Building Data Warehouse and Dashboard of Church Congregation Data

Ragil Yoga Irawan[1], Budi Susanto[2], Yuan Lukito[3]
Informatics Department, Duta Wacana Christian University
Jl. Dr. Wahidin Sudirohusodo No. 5 - 25, Yogyakarta, Indonesia
[1]ragil.yoga@ti.ukdw.ac.id
[2]budsus@ti.ukdw.ac.id
[3]yuanlukito@ti.ukdw.ac.id

***Abstract*— A data warehouse is essential for an organization to process and analyze data coming from the organization. Hence, a data warehouse together with a dashboard to visualize the processed data are built to accommodate the need of the church administrator to analyze a large set of church congregation data. The data warehouse is built using the Kimball principle. This Kimball principle emphasizes the implementation of a dimensional model in the data warehouse, not a relational model used in a regular transactional database. An ETL process that contains extract, transform and load processes is used to retrieve all data from the regular transactional database and transform the data so the data can be loaded into the data warehouse. A dashboard is then built to visualize the data from the data warehouse so the users can view the processed data easily. Users can also export the processed data into an excel file that can be downloaded from the dashboard. A web service is built to get data from the data warehouse and return it to the dashboard.**

Keywords— data warehouse, Kimball principle, church congregation, etl, dimensional model, dashboard, web service

I. Introduction

GKI (Gereja Kristen Indonesia/The Indonesian Christian Church) is a group consisting of Protestant Christian churches in Indonesia headquartered in Jakarta, Indonesia [1]. GKI Sinode Wilayah Jawa Tengah or Jawa Tengah Regional Synod of GKI is a smaller group of GKI located in Jawa Tengah province of Indonesia headquartered in Magelang, Jawa Tengah, Indonesia. With a large size of data being generated consistently from all churches in Jawa Tengah Regional Synod of GKI, it raises the need to process and analyze those data, therefore, the church administrators can comprehend the data related to the states of their churches.

A data warehouse is chosen to store the data generated from the churches because of its ability to process and analyze a large size of data. Inmon [2] defined a data warehouse as a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions. A data warehouse is specially designed to store, filter, retrieve, and analyze very large collections of data [3].

Kimball principle is used to build a data warehouse [4]. Kimball principle emphasizes the use of dimensional modeling in the data warehouse. Data from a relational database will be loaded into the data warehouse using an ETL (Extract, Transform, Load) process. A web service will then be built to retrieve data from the data warehouse. The retrieved data will then be visualized through a dashboard. Users can view and analyze the processed data using this dashboard. The processed data can also be exported into an excel file using this dashboard.

Höpken, Fuchs, Höll, Keil, and Lexhagen [5] researched how to utilize abundant tourism destination data of Sweden that is unused. In this research, a DMIS (Destination Management Information System) is built using the implementation of business intelligence methods, data warehouse, and OLAP (Online Analytical Processing) functions. It resulted in the success of demonstrating the effectiveness of the multidimensional modeling paradigm in support of robust and flexible OLAP analysis.

Mali [6] researched how to use a data warehouse built using the Kimball method to analyze tennis players' performance. This research examined whether temperature, wind speed, court type, and court surface affect tennis players' performance. The data used in this research was the data of tennis tournament statistics for 2014.

The goal of this research is to build a data warehouse and a dashboard of church congregation data of Jawa Tengah Regional Synod of GKI. This research is expected to benefit the users, in this case, the church administrators, in the form of helping them analyze church congregation data and make an annual report in a form of PURJ (Perlawatan Umum Rutin Jemaat/Routine General Visitation of Church Congregation) document.

II. Theoretical Framework

Below are explanations of theories and methods used in this research.

A. Jawa Tengah Regional Synod of GKI

GKI Sinode Wilayah Jawa Tengah or Jawa Tengah Regional Synod of GKI is one of many synods of GKI located in the Jawa Tengah province of Indonesia headquartered in Magelang, Jawa Tengah, Indonesia. It consists of eight presbyteries, each of which has its churches.

B. Data Warehouse

Inmon [2] defined a data warehouse as a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions. It is subject-oriented because of its ability to be used to analyze a particular subject area, integrated because of its ability to

DOI 10.21460/jutei.2019.32.183

integrates data from multiple data sources, non-volatile because of the data inside a data warehouse will not change once the data is loaded into it, and time-variant because of its ability to keep historical data.

C. Kimball Principle

Kimball principle is a principle proposed by Kimball that is used to build a data warehouse. Kimball principle emphasizes the use of dimensional modeling in the data warehouse [7]. Figure 1 below is the data warehouse architecture proposed by Kimball.

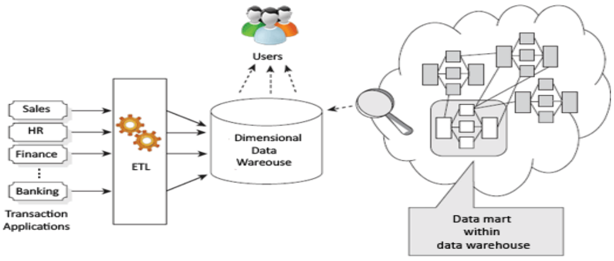


Figure 1. Data warehouse architecture by Kimball (http://www.zentut.com/data-warehouse/ralph-kimball-data-warehouse-architecture/))

D. Dimensional Model

Dimensional modeling is widely accepted as the preferred technique for presenting analytic data because it can deliver data that's understandable to the business users and can deliver fast query performance [7]. Dimensional model is also chosen to build a data warehouse in this research because unlike the relational model in which the goals are to do transactional activities such as adding, updating, and deleting data, the main goal of the dimensional model is to read data stored in a data warehouse. A dimensional model stores data in a denormalized form, or at least partly denormalized form, while a relational model stores data in a normalized form to avoid redundancy [8].

Dimensional model implementation in a data warehouse can be achieved using a schema called star schema. This schema is called star schema because of its shape that resembles a star as depicted in figure 2 below.

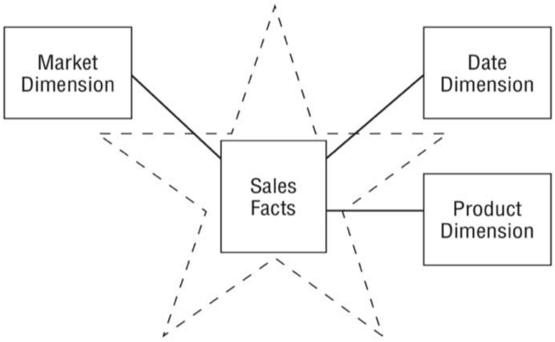


Figure 2. Star Schema [7]

A star schema mainly consists of dimensional tables, hence, is called a dimensional model, and fact tables. A dimensional table is a table consisting of dimensions/descriptions of data composing a fact table. A fact table is a table consisting of facts related to a certain business process. Besides star schema, there is another schema called snowflake schema because of its shape that resembles a snowflake as depicted in figure 3 below.

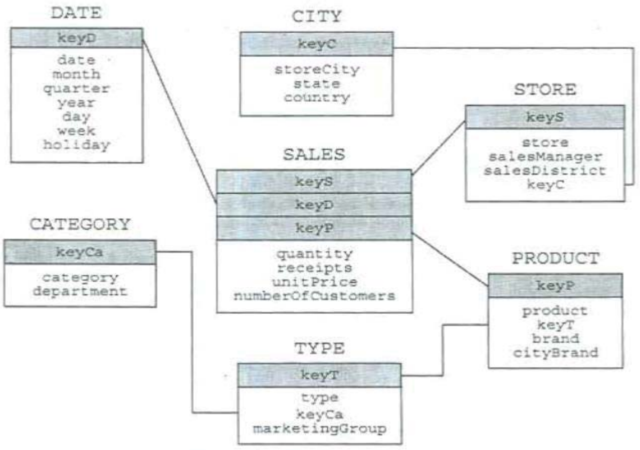


Figure 3. Snowflake schema

A snowflake schema is derived from a star schema by dividing one or some dimension tables into some smaller dimension table, therefore the dimension tables in this schema are partly normalized.

