Optimization of Spatial Visualization Module in SOLAP for Indonesian Agricultural Commodities

Ginanjar, A. R.,[1] Sitanggang, I. S.[2*] and Annisa[3]
Department of Computer Science, IPB University, Indonesia
Email: aseprahmatginanjar@gmail.com,[1] imas.sitanggang@ipb.ac.id,[2] annisa@apps.ipb.ac.id[3]
**Corresponding Author*

Abstract
Indonesia produces various agricultural commodities every year. To manage agricultural data, the Ministry of Agriculture of Indonesia provides a web-based application open to the public. The SOLAP for Indonesian Agricultural Commodities system has been successfully developed in the previous study for visualization and analysis of agricultural data. Still, the visualization module on the SOLAP takes a long time to display the map due to big size GeoJSON file. This study aims to simplify polygon data that represent administrative boundaries in Indonesia. Simplification on the polygon was done using two algorithms namely Douglas-Peucker and Visvalingam-Whyatt. This study showed that simplified polygon has successfully reduced access time of the SOLAP visualization module from 3.29 minutes to less than 15 seconds. The simplified polygon of the Douglas-Pecker algorithm has better shape and fewer coordinates. Hence, the results of Douglas-Pecker's algorithm simplification are implemented in SOLAP for Indonesian Agricultural Commodities.

1. Introduction

Data management of Indonesian agricultural commodity is committed by Ministry of Agriculture, Republic of Indonesia. The agricultural commodity is divided into 4 subsectors: food crop, horticulture, plantation, and livestock. The data collected by Ministry of Agriculture contain annual data from national level to regency level. These data are presented in tabular form which is less informative. Ward et al., (2015) stated that the analysis process of tabular data consumes more time than the analysis process of visualized data.

Several studies have been conducted to provide a more informative agricultural commodity data using the display of dynamic table and interactive map in a Geographic Information System (GIS). According to Bhat et al., (2011), GIS is a collection of tools that captures, stores, analyzes, manages, and presents data that are linked to geographical locations. Hidayah and Sitanggang (2014) had developed a GIS application to facilitate users in obtaining information about Indonesian soybean production. A GIS developed by Wijayanti and Sitanggang (2014) provides information about the distribution of rice production in Indonesia. Visualization of rice distribution covered production data, productivity, and harvest area. The data then were represented in the form of a table, map, and graph. The infirmity of those studies is only accommodated one agricultural commodity.

Besides GIS technology, data analysis and visualization can be performed with Spatial Online Analytical Processing (SOLAP). According to Bimonte et al., (2007), SOLAP is an Online Analytical Processing (OLAP) which supports spatial data. An OLAP development aims to provide a tool for retrieving specific data quickly from a significant amount of data (Lightstone et al., 2006). In order to store the data, OLAP is integrated to data warehouse. According to Han et al., (2012), a data warehouse is a semantically consistent data store that serves as a physical implementation of a decision support data model and stores the information on which an enterprise needs to make strategic decisions. In addition a data warehouse that could store spatial data type is named as spatial data warehouse (Malinowski and Zimányi, 2007).

A fact table is one of elements in data warehouse scheme. Bitmap indexing and partitioning fact tables in snowflake data warehouse scheme were implemented in order to get efficient queries on large data under the snowflake scheme (Benjelloun et al., 2017). Optimization of complex operations is one of challenges in development of efficient data warehouses. An ETL and MapReduce Hybrid architecture was proposed based on data filtering and processing to build an on-demand and dimensional big data which enables users to process relevant data in a big data warehouse (Houari et al., 2017). OLAP is an analytic tool which is integrated to databases or data warehouses. The NoSQL logical model was implemented in OLAP systems (Chevalier et al., 2015). The experimental works reveal that loading

Online ISSN 2673-0014/

and aggregate computation run faster under the NoSQL OLAP systems (Chevalier et al., 2015). In order to detect, explain and resolve bias in OLAP queries, the system HypDB was proposed (Salimi et al., 2018). The HypDB was also designed to automatically rewrite the biased query into an unbiased query by performing the hypothesis test (Salimi et al., 2018). Those four previous studies do not discuss the query optimization on a spatial data warehouse which is our focus in this study.

