dengan mudah. Pada *dashboard*, kita dapat melihat beberapa KPI yang digunakan sebagai acuan dalam pengukuran performa. Dengan demikian, kita dapat melihat apabila semua KPI dalam kondisi bagus (misalnya ditunjukkan dengan warna hijau), atau jika ada KPI yang belum mencapai target (ditampilkan dengan warna kuning atau merah).



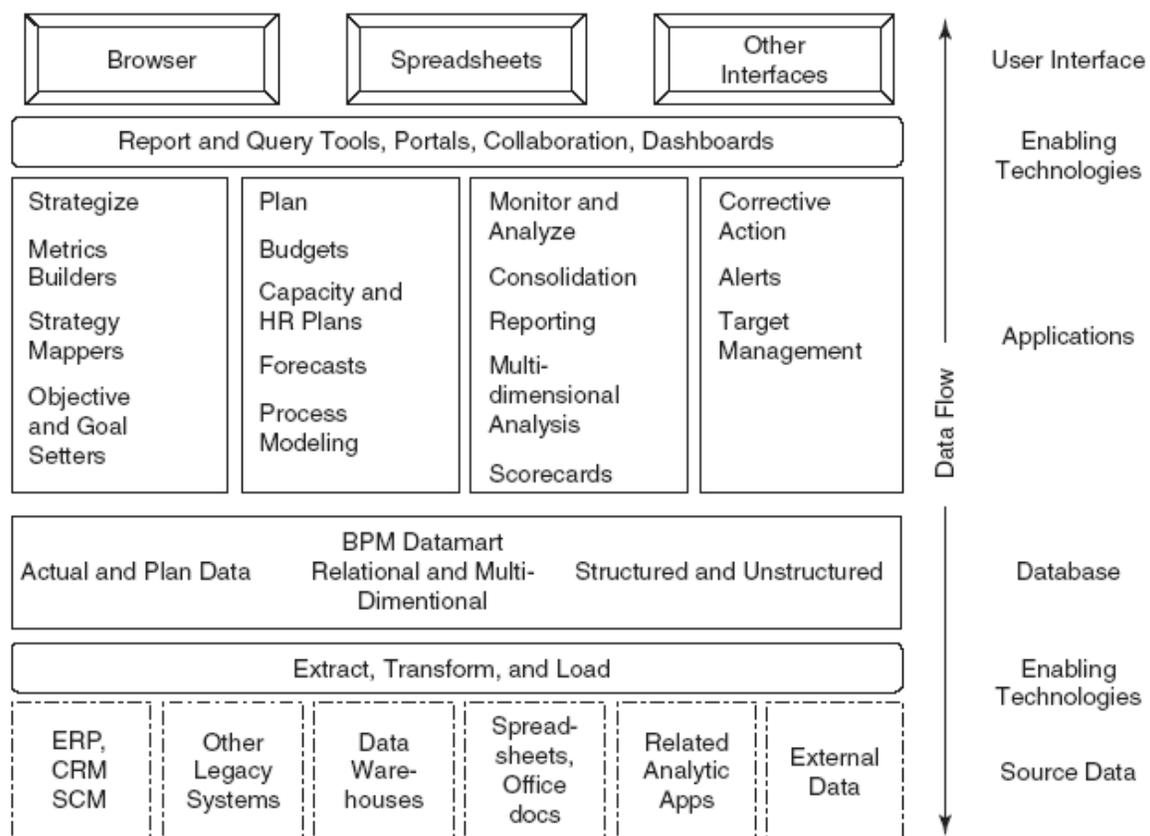
Gambar 5.4. *Performance Dashboard*

Tantangan mendasar dalam desain dashboard adalah menampilkan semua informasi yang dibutuhkan pada satu layar, dengan jelas dan tanpa gangguan, dengan tampilan yang dapat dipahami dengan cepat. Kita dapat menggunakan komponen visual (misalnya charts, performance bars, sparklines, gauges, meters, stoplights) untuk menyoroti data dengan pengecualian yang memerlukan tindakan. Dashboard harus transparan kepada pengguna, artinya mereka memerlukan tidak memerlukan pelatihan khusus dan sangat mudah digunakan. Dashboard dapat menggunakan data dari berbagai sumber ke dalam satu tampilan yang ringkas dan terpadu bagi pengguna bisnis. Pembahasan lebih mendalam mengenai dashboard ada pada bab berikutnya.

