

PERANCANGAN SISTEM METADATA UNTUK DATA WAREHOUSE DENGAN STUDI KASUS REVENUE TRACKING PADA PT. TELKOM DIVRE V JAWA TIMUR

Yudhi Purwananto, Rully Soelaiman, Aby Herwendo

Jurusan Teknik Informatika,

Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804

Email: yudhi@its-sby.edu, rully@its-sby.edu

ABSTRAK

Data warehouse merupakan media penyimpanan data dalam perusahaan yang diambil dari berbagai sistem dan dapat digunakan untuk berbagai keperluan seperti analisis dan pelaporan. Di PT Telkom Divre V Jawa Timur telah dibangun sebuah data warehouse yang disebut dengan Regional Database. Di Regional Database memerlukan sebuah komponen penting dalam data warehouse yaitu metadata. Definisi metadata secara sederhana adalah “data tentang data”. Dalam penelitian ini dirancang sistem metadata dengan studi kasus Revenue Tracking sebagai komponen analisis dan pelaporan pada Regional Database.

Metadata sangat perlu digunakan dalam pengelolaan dan memberikan informasi tentang data warehouse. Proses – proses di dalam data warehouse serta komponen – komponen yang berkaitan dengan data warehouse harus saling terintegrasi untuk mewujudkan karakteristik data warehouse yang subject-oriented, integrated, time-variant, dan non-volatile. Karena itu metadata juga harus memiliki kemampuan mempertukarkan informasi (exchange) antar komponen dalam data warehouse tersebut. Web service digunakan sebagai mekanisme pertukaran ini. Web service menggunakan teknologi XML dan protokol HTTP dalam berkomunikasi. Dengan web service, setiap komponen yang ada dalam data warehouse dapat bertukar informasi melalui metadata.

Implementasi web service dalam sistem metadata menggunakan Framework Microsoft .NET. Dalam tugas akhir ini dilakukan uji coba terhadap sistem metadata yang dibuat dengan membuat aplikasi yang memanfaatkan metadata. Dengan sistem metadata ini dapat dibentuk aplikasi sistem pendukung keputusan yang lain seperti aplikasi OLAP dan sistem pengelolaan metadata.

Kata kunci : *data warehouse, metadata, exchange, web service, xml*

1. PENDAHULUAN

Sistem informasi telah menjadi salah satu faktor yang menentukan dalam kesuksesan pengelolaan bisnis modern secara kompetitif. Konsep dalam sistem informasi seperti *Decision Support System (DSS)*, *Management Information System (MIS)*, *Executive Information System (EIS)*, dan sistem atau aplikasi lainnya diharapkan mampu memberikan analisis - analisis bisnis yang diperlukan. Data warehouse merupakan sarana yang dapat mendukung dalam memberikan ketersediaan data untuk keperluan analisis.

Di PT. Telkom Divre V Jawa Timur telah dibangun data warehouse yang disebut dengan Sistem Regional Database (RDB) Divre V Jawa Timur, yang menyediakan data dan informasi media pengolahan dan penyimpanan data, serta tools untuk akses, pengolahan, dan analisis data. Dari segi arsitektur, sistem ini dirancang untuk dapat menyimpan data dari berbagai sistem database, seperti Oracle, Cobol, *plain text*, dan sebagainya dengan sistem operasi Unix, VMS, dan lain - lain. Sedangkan pemanfaatan data warehouse sebagai sumber informasi, sistem ini

dilengkapi dengan sistem *on-line analytical processing (OLAP)* dan sistem *analyzer*. Arsitektur ini juga memungkinkan untuk diintegrasikan dengan sistem eksternal yang mungkin dikelola oleh unit / divisi lain yang membutuhkan akses database. Dalam prakteknya, data warehouse ini tumbuh dan berkembang membesar secara cepat. Begitu pula dengan *user*-nya yang semakin beragam dengan keperluannya masing - masing terhadap data, dari *developer*, *designer*, analis, administrator, hingga manajer. Guna memudahkan pengelolaan data dan akses data yang beragam di data warehouse ini dibutuhkan informasi tentang data itu sendiri.

Metadata dapat disebut sebagai data tentang data. Dalam implementasinya, metadata mendefinisikan struktur dan arti data yang digunakan untuk aplikasi, database, tools, dan sistem informasi lainnya. Metadata sangat penting karena representasi terhadap obyek database yang berbeda - beda bagi tiap user, selain itu digunakan juga untuk melakukan pengelolaan terhadap obyek - obyek dalam database atau komponen - komponen pendukung dalam data warehouse.

Pemahaman tentang metadata yang sangat kompleks mengakibatkan metadata kurang begitu populer di dunia teknologi informasi. Kompleksitas metadata ini mengakibatkan kurangnya keberhasilan implementasi metadata pada suatu perusahaan. Implementasi metadata dalam data warehouse di PT. Telkom Divre V Jawa Timur ini sendiri hanya sebatas implementasi sederhana untuk membantu operasional dan menjelaskan kondisi data warehouse. Metadata yang diimplementasikan belum terintegrasi sepenuhnya dengan lingkungan data warehouse, masih bersifat pasif dan belum dapat dimanfaatkan sepenuhnya untuk mendukung data warehouse tersebut. Oleh karena itu diperlukan sebuah sistem metadata yang terintegrasi dengan data warehouse dan diharapkan dapat membantu dalam perancangan dan pengembangan lebih lanjut sistem informasi dalam data warehouse ini.