E. ETL Process

An ETL (Extract, Transform, Load) process is used to load church congregation data from the source database into the data warehouse. ETL processes are responsible for the extraction of the appropriate data from the sources, their transportation to a special-purpose area of the data warehouse where they will be processed, the transformation of the source data and the computation of new values (and, possibly records) in order to obey the structure of the data warehouse relation to which they are targeted, the isolation and cleansing of problematic tuples, in order to guarantee that business rules and databases constraints are respected and the loading of the cleansed, transformed data to the appropriate relation in the warehouse, along with the refreshment of its accompanying indexes and materialized views [9].

F. Dashboard

A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance [10]. The data stored in the data warehouse will be processed into useful information that will be visualized in a dashboard in forms of tables, diagrams, and graphs. Figure 4 below depicts an example of a dashboard.

G. Greenplum Database

According to Pivotal Software, Inc. [11], Greenplum Database is a massively parallel processing (MPP) database server with an architecture specially designed to manage large-scale analytic data warehouses and business intelligence workloads. MPP (also known as a shared-nothing architecture) refers to systems with two or more processors that cooperate to carry out an operation, each processor with its own memory, operating system and disks [11]. Greenplum Database is chosen in this research because of its ability to handle and process a large quantity of data.

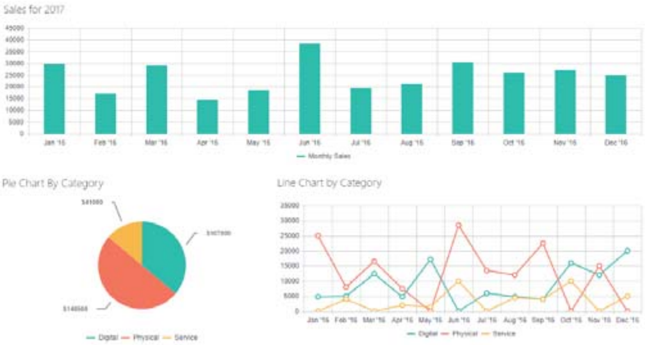


Figure 4. A dashboard example (https://plumsail.com/sharepoint-dashboard-designer/documentation/build-simple-charts-sales-report-sharepoint/)

H. *Pentaho Data Integration*

Pentaho Data Integration (PDI) provides the Extract, Transform, and Load (ETL) capabilities that facilitate the process of capturing, cleansing, and storing data using a uniform and consistent format that is accessible and relevant to end-users and IoT technologies [12]. PDI, which also referred to as Kettle, provides a set of tools consisting of Spoon, Pan, Kitchen, and Carte. Spoon is the graphical interface used to create transformation and jobs. Pan is a batch-style command-line tool used to run transformations. Kitchen is a batch-style command-line tool used to run jobs. Carte is a web server that can be used to run jobs on remote servers. A transformation is a set of ETL processes that consists of retrieving data from the source databases, transforming the data, and loading the data into the target data storage. A job is a set of transformations.

I. *Web Service*

A web service is a software system designed to support interoperable machine-to-machine interaction over a network [13]. A web service is designed to communicate with another machine/application, unlike a regular web application that is designed to communicate directly with the users. A web service can return data in many formats such as XML (Extensible Markup Language) and JSON (JavaScript Object Notation). The web service built in this research is a RESTful web service, a web service that is based on REST (REpresentational State Transfer). A REST is an architectural style for networked hypermedia applications, primarily used to build web services that are lightweight, maintainable, and scalable [14]. The protocols used in a RESTful web service are the HTTP (HyperText Transfer Protocol) protocols such as GET, POST, PUT, and DELETE.

J. *PURJ Document*

PURJ (Perlawatan Umum Rutin Jemaat/Routine General Visitation of Church Congregation) document is an official annual church document consisting of church and the congregation data. This document is made to help the church administrators to analyze the church congregation data produced in one year.

III. Research Methodology

A. *System Requirements*

The system of data warehouse and dashboard must meet the non-functional requirements as explained below.

1. The ETL process can be executed automatically without having the users to execute it manually.
2. The dashboard can be accessed using most web browsers.
3. The dashboard is using services provided by Auth0 to handle the user's authentication process.

The system of data warehouse and dashboard must meet the non-functional requirements as explained below.

1. The system can execute an ETL process from the source database into the data warehouse.
2. The system can display the data about a church in one year in the form of a table.
3. Users can add or edit data about visi (vision), misi (mission), and strategi pencapaian (achievement strategy) of the church.
4. The system can display the general data of the church congregation in one year in the form of a table.
5. The system can display the data of the church congregation on an age category, education category, job category, and ethnic group category in one year in the form of a table and a graph.
6. The system can display the data of the average attendance number in a church service and activity in one year in the form of a table and a graph.
7. The system can display the data of the amount of the church administrators and the church reverends in one year in the form of a table and a graph.
8. Users can download all of the data displayed in the dashboard in the form of an excel file called PURJ document.

B. *Research Materials*

The data used in this research is a sample data of the church congregation in the Jawa Tengah Regional Synod of GKI. This sample data contains information relevant to the church congregation and the church itself such as information about name, address, date of birth, family, and educational background of a church congregation. The data also contains information about the church's profile, information about the fields, the commissions or the teams, services' schedules, reverends, administrators of the church, and also information about events such as attestations, marriages, births, deaths, and baptism. Figure 5 below depicts the ERD of the source database.

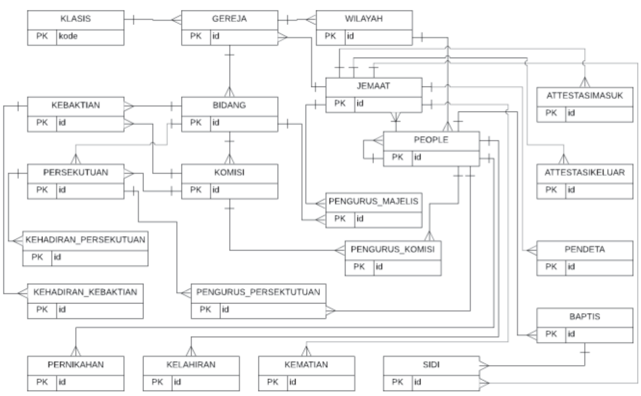


Figure 5. The source database's ERD

C. System Design

1) Flow Diagram: Figure 6 below depicts the flow diagram of the system.

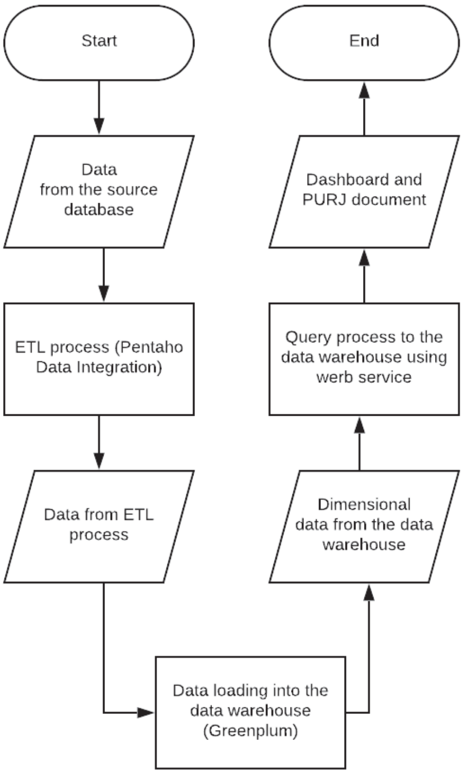

Start
End
Data from the source database
Dashboard and PURJ document
ETL process (Pentaho Data Integration)
Query process to the data warehouse using werb service
Data from ETL process
Dimensional data from the data warehouse
Data loading into the data warehouse (Greenplum)

Figure 6. System flow diagram

First, the data will be retrieved from the source database and will be transformed and loaded into the data warehouse by using the ETL process. Then, the data from the data warehouse will be visualized through a dashboard in the forms of tables and graphs. The data visualized in the dashboard is extracted from the data warehouse by using a RESTful web service.