5.7. Arsitektur BPM

Agar dapat bekerja dengan baik, BPM memiliki arsitektur yang menggambarkan bagaimana komponen-komponen yang ada di dalamnya, seperti dapat dilihat pada gambar 5.5. Sebuah sistem BPM membutuhkan tiga komponen agar dapat berkontribusi dalam implementasi strategi perusahaan. Tiga komponen tersebut adalah:

- *Database tier.* Ini merupakan kerangka dari sistem BPM. Melalui komponen inilah BPM dapat bekerja. *Database tier* dapat terdiri dari *Transactional data stores*, *Application data marts*, dan *Centralized data warehouse*
- *Application tier.* Ini merupakan mesin utama dalam sistem BPM. Di sini proses BPM berjalan dengan mentransformasikan interaksi pengguna dan sumber data menjadi laporan atau hasil analisis.
- *User interface tier.* Antarmuka pengguna adalah jembatan antara aplikasi BPM dan pengguna akhir. *Web browser* saat ini merupakan alat utama untuk mengakses informasi dalam sistem BPM. *Spreadsheet* adalah alternatif yang populer bila diperlukan antarmuka pengguna untuk mendukung kebutuhan analitis dan perhitungan pengguna. Antarmuka BPM juga harus memberikan panduan kepada pengguna akhir.



Gambar 5.5. Arsitektur BPM

5.8. Rangkuman

- BPM mengacu pada proses, metodologi, metrik, dan teknologi yang digunakan oleh perusahaan untuk mengukur, memantau, dan mengelola kinerja bisnis.
- BPM adalah hasil dari BI, dan menggabungkan banyak teknologi, aplikasi, dan tekniknya.
- Praktik dan perangkat lunak BI hampir selalu merupakan bagian dari solusi BPM secara keseluruhan.
- BPM mencakup serangkaian proses loop tertutup yang menghubungkan strategi dengan eksekusi agar dapat mengoptimalkan kinerja bisnis.
- Proses kunci dalam BPM adalah menyusun strategi, merencanakan, memantau, bertindak, dan menyesuaikan.
- Sasaran strategis dan metrik utama organisasi harus menjadi penggerak top-down untuk alokasi aset berwujud dan tidak berwujud organisasi.
- BSC, dasbor kinerja, sistem pemantauan proyek, sistem sumber daya manusia, dan sistem pelaporan keuangan adalah contoh sistem kontrol diagnostik.
- Ukuran kinerja perlu diambil dari strategi perusahaan dan unit bisnis dan dari analisis proses bisnis utama yang diperlukan untuk mencapai strategi tersebut.
- Sistem manajemen kinerja yang paling terkenal dan paling banyak digunakan adalah BSC.

5.9. Latihan Soal

1. Ambil sebuah perusahaan sebagai studi kasus, carilah contoh BSC yang digunakan.
2. Apakah perbedaan CSF dengan KPI?
3. Bagaimana sebuah dashboard dapat dibuat dengan acuan dari BPM?

BAB VI

DASHBOARD SYSTEM

Kompetensi

Mahasiswa dapat merancang prototype BPM (minggu 13 dan 14)

Bahan Kajian

- Pengertian Dashboard
- Manfaat Dashboard
- Tipe-tipe Dashboard
- Pembuatan Dashboard

6.1. Pengertian *Dashboard*

Bagi kita yang masih belum familiar dengan dashboard, penting untuk memahami terlebih dahulu tentang dashboard. Setelah ini kita akan mempelajari pengertian dashboard, manfaatnya, tipe-tipe dashboard, serta format data yang digunakan pada dashboard. Secara umum, dashboard adalah sebuah user interface yang mudah dipahami, biasanya dalam satu halaman, yang menampilkan informasi visual terkait status atau performa organisasi atau Key Performance Indicators (KPI). Adanya dashboard ini akan membantu dalam pengambilan keputusan secara cepat dan tepat.

Dashboard mengambil nama dari automobile dashboard, yaitu panel pada mobil yang menunjukkan bagaimana performa dari kendaraan secara ringkas. Pada dashboard mobil, kita dapat melihat visualisasi informasi mengenai indikator-indikator performa kendaraan (kecepatan, putaran mesin, level bahan bakar, dan sebagainya). Indikator ini dapat dilihat dengan mudah sambil kita berkendara. Serupa dengan ini, sebuah business dashboard juga bisa digunakan untuk memonitor performa organisasi dengan mudah.