2. DATA WAREHOUSE DAN METADATA

2.1. DATA WAREHOUSE

Data warehouse adalah database berupa kumpulan data dari berbagai sistem dan ditujukan untuk mendukung pelaporan dan pengambilan keputusan pada manajemen [2]. Sebuah data warehouse dalam persepsi sederhana merupakan kumpulan dari potongan – potongan kunci informasi yang digunakan untuk mengelola dan melangsungkan kegiatan bisnis untuk memperoleh hasil yang paling menguntungkan. Definisi lain mengenai data warehouse terfokus pada data. Lebih dari itu, data warehouse juga merupakan proses yang melibatkan cara untuk mendapatkan data dari asal data ke dalam data warehouse, dan mengolahnya untuk suatu analisis. Dengan kata lain, data warehouse adalah data (*meta / fact / dimension / aggregation*) dan pengelolaan proses (*load / warehouse / query*) yang memungkinkan informasi tersedia, sehingga memudahkan orang untuk mengambil keputusan [1]. Data warehouse juga merupakan salah satu Sistem Pendukung Keputusan, yang menyimpan data dari berbagai sumber, mengorganisasikan data untuk *reporting* (pelaporan) dan analisis dalam bentuk *Online Analytical Processing* (OLAP).

Dalam data warehouse, database berisi *summary* (ringkasan) dan *aggregate* (kumpulan) data dalam hubungan *facts* (fakta, misalnya transaksi penjualan) dan *dimensions* (dimensi, misalnya produk, wilayah, waktu). *Fact* dan *dimension* biasanya dikombinasikan dalam sebuah desain *star schema*. Kombinasi *fact* dan *dimension* dalam bentuk *star schema* inilah yang menjadi *data mart*.

Data yang masuk ke dalam data warehouse datang dari lingkungan operasional hampir dalam segala hal. Data warehouse merupakan penyimpanan fisik data yang ditransformasikan dari data aplikasi yang berada dalam lingkungan operasional. Bill Inmon menyatakan karakteristik data warehouse

antara lain: *subject oriented, integrated, time variant*, dan *non volatile* [4].

2.2. METADATA

Metadata secara sederhana didefinisikan dalam bahasa Inggris sebagai *data about data*, atau data tentang data [5]. Kesuksesan operasi dan penggunaan data warehouse tergantung pada efektifitas pengelolaan metadata [3]. Metadata ini dibutuhkan untuk mendeskripsikan seluruh aspek relevan pada data warehouse.

Contoh sederhana metadata dalam ruang lingkup database adalah nama, lokasi database, nama *table*, nama dan tipe field, dan lain sebagainya. Dalam ruang lingkup data warehouse khususnya, metadata dikaitkan dengan lingkungan sekitar data warehouse, misalnya OLAP (*Schema, Dimension, Cube, Hierarchy*), proses ETL, dan lain sebagainya. Bill Inmon memaparkan jenis / variasi metadata antara lain sebagai berikut [6] :

- *Direct descriptive metadata*
- *Indirect descriptive metadata*
- *Direct relationship metadata*
- *Indirect relationship metadata*
- *Business and technical metadata*
- *Static metadata and time variant metadata*

2.2.1. Arsitektur Metadata

Lokasi dimana metadata disimpan disebut dengan *metadata repository*. Ada beberapa arsitektur untuk mengakses dan mendapatkan data pada *metadata repository* ini, antara lain [3]:

- Centralized approach
- Decentralized approach
- Shared approach
- Mixed approach

Perkembangan teknologi dalam metadata repository dapat dibagi menjadi dua bagian, yaitu teknologi yang lama dan teknologi yang baru. Beberapa ciri teknologi *repository* yang lama antara lain [10]:

- *data dictionary* yang pasif
- berdiri sendiri (*standalone*)
- seluruh produk repository dari suatu vendor yang tersentral dan menawarkan bermacam model dan pilihan antarmuka, dan secara keseluruhan dibangun oleh vendor itu sendiri.

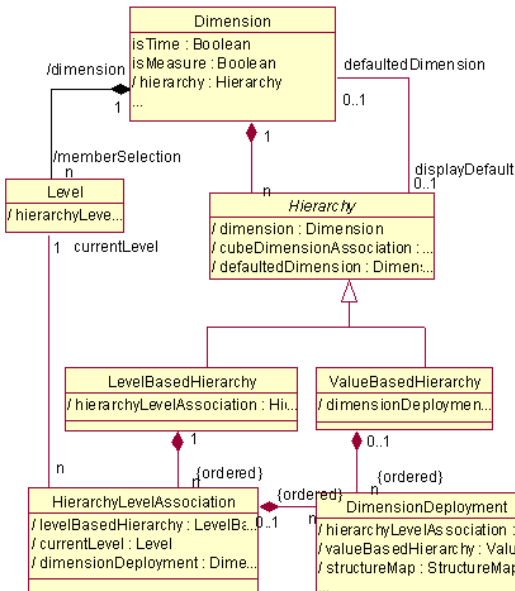
Generasi baru dalam teknologi *repository* dilengkapi dengan:

- penggunaan kembali (reuse) komponen – komponen dalam aplikasi
- standardisasi dan exchange metadata
- repository yang terintegrasi.

