

# PENGEMBANGAN APLIKASI PEMODELAN DATA MULTIDIMENSI BERBASIS JAVA PADA PostgreSQL

*by Allen F. Aritonang Mewati Ayub*

---

**Submission date:** 12-Aug-2021 12:39PM (UTC+0700)

**Submission ID:** 1630517922

**File name:** 010\_PENGEMBANGAN\_APLIKASI\_PEMODELAN\_DATA\_MULTIDIMENSI.pdf (742.54K)

**Word count:** 2471

**Character count:** 16055

# PENGEMBANGAN APLIKASI PEMODELAN DATA MULTIDIMENSI BERBASIS JAVA PADA *PostgreSQL*

Allen F. Arironang<sup>1</sup>, Mewati Ayub<sup>2</sup>

Jurusan Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri no. 65 Bandung 40164

Email : allen\_f\_arironang@yahoo.com,

mewatia@yahoo.com

## ABSTRACT

*Analyzing data in a large volume can be done with multi-dimensional data modeling, which is called data cube. The goal of this application is providing tools for users to analyze data with multidimensional model. Data cube operators, like roll-up, drill-down, slice, dice and pivot or rotate, can help users to analyze data in flexible ways. This application was built in Java and use PostgreSQL database to store data. Data cube is implemented as a query result from relational data base. Testing results indicated that this application can be running well with different data sources. Roll-up, drill-down and rotate operators can be combined each other, but there are two operators that cannot be combined, they are slice and dice operators.*

**Keywords :** data cube, multi-dimensional data modelling, data cube operator.

## ABSTRAK

*Analisis terhadap data bervolume besar dapat dilakukan melalui pemodelan data multi dimensi, yang biasa disebut sebagai kubus data (data cube). Aplikasi ini bertujuan menyediakan perangkat untuk memudahkan pengguna dalam melakukan analisis data dengan model multidimensi. Sejumlah operator data cube seperti roll-up, drill-down, slice, dice dan pivot atau rotate, dapat membantu pengguna untuk menganalisis data secara fleksibel. Aplikasi ini dibangun dengan bahasa pemrograman Java dan menggunakan basis data PostgreSQL untuk penyimpanan data. Data cube diimplementasikan sebagai hasil query terhadap suatu basis data relasional. Hasil pengujian menunjukkan bahwa aplikasi ini dapat berjalan dengan baik untuk berbagai data sumber yang berbeda. Operator roll-up, drill-down dan rotate dapat dikombinasikan satu sama lain, namun terdapat dua operator yang tidak dapat dikombinasikan, yaitu operator slice dan dice.*

**Kata kunci:** data cube, pemodelan data multidimensi, operator data cube.

## 1. PENDAHULUAN

Penggunaan teknologi informasi di hampir semua bidang kehidupan menghasilkan data dengan volume yang sangat besar. Perangkat lunak yang dapat melakukan analisis terhadap kumpulan data dengan volume yang sangat besar akan sangat memudahkan pengguna dalam menganalisis data dari beberapa dimensi yang berbeda. Perangkat tersebut dapat direalisasikan melalui penggunaan *data warehouse* dengan perangkat *on-line analytical processing* (OLAP) [3]. Dengan perangkat OLAP, pengguna dapat melakukan analisis data secara interaktif dan juga secara multidimensi, sehingga dapat menganalisis data dari beberapa tingkat abstraksi yang berbeda.

Saat ini aplikasi OLAP dalam *data warehouse* masih ditujukan untuk *database* komersial seperti *Oracle* dan *SQL Server*, sedangkan aplikasi OLAP yang bersifat *open source* barulah *Mondrian* yang saat ini sudah dikembangkan sampai dengan versi 3.1.2.1. (<http://mondrian.pentaho.org/>).

Adapun penelitian yang kami lakukan adalah mengembangkan aplikasi OLAP yang bersifat *open-source* dengan menggunakan bahasa pemrograman *Java* pada basis data *PostgreSQL*.