Dashboard dapat didefinisikan sebagai sebuah tampilan visual yang berisi informasi penting yang dibutuhkan untuk mencapai target/tujuan dalam satu layar komputer sehingga dapat dimonitor dengan mudah. Dashboard berisi metriks dan Key Performance Indicators (KPI) yang menyederhanakan kumpulan data kompleks untuk memfasilitasi pengguna dalam memahami kondisi/performa organisasi secara cepat. Pengguna dashboard bisa sangat beragam, tergantung departemen atau jabatan dalam organisasi, oleh karena itu pembuatan dashboard harus disesuaikan dengan kebutuhan penggunanya.

Penentuan parameter atau KPI yang tepat untuk ditampilkan pada dashboard sangat penting untuk menunjang kesuksesan. Maka, kita harus memahami untuk siapa suatu dashboard akan dibuat. Dashboard yang diperlukan oleh seorang manajer produksi tentunya berbeda dengan yang diperlukan manajer keuangan. Pada bab sebelumnya, kita sudah mempelajari penyusunan kerangka Balanced Scorecards untuk mengetahui KPI yang diperlukan pada suatu organisasi bisnis. Setelah mengetahui KPI apa saja yang tepat, barulah kita dapat menyusun sebuah dashboard yang sesuai untuk organisasi tersebut.

6.2. Manfaat *Dashboard*

Dashboard dapat membantu manajer dalam memantau perkembangan performa dari setiap departemen dalam organisasi. Sebagai gambaran umum performa perusahaan, dashboard dapat menampilkan perbandingan performa saat ini dengan periode sebelumnya. Adapun manfaat lain dari dashboard adalah sebagai berikut:

- Presentasi visual dari pengukuran performa
- Dapat mengidentifikasi tren negatif dan melakukan koreksi
- Mengukur efisiensi/inefisiensi
- Pendukung pengambilan keputusan berbasis informasi
- Keselarasan dengan strategi dan tujuan organisasi
- Cepat mengidentifikasi kejanggalan data
- Menghemat waktu dibandingkan dengan mengamati berbagai laporan

6.3. Tipe-tipe *Dashboard*

Dashboard dapat dikelompokkan berdasarkan fungsinya sebagai berikut:

- *Strategic Dashboard*
- *Analytical Dashboard*
- *Operational Dashboard*
- *Informational Dashboard*

Strategic Dashboard. Dashboard strategis mendukung para manajer pada tingkat manapun dalam sebuah organisasi untuk pengambilan keputusan. Mereka menyediakan snapshot data, menampilkan kesehatan dan peluang bisnis, dengan fokus pada ukuran kinerja dan prakiraan tingkat tinggi. Dasbor strategis memerlukan data snapshot berkala dan statis (misalnya harian, mingguan, bulanan, kuartalan dan tahunan). Mereka tidak perlu terus berubah dari satu momen ke momen berikutnya dan memerlukan update pada interval waktu yang ditentukan. Mereka hanya menggambarkan data tingkat tinggi yang tidak harus memberikan rinciannya. Mereka dapat interaktif untuk memfasilitasi perbandingan dan pandangan yang berbeda dalam hal set data yang

besar dengan sekali klik sebuah tombol. Tapi, tidak perlu menyediakan fitur interaktif di dasbor ini.

Analytical dashboard mencakup lebih banyak konteks, perbandingan, dan sejarah. Mereka fokus pada berbagai aspek data yang dibutuhkan untuk analisis. Dasbor analitis biasanya mendukung interaksi dengan data, seperti mengebor ke detail yang mendasarinya dan karenanya harus interaktif. Contoh dasbor analitis meliputi dasbor Manajemen Keuangan dan dasbor Manajemen Penjualan.

Operational dashboard adalah untuk memantau operasi secara konstan. Mereka sering dirancang secara berbeda dari dasbor strategis atau analitik dan fokus pada pemantauan aktivitas dan kejadian yang terus berubah dan mungkin memerlukan perhatian dan tanggapan pada saat itu juga. Dengan demikian, dasbor operasional memerlukan data live dan up to date yang tersedia setiap saat dan karenanya harus dinamis. Contoh dasbor operasi bisa berupa dasbor sistem pendukung, menampilkan data langsung pada tiket layanan yang memerlukan tindakan segera dari pengawas pada tiket dengan prioritas tinggi.

