SeTISI 2015

SEMINAR TEKNIK INFORMATIKA & SISTEM INFORMAS

PROSIDING

PERAN KEAMANAN INFORMASI MENUJU INDONESIA HEBAT DALAM MENGHADAPI ASEAN ECONOMIC COMMUNITY 2015

1965 - 2015 Universitas Kristen Maranatha

PROSIDING

SeTISI 2015 Seminar Teknik Informatika dan Sistem Informasi

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Editor: **Robby Tan, Hapnes Toba**

Desain Sampul: **Risal**

Penerbit:

Maranatha University Press (MUP)

Jl. Prof. Drg. Suria Sumantri, MPH No. 65

Bandung 40164

Cetakan pertama, 2015

ISBN: 978-602-72127-1-8

KOMITE

Pelindung
Rektor Universitas Kristen Maranatha

Penanggung Jawab
Dekan Fakultas Teknologi Informasi
Universitas Kristen Maranatha

Ketua Pelaksana
Ir. Teddy Marcus Zakaria, M.T.

Komite Program
Dr. Andi Wahju Rahardjo Emanuel, BSEE, MSSE (UKM)
Ir. Dana Indra Sensuse, MLIS., Ph.D. (UI)
Dr. Hapnes Toba (UKM)
Ito Warsito, Ph.D (UI)
Ir. Kridanto Surendro, M.Sc., Ph.D. (ITB)
Dr. Ir. Mewati Ayub, M.T. (UKM)
DR. dr. Oerip Setiono Iman Santoso, M.Sc. (ITB)
Drs. Retantyo Wardoyo, M.Sc., Ph.D. (UGM)
Prof. Dr. dra. Sri Hartati, M.Sc. (UGM)
Prof. Dr. Wiranto Herry Utomo (UKSW)
Yenni M.Djajalaksana, Ph.D. (UKM)

Komite Pelaksana
Adelia, S.Kom., M.T.
Dr. Andi Wahyu Rahardjo Emannuel, BSEE., MSSE.
Daniel Jahja Surjawan, S.Kom., M.T.
Djoni Setiawan K., S.T., M.T.
Diana Trivena Yulianti, S.Kom., M.T.
Doro Edi, S.T., M.Kom.
Erico Darmawan Handoyo, S.Kom., M.T.
Dr. Hapnes Toba
Maresha Caroline Wijanto, S.Kom., M.T.
Meliana Christianti J., S.Kom., M.T.
Dr. Ir. Mewati Ayub, M.T.
Niko Ibrahim, S.Kom., MIT
Oscar Karnalim, S.T., M.T.
Oscar Wongso, S.Kom., M.T.
Radiant Victor Imbar, S.Kom., M.T.
Risal, S.T., M.T.
Robby Tan, S.T., M.Kom.
Saron K. Yefta, S.Kom., M.T.

DAFTAR ISI

ISBN: 978-602-72127-1-8

ISBN: 978-602-72127-1-8

Prosiding Seminar Teknik Informatika dan Sistem Informasi
Bandung, 9 April 2015

ISBN: 978-602-72127-1-8

Implementation of Niemi's *Algorithms in* OLAP *Cube to Optimize Student Data Analysis*

Lilian Aymee Natalia[1], Maresha Caroline[2], Mewati Ayub[3]

Department of Informatics Engineering, Faculty of Information Technology
Maranatha Christian University, Bandung
[1]circularlogic@windowslive.com
[2]maresha.caroline@itmaranatha.org
[3]mewati.ayub@it.maranatha.edu

Abstract — **Ketersediaan sejumlah besar data yang disimpan dalam *data warehouse* atau *data mart* akan mendukung proses pengambilan keputusan. Analisis data multidimensi dilakukan dengan menggunakan operator OLAP pada *data cube*. Dalam pemrosesan *query* MDX, rancangan *data cube* akan menentukan efektivitas dari analisis data. Penelitian ini mengimplementasikan algoritma Niemi dalam membuat data cube untuk mengoptimalkan pemrosesan *query* dalam analisis data multidimensi. Implementasi ini bertujuan membantu para analis untuk mendapatkan *cube* yang optimal yang relevan hanya untuk masalah tertentu. Dengan metode ini, analis dapat memperoleh *cube* dalam bentuk yang optimal dan menggunakannya untuk menganalisis masalah yang lebih spesifik.**

Keywords — **optimasi MDX, OLAP, *cube*, analisis data**

I. Introduction

Currently, decision support systems are supported by data warehouse for availability of huge data in decision making process [1]. Datawarehouse model enables multidimensional data analysis using data cube and OLAP tools such as pivot, roll-up, drill down, slicing, and dicing. In consequence of huge data volume involved in multidimensional data analysis, data cube design will determine data cube computation in query processing [2].

