Implementation of online analytical processing for Indonesia agricultural commodities using JavaScript

R Trisminingsih¹*, M F Rahman¹, and I S Sitanggang¹
¹Computer Science Department, Bogor Agricultural University, Bogor, Indonesia

*email: rina.ilkomipb@gmail.com

Abstract. Ministry of Agriculture, Republic of Indonesia provide Indonesia agricultural commodities data which presented in the simple tabular format. It is a challenging task to analyze this growing data and generate data summarization. This study was conducted to implement online analytical processing (OLAP) application to provide an analytical tool for agricultural commodities data. The OLAP application is built using RESTful website architecture with Node.Js and Angular.Js. The back-end side uses Node.Js as a server and PostgreSQL as data storage. In front-end side, we use HTML5, Wijmo, and Angular.Js for visualizing data summary in the form of crosstab table and chart. OLAP operations are used in this application to perform analysis of some feature of agricultural commodities data. The features include crop production, productivity, and harvested area.

1. Introduction
Data on agricultural commodities in Indonesia are presented online to the public by the Ministry of Agriculture, Republic of Indonesia. The public can access agricultural commodities data at http://aplikasi.pertanian.go.id/bdsp/index.asp. Those data are presented in the table based on location, time, and commodity. In addition, the application provides agricultural commodities data in the form of spreadsheets but it does not allow users to quickly obtain a summary of historical data. Therefore, a data analysis tool like online analytical processing (OLAP) that implements data warehouse technology is necessary to present dynamic analysis and summary of the agricultural commodities data. Data warehousing and OLAP are essential elements of decision support [1]. Data warehouses provide OLAP tools for the interactive analysis of multidimensional data of varied granularities. According to [2], the advantages of OLAP are allowing users to gain insight and conclude about various aspects of data quickly and consistently and also providing interactively data accessing from various views.

Previous studies related to developing OLAP application that process and analyze agricultural commodities data have been done by Dwiprianti [3] and Putri and Sitanggang [4]. Dwiprianti [3] have built a web-based OLAP for horticultural crops data using Palo. Palo is multidimensional OLAP server and typically used as a business intelligence tool. Putri and Sitanggang [4] have used SpagoBI, an open source business intelligence and reporting tool, to create OLAP application for agricultural commodities. SpagoBI is complete because it covers all the needs of business intelligence with innovative solutions and provides a variety of analysis tools [5]. However those studies were developed using business intelligence framework that need to complex system configuration. Moreover, the built-in OLAP application is difficult to develop further because it depends on the
capabilities of the business intelligence framework. Therefore, further development of OLAP application that do not depend on complex framework is required. This study aims to develop an OLAP application with a simple architecture that requires no complex system set up and configuration. We build the OLAP application using RESTful website architecture with Node.Js and Angular.Js. Representational State Transfer (REST) architectural style is used in this study because it is simple, can make easy use of HTTP cache and proxy server to help for handling high load request, and helps organizing even a very complex application into simple resources [6]. This architecture consists of two sides as server (back-end) and client (front-end). The back-end side uses Node.Js as server and PostgreSQL as data storage. In front-end side, we use HTML5, Wijmo, and Angular.Js for visualizing data summary in the form of crosstab table and chart. The OLAP server is not used in this application. When we perform OLAP operations, Wijmo 5 can visualize the results on crosstab table and real time graphic without reloading the web page. Another benefit of using Wijmo 5 is that module can renders JSON on client side (browser) for data visualization.

2. Methods

2.1. Data pre-processing
The data used in this study are agricultural commodities data which were collected from Agricultural Statistics Database websites, that are publically provided at http://aplikasi.pertanian.go.id/bdsp/index.asp. The agricultural commodities used are food crops, plantations, and livestock from 2000 to 2014 at the district level in Indonesia.

Those data of agricultural commodities have been pre-processed in data warehouse by Putri and Sitanggang [4]. The data were modelled by multidimensional data structured, in the form of galaxy scheme. In this scheme, there are four fact tables and seven dimensions as shown in figure 1. In this study, we also need to add a new attribute in fact table to define commodity category.

![Figure 1. Galaxy scheme for data warehouse of agricultural commodities.](image-url)

2.2. Design and implementation of OLAP
This study adopts RESTful website architecture to develop the OLAP application. This architecture separates the server from its client. The purpose of separating client-server is to maximize the division
of labour and minimize overlap between the two. For communication between server and client, we use Representational State Transfer (REST) API, a standard web architecture which uses HTTP protocol for data exchange. The OLAP architecture for agricultural commodities data is given in figure 2.