Informational dashboard hanya untuk menampilkan figur, fakta dan / atau statistik. Mereka bisa statis atau dinamis dengan data live tapi tidak interaktif. Misalnya, dashboard informasi kedatangan / keberangkatan di bandara.

6.4. Pembuatan *Dashboard* pada Excel

Kita dapat membuat dashboard di Excel menggunakan berbagai fitur yang membantu kita dalam membuat visualisasi data, yang merupakan karakteristik utama dari dashboard. Kita dapat menampilkan data dalam tabel dengan format kondisional untuk menyoroti hasil baik dan buruk, kita dapat meringkas data dalam grafik dan PivotTables, kita dapat menambahkan kontrol interaktif, dan kita dapat menentukan dan mengelola KPI.

Excel Tables

Komponen terpenting dashboard adalah datanya. Data bisa dari sumber tunggal atau banyak sumber. Tabel Excel sangat sesuai untuk mendapatkan data ke dalam workbook, di mana Anda ingin membuat dashboard. Ada beberapa cara untuk mengimpor data ke dalam Excel, dengan membuat koneksi ke berbagai sumber. Hal ini memungkinkan untuk me-refresh data dalam workbook kita setiap kali sumber data diperbarui.

Anda dapat memberi nama tabel Excel dan menggunakan nama tersebut untuk merujuk data Anda di dashboard. Ini akan lebih mudah daripada mereferensi pada range data dengan referensi sel. Tabel Excel ini menjadi tabel kerja Anda yang berisi data mentah.

Menampilkan Tren dengan Sparkline pada tabel

Sparklines adalah grafik kecil yang ditempatkan pada sel tunggal, masing-masing mewakili deretan data sesuai pilihan kita. Sparklines membantu dalam melihat tren secara cepat. Di Excel, kita bisa memilih Line Sparklines, Column Sparklines atau Win / Loss Sparklines. Kita bisa menambahkan Sparklines ke dalam tabel dengan cepat dengan Quick Analysis tool.

- Identifikasi data yang ingin Anda tambahkan Sparklines.
- Siapkan kolom kosong ke sisi kanan data dan beri nama kolomnya. Sparklines akan ditempatkan di kolom ini.
- Pilih data. Tombol Quick Analysis akan muncul di sudut kanan bawah data yang sudah dipilih.

	A	B	C	D	E	F
1	% Profits Region-wise and Quarter-wise					
2		Qtr1	Qtr2	Qtr3	Qtr4	Trends
3	East	87%	90%	79%	96%	
4	North	92%	94%	85%	97%	
5	South	88%	95%	75%	80%	
6	West	85%	87%	87%	88%	
7						
8						
9						
10						
11						
12						