Putri and Sitanggang (2017a) had successfully developed the SOLAP system using SpagoBI for horticultural crops in Indonesia. The system stores and manages horticultural crops data which consist of vegetables, decorative plants, medicinal plants, and fruits from 2000 until 2013. The study was then continued by Putri and Sitanggang (2017b). Data cube using the galaxy scheme had been appropriately implemented on data of agricultural commodities which include food crop, horticulture, plantation, and livestock. However, both studies are still carried out issues. The output of SOLAP operations represented in the crosstab had not been able to be integrated with a map. In addition there a limitation of data quantity for the location intelligence module in SpagoBI which the maximum amount of records that can be processed is 182,250.

Di Martino et al., (2011) has successfully developed a SOLAP application that combines Mondrian as OLAP server, Google Earth for spatial data visualization, and JPivot for tabular data visualization. This study is based on a web application named Google OLAP (GooLAP). The data used in this study are sourced from the data warehouse air pollution in Italy. This study has successfully processed further output from OLAP operations which contained spatial data. Moreover, the results of the analysis can be represented in Google Earth 3D maps interactively. The study related to SOLAP is the SOLAP Indonesian Agricultural Commodities by Sitanggang et al., (2017). The SOLAP system that was built adapting the GooLAP application architecture proposed by Di Martino et al., (2011). The commodity data that used include four agricultural sub-sectors consisting of horticulture, food crops, plantations and livestock sub-sectors. The primacy of this system is located on the visualization results of SOLAP operations in crosstab that able to synchronize with graphics and maps so there is no limit of maximum data amount as in SOLAP SpagoBI (Putri and Sitanggang, 2017a and 2017b). However, the map visualization module on the system takes a quite long time to display the maps. This is caused by the large size of Geospatial JavaScript Object Notation (GeoJSON) file that represents the polygon of administrative boundaries in Indonesia. The file size of GeoJSON on SOLAP results that displays all districts on a map is 91.30 MB.

The number of coordinates in the polygon can be reduced by the simplification process (Arteaga, 2013). Simplification can be done using the Douglas-Peucker and Visvalingam-Whyatt algorithms. Shi and Cheung (2006) used Douglas-Peucker' algorithm to simplify the polyline of Hong Kong Island coastline and Kowloon coastline in Tiongkok. Meanwhile, Song and Miao (2016) uses Douglas-Pecker's algorithm to simplify the polyline of railway data in Tiongkok. Both types of study acquired precise simplification result. Moreover, according to the study by Song and Miao (2016), the number of polylines that composes coordinates decrease by 85%. Visvalingam and Williamson (1995) compared two simplification algorithms namely Douglas-Peucker and Visvalingam-Whyatt. The study simplifies the polygon which represented street shape.

Dynamic merging of spatial objects is one of challenges in OLAP and spatial data warehouse technology. This operation requires expensive computation and much storage space. Optimization of spatial operations provides an alternative solution for this problem therefore OLAP queries can be efficiently implemented on a spatial data warehouse. Sitanggang et al., (2017) developed a web-based system namely SOLAP for Indonesian Agricultural Commodities. The SOLAP operation results are represented in forms of maps, crosstab, and graphs. The system is using web platform so that it can be accessed by the public. However, the system is constrained by a large GeoJSON file size, which reaches 91.30 MB. Large size of GeoJSON file in visualization map module of SOLAP Indonesian Agricultural Commodities is caused by the high number of coordinates. Therefore, it is needed to simplify the polygons in the SOLAP system in order to make the coordinates of polygons reduced. This study aims to simplify polygon data that represent administrative boundaries in Indonesia. Simplification on the polygon uses two algorithms namely Douglas-Peucker and Visvalingam-Whyatt. The result of this study is expected to produce an optimal administrative boundary polygon in Indonesia in term of file size and its shape. The optimized polygon is implemented in the visualization map module of SOLAP for Indonesian Agricultural Commodities, so that the visualization module can be accessed more quickly.

2. Methodology

There are six main steps of this study. These steps include analysis of SOLAP for Indonesian Agricultural Commodities, data preprocessing, polygon simplification, analysis of simplification result, measurement of map visualization module access time, and implementation of simplified polygon on SOLAP for Indonesian Agricultural Commodities.

2.1 Data

This study uses spatial data of Indonesia's administrative border in polygon format. The spatial data are available for district and province level. Data were collected from Geospatial Information Agency (BIG), Indonesia and are stored in the shapefile ESRI format. The number of polygons is 15,798 which consist of 3,427,021 coordinates. In addition to spatial data, this study uses non-spatial data of Indonesian agriculture commodities that were obtained from Agricultural Statistics Database provided by Ministry of Agriculture, Republic of Indonesia. We collected the data from http://aplikasi.pertanian.go.id/bdsp/index.asp (accessed on February 2017). The attributes of the non-spatial data include subsector, indicator (harvest area, production, and productivity), data level (national, province or district), and year. Subsector consists of horticulture, food crops, plantation, and livestock.