Study on OLAP cube had been performed by some researchers. In [3], OLAP cube was constructed from web data in XML format, which considered dimension hierarchies and correct aggregation. Works on conditions for summarization operations on multidimensional data was explored in [4]. Niemi [5][6] proposed a logical method to design OLAP cube optimally based on queries.

In this research, an algorithm developed by Niemi [5] to construct a data cube based on queries is implemented to optimize query processing in data analysis. With this algorithm, data cube used for query processing can be dynamic based on queries. This research is a continuing of Natalia's research [7] which is part of the Hibah Bersaing of Ayub [8][9]. In previous study, students data marts were implemented using star schema and snowflakes schema. In this study, we used the student data marts as a study case for the data source for data cube. Hopefully the result of this research can present information about students optimally, so that it can be used in the process of analysis and decision making.

II. Literature Study

A data cube is built from a set of data extracted from a data warehouse that can be viewed in multiple dimensions. Each dimension of a data cube may have a hierarchy to map from a set of low level concepts to higher level. Online Analytical Processing (OLAP) operations on a data cube enable users to view data from different perspectives, thus can support decision-making process [1][2].

OLAP works by manipulating data cube using Multidimensional Expression (MDX) query designed by Microsoft [10]. Fig 1 shows general syntax of MDX query.

```
[ WITH <SELECT WITH clause>
 [, <SELECT WITH clause>... ] ]
SELECT [ * | ( <SELECT query axis clause>
 [, <SELECT query axis clause>... ] ) ]
FROM <SELECT subcube clause>
[ <SELECT slicer axis clause> ]
[ <SELECT cell property list clause> ]
```

Fig 1. Syntax of MDX Query

Niemi [5] proposed three criteria to determine the quality of an OLAP cube design. These three criteria are completeness and minimalism, correct aggregations, and minimal sparsity. Completeness and minimalism is a state where a cube has all information that is relevant to certain queries. It means, the cube can answer certain queries in effective size [5][11]. There are three necessary conditions to define correct aggregations of a cube, those are disjointness of categories in hierarchies, completeness in hierarchies, and correct use of measure attributes [5][4]. In general, sparsity means ratio between empty cells and total cells of a cube. The less the value of the sparsity, the better the cube is [5].

In Fig 2, we shown an algorithm that had been developed by Niemi [5][6] to build optimal cubes based on given queries. Basically, the algorithm combine attributes of similar queries to build a new cube. The meaning of similar queries is that they share a dimension and operate on the same hierarchy level [5].

1. For each query Q_i in a set of Q, construct set Xi by placing attributes of SELECT clause from Q_i.
2. For each set X_i, construct set Y_i by placing the dimension key of each attribute in X_i.
3. Construct equivalence classes for queries as follows:
   Two queries Q and Q' belong to the same equivalence class E if we can form a sequence of existing queries $<Q_0 = Q, Q_1, \ldots, Q_n, Q' = Q_{n+1}>$ *such that* $Y_i \cap Y_{i-1} \neq \emptyset$, $0 \leq i \leq n$, *where* Y_i *denotes the dimension key set of the query* Q_i.
4. (An additional phase to improve efficiency for analysing less detailed cubes):
   For each equivalence class E_i:
   For each query Q in E_i:
   If there is a query Q'ε E_i with the same dimensions as Q but some dimensions of Q' are in more detailed levels than in Q, then construct a new equivalence class E' as follows: E'=E_i - {Q'}.
5. Output the set of equivalence classes obtained.

Fig 2. Algorithm to build a cube based on queries by Niemi [5]

The next phase is to normalize equivalence classes obtained from the algorithm. The cube normalization consists of dimension decomposition and cube decomposition [5][6]. After the equivalence classes are constructed, cube schemata can be built based on the attributes of dimension in the classes.

III. Methodology

The research methodology used in this study consists of three steps as follows:

A. Data preparation

The data cube used in this study was built from student data mart schema that resulted from prior studies [7][8][9] as a data resource. The students' data schemata were new students data schema, active students data schema, and graduated students data schema.

B. Cube Construction

We begin with some 'similar' queries written in MDX format that represent a common cube [5]. Each query will build a data cube. To achieve completeness and minimalism in designing an OLAP cube, we execute algorithm in Fig 2 based on those queries. The result of the execution is a set of equivalence classes.