2.2.2. Model Driven Architecture (MDA)

Model Driven Architecture merupakan pendekatan terhadap pengintegrasian sistem

perusahaan yang terdiri dari *software, hardware, user,* dan praktek bisnis. Spesifikasi MDA oleh OMG yang juga didukung beberapa standard, antara lain: *Unified Modelling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI), dan Common Warehouse Metamodel (CWM) [7].*



Gambar 1. Representasi Model Sistem OLAP dalam CWM

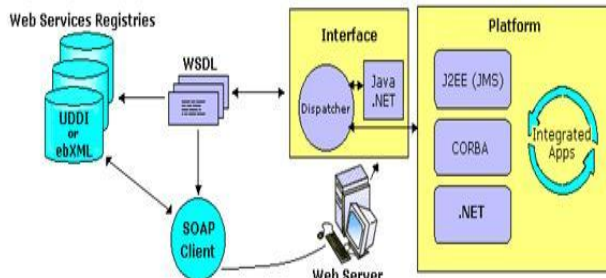
2.3. COMMON WAREHOUSE METAMODEL (CWM)

Common Warehouse Metamodel (CWM) mendefinisikan sebuah pemodelan yang merepresentasikan baik *business* maupun *technical metadata* yang banyak ditemui dalam data warehouse maupun area analisis bisnis [7]. Sistem yang berdasarkan spesifikasi ini saling bertukar metadata dalam format yang konsisten. CWM dinyatakan dalam bentuk notasi UML. CWM terdiri dari beberapa unsur pokok yang merepresentasikan *data resources, analisis, pengelolaan warehouse,* dan komponen dasar pada data warehouse di lingkungan *business intelligence.* Gambar 1 menunjukkan representasi model sistem OLAP dalam CWM yang diimplementasikan dalam sistem metadata ini.

3. WEB SERVICE DAN FRAMEWORK MICROSOFT .NET

Web service menggunakan teknologi dan protokol dalam web, yaitu protokol HTTP dan format data XML untuk membentuk komponen yang dapat diakses oleh aplikasi lain.

Untuk melakukan akses terhadap web service hanya memerlukan protokol *Hypertext Transfer Protocol (HTTP)* dan format data *Extensible Markup Language (XML).* Web service dapat diimplementasikan pada platform dan bahasa pemrograman yang berbeda dan tidak terhalang *firewall* karena menggunakan port 80 (port HTTP).



Gambar 2. Arsitektur Web Service

3.1. EXTENSIBLE MARKUP LANGUAGE (XML)

Web service sangat tergantung pada kemampuan aplikasi tersebut dalam menghasilkan, mengirimkan ke tujuan, dan menginterpretasikan file XML. XML merupakan gabungan antara *Standard Generalized Markup Language (SGML)* dan *Hypertext Markup Language (HTML).* XML memiliki *syntax* seperti HTML, namun *syntax* dalam XML harus “*well-formed*”.

3.2. WEB SERVICES DESCRIPTION LANGUAGE (WSDL)

Web Services Description Language (WSDL) adalah spesifikasi untuk mendeskripsikan web service. Dengan WSDL ini, web service dapat mendeskripsikan dirinya ke client yang akan mengakses web service.

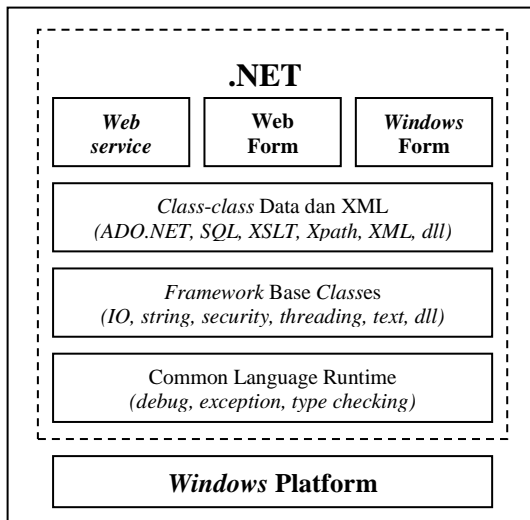
3.3. SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

SOAP adalah protokol untuk pertukaran dokumen XML dalam sebuah pesan. SOAP merupakan protokol kelas ringan untuk pertukaran informasi dalam sebuah lingkungan terdistribusi.