2) Use Case Diagram: Figure 7 below depicts the use case diagram of the system.

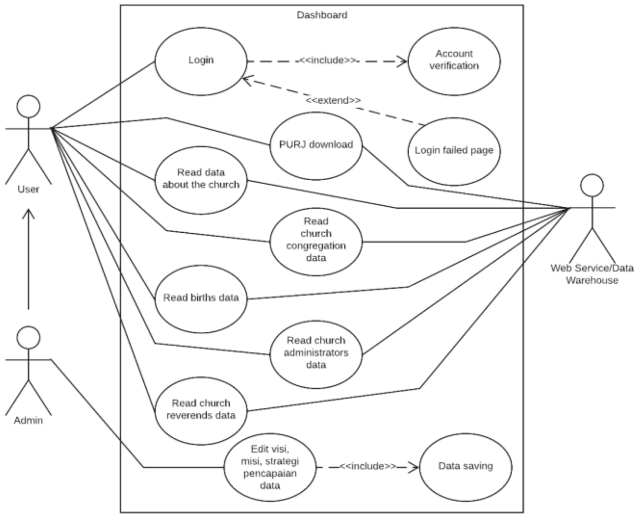

Dashboard
Login
<<include>>
Account verification
<<extend>>
PURJ download
Login failed page
Read data about the church
Read church congregation data
Read births data
Read church administrators data
Read church reverends data
Edit visi, misi, strategi pencapaian data
<<include>>
Data saving
User
Admin
Web Service/Data Warehouse

Figure 7. Use case diagram

There are two types of user in this system, an administrator and a regular user. A regular user can access some functions such as login, PURJ download, view data about the church, view data about the church congregation, view data about the attendance number, view data about the church administrators and view data about the church reverends. Meanwhile, an administrator can access all functions a regular user can access with the additions of being able to edit visi (vision), misi (mission), and strategi pencapaian (achievement strategy) data.

The login process will verify and authenticate login attempts and display a failed login page if the verification and authentication process failed. The edit data about visi (vision), misi (mission), and strategi pencapaian (achievement strategy) process will save the edited data into a database in the system. The data warehouse will provide the data for the PURJ download process, view data about the church process, view data about the church congregation process, view data about the attendance number process, view data about the church administrators process, and view data about the church reverends process.

3) System Architecture: Figure 8 below depicts the system architecture of the data warehouse and the dashboard.

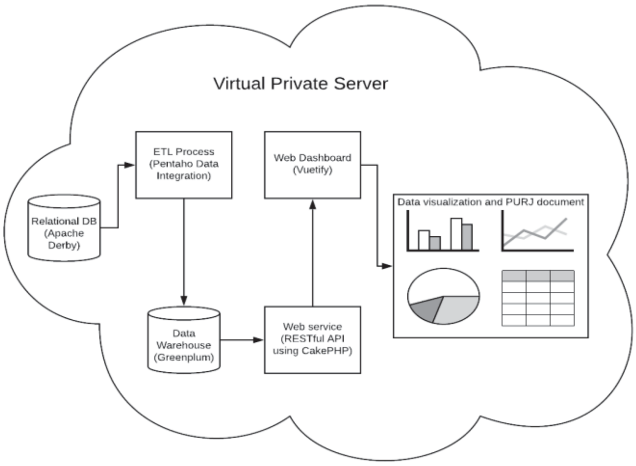

Virtual Private Server
ETL Process (Pentaho Data Integration)
Web Dashboard (Vuetify)
Relational DB (Apache Derby)
Data visualization and PURJ document
Data Warehouse (Greenplum)
Web service (RESTful API using CakePHP)

Figure 8. System architecture

The ETL process will retrieve data from a relational database built with Apache Derby. Through the ETL process, relational data from the source database will be transformed into dimensional data and loaded into the data warehouse built with Greenplum Database. A RESTful web service is built using CakePHP, a PHP framework, to facilitate the retrieval of data from the data warehouse using SQL queries. Finally, a dashboard built using VueJS with Vuetify, a javascript framework with a material design will provide the data returned from the web service to the users and visualize it into tables and graphs. A service from Auth0 will be used to handle the authentication mechanism of the dashboard.

4) ERD (Entity Relationship Diagram): Figure 9 below depicts the ERD of the data warehouse but only with its primary keys.

The ERD of the data warehouse uses snowflake schema, that has a fact table (fact_kehadiran) and some dimensional tables (dim_gereja, dim_kebaktian, dim_kegiatan, and dim_waktu) that provide details to the fact table. Some of the dimensional tables also have their dimensional tables, to provide details to the dimensional table itself. This ERD model conforms to the Kimball principle for building a data warehouse, where a dimensional model is used for the data inside the data warehouse [7]. A more detailed ERD of the data warehouse with each of its attributes is depicted in figure 10 below.

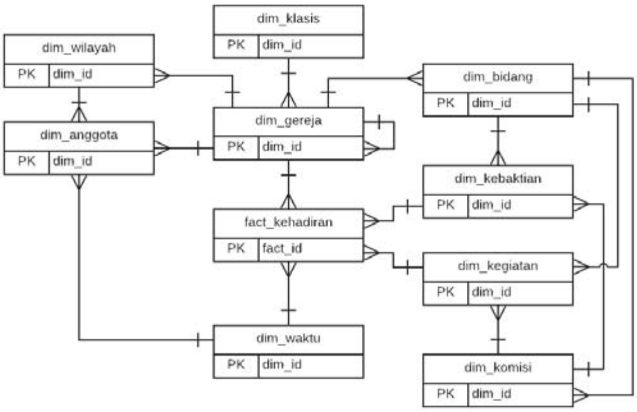


Figure 9. Simple data warehouse ERD

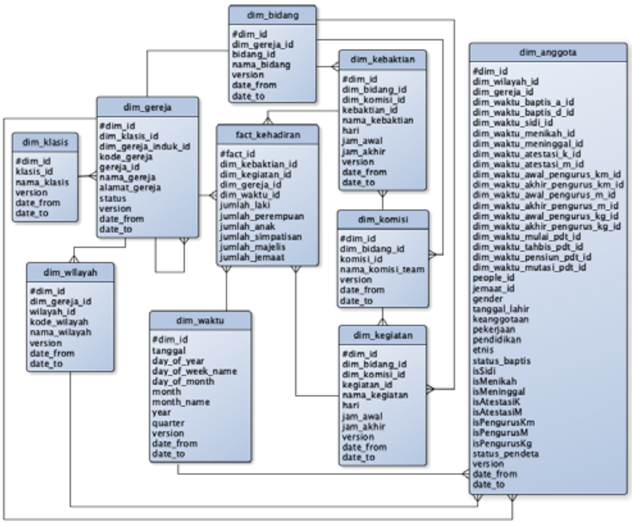


Figure 10. More detailed data warehouse ERD

There is also a regular relational database built using MariaDB alongside the data warehouse. This database is built for storing data about users, visi (vision), misi (mission), and strategi pencapaian (achievement strategy). The ERD of the said database is depicted in figure 11 below.