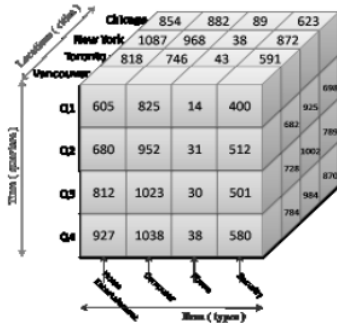
## 2. PEMODELAN, DESAIN, dan IMPLEMENTASI

### 2.1 Pemodelan Data Multidimensi

Pemodelan data pada *data warehouse* didasarkan atas pemodelan data multidimensi. Dalam pemodelan tersebut, data ditampilkan dalam bentuk *data cube*, yang memungkinkan data dipandang dari beberapa dimensi yang berbeda [3][4][5]. Pada gambar 1 ditunjukkan suatu contoh *data cube*.

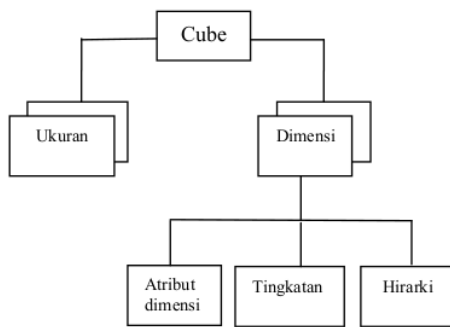
Pemodelan data multidimensi tersusun atas data cube, ukuran (measures), dimensi, hirarki, tingkatan (level), dan atribut [5]. Data cubes menyediakan perangkat untuk mengorganisasi ukuran-ukuran

pada dimensi yang sama. Ukuran akan menghubungkan sel-sel dari suatu data cube dengan fakta yang dikumpulkan dari operasi bisnis. Dimensi mengandung kumpulan nilai yang unik yang mengidentifikasi dan mengelompokkan data. Hirarki merupakan cara untuk mengorganisasi data pada beberapa tingkatan agregasi yang berbeda. Setiap tingkatan merepresentasikan suatu posisi dalam hirarki. Atribut menyediakan informasi tambahan mengenai data.



Gambar 1 Data Cube [3].

Pada gambar 2 ditunjukkan diagram dari pemodelan data multidimensi.

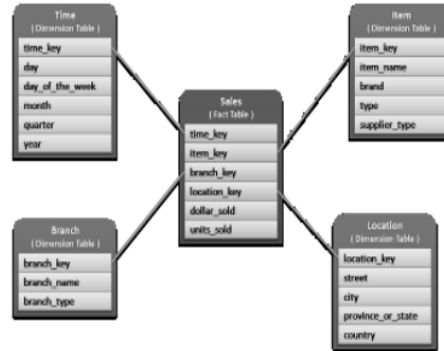


Gambar 2 Diagram Pemodelan Data Multidimensi [5].

**11** *Data warehouse* memerlukan skema yang ringkas dan berorientasi subjek yang dapat mendukung pelaksanaan operasi OLAP. Terdapat tiga jenis skema yang dapat digunakan untuk memodelkan data multidimensi pada *data warehouse* yaitu, *Star Schema*, *Snowflake Schema*, dan *Fact Constellation Schema* [3][5].

*Star schema* adalah skema yang tersusun atas satu tabel pusat yang disebut tabel *fact* yang menyimpan data tanpa redundansi, serta sekumpulan tabel pendukung yang disebut tabel dimensi, dimana setiap tabel mewakili satu dimensi. Pada gambar 3

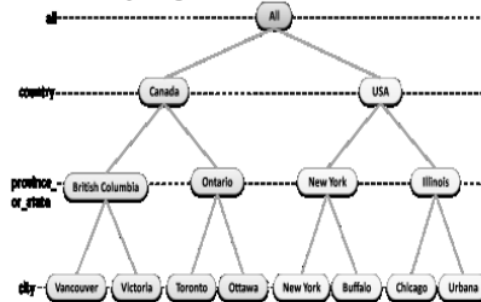
ditunjukkan contoh suatu *star schema*, dengan Sales sebagai tabel *fact*, dan empat tabel dimensi, yaitu *Time*, *Item*, *Branch*, dan *Location*.



Gambar 3 Star Schema [4]

Skema *snowflake* adalah skema *star* dengan normalisasi data pada semua tabel dimensinya. Sedangkan *fact constellation* adalah beberapa skema *star* yang digabungkan sebagai satu skema.