2.2 Analysis of SOLAP for Indonesian Agricultural Commodities

The previous study by Sitanggang et al., (2017) has successfully developed a web-based SOLAP system of Indonesian Agricultural Commodity. The agricultural commodity data were collected from the Ministry of Agriculture Indonesia. The data warehouse on this system consists of 7 fact tables and 3 dimension tables. Those fact tables include horticultural, decoration plants, flowers, plantation, food crop, livestock population, and livestock production. Meanwhile, the three dimension tables are location, time, and commodity. Moreover, this study uses the galaxy scheme for its data warehouse scheme. This system could visualize the result of SOLAP operations in the form of crosstab, graphic, and map.

2.3 Data Preprocessing

Polygon data of administrative area border are imported into the database in PostgreSQL with the PostGIS extension. The database used in this study is the database of SOLAP Indonesian Agricultural Commodities. Within the data, there are 8 districts which do not have province name, so this study fill the province name over these 8 districts based on data from Statistics Indonesia. The adjustment of districts name was conducted for spatial data from BIG and data which are located in the SOLAP system. The purpose of this step is to make a consistent districts name which was referenced from Statistics Indonesia. In addition, in this step island level polygons are formed by merging all district level polygons in provinces.

2.4 Polygon Simplification

The main phase of this study is a polygon simplification process. Douglas-Peucker and Visvalingam-Whyatt algorithms are used to simplify polygon. Douglas and Peucker algorithms will remove a certain coordinate if the distance of the coordinate and line segment is less than the threshold value (Barriot et al., 2001). Line segment is the farthest coordinate which is calculated from its initial polygon coordinate, meanwhile threshold is a distance which determines whether a specific coordinate should be kept or removed. The idea of Visvalingam-Whyatt algorithm is to rank coordinate from a certain polygon based on its level of coordinate significance (Visvalingam, 2016). The significance of a coordinate is calculated based on the area formation from the coordinate itself and two other coordinates until they create a triangle area. The threshold value is the area which is used to determine whether the coordinate should be kept or removed. According to study by Xiao-Li and De (2010), 0.0022 is an optimal threshold for the Douglas-Peucker algorithm for simplifying polygon street shape in Tiongkok. In this study, simplification process uses threshold values from 0.00001 until 0.04981. The value of 0.00001 in degree is about 1.11 meter of length in the Douglas and Peucker algorithm and it is about 1.11 meter^2 of the area in the Visvalingam-Whyatt algorithm.

2.5 Douglas-Peucker Algorithm

Douglas-Peucker algorithm was first introduced by David Douglas and Thomas Peucker in 1973. This algorithm is used to reduce the number of points to represent a line (Douglas and Peucker, 1973). Figure 1 shows polyline simplification using the Douglas-Peucker algorithm.

Figure 1 presents a polyline simplification step using the Douglas-Peucker algorithm. The first step is to determine the threshold value. This value is used as a reference in the process of determining whether a coordinate will still be used as a part of polygons or not. Later, the polyline is connected to a line from the first coordinate point to the last coordinate point. The illustration of the segment line is presented in Figure 1 (a). The line is called the segment line. The distance of all coordinates of the polyline except the first and last coordinates are calculated perpendicular to the segment line. Coordinates with farthest distance to the segment line and its distance greater or equal to the threshold value will be used as coordinates reference to divide the polyline into two parts. In both parts of the polyline, the segment line is formed and distance of coordinates perpendicular to the segment line is calculated. When a distance found to be greater than or equal to the threshold value, then the division of polyline is re-divided into two parts. The illustration of the division of polyline into two parts is presented in Figure 1 (b). Whereas when all coordinate distances to the segment line are less than the threshold value, the coordinates will be removed from the polyline. Simplified results of a separate polyline will be recombined into one polyline. The results of merging simplified polyline is presented in Figure 1 (c).

2.6 *Visvalingam-Whyatt Algorithm*

The Visvalingam-Whyatt algorithm is used to reduce the number of points to represent a line using a significance value (Visvalingam, 2016). According to Visvalingam (2016) each coordinate has a significance value. This algorithm will eliminate coordinates that have a significance value less than a certain value limit. Figure 2 shows polyline simplification using the Visvalingam-Whyatt algorithm.