3.3.1. Framework Microsoft .NET

Framework Microsoft .NET atau biasa disebut dengan *Framework .NET* saja adalah sebuah *platform computing* baru yang dirancang untuk menyederhanakan pengembangan aplikasi pada lingkungan terdistribusi pada internet [11]. Bahasa yang didukung oleh *Framework .NET* adalah Visual Basic.NET, C++, ASP.NET, Jscript.NET dan C#.

Pada gambar 3, terlihat *Framework .NET* berada diatas sistem operasi, yang berupa sistem operasi buatan Microsoft, yaitu *Windows 2000 (Server dan profesional), Windows NT 4.0 (Server dan workstation), Windows Millenium Edition, Windows 98 dan Windows XP Profesional.*



Gambar 3. Framework .NET

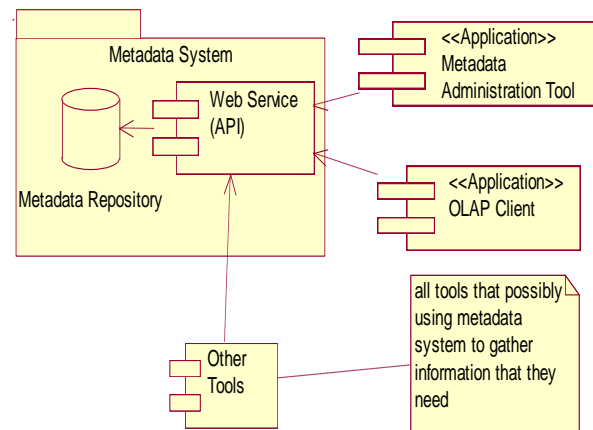
4. PERANCANGAN DAN IMPLEMENTASI

4.1. ARSITEKTUR SISTEM METADATA

Sistem metadata merupakan bagian dari arsitektur sistem n-tier yang terdiri dari 2-tier, yaitu *business-logic tier* dan *data access tier*. Sedangkan *presentation tier*-nya adalah sistem lain yang menggunakan fasilitas yang ada dalam sistem metadata. Arsitektur sistem metadata ini ditunjukkan oleh diagram pada gambar 4.

Data access tier pada arsitektur ini direpresentasikan oleh *metadata repository* yang menjadi media penyimpanan data untuk metadata tersebut. Sistem yang berada di *presentation tier* dapat berupa aplikasi untuk mengelola metadata, OLAP Client, ataupun aplikasi lain yang memanfaatkan sistem metadata ini untuk berbagai keperluan, misalnya *reporting*, *monitoring*, atau proses warehouse. Seluruh sistem dalam *presentation tier* harus memiliki pengetahuan tentang sistem metadata ini sehingga dapat berkomunikasi dengan sistem metadata. Oleh karena itu komunikasi dengan sistem metadata ini diatur oleh *business-logic tier*.

Dalam berkomunikasi dengan sistem metadata, *presentation tier* cukup memanggil fungsi – fungsi yang terdapat dalam *business-logic tier*. Pada arsitektur ini *business-logic tier* direpresentasikan oleh web service sebagai *Application Programming Interface (API)*. Web service ini menyediakan API bagi *presentation tier*. Selain itu web service ini juga mendokumentasikan seluruh fungsional dan arsitektur yang terdapat di dalam sistem metadata, misalnya: fungsi – fungsi yang ada dengan parameter serta *return value*-nya, dokumentasi obyek dalam metadata, beserta cara penggunaannya.

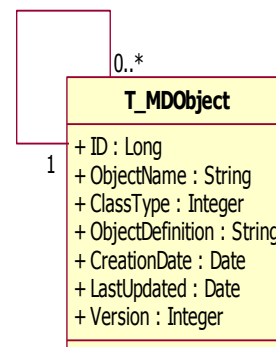


Gambar 4. Arsitektur Sistem Metadata

4.2. PEMODELAN DATA / METADATA REPOSITORY

Dalam arsitektur sistem metadata ini akan digunakan *centralized approach*, maksudnya adalah repository yang terpusat untuk keseluruhan metadata. Penggunaan *centralized approach* ini dimaksudkan agar mempermudah dalam pengelolaan metadata.

Setiap *object* di dalam model CWM belum tentu direpresentasikan dalam sebuah *table* di dalam database, atau setiap relasi/ asosiasi antar *object* merupakan relasi dalam database. Dalam desain model database, terdapat *table* dengan *field – field* yang ada di dalam *table* dapat merepresentasikan *object – object* dalam model logis CWM. Gambar 5 merupakan diagram konseptual model data (*conceptual data model*) yang digunakan dalam sistem ini.



Gambar 5. Conceptual Data Model : Table T_MDObject

Table utama dalam sistem ini adalah table T_MDObject yang menyimpan keseluruhan *object* dalam sistem metadata. Keterangan untuk setiap field – field dijelaskan pada Tabel 1.