Atribut di dalam tabel dimensi mempunyai hirarki, yang didefinisikan sebagai serangkaian pemetaan dari sekumpulan atribut pada tingkat rendah ke tingkat yang lebih tinggi, atau konsep yang lebih umum. Sebagai contoh, pada dimensi *location* seperti pada gambar 3, terdapat hirarki *street < city < province\_or\_state < country*. Hirarki tersebut diilustrasikan pada gambar 4.



Gambar 4 Hirarki pada tabel dimensi Location [3].

Analisis data dalam *data cube* dilakukan dengan menggunakan operasi yang disebut sebagai operasi OLAP, yang terdiri atas *roll-up*, *drill-down*, *slice*, *dice*, dan *pivot* [2][3].

Operasi *roll-up* menjalankan operasi agregasi pada *data cube* dengan menaikkan tingkat hirarki (*climbing up*) untuk suatu dimensi atau melakukan reduksi dimensi.

*Drill-down* adalah kebalikan dari operasi *roll-up*, *drill-down* memecah data dari yang umum ke data

yang lebih rinci. Operasi *drill-down* dapat direalisasikan dengan menurunkan hirarki atau menambah dimensi baru.

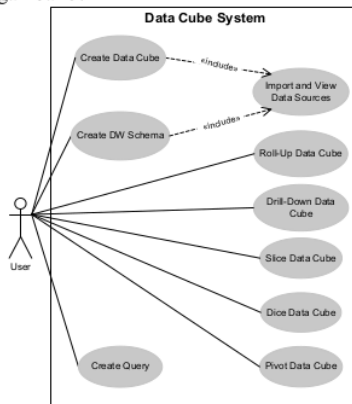
Operasi *slice* melakukan seleksi terhadap satu dimensi dari *data cube*, kemudian melakukan pemotongan terhadap dimensi tersebut, sehingga dihasilkan *sub cube* dalam bentuk berdimensi dua.

Operasi *dice* melakukan operasi yang mirip dengan *slice* hanya saja menghasilkan *sub cube* dalam bentuk berdimensi tiga.

*Pivot* adalah operasi pada *data cube* dengan melakukan rotasi pada dimensi *data cube*, hal ini dimaksudkan untuk memberi alternatif kepada pengguna dalam menampilkan data melalui sudut pandang yang lain.

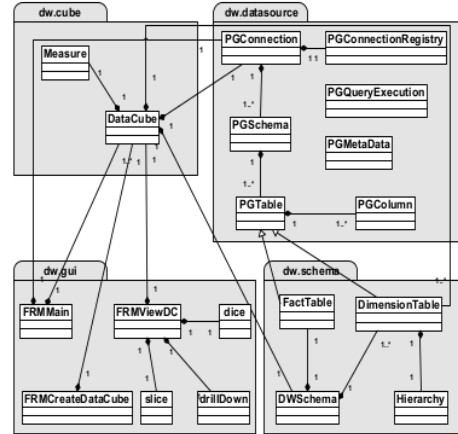
## 2.2 Desain Aplikasi

Aplikasi pemodelan data multidimensi dirancang sebagai suatu aplikasi yang mempunyai sejumlah fitur yang diperlukan untuk pembentukan *data cube* dan pelaksanaan operasi OLAP seperti ditampilkan pada gambar 5.



Gambar 5 Diagram Use Case Aplikasi

Fitur yang disediakan oleh perangkat lunak untuk memodelkan data multidimensi adalah fitur untuk membuat skema *data warehouse* (*Create DW Schema*), fitur untuk membuat *data cube* (*Create Data Cube*), fitur untuk setiap operator OLAP (*Roll-up*, *Drill-down*, *Slice*, *Dice*, *Pivot*), serta fitur untuk melakukan *query* (*Create Query*). Terdapat juga fitur untuk mengimpor data sumber yang akan dibuat skemanya atau *data cube*-nya, yaitu *Import and View Data Sources*.