In Figure 2 the threshold value in the form of area is determined before the simplification is carried out. This value is used as a reference in the process of determining whether a coordinate will still be used as a constituent of polygons or not. Then each coordinate on the polyline is calculated as the significance level. The calculation is done by connecting a coordinate with two neighboring coordinates to form a triangle area. The area of the triangle is used as a significance value. The illustration of triangle area for each coordinate in a polyline is shown in Figure 2 (b). When the coordinates have a significance value less or equal to the threshold value, the coordinates will be omitted from the polyline. The process is done repeatedly until the significance value of each coordinate is greater than the threshold value. In Figure 2 (c) it is found that coordinates in segment 3 and segment 4 are removed from the polyline. This is because the significance value in both segments is less than the threshold value.

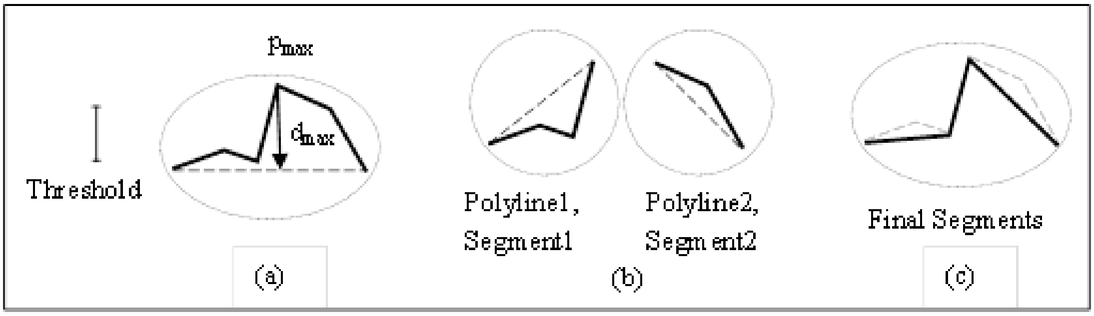

Figure 1: Douglas-Peucker algorithm illustration (Barriot et al., 2001) (a) Calculating distance between coordinates and line segments (b) Polyline is divided into two lines based on the threshold value (c) Simplification results

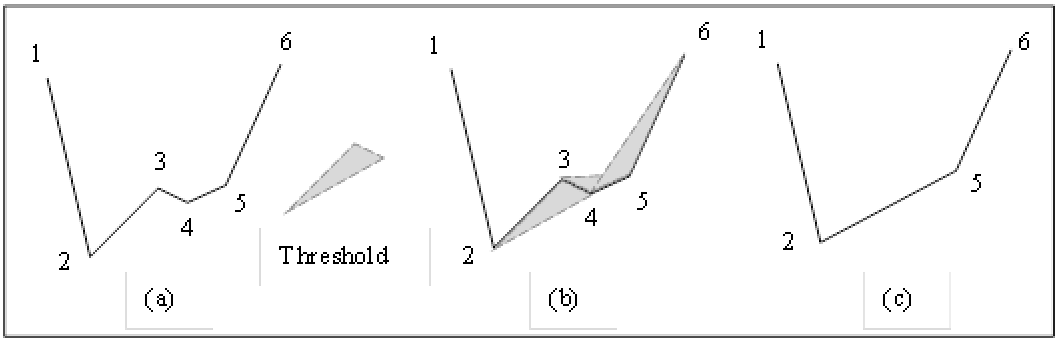

Figure 2: Visvalingam-Whyatt algorithm illustration (Visvalingam & Whyatt, 1993) (a) Original polyline (b) Calculating significant area (c) Simplified polyline

2.7 GeoJSON
JavaScript Object Notation (JSON) is a lightweight, text-based, language-independent data interchange format (Crockford, 2006). JSON defines a small set of formatting rules for the portable representation of structured data. While GeoJSON is a geospatial data interchange format based on JSON (Butler et al., 2016). GeoJSON extends the JSON format to include geometric features including Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection (Li et al., 2015). Each feature in GeoJSON has two parts: vector and attribute data. GeoJSON uses a geographic coordinate reference system namely World Geodetic System 1984 and units of decimal degrees. The advantage of GeoJSON is to enable user to add additional information to geographic objects easily (Gaffuri,2012).

2.8 *Analysis of Simplification Result*
The analysis is conducted on polygons which are resulted from simplification. This study analyzes the number of coordinates, number of polygons, shape, and size of GeoJSON file. In this phase, the optimal threshold value is selected for each algorithm.