Tabel 1 Field - field pada table T_MDObject

Nama Field	Type	Keterangan
ID	Long	Identitas <i>Object</i> Metadata yang <i>unique</i> .
ObjectName	String	Nama <i>Object</i> Metadata. Nama tidaklah harus <i>unique</i> .
ClassType	Integer	Tipe <i>Object</i> / Class Metadata. Tipe <i>object</i> ini harus merferensi kepada <i>object</i> metadata yang lain.
ObjectDefinition	String	Definisi <i>object</i> metadata. Definisi ini berisi dokumen XML yang merupakan serialisasi dari <i>object</i> metadata.
CreationDate	Date	Waktu saat <i>object</i> metadata dibuat pertama kali.
LastUpdated	Date	Waktu dimana <i>object</i> metadata mengalami perubahan.
Version	Integer	Versi <i>object</i> . Versi ini digunakan untuk menjelaskan perbedaan yang ada antara model <i>object</i>

Dalam *table* ini terdapat relasi antar *table* sendiri. Relasi ini adalah relasi untuk menunjukkan turunan *object*, maksudnya adalah setiap *object* merupakan turunan dari *object* yang lain, kecuali *base class*. *Base class* ini adalah *MetadataObject* yang memiliki ID dan relasi referensi ID sama.

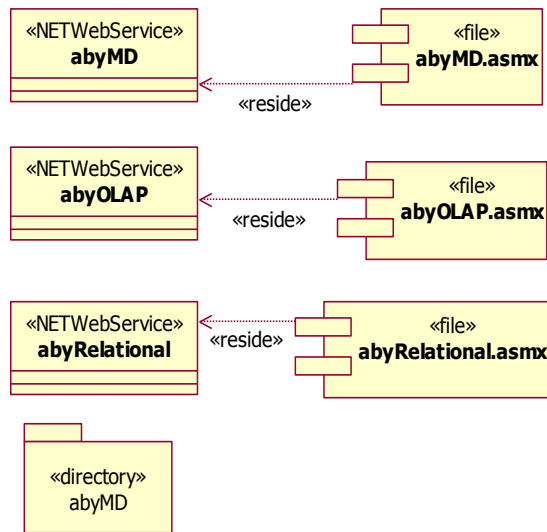
4.3. IMPLEMENTASI RANCANGAN SISTEM METADATA

Dari beberapa model hasil rancangan web service ini, kemudian tahap selanjutnya adalah melakukan *deployment* terhadap aplikasi fisik yang akan dibuat. Gambar 6 menunjukkan *deployment diagram* dari hasil rancangan yang telah dibuat.

Setiap web service diwujudkan menjadi sebuah file dengan *extension* .asmx. misalnya untuk web service *abyOLAP* menjadi file *abyOLAP.asmx*, *abyRelational* menjadi *abyRelational.asmx*, dan web service *abyMD* menjadi *abyMD.asmx*. Seluruh file web service tersebut berada dalam sebuah direktori, yaitu direktori *abyMD*. Semua file yang ada dalam satu direktori ini ditaruh dalam sebuah web server dimana web service ini akan diakses oleh *client*-nya. Web service juga harus memiliki *namespace* berupa url untuk mendefinisikan web service tersebut. Meskipun demikian, *namespace* ini tidak harus ada, tetapi hanya sebagai identifier untuk web service.

Untuk mengimplementasikan rancangan sistem metadata ini, khususnya untuk sistem *exchange*, tools **Microsoft Visual Studio .NET** digunakan dalam membuat web service. Bahasa pemrograman yang digunakan adalah **C# .NET**. Web service tidak

menghasilkan bentuk antarmuka melainkan hanya API bagi sistem lainnya.



Gambar 6. Deployment Diagram

5. HASIL UJI COBA

Ujicoba ini dimaksudkan untuk mengetahui kemungkinan implementasi sistem ini dapat digunakan oleh aplikasi lain sebagai bagian dalam sistem aplikasi. Aplikasi yang menggunakan sistem metadata ini merupakan aplikasi *presentation layer* pada arsitektur n-tier.

Skenario ujicoba yang disini adalah ujicoba sistem *exchange* / pertukaran informasi dalam sistem metadata serta ujicoba untuk mengakses dan menggunakan sistem metadata untuk berbagai keperluan. Ujicoba sistem *exchange* menggunakan PHP Hypertext Preprocessor yang akan digunakan untuk mengakses sistem metadata. Sedangkan ujicoba contoh client yang menggunakan metadata adalah contoh aplikasi OLAP sederhana dengan menggunakan Microsoft Office Pivot Table Component.