Gambar 6 Diagram Kelas

Gambar 6 memperlihatkan diagram kelas dari setiap *package* untuk membangun aplikasi yang dikembangkan. Adapun *package-package* tersebut adalah *package* *dw.cube*, *dw.datasource*, *dw.gui*, serta *dw.schema*.

*Package* *dw.cube* berfungsi untuk merepresentasikan *data cube* ke dalam sistem dan terdiri dari kelas *DataCube* dan *Measure*. *Package* *dw.datasource* berfungsi untuk melakukan koneksi, mengeksekusi *query* dan membaca *metadata* dari database PostgreSQL, dan terdiri dari *PGConnection*, *PGConnectionRegistry*, *PGSchema*, *GTable*, *PGColumn*, *PGMetaData*, *PGQueryExecution*. *Package* *dw.schema* menyediakan fungsi-fungsi untuk membuat skema *data warehouse* dan terdiri dari kelas *FactTable*, *DimensionTable*, *DWSchema*, dan *Hierarchy*. *Package* *dw.gui* merupakan antarmuka aplikasi dengan pengguna dan terdiri dari kelas *FRMMain*, *FRMCreateDataCube*, *FRMViewDC*, *slice*, *dice*, *Drilldown*.

## 2.3 Implementasi Aplikasi

Implementasi aplikasi menyangkut dua hal, yaitu implementasi *data cube* dan implementasi kelas-kelas Java untuk setiap rancangan fitur. *Data cube* dapat diimplementasikan dengan dua cara yaitu dengan menggunakan basis data relasional atau dengan menggunakan *array* multidimensi [3][6].

Aplikasi yang dibahas dikembangkan dengan menggunakan basis data relasional, sehingga *data cube* dibentuk melalui eksekusi *query SQL*. Adapun skema *data warehouse* yang digunakan adalah skema *star*. Keuntungan dari penggunaan basis data relasional adalah penanganan data tidak perlu dilakukan oleh aplikasi dan menyerahkan masalah penanganan data kepada pengelola basis data[1][5].

Berikut ini diberikan contoh bagaimana m

menampilkan *data cube* sebagai tabel berdimensi tiga. Sebagai contoh, dengan menggunakan skema *star* seperti pada gambar 3, akan dibentuk *data cube* untuk menampilkan jumlah penjualan per tahun untuk setiap *country* dan untuk setiap *type*, dengan tabel *item* sebagai dimensi x, tabel *location* sebagai dimensi y, dan tabel *time* sebagai dimensi z. Pada tabel 1, ditunjukkan tampilan yang ingin dihasilkan.

Tabel 1 Tampilan *Data Cube* dalam Tabel Berdimensi Tiga

year = '2002'		
country	Type	
	Computer	Home Entertainment
US	200000	12900
Canada	45000	7000

Tampilan data seperti pada tabel 1 memerlukan suatu fungsi *cube\_slice* sebagai berikut :

```
CREATE FUNCTION cube_slice(text, text,
text)
RETURNS double precision AS
$BODY$
SELECT (SELECT dolar_sold
FROM data_cube
WHERE year = DC.year
AND country = DC.country
AND type = DC.type
) FROM data_cube DC
WHERE DC.year = $1
AND DC.country = $2
AND DC.type = $3;
$BODY$
LANGUAGE 'sql' VOLATILE COST 100;
```

Fungsi *cube\_slice* di atas diaplikasikan dalam *query* berikut ini :

```
SELECT DISTINCT
CUBE.country AS country,
cube_slice(
CUBE.year,
CUBE.country,
'Computer'
) AS "Computer",
cube_slice(
CUBE.year,
CUBE.country,
'Home Entertainment'
) AS "Home Entertainment",
...
FROM data_cube CUBE
WHERE CUBE.year ='2002';
```