2.9 Measuring Access Time of Map Visualization Module
The polygon which has been simplified using optimal threshold value is implemented on SOLAP system for Indonesian Agricultural Commodity. During this phase, polygon formation at province level and island level are conducted. Moreover, this phase measures the system's access time of map visualization module. The access time of simplified polygon is compared to those of initial polygon. Measurement of access time is performed ten times of repetition for each algorithm and each polygon level. The internet connectivity speed during measurement is 6.29 Mbps.

3. Result and Discussion

3.1 Number of Coordinate in Simplified Polygon
The number of coordinates in the maps for all districts in Indonesia is 3,427,021. After the polygon simplification is performed, the number of coordinates decreases. Figure 3 shows number of coordinates of simplified polygon as the results of Douglas-Peucker and Visvalingam-Whyatt algorithms. Based on Figure 3, it is known that the use of 0.00001 threshold value creates a significant gap in two algorithms. There are 2,681,198 coordinates in simplified polygon as the result of the Douglas-Peucker algorithm while there are only 201,587 coordinates as the result of the Visvalingam-Whyatt algorithm. There is a significant reduction of coordinates in the Douglas-Peucker algorithm at the threshold value of 0.00001 until 0.00119. Meanwhile, in the Visvalingam-Whyatt algorithm number of coordinates significantly reduced at the threshold value of 0.00001 until 0.00114.

3.2 Size of GeoJSON file
Figure 4 shows comparison of threshold values and GeoJSON file size which represents polygon for all districts in Indonesia. The size of GeoJSON file is linearly related to number of coordinates in polygon and threshold value. The higher threshold value is, the less number of coordinates will be, and the size of GeoJSON file will decrease. At threshold value of 0.00001, the size of GeoJSON file using the Douglas-Peucker algorithm is 65.56 MB. Meanwhile, Visvalingam-Whyatt algorithm produces a smaller file size of 5.23 MB. The size of GeoJSON file using the Visvalingam-Whyatt algorithm is always larger than 1.5 MB. Meanwhile, the smallest size of GeoJSON file in Douglas-Peucker is 0.26 MB.

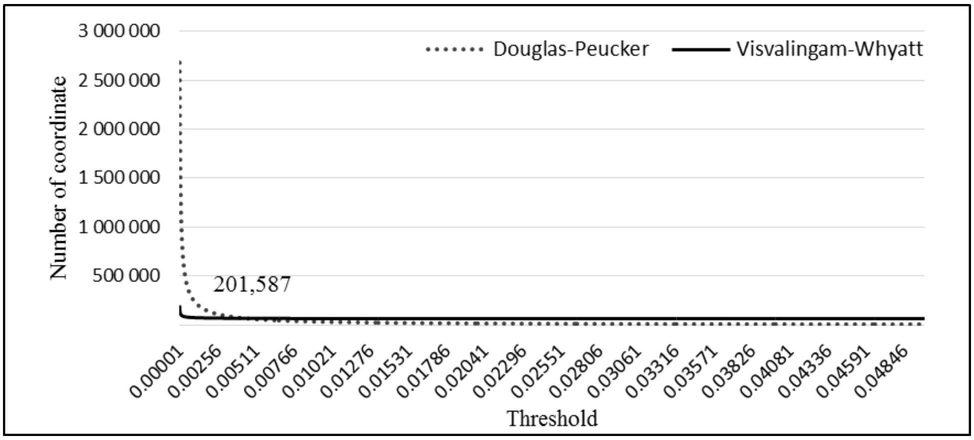

Figure 3: Comparison of the threshold value and number of coordinate in polygon for all districts in Indonesia