The server, or back end, is responsible for storing persistent data of application, along with all the business logic required to interact with it. In this study, application server developed using HTTP protocol that has embedded in Node.Js. Node. Js is a Javascript runtime environment for executing Javascript code in server side.

![Figure 2. The architecture of agricultural commodities OLAP application.](image)

The client, or front end side, is responsible for making requests to the service, then doing something meaningful with the response that it receives. This OLAP architecture uses Angular.Js for implementing application interface. Angular.Js is a Javascript front end framework that developed by Google. Angular.Js use Model View Controller (MVC) architecture to develop good and clean code. For implementing OLAP and then visualizing data in crosstab table and several graphic forms, we used Wijmo 5 module. Wijmo 5 is a Javascript based module that developed to support data analytics with good performance and no need third-party library.

2.3. OLAP testing
Data verification has been done before the system was tested to evaluate the data provided by OLAP application. In addition, OLAP testing was performed on OLAP operations including roll up, drill down, slice, dice, and pivot on each data cube using Black box method. Some functions in OLAP application were tested including displaying crosstab tables and chart for selected dimensions and measure. Application testing was done to find the functions that could not work properly. Moreover, testing results are important for further system development. Then data warehouse testing was presented, by comparing the results of OLAP operations, SQL queries in the database, and initial data in website. Data warehouse testing is successful if all these results displaying the same value.

3. Results and Discussion

3.1. Data pre-processing
This study uses agricultural commodity data that has been pre-processed by previous researcher [4]. The study by Putri and Sitanggang [4] has converted data in Agricultural Statistic Database websites into data warehouse in PostgreSQL. However, this study needs to do advanced pre-processing to meet current application requirements. Data pre-processing started by adding new column in each fact table for storing categories of agricultural data, in order to group data based on the agricultural sector.
Besides that, there are some null columns in the fact table, whereas Wijmo 5 module cannot accept input which have null value. Crosstab table will display empty if it receives null input. Therefore, we need to replace all of null column with 0 value using default function in PostgreSQL, Coalesce function, when displaying data.

3.2. Design and implementation of OLAP

In back end side, we encode server using port 9090 to create application server and generate some function for querying in database. These functions represent the required data from client side or other application interface. This study uses REST API architecture that consists of four main components, namely URL design, HTTP verbs, HTTP response code, and response format.

3.2.1. URL Design. REST API can be accessed by using the protocol HTTP. Each communication between client and server is done through the URL. In this study, communication between client and server need two requests, they are request to get list of location and request to get data according filter selection. Table 1 describes list of created URL in OLAP application.

Table 1. List of URL in OLAP application.

| URL                  | Methods | Description                                      |
|----------------------|---------|--------------------------------------------------|
| localhost:9090/api/lokasi/get | GET     | This URL will return location data based on the user selection of filter in JSON form. |
| localhost:9090/api/post       | POST    | This URL will receive JSON sent from the client side that defines user required data, after querying the database and getting data from the database, server will send this data in the form of JSON to the client side. |

In this study, JSON format is used for data exchange between back end and front end. This format is useful in application development because the function of back end side can be accessed by various platform through URL without depend on data format standard in each platform.

3.2.2. HTTP Verbs. Each request of user uses certain method for the server to understand what the user wants. There are several methods that commonly used, such as GET, POST, DELETE and PUT. In this study, we only use two methods, which are GET method that is used in URL to get location data and POST method that is used for client to send JSON filter that will be respond by server with resending JSON format from the database.

3.2.3. HTTP Response Code. HTTP uses certain numbers code to indicate whether a request has failed or succeeded. Some HTTP codes requires meta information in the header of response as an additional information. Meta information is also used to indicate that the client requesting to the server has authority for data requesting. In this study, the HTTP codes used are 200 and 500. The HTTP code of 200 indicates that the request is successful and data have been sent to the client, while the HTTP code of 500 is used to indicate that an error occurred on the server and data are not sent to the client.

3.2.4. Response Format. Every request that made by client side will get a response from the server, the response format can be either XML or JSON form. In this study the response format is JSON with the header that has been entered meta information. Figure 3 describes response sample in JSON format.
The main module in front end side is Wijmo 5. Wijmo 5 provides OLAP templates to visualize data in crosstabs table and graphics form. Therefore, OLAP server is not used in this application. All of OLAP operations are handled by Wijmo 5 that process JSON response from server side according to request in filter function. For handling OLAP operations, Wijmo 5 needs several libraries that must be downloaded and configured. List of Wijmo 5 libraries that used in this application illustrated in figure 4.