5.1. AKSES WEB SERVICE MELALUI HALAMAN DEFAULT

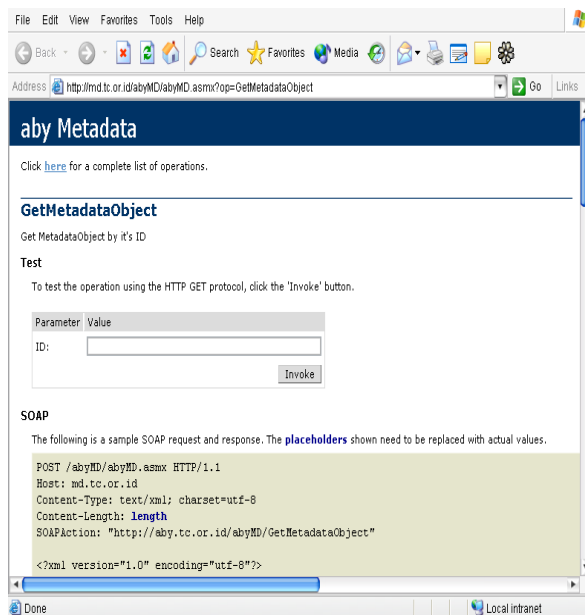
Akses web service secara langsung dapat dilakukan dari halaman default web service. Pembuatan web service dengan menggunakan *Framework .NET* menghasilkan *binding* web service melalui beberapa protokol, yaitu HTTP POST, HTTP GET, dan SOAP. Pada beberapa fungsi - fungsi yang dihasilkan web service, ada beberapa fungsi yang dapat di-*invoke* melalui HTTP GET, sehingga hasilnya dapat ditampilkan langsung ke web browser dengan jalan *query* melalui *url address*. Fungsi – fungsi yang dapat di-*invoke* melalui HTTP GET hanyalah fungsi – fungsi yang tidak memiliki parameter, atau memiliki parameter dengan tipe *native* / *standard* (misalnya tipe string, integer, atau boolean). Untuk tipe – tipe kompleks, misalnya struct

atau class, akses web service harus menggunakan protokol POST atau SOAP.

Pada skenario ujicoba ini terdapat dua fungsi pada web service yang dapat diakses dengan menggunakan protokol HTTP GET, yaitu fungsi `GetMetadataObject` dan fungsi `GetMetadataObjects`. Gambar 7 menunjukkan halaman default web service untuk fungsi `GetMetadataObject`.

Akses url ke halaman ini untuk fungsi `GetMetadataObject` adalah `http://md.tc.or.id/abyMD/abyMD.asmx/GetMetadataObject?ID=0` yang artinya *invoke* fungsi `GetMetadataObject` dengan parameter yaitu `ID = 0`. Fungsi ini akan mengembalikan hasil berupa dokumen XML seperti yang ditampilkan pada Gambar . Dokumen XML tersebut menjelaskan *return value* dari fungsi ini, yaitu `MetadataObject`, beserta atribut – atributnya, seperti `ID`, `ParentID`,

Return value ini berlainan untuk setiap parameter yang diberikan. Misalnya untuk parameter `ID = 123` maka akan ditampilkan hasil *null*, seperti yang ditunjukkan oleh dokumen XML pada gambar 8.



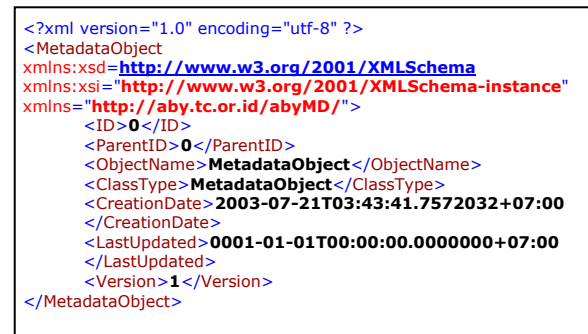
Gambar 7. Invoke `GetMetadataObject` melalui protokol HTTP GET

Selain mengembalikan *return value*, web service juga dapat mengembalikan *error exception* apabila terdapat kesalahan dalam memberikan tipe parameter atau *error exception* yang dikeluarkan sistem apabila terdapat kesalahan dalam memberikan nilai dalam parameter.

5.2. AKSES SISTEM METADATA DENGAN MENGGUNAKAN PHP

Aplikasi PHP yang dibuat sebagai ujicoba ini berfungsi sebagai sistem eksplorasi dan pengelolaan metadata. Sistem ini menggunakan web service sebagai *business logic tier* dalam mengakses dan melakukan pengelolaan metadata. Aplikasi ini

merupakan *presentation tier* pada arsitektur n-tier. Gambar 9 adalah tampilan halaman utama untuk sistem eksplorasi metadata ini.



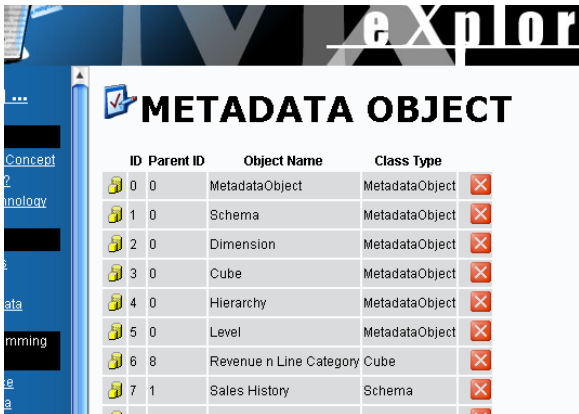
Gambar 8. Dokumen XML hasil output pada fungsi `GetMetadataObject`