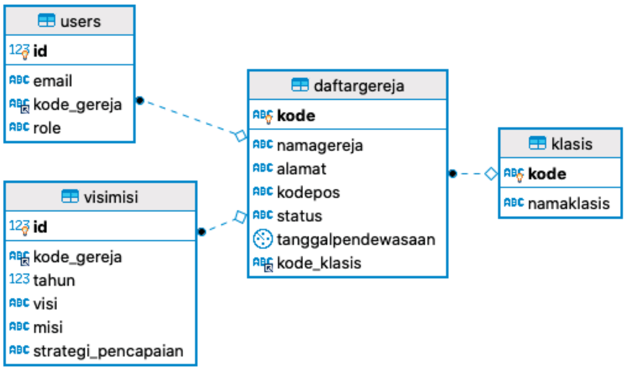


Figure 11. Database ERD in the system to store data about visi (vision), misi (mission), and strategi pencapaian (achievement strategy)

IV. Results and Discussion

The first and foremost thing to do is to make the tables in the data warehouse using the Kimball principle. As explained before, the Kimball principle is used to build a data warehouse where the data inside the data warehouse is stored in a dimensional model [7]. Thus, there will be a fact table and dimensional tables in the data warehouse to store the data from the source database after running an SQL script containing create dimensional tables and fact table commands.

Then, ETL processes will be executed to load the data from the source database to the data warehouse. Each of the tables on the data warehouse will have its transformation process to load data from the corresponding tables in the source database. Figure 12 below is one of those transformation processes, this transformation process belongs to dim_gereja table on the data warehouse.

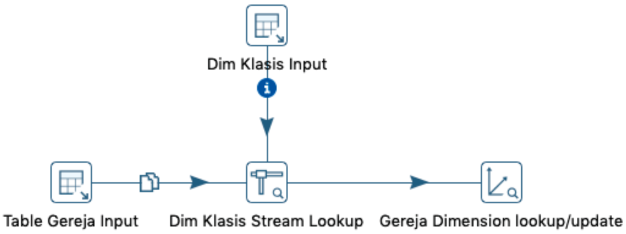
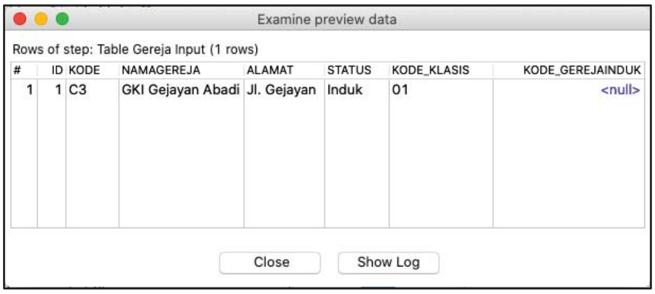


Figure 12. Dim_gereja transformation

The "Table Gereja Input" process will retrieve data from gereja table in the source database. The data retrieved from gereja table in the source database is depicted in figure 13 below.

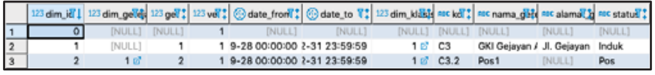

Examine preview data

Rows of step: Table Gereja Input (1 rows)

#	ID	KODE	NAMAGEREJA	ALAMAT	STATUS	KODE_KLASIS	KODE_GEREJAINDUK
1	1	C3	GKI Gejayan Abadi	Jl. Gejayan	Induk	01	<null>

Figure 13. Data from gereja table

The next process is to do look up with the data from dim_klasis table in the data warehouse using the "Dim Klasis Stream Lookup" process. The purpose of this process is to retrieve the dimension key from dim_klasis table in the warehouse that matches the reference key of the klasis data from the data of the gereja table in the source database. Then, the "Dimension lookup/table" process will load the data from the previous process to the dim_gereja table in the data warehouse. The data loaded into the dim_gereja table in the data warehouse is depicted in figure 14 below.



	dim_id	dim_ge	ge	ve	date_from	date_to	dim_klasi	kd	nama_g	alama	statu
1	0	[NULL]	[NULL]	1	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
2	1	[NULL]	1	1	9-28 00:00:00	2-31 23:59:59	1	C3	GKI Gejayan /	Jl. Gejayan	Induk
3	2	1	2	1	9-28 00:00:00	2-31 23:59:59	1	C3.2	Pos1	[NULL]	Pos

Figure 14. Data loaded into dim_gereja table

For the fact table in the data warehouse (fact_kehadiran), it requires two transformation processes, because it has two source tables from the source database. Figure 15 and 16 below depicts one of the transformation processes of fact_kehadiran table in the data warehouse from the kehadiran_kebaktian table in the source database.

The "Kehadiran Kebaktian Table Input" process will retrieve data from kehadiran_kebaktian table in the source database. The data from kehadiran_kebaktian table in the source database is depicted in figure 17 below.

The next process is to do look up with the data from dim_klasis and dim_gereja table in the data warehouse to retrieve the dimension key of dim_klasis and dim_gereja table in the data warehouse using "Dim_Klasis Stream Lookup" and "Dim Gereja Stream Lookup" process. The "Filter Bidang or Komisi" process will separate the types of kehadiran data retrieved from the source database. If it's a bidang type, then a look up will be executed with dim_bidang table in the data warehouse. If it's a komisi type, then a look up will be executed with dim_komisi table in the data warehouse. This process is intended to retrieve the dimension key according to the type of kehadiran data from the source table. Then, a look up process will be executed again with the dim_kebaktian, dim_gereja, and dim_waktu table to get the dimension key of the dim_kebaktian, dim_gereja, and dim_waktu. The last process is to load the data up until the last process to the fact_kehadiran table in the data warehouse. Figure 18 depicts the data loaded to the fact_kehadiran table in the data warehouse.

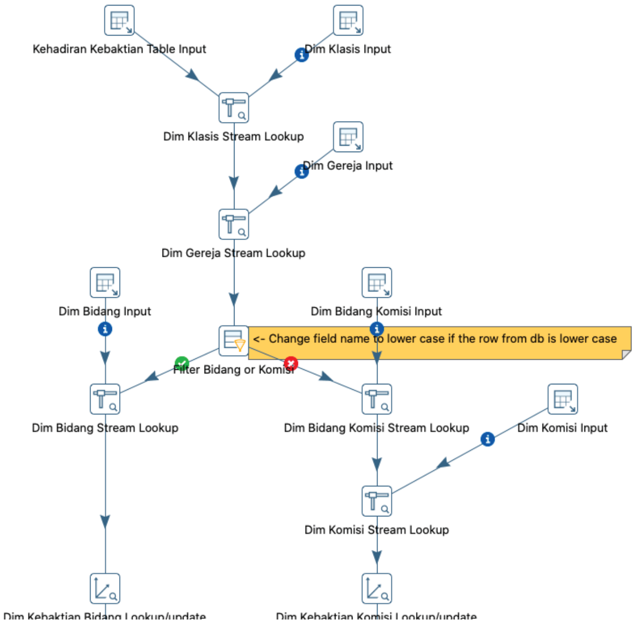


Figure 15. Fact_kehadiran transformation

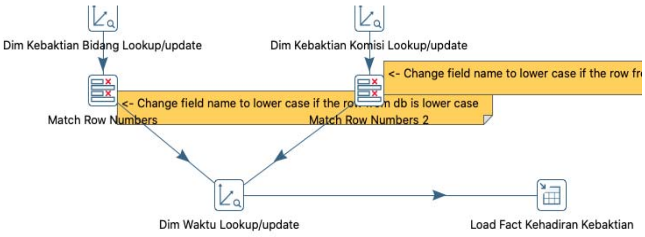
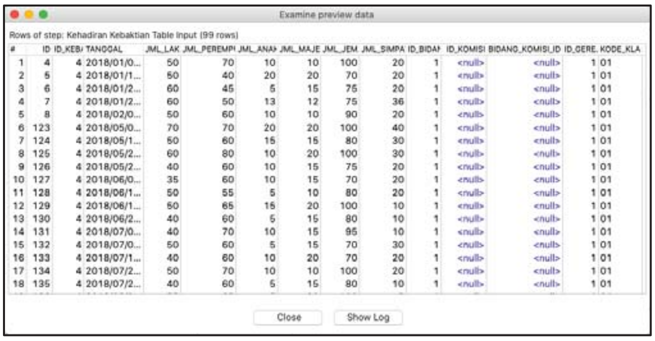