The other function that not handled by Wijmo 5 is implemented using AngularJS framework. AngularJS is client side Javascript framework used for developing dynamic web application. AngularJS is used for develop OLAP application user interface because this framework have flexibility filters. In OLAP application, there is filter function that created for selecting OLAP analysis parameters, ie sector category, indicator, number and location status. Filter function use checkbox design that allow user to select one or more parameter that will be displayed in crosstabs table and graphical interface. Figure 5 illustrates OLAP application user interface that developed using AngularJS.

For processing front-end side request, we have to encode this request to server. At first, all of functions are created to post data to the application server. The result of the user filter is sent to the server with API url: localhost:9090/api/post. Data which returned from application server is used by Wijmo 5 displaying in cross-tabs table and graphs. Wijmo 5 is processed data from server in JSON format. Figure 6 show cross-tab table and graphs that displayed by Wijmo 5.
3.3. OLAP testing
At this stage, there are two testing section, namely data verification and application testing.

- Data verification. In this test compared results on queries on the database and OLAP operations are valid and correspond to the original data that are available in website of Ministry of Agriculture.
- OLAP testing. This testing is performed using Black box method that conducted by selecting some parameters on the filter function and then display the data based on the selected

**Figure 5.** OLAP application user interface.

**Figure 6.** Crosstab table and graph resulted in OLAP application using Wijmo 5.
parameter. Table 2 shows the test scenarios for OLAP application for agricultural commodities. The testing results on the OLAP application for agricultural commodities show that all functions on the system work properly.

**Table 2. Test scenarios.**

| Features                                      | Test Scenarios                                                                 | Expected Output                  | Work Properly? |
|-----------------------------------------------|-------------------------------------------------------------------------------|----------------------------------|----------------|
| **Displaying data in crosstab table**         | Users select one or more parameter in OLAP navigator (filter function) and click “ok” to display the crosstab table | Data are represented in crosstab table | Yes            |
| **Displaying data in graph**                  | Users select one or more parameter in OLAP navigator (filter function) and click “Grafik” menu to display data in graph | Data are represented in selection graph | Yes            |
| **Filtering data for selection category and indicator** | Users examine each category of fact table and indicator of measures | Data are displaying according to selected category | Yes            |
| **Displaying data as the result of OLAP operations** | Users select OLAP operations task, ie drill down, roll up, slice, dice, and pivoting. | Data are displayed according to selected OLAP operations | Yes            |

4. Conclusion
This study has successfully implemented an OLAP application for agricultural commodities which are built using RESTful website architecture with Javascript. The back-end side use Node.Js as server and PostgreSQL as data storage. The front-end side use Wijmo 5, HTML5, and Angular.Js to visualize data summarization in crosstab tables and chart. Several OLAP operations, like drill down, roll up, slice, and pivoting, are well performed in this application so users can easily explore and analyze agricultural commodities. Further works include developing the map visualization modules for exploring the distribution of agricultural commodities in the different level area.

Acknowledgments
The authors would like to thank Directorate of Research and Community Service, Ministry of Research, Technology and Higher Education, the Republic of Indonesia for the research grant.

References
[1] Reddy, G., Srinivasu, R., Rao, M., Rikkula, S. 2010. Data Warehousing, Data Mining, OLAP and OLTP Technologies are Essential Elements to Support Decision-Making Process in Industries. International Journal on Computer Science and Engineering. 2(9):2866
[2] Connolly, T dan C Begg. 2005. Database Systems. A Practical to Design, Implementation and Management, Fourth Edition. Pearson Education Limited.
[3] Dwiprianti F. 2015. Data warehouse dan OLAP berbasis web untuk tanaman hortikultura menggunakan Palo. Undergraduate Thesis. Bogor Agricultural University.
[4] A.I. Putri, and I.S. Sitanggang, "Data cubes integration in spatial OLAP for agricultural commodities," IOP Conf. Series: Earth and Environmental Science, vol. 58, 2017. doi:10.1088/1755-1315/58/1/012034.
[5] Cazzin G 2012 Business Intelligence with SpagoBI (Padua (IT): SpagoBI Competency Center)
[6] Fielding, RT. 2000. Architectural Styles and the Design of Network-based Software Architectures, Ph.D. dissertation, in University of California, Irvine.