C. Cube Normalization

Normalization is needed to reduce sparsity and incorrect aggregations [5]. There are two steps to normalize a cube, which are dimension decomposition and cube decomposition. Dimension decomposition must be ensured by separating attributes X and Y in different dimensions if there does not exist an attribute that functionally resolves both X and Y. Cube decomposition is done by removing dimension conflict from all cubes [5].

IV. Implementation Result

To explain implementation of optimization of MDX queries based on Niemi's algorithms, we use cubes built from a new students data mart based on some queries for the example. The new students data mart consists of one fact table and six dimension tables. The attributes in fact table are NRP, IdProdi, IdGelombang, IdKelompokNilaiUSM, IdJurusanSMA, IdProyekPendidikan, IdSekolah, and NilaiUSM. The dimension tables are Prodi, Gelombang, Sekolah, KelompokNilaiUSM, JurusanSMA, and ProyekPendidikan.

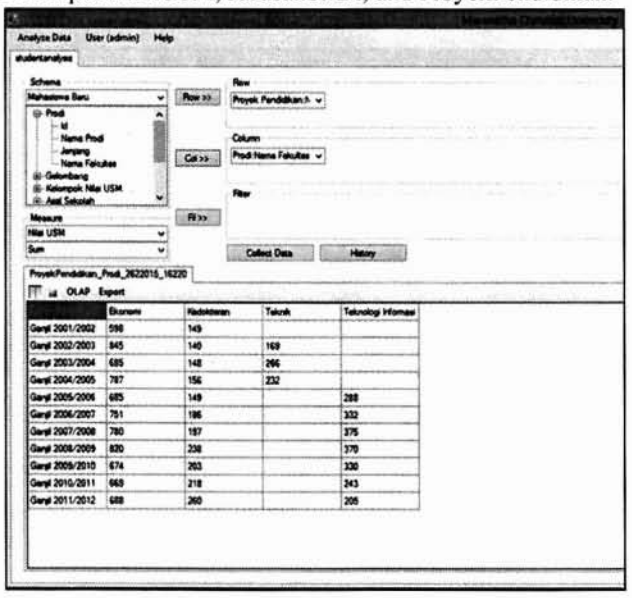

Fig 3. User interface to build cubes

User interface used to form a cube is shown in Fig 3. To build a cube, user can choose a dimension attribute for the row and another dimension attribute for the column. System will generate MDX query automatically based on user's choice.

In the following explanation, we will show a scenario to form a cube without and with optimization. We begin with three MDX queries and resulted cubes in Fig. 4 until Fig. 9 that are executed without optimization.

```
SELECT
 {[Prodi].[Nama Fakultas].Members}
on columns,
 {[Proyek Pendidikan].[Nama].Members}
 on rows
```

ISBN: 978-602-72127-1-8

```
FROM MahasiswaBaru
```

Fig 4. MDX Query -1

Proyek Pendidikan_Prodi_2622015_16220

OLAP Export

	Ekonomi	Kedokteran	Teknik	Teknologi Informasi
Ganjil 2001/2002	598	149		
Ganjil 2002/2003	845	140	169	
Ganjil 2003/2004	685	148	266	
Ganjil 2004/2005	787	156	232	
Ganjil 2005/2006	685	149		288
Ganjil 2006/2007	751	186		332
Ganjil 2007/2008	780	197		375
Ganjil 2008/2009	820	238		370
Ganjil 2009/2010	674	203		330
Ganjil 2010/2011	669	218		243
Ganjil 2011/2012	688	260		205

Fig 5. Resulted Cube of MDX Query-1

```
SELECT
 {[Prodi].[Nama Prodi].Members}
on columns,
 {[Proyek Pendidikan].[Nama].Members}
 on rows
FROM MahasiswaBaru
```

Fig 6. MDX Query -2

ProyekPendidikan_Prodi_2622015_16247

OLAP Export

	Akuntansi	D3 Teknik Informatika	Kedokteran Gigi	Kedokteran Umum	Manajemen	Sistem Informasi	Teknik Informatika
Ganjil 2001/2002	237			149	361		
Ganjil 2002/2003	365			140	480		
Ganjil 2003/2004	306			148	379		
Ganjil 2004/2005	370			156	417		
Ganjil 2005/2006	326	22		149	359	73	193
Ganjil 2006/2007	441	26		186	310	84	222
Ganjil 2007/2008	403	14		197	377	92	269
Ganjil 2008/2009	429	12	31	207	391	83	275
Ganjil 2009/2010	338	17	32	171	336	82	231
Ganjil 2010/2011	351	10	46	172	318	49	184
Ganjil 2011/2012	360	8	44	216	326	48	149