Dim Kebaktian Bidang Lookup/update
Dim Kebaktian Komisi Lookup/update
<- Change field name to lower case if the row fr
<- Change field name to lower case if the row from db is lower case
Match Row Numbers
Match Row Numbers 2
Dim Waktu Lookup/update
Load Fact Kehadiran Kebaktian

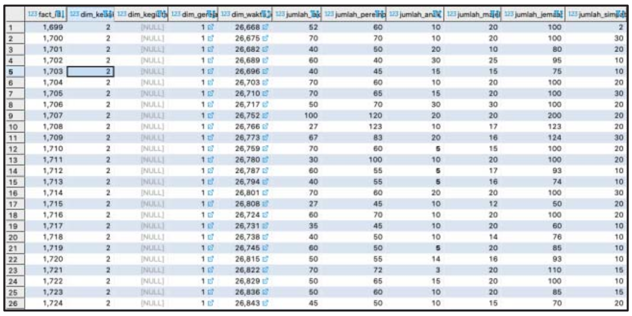
Figure 16. Fact_kehadiran transformation (continued)



Rows of step: Kehadiran Kebaktian Table Input (99 rows)

#	ID	ID_KEB/	TANGGAL	JML_LAK	JML_PEREMPI	JML_ANAK	JML_MAJE	JML_JEM	JML_SIMPA	ID_BIDAN	ID_KOMISI	BIDANG_KOMISI_ID	ID_GERE.	KODE_KLA
1	4	4	2018/01/0...	50	70	10	10	100	20	1	<null>	<null>	1	01
2	5	4	2018/01/1...	50	40	20	20	70	20	1	<null>	<null>	1	01
3	6	4	2018/01/2...	60	45	5	15	75	20	1	<null>	<null>	1	01
4	7	4	2018/01/2...	60	50	13	12	75	36	1	<null>	<null>	1	01
5	8	4	2018/02/0...	50	60	10	10	90	20	1	<null>	<null>	1	01
6	123	4	2018/05/0...	70	70	20	20	100	40	1	<null>	<null>	1	01
7	124	4	2018/05/1...	50	60	15	15	80	30	1	<null>	<null>	1	01
8	125	4	2018/05/2...	60	80	10	20	100	30	1	<null>	<null>	1	01
9	126	4	2018/05/2...	40	60	10	15	75	20	1	<null>	<null>	1	01
10	127	4	2018/06/0...	35	60	10	15	70	20	1	<null>	<null>	1	01
11	128	4	2018/06/1...	50	55	5	10	80	20	1	<null>	<null>	1	01
12	129	4	2018/06/1...	50	65	15	20	100	10	1	<null>	<null>	1	01
13	130	4	2018/06/2...	40	60	5	15	80	10	1	<null>	<null>	1	01
14	131	4	2018/07/0...	40	70	10	15	95	10	1	<null>	<null>	1	01
15	132	4	2018/07/0...	50	60	5	15	70	30	1	<null>	<null>	1	01
16	133	4	2018/07/1...	40	60	10	20	70	20	1	<null>	<null>	1	01
17	134	4	2018/07/2...	50	70	10	10	100	20	1	<null>	<null>	1	01
18	135	4	2018/07/2...	40	60	5	15	80	10	1	<null>	<null>	1	01

Figure 17. Data from kehadiran_kebaktian table



	fact_i	dim_keba	dim_kegiata	dim_gereja	dim_waktu	jumlah_lak	jumlah_peremp	jumlah_anak	jumlah_maje	jumlah_jemaat	jumlah_simpa
1	1,699	2	[NULL]	1	26,668	52	60	10	20	100	2
2	1,700	2	[NULL]	1	26,675	70	70	10	20	100	30
3	1,701	2	[NULL]	1	26,682	40	50	20	10	80	20
4	1,702	2	[NULL]	1	26,689	60	40	30	25	95	10
5	1,703	2	[NULL]	1	26,696	40	45	15	15	75	10
6	1,704	2	[NULL]	1	26,703	70	60	10	20	100	20
7	1,705	2	[NULL]	1	26,710	70	65	15	20	100	30
8	1,706	2	[NULL]	1	26,717	50	70	30	30	100	20
9	1,707	2	[NULL]	1	26,752	100	120	20	20	200	20
10	1,708	2	[NULL]	1	26,766	27	123	10	17	123	20
11	1,709	2	[NULL]	1	26,773	67	83	20	16	124	30
12	1,710	2	[NULL]	1	26,759	70	60	5	15	100	20
13	1,711	2	[NULL]	1	26,780	30	100	10	20	100	20
14	1,712	2	[NULL]	1	26,787	60	55	5	17	93	10
15	1,713	2	[NULL]	1	26,794	40	55	5	16	74	10
16	1,714	2	[NULL]	1	26,801	70	60	20	20	100	30
17	1,715	2	[NULL]	1	26,808	27	45	10	12	50	20
18	1,716	2	[NULL]	1	26,724	60	70	10	20	100	20
19	1,717	2	[NULL]	1	26,731	35	45	10	20	60	10
20	1,718	2	[NULL]	1	26,738	40	50	10	14	76	10
21	1,719	2	[NULL]	1	26,745	60	50	5	20	85	10
22	1,720	2	[NULL]	1	26,815	50	55	14	16	93	10
23	1,721	2	[NULL]	1	26,822	70	72	3	20	110	15
24	1,722	2	[NULL]	1	26,829	50	65	15	20	100	10
25	1,723	2	[NULL]	1	26,836	50	60	10	20	85	15
26	1,724	2	[NULL]	1	26,843	45	50	10	15	70	20

Figure 18. Data loaded into fact_kehadiran table

A job is then created to run all transformation processes of all the tables in the data warehouse sequentially. This process begins with setting the connection attributes of the source database and then using those attributes to connect with the said database. After the connections are successfully established, all of the transformation processes of all the tables in the data warehouse will be executed sequentially. Figure 19 below depicts the job explained before.