Quick Analysis

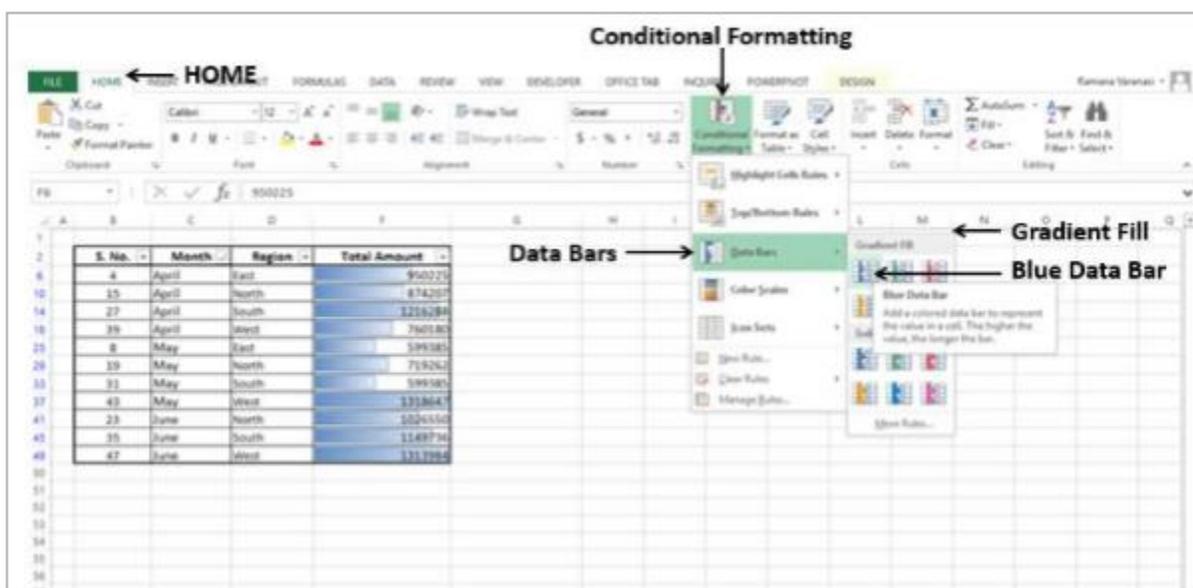
Gambar 6.1. Pembuatan *sparkline*

- Klik pada tombol Quick Analysis. Tool Quick Analysis muncul.
- Klik pada SPARKLINES. Pilihan grafik muncul.
- Klik pada Line. Diagram baris akan ditampilkan untuk setiap baris dalam data yang dipilih.
- Klik Kolom. Diagram kolom akan ditampilkan untuk setiap baris dalam data yang dipilih.
- Grafik Win / Loss tidak sesuai untuk data ini. Perhatikan data berikut untuk memahami bagaimana grafik Win / Loss terlihat.

Data Bars

Anda dapat menggunakan *Data Bars* berwarna untuk melihat suatu nilai relatif terhadap nilai lainnya. Panjang *Data Bar* mewakili besarnya suatu nilai. *Bar* yang lebih panjang mewakili nilai yang lebih tinggi, dan *Bar* yang lebih pendek mewakili nilai yang lebih rendah. Anda bisa menggunakan warna solid atau warna gradien untuk *Data Bars*.

- Pilih kolom - Total Amount.
- Klik *Conditional Formatting* di grup *Styles* di bawah tab *Home*.
- Klik pada *Data Bars* dalam daftar *drop-down*.
- Klik *Blue Data Bar* di bawah *Gradient Fill* pada *dropdown* kedua yang muncul.



Gambar 6.2. Pembuatan *data bars*

Memilih grafik yang sesuai

Untuk menampilkan data dengan bagan di *dashboard* Anda, pertama tentukan tujuan bagan. Begitu Anda memiliki kejelasan tentang apa yang ingin Anda wakili oleh bagan, Anda dapat memilih jenis grafik terbaik yang menggambarkan pesan Anda.

Berikut adalah beberapa saran untuk memilih jenis bagan:

- Jika ingin membandingkan nilai data, Anda bisa memilih chart bar, pie chart, line chart, atau scatter chart.
- Jika ingin menunjukkan distribusi, Anda bisa melakukannya dengan grafik kolom, scatter chart atau line chart.

- Jika Anda ingin menunjukkan tren dari waktu ke waktu, Anda dapat menggunakan diagram garis.
- Jika Anda ingin mewakili bagian dari keseluruhan, diagram lingkaran bisa menjadi pilihan. Tapi, saat Anda menggunakan diagram lingkaran, ingat bahwa hanya dua sampai tiga titik data berbeda dengan nilai data yang sangat berbeda dapat digambarkan secara efektif dengan berbagai ukuran irisan Pie. Jika Anda mencoba untuk menggambarkan lebih banyak jumlah titik data dalam diagram Pie, akan sulit untuk mendapatkan perbandingannya.
- Anda dapat menggunakan tabel Scatter jika salah satu dari berikut ini adalah tujuan - Anda ingin menunjukkan kesamaan antara kumpulan data yang besar, bukan perbedaan antara titik data.
Anda ingin membandingkan banyak titik data tanpa memperhatikan waktu. Semakin banyak data yang Anda sertakan dalam grafik Scatter, semakin baik perbandingan yang bisa Anda buat.
- Bagan yang disarankan di Excel membantu Anda menemukan jenis bagan yang sesuai dengan data Anda.
Di Excel, Anda dapat membuat bagan dengan jenis bagan dan mengubahnya kapan saja dengan mudah.