Fig 7. Resulted Cube of MDX Query-2

```
SELECT
 {[Prodi].[Nama Prodi].Members} on columns,
 {[Proyek Pendidikan].[Nama].Members
 * [Jurusan SMA].[Nama].Members} on rows
FROM MahasiswaBaru
```

Fig 8. MDX Query -3

ProyekPendidikan_JurusanSMA_Prodi_2722015_141511

OLAP Export

		Akuntansi	D3 Teknik Informatika	Kedokteran Gigi	Kedokteran Umum	Manajemen	Sistem Informasi	Teknik Informatika
Ganjil 2005/2006	Bahasa					1		
Ganjil 2006/2009	IPA	155	4	31	207	107	26	162
Ganjil 2006/2009	IPS	268	7			275	53	101
Ganjil 2009/2010	Bahasa						1	
Ganjil 2009/2010	IPA	117	7	32	171	105	24	115
Ganjil 2009/2010	IPS	219	9			224	56	111
Ganjil 2010/2011	Bahasa					1		1
Ganjil 2010/2011	IPA	144	5	46	172	95	12	95
Ganjil 2010/2011	IPS	159	3			218	35	84

Fig 9. Resulted Cube of MDX Query-3

To optimize the three cubes, we perform some steps as follows:

1. Choose the cubes that will be optimized as shown in Fig 10.

Fig 10. User interface to optimize query

2. After the optimize button is chosen, system will execute optimization and generate two new cubes as shown in Fig 11.

Fig 11. User interface of optimization result

3. Based on Niemi's algorithms[5], the optimization is done as follows:
   a. Construct set X for each query by taking all attributes from SELECT clause, which are X_1 from $Query_1$, X_2 from $Query_2$, and X_3 from $Query_3$.
   X_1 = {nama fakultas, nama proyek pendidikan}
   X_2 = {nama prodi, nama proyek pendidikan}
   X_3 = {nama prodi, nama proyek pendidikan, nama jurusan SMA}

b. Construct set Y by placing the dimension key or name of each attribute in set X, which are Y_1 from X_1, Y_2 from X_2, and Y_3 from X_3.
Y_1 = {prodi, proyek pendidikan}
Y_2 = {prodi, proyek pendidikan}
Y_3 = {prodi, proyek pendidikan, jurusan SMA}
c. Construct all possible equivalence classes. $Query_1$ is equivalence with $Query_2$ if set Y_1 and set Y_2 have an intersection. In this case, we deceive two classes, which are E_1 = {$Query_1$, $Query_2$} and E_2 = {$Query_1$, $Query_3$} or E_2={$Query_2$, $Query_3$}.
d. The normalized cubes resulted from the process are C1 from E_1 and C2 from E_2. C1 = {namaprodi, namaFakultas, namaProyekPendidikan}; C2 = {namaprodi, namaFakultas, namaProyekPendidikan, namaJurusanSMA}. C1 and C2 are shown in Fig.12 and Fig. 13.

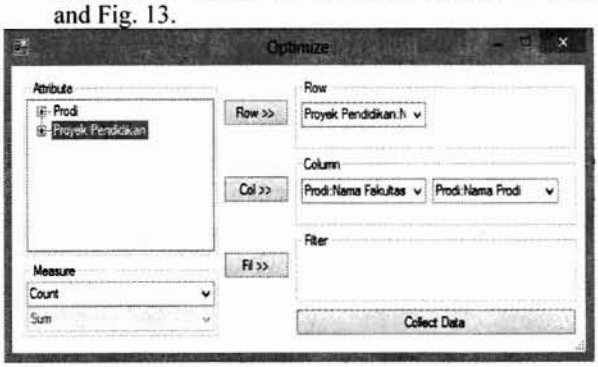

Fig 12. User interface for cube C1

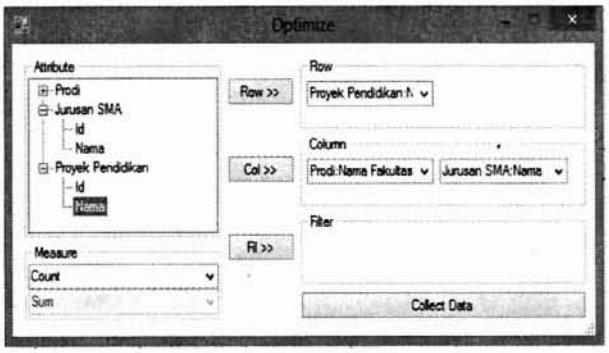

Fig 13. User interface for cube C2

We can generate a new query from C1, for example we choose an attribute NamaProyek from dimension ProyekPendidikan for the row and attributes NamaFakultas and NamaProdi from dimension Prodi for the column. The resulted cube of the query is shown in Fig. 14.