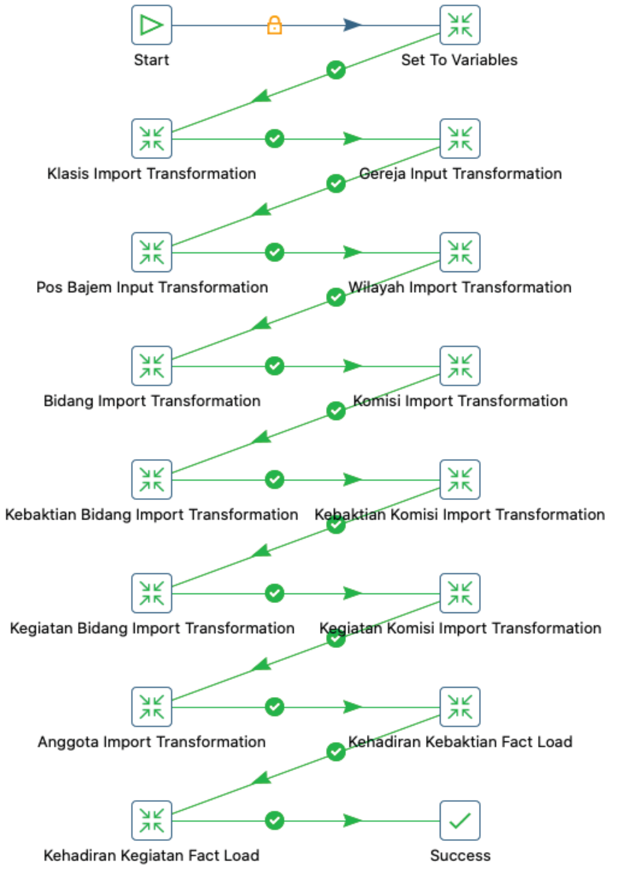

Start
Set To Variables
Klasis Import Transformation
Gereja Input Transformation
Pos Bajem Input Transformation
Wilayah Import Transformation
Bidang Import Transformation
Komisi Import Transformation
Kebaktian Bidang Import Transformation
Kebaktian Komisi Import Transformation
Kegiatan Bidang Import Transformation
Kegiatan Komisi Import Transformation
Anggota Import Transformation
Kehadiran Kebaktian Fact Load
Kehadiran Kegiatan Fact Load
Success

Figure 19. Job to run all transformations

Lastly, one more job is created to run the complete ETL process. Figure 20 depicts the job to run the complete ETL process.

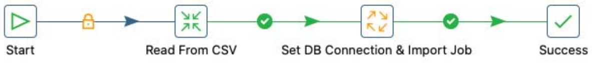


Figure 20. Job to run the complete ETL process

This job starts with reading a CSV file containing the source database connections attributes, such as hostnames, database names, usernames, and passwords as depicted below.

```
host,database,username,password
localhost,siswagki,siswa,7f458f88250999c5239382589b94e3
88fe39bb1ab7fdf255f92b636e1198793dd1f256a81b4a40e68
ae6ac5853ca3bcc5da25167969241cc9788babd2e42dff9
```

After getting those attributes, this job will run the job in figure 19. After the job is successfully executed, the whole ETL process is done. The job in figure 20 can be executed automatically without having the user manually executing it using a tool provided by Pentaho Data Integrator called kitchen. The users can run the job in figure 20 using below syntax.

```
./kitchen.sh -
file:/Users/snowvax/Documents/Work/dw/etl_siswagki/jobs/co
mplete_import_job
```

After all the ETL processes are done, a RESTful web service is built using CakePHP, a PHP framework, with a purpose to communicate directly with the data warehouse and the database containing data about visi (vision), misi (mission), and strategi pencapaian (achievement strategy). The source code below is the connections attributes in the web service to connect to the data warehouse and the said database.

```
    'default' => [
        'className' => Connection::class,
        'driver' => Postgres::class,
        'persistent' => false,
        'host' => 'gpdbcentos7',
        'port' => '5432',
        'username' => 'gpadmin',
        'password' => 'pivotal',
        'database' => 'dwsiswagki',
        'timezone' => 'UTC',
        'flags' => [],
        'cacheMetadata' => true,
        'log' => false,
        'quoteIdentifiers' => false,
        'url' => env('DATABASE_URL', null),
    ],
    'mysqlconn' => [
        'className' => Connection::class,
        'driver' => Mysql::class,
        'persistent' => false,
        'host' => 'gpdbcentos7',
        'username' => 'admin',
        'password' => 'starbear',
        'database' => 'siswa',
        'timezone' => 'UTC',
        'flags' => [],
        'cacheMetadata' => true,
        'log' => false,
        'quoteIdentifiers' => false,
        'url' => env('DATABASE_URL', null),
    ]
```

This web service accepts HTTP request methods like GET, POST, PUT, and DELETE and then returns a JSON data according to the request method it gets. Below is the example of HTTP GET request that will return the data about the church.

```
http://localhost:8765/dimgereja/get-all-tentang-
gereja.json?kode_gereja=[kode gereja]&tahun=[tahun]
```

Then, a dashboard is also built using Vue.js, a progressive javascript framework, accompanied by Vuetify, a material design library to serve the purpose of providing the users with visualized data retrieved from the data warehouse by sending HTTP requests to the RESTful web service explained before. Figure 21 below is the login page for the dashboard, built using the service provided by Auth0.

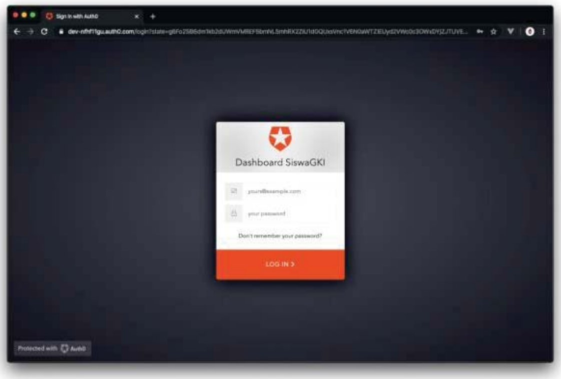
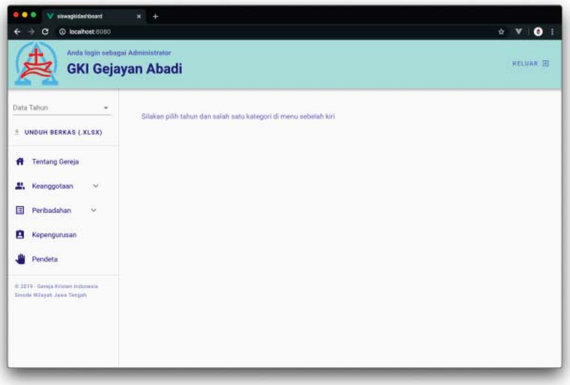


Figure 21. Dashboard login page

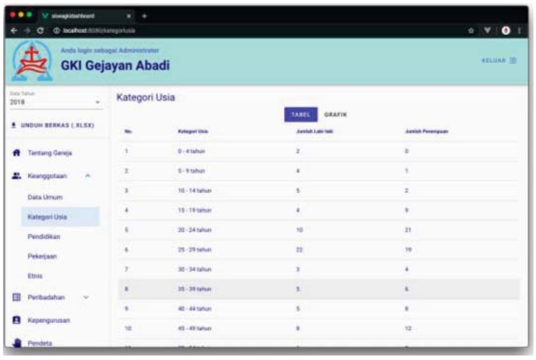
Figure 22 below is the main page of the dashboard after the user has been successfully logged in and when the user has not chosen any year and menu.



Silakan pilih tahun dan salah satu kategori di menu sebelah kiri

Figure 22. Dashboard main page

Figure 23 below is one example of the dashboard page containing visualized data from the data warehouse in a table form.



Kategori Usia

No.	Kategori Usia	Jumlah Laki-laki	Jumlah Perempuan
1	0 - 4 tahun	2	0
2	5 - 9 tahun	4	1
3	10 - 14 tahun	5	2
4	15 - 19 tahun	4	9
5	20 - 24 tahun	10	21
6	25 - 29 tahun	22	19
7	30 - 34 tahun	3	4
8	35 - 39 tahun	5	6
9	40 - 44 tahun	5	8
10	45 - 49 tahun	8	12

Figure 23. Data visualization in table form in the dashboard

Figure 24 below is one example of the dashboard page containing visualized data from the data warehouse in a graph form.