Berikut ini ditunjukkan bagaimana implementasi metode untuk membangun skema *star*, membuat tabel *fact* dan tabel dimensi dalam bentuk *wizard*.

```
private void createDWSAction(){
String catalogName =
pgConn.getCatalogName();
String userName =
pgConn.getConnectionRegistry().
getUserName();
String schemaName =
cbSchema.getModel().
getSelectedItem().toString();
sqlDWSchema = "--Data Warehouse
Schema : dws_" + catalogName +
"\n";
sqlDWSchema+="DROP SCHEMA IF EXISTS
dws_" + catalogName +
"CASCADE;\n" +
"CREATE SCHEMA dws_" +
catalogName
+ " AUTHORIZATION " + userName +
";\n";
"GRANT ALL ON SCHEMA dws_" +
catalogName +
" TO " + userName + ";\n" +
"GRANT ALL ON SCHEMA dws_" +
catalogName + " TO "
+ schemaName + ";\n";
txtSQL.setText(sqlDWSchema);
tabWizard.setSelectedIndex(1);
}
```

Program 1 Metode untuk Membuat Skema *Star*

Program 1 menunjukkan pembuatan skema *star*, program 2 menunjukkan pembentukan tabel *fact*, dan program 3 untuk pembuatan tabel dimensi.

```
private String createFactTable(){
String fromTable =
cbDSTable.getSelectedItem().toString();
String factTable = "fact_" + fromTable;
String sql="";
sql+="\n--Fact Table : " + dwSchema + ". " +
factTable + " \n";
sql+= "CREATE TABLE " + dwSchema + ". "
+ factTable + " AS \n";
sql+= "\tSELECT ";
for(i=0;i<tableModel.getRowCount();i++){
String columnType = tableModel.
getValueAt(i, 0).toString();
String columnName = tableModel.
getValueAt(i, 1).toString();
String columnAlias = tableModel.
getValueAt(i, 2).toString();
if(columnAlias.equals("")){
sql+=columnName + ", ";
}else{
sql+=columnName + " AS "
+ columnAlias + ", ";
}
}
sql=sql.substring(0, sql.length()-2);
sql+="\n\tFROM " + fromTable + ";\n";
return sql;
}
```

Program 2 Metode untuk Membuat Tabel *Fact*

```

private void createAllDTAction() {
    for(int i=0;i<sqlSeq.size();i++){
        String sql = sqlSeq.get(i);
        if(!sql.equals(null)||
            !sql.equals("")){
            int result = PGQueryExecution.
                executeUpdate
                (pgConn.getConnection(), sql);
        }
    }
}

```

Program 3 Metode untuk Membuat Tabel Dimensi

```

private void createDataCube(){
    int dcid=nextDCID();
    String dimTableX =
        cboModelX.getSelectedItem().
            toString();
    String dimTableY =
        cboModelY.getSelectedItem(). toString();
    String dimTableZ =
        cboModelZ.getSelectedItem(). toString();
    String measure =
        cboMeasure.getSelectedItem().
            toString();
    String aggregate =
        cboAggregat.getSelectedItem().
            toString().toLowerCase();
    String dataCubeName = "cube_" + aggregate +
        "_" + measure + "_" + dcid;
    String functionName = "cs_" + aggregate + "_"
        + measure + "_" + dcid;

    dataCube = new DataCube();
    dataCube.setPgConn(pgConn);
    dataCube.setDwSchema(dwSchema);
    dataCube.setDataCubeID(dcid);
    dataCube.setName(dataCubeName);
    dataCube.setFunctionName(functionName);
    dataCube.setDimensionX
        (getDimTableObject(dimTableX));
    dataCube.setDimensionY
        (getDimTableObject(dimTableY));
    dataCube.setDimensionZ
        (getDimTableObject(dimTableZ));
    dataCube.setMeasure(new Measure(measure,
        aggregate));
    dataCube.initLevel();

    String sql = saveDataCubeInfo(dcid,
        dataCubeName, measure, functionName,
        aggregate);

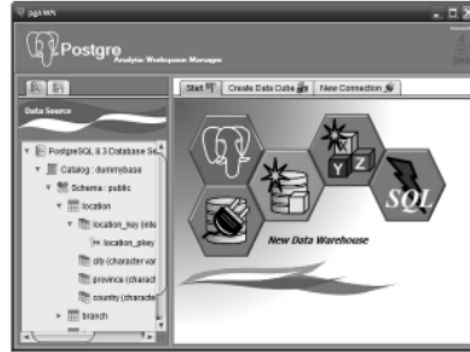
    sql+=dataCube.createDataCubeSQL();
    sql+=dataCube.updateLevel();
    int result = PGQueryExecution.
        executeUpdate(pgConn.getConnection()
        , sql);
    FRMMain frmMain = (FRMMain)GUIUtil.
        getCreatedWindows("dw.gui.FRMMain");
    frmMain.addDataCube(dataCube);
}