	Ekonomi	Ekonomi	Kedokteran	Kedokteran	Teknologi Informasi	Teknologi Informasi	Teknologi Informasi
	Akuntansi	Manajemen	Kedokteran Gigi	Kedokteran Umum	D3 Teknik Informatika	Sistem Informasi	Teknik Informatika
Ganjil 2001/2002	237	361		149			
Ganjil 2002/2003	365	480		140			
Ganjil 2003/2004	306	375		148			
Ganjil 2004/2005	370	417		156			
Ganjil 2005/2006	326	359		149	22	73	153
Ganjil 2006/2007	441	310		186	26	84	222
Ganjil 2007/2008	403	377		197	14	92	269
Ganjil 2008/2009	429	391	31	207	12	93	275
Ganjil 2009/2010	338	336	32	171	17	82	231
Ganjil 2010/2011	351	318	46	172	10	49	184
Ganjil 2011/2012	360	328	44	216	8	48	149

Fig. 14. Resulted cube of a query to C1

With the same manner, we can also generate a new query from C2. In this case, we choose an attribute NamaProyek from dimension ProyekPendidikan for the row and an attribute NamaFakultas from dimension Prodi and an attribute NamaJurusan from dimension JurusanSMA for the column. The resulted cube of the query is shown in Fig 15.

	Ekonom	Ekonom	Ekonom	Kedokteran	Teknik	Teknik	Teknolog Informas	Teknolog Informas	Teknolog Informas
	Bahasa	IPA	IPS	IPA	IPA	IPS	Bahasa	IPA	IPS
Ganjil 2001/2002	12	120	449	149					
Ganjil 2002/2003	2	257	576	140	87	75			
Ganjil 2003/2004	2	150	525	148	126	133			
Ganjil 2004/2005	2	200	568	156	110	120			
Ganjil 2005/2006	1	179	497	149				140	136
Ganjil 2006/2007		210	531	186				171	152
Ganjil 2007/2008		245	522	197				175	187
Ganjil 2008/2009	1	262	544	228				192	161
Ganjil 2009/2010		223	443	203			1	146	176
Ganjil 2010/2011	1	209	417	216			1	132	122
Ganjil 2011/2012	6	263	407	260			4	94	98

Fig. 15. Resulted cube of a query to C2

Without optimization, the application will generate a query or a cube from data mart schema as shown in Fig 3. With optimization to some similar queries, the application can form new cubes that satisfy Niemi's criteria for optimal cubes. The size of data mart schema usually is larger than cubes size formed from the schema. A group of similar queries represent a common cube, therefore it is more effective to form an optimal cube from those queries. Because the cube has all information that is relevant to certain queries, the cube can answer certain queries in effective size compared with data mart size.

In the previous study [7], cubes used for query processing were static as shown in Fig. 16. In the implementation of this research, the cubes can be dynamic as shown in Fig. 3, because the cubes can be optimized based on queries. Actually the cube that can be generated in this application can be in the form of tables and graphs. For example, Fig. 3 shows a cube in the form of tables.

ISBN: 978-602-72127-1-8

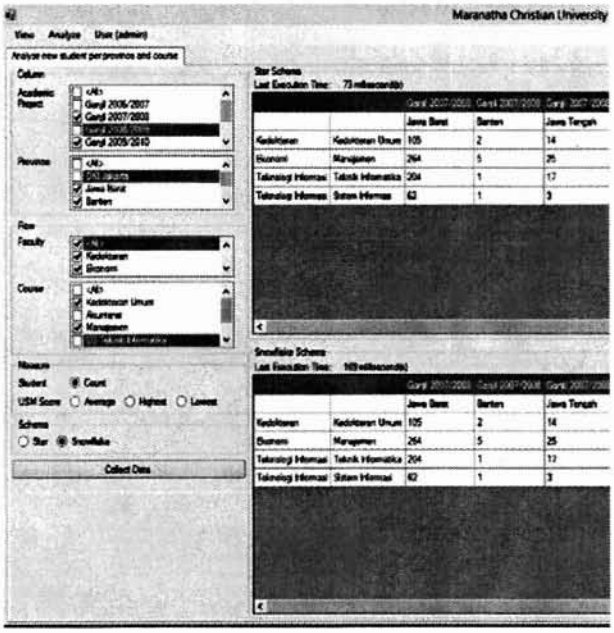

Fig. 16. User interface of data analysis (static)