6.5. Rangkuman

- Data yang dibutuhkan untuk *dashboard* bergantung pada kategorinya.
- Premis untuk data adalah bahwa hal itu harus relevan, bebas dari kesalahan, *up to date* dan tersedia jika diperlukan. Data mungkin berasal dari berbagai sumber dan format yang berbeda (*Spreadsheets, text files, web pages, organizational database*, dan lain-lain).
- Hasil yang ditampilkan di dasbor harus asli, benar dan tepat. Hal ini penting karena informasi di dasbor akan mengarah pada keputusan, tindakan dan / atau kesimpulan. Dengan demikian, bersamaan dengan data yang ditampilkan, medium yang dipilih untuk tampilan sama pentingnya karena seharusnya tidak memberikan kesan yang keliru dalam penggambaran data. Fokusnya harus pada kemampuan visualisasi data yang secara jelas akan memproyeksikan kesimpulan.

- *Data warehouse* dan OLAP memungkinkan penyegaran *dashboard* dinamis dengan data secara langsung. Hal ini juga membuat perancangan *dashboard* tidak bergantung pada departemen TI organisasi untuk mendapatkan data.
- *Dashboard* telah menjadi media yang paling banyak dicari dari manajemen *top-level* hingga pengguna biasa.

6.6. Latihan Soal

1. Mengapa *dashbord* sudah menjadi alat bantu manajemen yang populer saat ini?
2. Carilah contoh-contoh dari setiap tipe *dashboard*
3. Buatlah sebuah *dashboard system* menggunakan data dari *data warehouse* & OLAP yang sudah Anda miliki.

DAFTAR PUSTAKA

- Adamson, C. 2009. *The Star Schema Handbook: The Complete Reference to Dimensional Data Warehouse Design*. Hoboken, NJ: Wiley.
- CRISP-DM. 2009. "Cross-Industry Standard Process for Data Mining (CRISP-DM)." crisp-dm.eu diakses pada tanggal 30 Juli 2017
- Eckerson, W. 2009. "Performance Management Strategies: How to Create and Deploy Effective Metrics." TDWI Best Practices Report.
- Fouche, Guy. 2012. "Foundations of SQL Server 2008 R2 Business Intelligence". Apress.
- Hossain, Muhammad Muazzem. 2016. "Towards developing a business performance management model using causal latent semantic analysis". International Journal of Business Performance Management (IJBPM), Vol. 17, No. 2, 2016.
- Marr, Bernard. 2016. *25 Key Performance Indicators yang harus diketahui*. Jakarta: PT Elex Media Komputindo
- Sauro, Jeff. 2015. *Customer Analytics for Dummies*. New Jersey: John Wiley & Sons, Inc.
- Turban, Efraim. 2017. *Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition*. New Jersey: Pearson Education.
- Wirama, Kasim. 2012. *The Essential Business Intelligence in SQL Server 2008*. Indonesia: .NET Developer Community.
- Witten, Ian H. 2016. *Data Mining: Practical machine learning tools and techniques. 4th Edition* Massachusetts: Morgan Kaufmann.

INDEKS

A

akurasi, 121
alert, 8
Analytics, 51
asosiasi, 101
Automated Decision Support (ADS), 62
Automation Decision System, 13

B

Balanced Scorecards (BSC), 130
BI Competency Center (BICC), 14
Business Analytics (BA), 51
Business Intelligence, 5
Business Intelligence Development Studio (BIDS), 33
Business Performance Management (BPM), 126

C

clustering, 101
confusion matrix, 122
cost-benefit analysis, 15
CRISP-DM, 105
Critical Success Factors (CSF), 128
Cube, 57
Customer analytics, 61
Customer Relationship Management (CRM), 103

D

dashboard, 137
Data Flow, 31
Data kategoris, 99
Data mart, 21
data mining, 97
data numerik, 99

Data Profiling, 31
Data source, 20
Data Source View (DSV), 38
Data Warehouse, 19
Data warehouse administrator, 28
database administrator, 28
Decision Tree, 111
Dice, 53
Dimension, 56
drill down, 53

E

Enterprise Data Warehouse, 21
ETL, 24
Executive Information System, 7
Extraction, Transformation, and Load, 25