```

Program 4 Metode untuk Membuat Data Cube

Program 4 menunjukkan cara pembentukan *data cube*, yaitu dengan menetapkan parameter dimensi x, y, z, serta fungsi agregasi yang akan dipakai. Kemudian metode *createDataCube* akan berhubungan dengan basis data untuk mengakses skema *star* yang sebelumnya sudah dibuat.

Implementasi antarmuka utama dari aplikasi dapat dilihat pada gambar 7, yang menampilkan semua fitur yang disediakan oleh aplikasi.



Gambar 7 Form Utama Aplikasi

Gambar 8 memperlihatkan tampilan dari *form view data cube* untuk menampilkan *data cube* dan melakukan operasi OLAP terhadap *data cube*. Pada *form* ini terdapat *list box* yang menampilkan dimensi z dari *data cube*, komponen tabel yang menampilkan dimensi x dan y, serta komponen *tab* yang berisi tombol-tombol operator OLAP.

year	No	branch_name	brand	total_sold	total_sold	
2002	1	AEGIS	1.554x10 <sup>5</sup>	5.493x10 <sup>5</sup>	1.548x10 <sup>5</sup>	21600
2004	2	AEGON	20500	2.224x10 <sup>5</sup>	33700	21840
2005	3	AMANTING	77050	2.348x10 <sup>5</sup>	92000	10720
2006	4	ANTARKASA	1.103x10 <sup>5</sup>	4.533x10 <sup>5</sup>	1.108x10 <sup>5</sup>	27480
2007	5	BEAT HARBINGHY	62000	2.970x10 <sup>5</sup>	80000	12040
2008	6	BUDI MEGAN JAYA	81180	1.720x10 <sup>5</sup>	35000	12200
	7	G-PHASE	1.248x10 <sup>5</sup>	2.933x10 <sup>5</sup>	44700	9520
	8	INDO UNIVERSAL	80250	2.220x10 <sup>5</sup>	1.932x10 <sup>5</sup>	10200
	9	MARK B	1.071x10 <sup>5</sup>	3.370x10 <sup>5</sup>	40000	17880
	10	RESILENCE	79610	1.554x10 <sup>5</sup>	37000	10360

Gambar 8 Form View Data Cube

### 3. HASIL EVALUASI

Fungsionalitas dari *class-class* yang terdapat pada *package dw.cube*, *dw.datasource*, dan *dw.schema* telah diuji dengan *unit testing* menggunakan modul *JUnit* versi 4.0 yang sudah terintegrasi pada *IDE NetBeans 6.5* Hasil pengujian menunjukkan bahwa

ketiga *package* tersebut berfungsi sesuai spesifikasi.

Aplikasi ini juga telah diuji secara *black box* untuk memastikan setiap fitur dapat berfungsi sesuai dengan spesifikasinya. Pengujian juga dilakukan terhadap setiap operator OLAP, yaitu *roll-up*, *drill-down*, *slice*, *dice* dan *pivot* serta kombinasinya. Hasil pengujian menunjukkan bahwa ada beberapa operator yang tidak dapat dikombinasikan, yaitu kombinasi *slice* dan *dice*, kombinasi *slice* dengan *drill-down* dan kombinasi *dice* dengan *roll-up*.

Aplikasi juga sudah diuji dengan beberapa data sumber yang berbeda dan hasil pengujian menunjukkan bahwa aplikasi dapat bekerja dengan baik.

#### 4. KESIMPULAN

Aplikasi pemodelan data multidimensi ini dapat membantu pengguna dalam menganalisis data yang berbentuk *data cube* dengan memanfaatkan kelima operator OLAP. Melalui operator OLAP, pengguna dapat secara fleksibel menganalisis data dari berbagai sudut pandang serta dalam berbagai tingkat abstraksi data. Aplikasi ini hanya dapat menangani *data cube* dari dimensi satu sampai dengan dimensi tiga.