The OLAP operators interface used for query is viewed in Fig. 17. User can perform OLAP operations such as roll-up, drill-down, slice and dice, and pivot. In the implementation, MDX query was generated by application based on user selection in the interface.

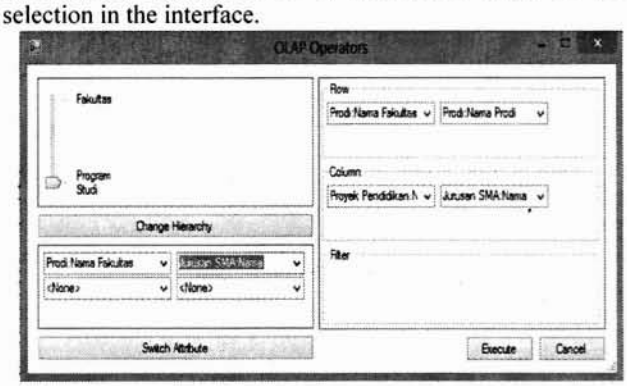

Fig. 17. The OLAP operators interface

V. CONCLUSION

In this paper, we have presented the implementation of Niemi's algorithm to construct optimal cubes based on given queries. The implementation process consists of the combination of attributes of similar queries to build a new cube, and the normalization of the cube. As the result, we can construct a dynamic cube based on given queries. To complete the data analysis, this study also implemented OLAP operators, such as roll-up, drill down, slicing, dicing, and pivot to work on the data cube.

REFERENCES

[1] W.H. Inmon, *Building The Data Warehouse*, 4th ed., Indianapolis: Wiley Publishing, Inc., 2005.

[2] J. Han, M. Kamber, J.Pei, *Data Mining Concepts and Techniques*, 3rd ed., Waltham: Elsevier, Inc, 2012.

[3] M.R. Jensen, T.H. Moller, T.B. Pedersen, "Specifying OLAP Cubes on XML Data", *Journal of Intelligent Information Systems*, vol.17, pp. 255 – 280, Dec.2001.

[4] H.J. Lenz, & A. Shoshani, "Summarizability in OLAP and Statistical Data Bases," [Online]. Tersedia: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.9702&rep=rep1&type=pdf.

[5] T. Niemi, J. Nummenmaa, & P. Thanisch (2001), *Constructing OLAP Cubes Based on Queries*. [Online]. Tersedia: http://www.cis.drexel.edu/faculty/song/dolap/dolap01/paper/Niemi%20-%202.pdf.

[6] T. Niemi, "Methods for Logical OLAP Designs," Academic Dissertation of Department of Computer and Information Sciences University of Tampere, Tampere, Finland, 2001.

[7] L.A. Natalia, Implementasi Skema Star dan Snowflake Dalam Pembuatan Data Mart Dengan Menggunakan Data Mahasiswa dan Dosen di Universitas Kristen Maranatha, Universitas Kristen Maranatha, Laporan Kerja Praktek, 2014.

[8] M. Ayub, & T. Kristanti, "Model Analisis Classification dan Clustering untuk Data Mahasiswa dan Dosen di Perguruan Tinggi," Universitas Kristen Maranatha, Laporan Penelitian Hibah Bersaing DIKTI tahun anggaran 2013, 2013.

[9] M. Ayub, T. Kristanti, & M. Caroline, "Data Warehouse Sebagai Basis Analisis Data Akademik Perguruan Tinggi," *Prosiding Seminar Nasional Teknologi Informasi, Fakultas Teknologi Informasi Universitas Tarumanagara*, pp. 18 -23, Nov.2013.

[10] Microsoft. (2014). MDX Query Fundamentals (Analysis Services). [Online]. Tersedia: https://msdn.microsoft.com/en-us/library/ms145514.aspx

[11] L. Cabibbo, & R. Torlone, "A logical approach to multidimensional databases," [Online]. Tersedia: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.9702&rep=rep1&type=pdf.