F

fact table, 58
Flat File, 31

G

Geographical information system (GIS), 60
Globalisasi, 2

H

hierarchies, 58

I

Integrasi data, 24

K

Kecerdasan Bisnis, 5
Key Performance Indicators (KPI), 128
klasifikasi, 102

L

lingkungan bisnis, 2, 3

Lookup, 32

M

Measures, 56

Metadata, 20

Model Inmon, 27

Model Kimball, 27

multidimensi, 29

Multidimensionality, 55

N

Naïve Bayes, 115

nonvolatile, 19

O

Online analytical processing (OLAP), 52

Online Transactional Processing (OLTP)., 53

Operational data stores (ODS), 21

P

Pivot, 54

prediksi, 101

R

Raw File, 31

S

sistem pendukung keputusan, 5

Skema, 28

Slice, 53

snowflakes schema, 28

spreadsheet, 6

SSIS, 30

star schema, 28

T

Transaction Processing Systems, 13

W

web server, 23

Web-services, 25

WEKA, 118

Workflow, 8

X

XML, 31

BUKU AJAR

Kecerdasan Bisnis

Kecerdasan bisnis (*Business Intelligence*) adalah suatu ilmu yang mendukung pengambilan keputusan pada organisasi bisnis. Tujuan utamanya adalah menyediakan akses interaktif terhadap data, memungkinkan manipulasi data, dan memberi manajer kemampuan untuk melakukan analisis bisnis. Dengan kemampuan untuk melakukan analisis terhadap data, performa, dan situasi lingkungan bisnis, maka manajer dapat melakukan pengambilan keputusan yang lebih baik.

Dengan buku ajar kecerdasan bisnis ini, pembaca khususnya mahasiswa diharapkan dapat mengetahui dan memahami kecerdasan bisnis dengan lebih terkonsep. Buku ini membahas mengenai pengantar kecerdasan bisnis, *data warehousing*, *business analytics* dan *OLAP*, *data mining*, *business performance management*, serta *dashboard system*. Selain berisi tentang konsep dasar, di dalam buku ajar ini juga terdapat panduan praktik menggunakan beberapa aplikasi / *tools* serta latihan-latihan untuk memperkuat pemahaman konsep.



Valentinus Roby Hananto, S.Kom., M.Sc., OCA. adalah dosen tetap program studi Sistem Informasi di Institut Bisnis dan Informatika Stikom Surabaya. Penulis merupakan lulusan S1 Sistem Informasi di Stikom Surabaya dan S2 Engineering Management di University of South Florida, USA. Beberapa mata kuliah yang sudah pernah diampu yaitu Kecerdasan Bisnis, Konsep Sistem Informasi, Sistem Pendukung Keputusan, dan Sistem Informasi Manajemen.



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Menteri Hukum dan Hak Asasi Manusia Republik Indonesia, berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta yaitu Undang-Undang tentang perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra (tidak melindungi hak kekayaan intelektual lainnya), dengan ini menerangkan bahwa hal-hal tersebut di bawah ini telah tercatat dalam Daftar Umum Ciptaan:

- I. Nomor dan tanggal permohonan : EC00201705414, 10 November 2017
- II. Pencipta
Nama : **Valentinus Roby Hananto**
Alamat : Wisma Sarinadi Blok K-07 RT29 RW6 Magersari Sidoarjo ,
Sidoarjo, JAWA TIMUR, 61211
Kewarganegaraan : Indonesia
- III. Pemegang Hak Cipta
Nama : **Institut Bisnis dan Informatika Stikom Surabaya**
Alamat : Jalan Raya Kedung Baruk 98, Surabaya, Jawa Timur,
Indonesia, Surabaya, JAWA TIMUR, 60298
Kewarganegaraan : Indonesia
- IV. Jenis Ciptaan : Buku
- V. Judul Ciptaan : **Buku Ajar "Kecerdasan Bisnis"**
- VI. Tanggal dan tempat diumumkan : 1 September 2017, di Surabaya
untuk pertama kali di wilayah
Indonesia atau di luar wilayah
Indonesia
- VII. Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut
pertama kali dilakukan Pengumuman.
- VIII. Nomor pencatatan : 04900

Pencatatan Ciptaan atau produk Hak Terkait dalam Daftar Umum Ciptaan bukan merupakan pengesahan atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang dicatat. Menteri tidak bertanggung jawab atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang terdaftar. (Pasal 72 dan Penjelasan Pasal 72 Undang-undang Nomor 28 Tahun 2014 Tentang Hak Cipta)

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
REPUBLIK INDONESIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b.
DIREKTUR HAK CIPTA DAN DESAIN INDUSTRI

Dr. Dra. Erni Widhyastari, Apt., M.Si.
NIP. 196003181991032001