Hasil yang diberikan oleh setiap operator OLAP sudah sesuai dengan spesifikasi, namun evaluasi dari kombinasi operator OLAP menunjukkan bahwa tidak semua dari operator tersebut dapat dikombinasikan. Aplikasi ini hanya dapat melakukan kombinasi antara operator *roll-up*, *drill-down* dan *rotate*. Masih ada masalah teknis dalam mengkombinasikan operator *dice* dan *slice* dengan operator yang lainnya.

Implementasi *data cube* sebagai *array* multidimensi pada awal pengembangan menimbulkan masalah memori bila aplikasi menangani jutaan data, sehingga dicari solusi alternatif. Solusi yang selanjutnya diambil adalah menggunakan basis data relasi untuk penyimpanan data, yaitu dengan membentuk *data cube* sebagai hasil dari *query* terhadap tabel-tabel di dalam basis data.

Jika dibandingkan dengan aplikasi Mondrian, fitur-fitur yang diimplementasikan dalam aplikasi ini barulah fitur dasar, aplikasi masih belum dilengkapi dengan fitur untuk visualisasi hasil dalam bentuk grafik, ataupun bahasa *query* untuk data multidimensi.

#### 5. DAFTAR PUSTAKA

Chan, Patrick. (2002). The Java™ Developers Almanac 1.4. Volume 1. Indianapolis : Pearson Education.

- [1] Gajah Mada University : Master of Information Technology. (2001). Data Warehouse or Data Market. Retrieved from [http://mti.ugm.ac.id/~adji/courses/resources/CICC/Short%20Cicc/text/or8\\_text/ch1.htm](http://mti.ugm.ac.id/~adji/courses/resources/CICC/Short%20Cicc/text/or8_text/ch1.htm), diakses pada tanggal 28 November 2008.
- [2] Gray, Jim. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Netherlands : Kluwer Academic Publishers.
- [3] Han, Jiawei, & Kamber, Micheline. (2006). Data Mining Concepts and Techniques. 2<sup>nd</sup> Edition. San Fransisco: Morgan Kaufman.
- [4] Kay, Jossel. (2004). QuickStudy : Data Cubes. <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=91640>, diakses pada tanggal 7 Juni, 2008.
- [5] Oracle (2003). The Multidimensional Data Model. Oracle OLAP Application Developer's Guide 10g Release 1. <http://www.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm>, diakses pada tanggal 7 Juni 2008.
- [6] Pendse, Nigel. (2006). OLAP Architectures. <http://www.olapreport.com/achitectures.htm>, diakses pada tanggal 7 Juni 2008.
- [8] Zukowski, John. (2005). The Definitive Guide to Java Swing. 3<sup>rd</sup> Edition. Berkeley, CA : Apress.

# PENGEMBANGAN APLIKASI PEMODELAN DATA MULTIDIMENSI BERBASIS JAVA PADA PostgreSQL

## ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

3%

PUBLICATIONS

4%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://repo.unand.ac.id">repo.unand.ac.id</a> Internet Source	1%
2	<a href="http://www.scribd.com">www.scribd.com</a> Internet Source	1%
3	Submitted to Open University of Mauritius Student Paper	1%
4	Submitted to Southampton Solent University Student Paper	1%
5	Submitted to University of Technology, Sydney Student Paper	1%
6	<a href="http://doku.pub">doku.pub</a> Internet Source	1%
7	Submitted to Help University College Student Paper	1%
8	<a href="http://jurnal.iaii.or.id">jurnal.iaii.or.id</a> Internet Source	1%

University of Tennessee, Knoxville



9

Publication

<1 %

10

[pdfs.semanticscholar.org](https://pdfs.semanticscholar.org)

Internet Source

<1 %

11

Kresno Budi Nugroho. "Pengembangan Data Warehouse Penerimaan Mahasiswa Baru Untuk Informasi Strategik Pada Universitas BSI", Jurnal Kajian Ilmiah, 2018

Publication

<1 %

12

[cnjavaclub.com](http://cnjavaclub.com)

Internet Source

<1 %

13

[smnggeophysics.com](http://smnggeophysics.com)

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off