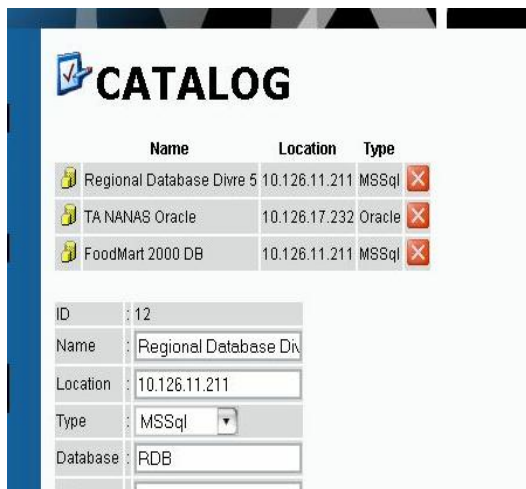
Gambar 9. Halaman utama tampilan web dengan kode PHP

Dari sistem ini dapat dilakukan eksplorasi terhadap metadata yang ada pada sistem metadata, misalnya metadata relational, metadata OLAP, atau seluruh object metadata. Gambar 10 menunjukkan daftar *object* dalam sistem metadata.

Selain berfungsi untuk eksplorasi, sistem ini juga berfungsi sebagai sarana pengelolaan metadata, yaitu dengan membuat *object* / data metadata baru dan / atau melakukan proses perubahan terhadap informasi metadata yang ada. Gambar 11 merupakan contoh data *catalog* database. Disini bisa dilakukan penambahan informasi tentang suatu database, misalnya lokasi database, jenis database, nama database, dan lain sebagainya. *Catalog* ini juga berfungsi sebagai definisi sumber data bagi *schema*, *cube*, *dimension*, atau metadata tentang OLAP lainnya.



Gambar 10. Halaman yang menampilkan daftar *object* dalam sistem metadata

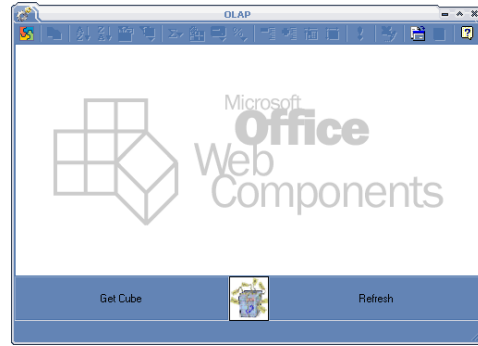


Gambar 11. Informasi tentang DBMS dalam catalog

5.3. QUERY OLAP DENGAN MENGGUNAKAN MICROSOFT OFFICE PIVOTTABLE COMPONENT

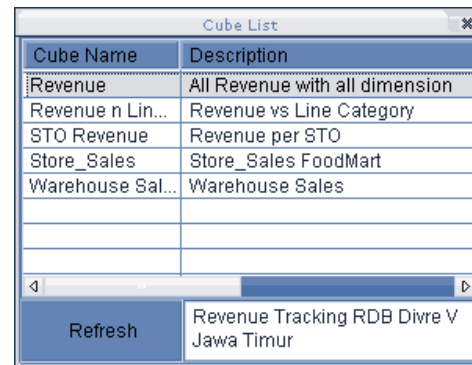
Pada scenario ujicoba kali ini akan dilakukan query terhadap data yang ada pada data warehouse untuk ditampilkan ke dalam sebuah aplikasi OLAP sederhana. Aplikasi OLAP ini dibuat dengan menggunakan Microsoft Visual Studio .NET (dengan bahasa pemrograman C# .NET) dan Microsoft Office PivotTable Component. Informasi untuk *query* dan akses terhadap data warehouse berasal dari sistem metadata. Tampilan aplikasi OLAP sederhana ini dapat dilihat pada gambar 12.

Database yang digunakan dalam sistem data warehouse ini adalah Microsoft SQL Server 2000. Sedangkan schema database yang digunakan mengacu pada *snowflake schema* untuk *factdata* revenue. Schema ini berfungsi untuk analisa dan pelaporan bisnis, yaitu Revenue Tracking pada PT. Telkom Divre V Jawa Timur.



Gambar 12. Tampilan aplikasi OLAP sederhana

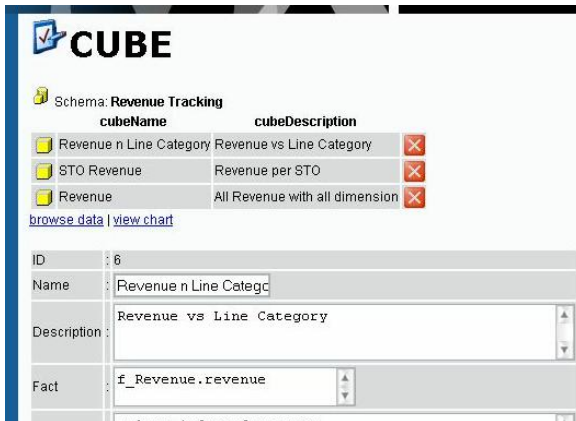
Sistem operasi yang digunakan pada client adalah Microsoft Windows NT 2000. Dalam melakukan akses database ini, di client harus telah terinstall Microsoft .NET Framework dan memiliki SQL OLEDB provider.