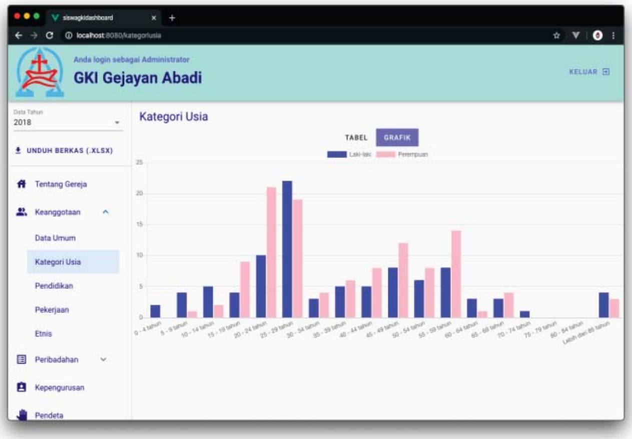
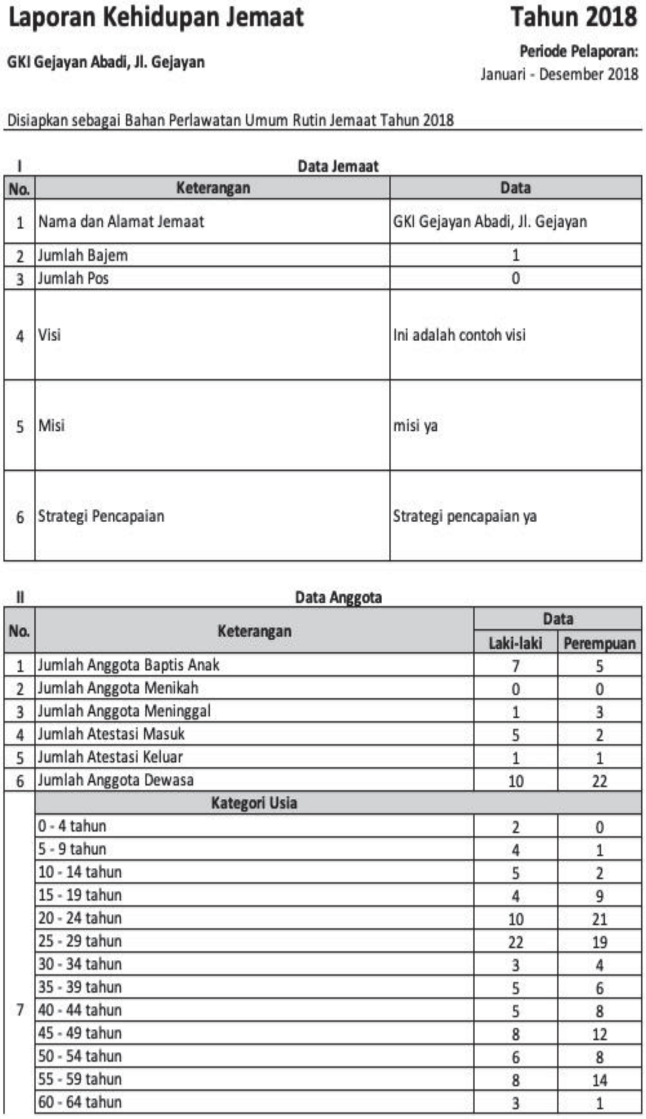


Figure 24. Data visualization in graph form in the dashboard

Users can also export the data visualized in this dashboard into an excel file in the form of a PURJ document. Figure 25, 26, and 27 below is the PURJ document downloaded from the dashboard.



Laporan Kehidupan Jemaat **Tahun 2018**

GKI Gejayan Abadi, Jl. Gejayan

Periode Pelaporan:
Januari - Desember 2018

Disiapkan sebagai Bahan Perlawatan Umum Rutin Jemaat Tahun 2018

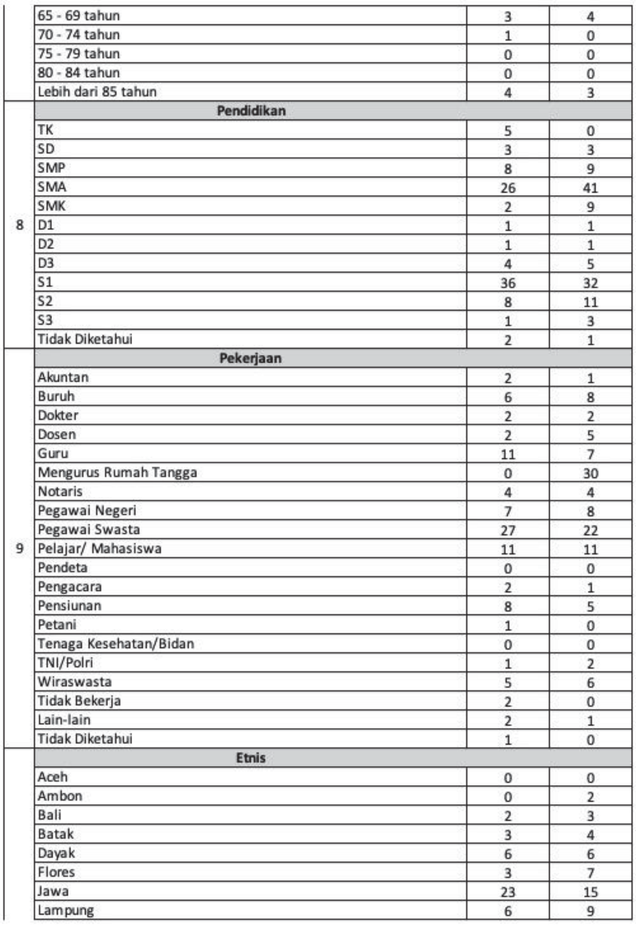
I **Data Jemaat**

No.	Keterangan	Data
1	Nama dan Alamat Jemaat	GKI Gejayan Abadi, Jl. Gejayan
2	Jumlah Bajem	1
3	Jumlah Pos	0
4	Visi	Ini adalah contoh visi
5	Misi	misi ya
6	Strategi Pencapaian	Strategi pencapaian ya

II **Data Anggota**

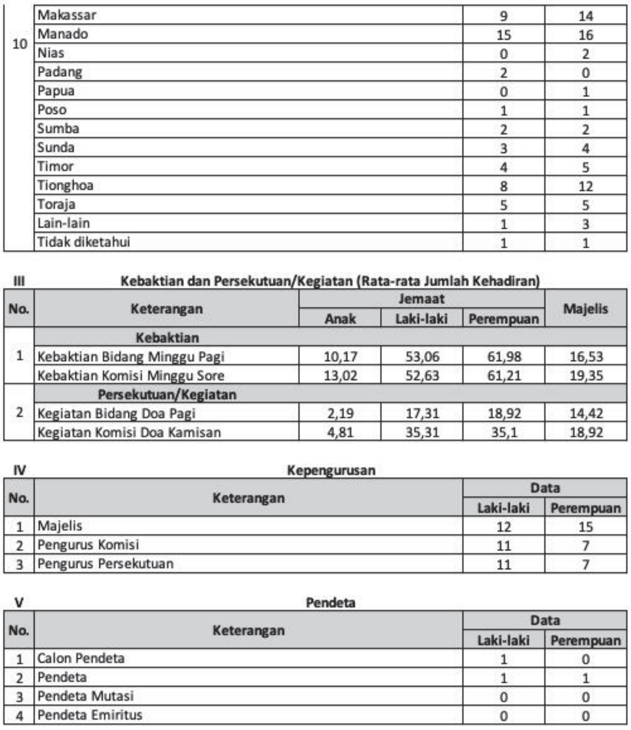
No.	Keterangan	Data	
		Laki-laki	Perempuan
1	Jumlah Anggota Baptis Anak	7	5
2	Jumlah Anggota Menikah	0	0
3	Jumlah Anggota Meninggal	1	3
4	Jumlah Atestasi Masuk	5	2
5	Jumlah Atestasi Keluar	1	1
6	Jumlah Anggota Dewasa	10	22
	Kategori Usia		
7	0 - 4 tahun	2	0
	5 - 9 tahun	4	1
	10 - 14 tahun	5	2
	15 - 19 tahun	4	9
	20 - 24 tahun	10	21
	25 - 29 tahun	22	19
	30 - 34 tahun	3	4
	35 - 39 tahun	5	6
	40 - 44 tahun	5	8
	45 - 49 tahun	8	12
	50 - 54 tahun	6	8
	55 - 59 tahun	8	14
	60 - 64 tahun	3	1