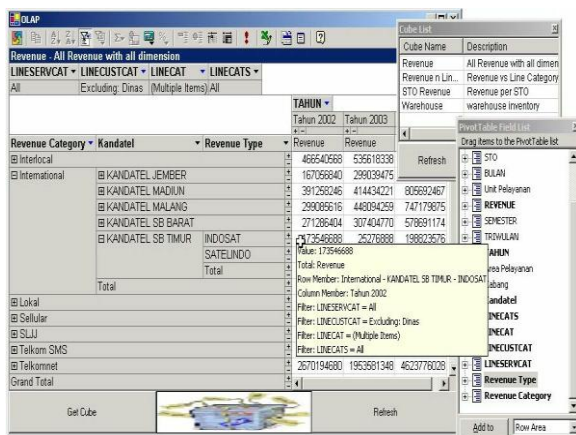
Gambar 13. Daftar cube pada aplikasi OLAP yang ada di sistem metadata

Daftar *cube* pada gambar 13 tersebut diambil dari sistem metadata. Dalam membuat *cube*, terlebih dahulu mengisi informasi tentang *cube* ke dalam metadata melalui form isian *cube* pada sistem pengelolaan metadata (aplikasi dengan PHP tersebut di atas). Gambar 14 adalah gambar yang menunjukkan daftar *cube* yang ada pada schema Revenue Tracking.

Ketika *cube* dipilih (dalam contoh ini adalah *cube* Revenue), maka aplikasi ini akan melakukan *query* ke database. Informasi tentang sumber data dan cara *query* didapatkan dari sistem metadata. Gambar 15 menunjukkan hasil *query* yang diperoleh setelah aplikasi OLAP melakukan koneksi ke database. Hasil *query* langsung disajikan ke dalam bentuk PivotTable. Dengan aplikasi OLAP ini dapat dilakukan *drilling* ke level yang lebih dalam atau dilakukan pengelompokkan terhadap kategori analisis pada *cube*.



Gambar 14. Daftar cube pada aplikasi PHP



Gambar 15. Hasil yang diperoleh dari query yang ditampilkan langsung ke pivottable

Hasil *query* dengan aplikasi OLAP sederhana ini dapat dikonversi menjadi file spreadsheet Microsoft Excel. File ini membentuk *data mart* yang dapat digunakan sebagai bahan analisis atau dipakai untuk tujuan pelaporan. Gambar 16 merupakan contoh hasil konversi data *query* aplikasi OLAP pada Gambar 15.

Dengan aplikasi ini dapat dibentuk *data mart* dalam format file spreadsheet excel yang didapatkan dari *query* ke data warehouse kemudian dikonversi dan disimpan. Selain itu dapat dilakukan juga pembentukan *cube - cube* baru dengan menggabungkan *dimension* lain yang terdapat pada PivotTable.

6. KESIMPULAN

Berdasarkan uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Penelitian ini telah dapat menghasilkan rancangan prototipe sistem metadata yang dapat digunakan untuk mendukung data warehouse.
2. Contoh implementasi rancangan prototipe berupa program eksplorasi metadata dalam PHP dapat digunakan sebagai sistem pengelolaan metadata .

3. Prototipe sistem metadata ini dapat digunakan sebagai alat bantu dalam mengembangkan sistem informasi dalam data warehouse, sebagai contohnya disini adalah aplikasi OLAP sederhana yang dapat menjadi aplikasi sistem pendukung keputusan.
4. Teknologi web service dapat digunakan untuk mendukung *metadata exchange* sehingga memungkinkan pengintegrasian antar sistem dengan metadata.

Gambar 16. Contoh hasil konversi data *query* aplikasi OLAP

7. DAFTAR PUSTAKA

1. Anahory, Sam. *Data Warehousing in the Real World: a practical guide for building decision support systems*. Addison-Wesley, 1997
2. Corey, Michael. *Oracle 8i Data Warehousing*. McGraw-Hill, 2001
3. Hai Do, Hong. *On Metadata Interoperability in Data Warehouses*. Department of Computer Science, University of Leipzig, 2000.
4. Inmon, William H. *What is Data Warehouse?* www.billinmon.com, 2000
5. Inmon, William H. *A Brief History of Metadata: From Master Files to Distributed Metadata*. www.billinmon.com, April 2001
6. Inmon, William H. *A Toward Unified Theory of Metadata*. www.billinmon.com, March 2002
7. Object Management Group, *The Common Warehouse Metamodel (specifications, papers, presentations)*, <http://www.cwmforum.org/>, <http://www.omg.org/cwm/>
8. PHP Hypertext Preprocessor Manual
9. Intranet Portal Regional Database PT. Telkom Divre 5 Jawa Timur
10. Tannenbaum, Adrienne. *Metadata solutions: using metamodels, repositories, XML, and enterprise portals to generate information on demand*. Addison-Wesley, 2002.
11. Thai, Thuan, Huang Q. Lam. *.NET Framework Essential*. 1st Edition, O'Reilly, 2001