Figure 25. PURJ document downloaded from the dashboard



	65 - 69 tahun	3	4
	70 - 74 tahun	1	0
	75 - 79 tahun	0	0
	80 - 84 tahun	0	0
	Lebih dari 85 tahun	4	3
	Pendidikan		
8	TK	5	0
	SD	3	3
	SMP	8	9
	SMA	26	41
	SMK	2	9
	D1	1	1
	D2	1	1
	D3	4	5
	S1	36	32
	S2	8	11
	S3	1	3
	Tidak Diketahui	2	1
	Pekerjaan		
9	Akuntan	2	1
	Buruh	6	8
	Dokter	2	2
	Dosen	2	5
	Guru	11	7
	Mengurus Rumah Tangga	0	30
	Notaris	4	4
	Pegawai Negeri	7	8
	Pegawai Swasta	27	22
	Pelajar/ Mahasiswa	11	11
	Pendeta	0	0
	Pengacara	2	1
	Pensiunan	8	5
	Petani	1	0
	Tenaga Kesehatan/Bidan	0	0
	TNI/Polri	1	2
	Wiraswasta	5	6
	Tidak Bekerja	2	0
	Lain-lain	2	1
	Tidak Diketahui	1	0
	Etnis		
	Aceh	0	0
	Ambon	0	2
	Bali	2	3
	Batak	3	4
	Dayak	6	6
	Flores	3	7
	Jawa	23	15
	Lampung	6	9

Figure 26. PURJ document downloaded from the dashboard (continued)



10	Makassar	9	14
	Manado	15	16
	Nias	0	2
	Padang	2	0
	Papua	0	1
	Poso	1	1
	Sumba	2	2
	Sunda	3	4
	Timor	4	5
	Tionghoa	8	12
	Toraja	5	5
	Lain-lain	1	3
	Tidak diketahui	1	1

III **Kebaktian dan Persekutuan/Kegiatan (Rata-rata Jumlah Kehadiran)**

No.	Keterangan	Jemaat			Majelis
		Anak	Laki-laki	Perempuan	
	Kebaktian				
1	Kebaktian Bidang Minggu Pagi	10,17	53,06	61,98	16,53
	Kebaktian Komisi Minggu Sore	13,02	52,63	61,21	19,35
	Persekutuan/Kegiatan				
2	Kegiatan Bidang Doa Pagi	2,19	17,31	18,92	14,42
	Kegiatan Komisi Doa Kamisan	4,81	35,31	35,1	18,92

IV **Kepengurusan**

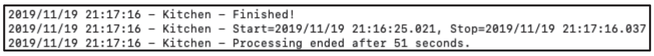
No.	Keterangan	Data	
		Laki-laki	Perempuan
1	Majelis	12	15
2	Pengurus Komisi	11	7
3	Pengurus Persekutuan	11	7

V **Pendeta**

No.	Keterangan	Data	
		Laki-laki	Perempuan
1	Calon Pendeta	1	0
2	Pendeta	1	1
3	Pendeta Mutasi	0	0
4	Pendeta Emiritus	0	0

Figure 27. PURJ document downloaded from the dashboard (continued)

The building of the data warehouse and dashboard of church congregation data in this research has been successfully done. The usage of the Greenplum Database together with the dimensional model of the data warehouse can accommodate the large size of data being loaded into the data warehouse from the source database. The ETL process from the source database to the data warehouse has been done successfully with the help of the tools provided by Pentaho Data Integration. From the test conducted with the ETL process, the ETL process took about 51 seconds to complete all the transformations and jobs as in figure 28 below.



```
2019/11/19 21:17:16 - Kitchen - Finished!
2019/11/19 21:17:16 - Kitchen - Start=2019/11/19 21:16:25.021, Stop=2019/11/19 21:17:16.037
2019/11/19 21:17:16 - Kitchen - Processing ended after 51 seconds.
```

Figure 28. ETL process executing time

All of the functional and non-functional requirements of the system have been successfully fulfilled. Although, from the test conducted to the dashboard, it revealed that the data loading process to be visualized from the data warehouse is taken too much time. This can happen because of the tables' designs in the data warehouse is not efficient enough, or the SQL select query to retrieve the data from the data warehouse is poorly structured. But still, the system built has been able to meet the needs of the users.

V. Conclusion

The data warehouse and the dashboard in this research has been successfully made and running in Greenplum Database. The data in the data warehouse is loaded into the data warehouse from the source database using an ETL process. This data is then visualized into tables and graphs in a dashboard using a web service as a bridge between the data warehouse and the dashboard. Users can also export the data from the data warehouse into an excel file in the form of a PURJ document.

The data warehouse and the dashboard built in this research are far from perfect. As explained before, the data loading process to be visualized in the dashboard took too much time. For future research, the researcher can focus on building more efficient tables' designs in the data warehouse and making the SQL select query more robust so the system of the data warehouse and the dashboard can run more smoothly and the users can be having a good time using the system.

Acknowledgements

We would like to express our biggest gratitude to the Jawa Tengah Regional Synod of GKI for the materials provided during this research was being conducted. We would also like to thank the teachers, professors, and staff of Duta Wacana Christian University for providing insight and expertise that have been very helpful for the research. Lastly, we also thank our colleagues at Duta Wacana Christian University for the mental supports given during this research was being conducted. For without anyone mentioned before, this research would not be possible to be completed successfully.

References

[1] "Gereja Kristen Indonesia Sinode Wilayah Jawa Tengah (GKI SW Jateng) - Deskripsi," 2014. [Online]. Available: https://www.gkiswjateng.org/sinodes#deskripsi.

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

[3] P. O'Donnell, D. Arnott, and M. Gibson, "Data Warehousing Development Methodologies: A Comparative Analysis," in *Decision Support in the Internet Age*, 2002, pp. 387–398.

[4] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Indianapolis, IN: John Wiley & Sons, Inc., 2013.

[5] W. Höpken, M. Fuchs, G. Höll, D. Keil, and M. Lexhagen, "Multi-Dimensional Data Modelling for a Tourism Destination Data Warehouse," in *Information and Communication Technologies in Tourism 2013*, 2013, pp. 157–169.

[6] S. Mali, "Data Warehouse implementations to analyse Tennis Player performance in 2014," 2016.

[7] P. Vassiliadis, "A Survey of Extract-Transform-Load Technology.," *Int. J. Data Warehous. Min.*, vol. 5, no. 3, pp. 1–27, 2009.

[8] J. Serra, "Building an Effective Data Wareheouse Architecture." 2013.

[9] Pivotal Software Inc., "About the Greenplum Architecture," 2019. .

[10] S. Few, "Dashboard Confusion Revisited," *Visual Business Intelligence Newsletter*, 2007.

[11] Pentaho Corporation, "Pentaho Data Integration," 2017. .

[12] W3C Working Group, "Web Services Glossary," 2004. .

[13] G. Alley, "Database vs Data Warehouse," 2018. [Online]. Available: https://www.alooma.com/blog/database-vs-data-warehouse.

[14] M. Vaqqas, "RESTful Web Services: A Tutorial," 2014. .