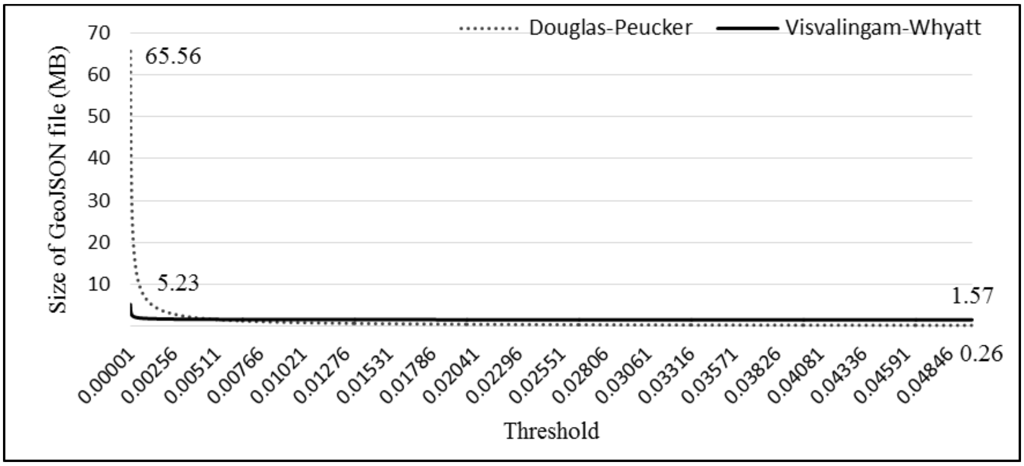

Figure 4: Comparison of threshold value and GeoJSON file size of all districts polygon in Indonesia

Table 1: Comparison of several threshold values and size of GeoJSON file

Size of GeoJSON file (MB)	Threshold value	
	Douglas-Peucker algorithm	Visvalingam-Whyatt algorithm
4	0.00178	0.00003
3	0.00251	0.00007
2	0.00405	0.00058
1	0.00893	-
0.5	0.02032	-
0.3	0.03974	-

3.3 Optimal Threshold Value

Selecting threshold value will result in a significantly different polygon shape than its original shape because there is a large number of coordinates that are eliminated. On the other hand, high threshold value will decrease the size of GeoJSON file. This condition will result in a more efficient time consumption to visualize map in SOLAP for Indonesian Agricultural Commodity. Table 1 shows comparison of several threshold values which produce the size of GeoJSON file less than 4 MB. According to Table 1, polygon simplification as the output of the Visvalingam-Whyatt algorithm with smallest threshold values of 0.04981 produces a 1.57 MB GeoJSON file. As an example, Figure 5 shows the simplified polygon of DKI Jakarta Province. According to visual observation on Figure 5, simplified polygon as the result of the Douglas-Peucker algorithm has more coordinates than simplified polygon as the result of the Visvalingam-Whyatt algorithm. The simplified polygon which is produced by the Douglas-Peucker algorithm is similar to the original shape. Figure 5 (c) and (f) shows that both GeoJSON files size is 2 MB (Table 1). However, the polygon in Figure 5 (f) has a different shape from its original shape. The factor which leads to this condition is the different number of the polygon. As an example, Figure 6 shows the differences polygon number of Riau Islands Districts.

Figure 6 shows that number of polygons as the result of the Visvalingam-Whyatt algorithm is higher than those resulted from the Douglas-Peucker algorithm. The number of polygons which compose Riau Island is 1,968. After simplification using the Douglas-Peucker algorithm with threshold value of 0.00405, the number of polygon decrease to 398 polygons. Meanwhile simplification using the Visvalingam-Whyatt algorithm did not reduce the number of polygons. Simplified polygons are then implemented in the SOLAP system of Indonesian Agricultural Commodity using several threshold values in Table 1. Visual observation on the map visualization module, and analysis of GeoJSON file size show that the optimal threshold value for the Douglas-Peucker algorithm is 0.00251. Meanwhile, a 0.00007 is the optimal threshold value for the Visvalingam-Whyatt algorithm.

3.4 Comparison of Polygons Before and After Simplification

The polygon simplification with the optimal threshold value was done at the district level. Based on the simplified polygon at district level, number of polygons at provincial and island levels should be analyzed. Provincial level polygons are formed by merging all district polygons in the province. Likewise, island level polygon is formed from by merging all provincial level polygons that are located in the island. Table 2 shows comparison of number of coordinates before and after simplification.

3.5 Comparison of Polygons Before and After Simplification

The polygon simplification with the optimal threshold value was done at the district level. Based on the simplified polygon at district level, number of polygons at provincial and island levels should be analyzed. Provincial level polygons are formed by merging all district polygons in the province. Likewise, island level polygon is formed from by merging all provincial level polygons that are located in the island. Table 2 shows comparison of number of coordinates before and after simplification. Based on Table 2, number of coordinates composing the polygon at the district level is higher than those at the provincial level. This is caused by the process of merging provincial polygons which removes coordinates on polygon's boundaries between districts in the province. Likewise, at the provincial level, the number of coordinates in polygons is higher than those at the island level polygon. Based on Table 2, it can be seen that coordinates of polygons resulted from simplification using the Douglas-Peucker algorithm are fewer than coordinates resulted from the Visvalingam-Whyatt algorithm. The difference is 2,753 coordinates at the district level, 18,401 coordinates at the provincial level, and 21,871 at the island level. Table 3 shows comparison of the number of polygons before and after simplification.

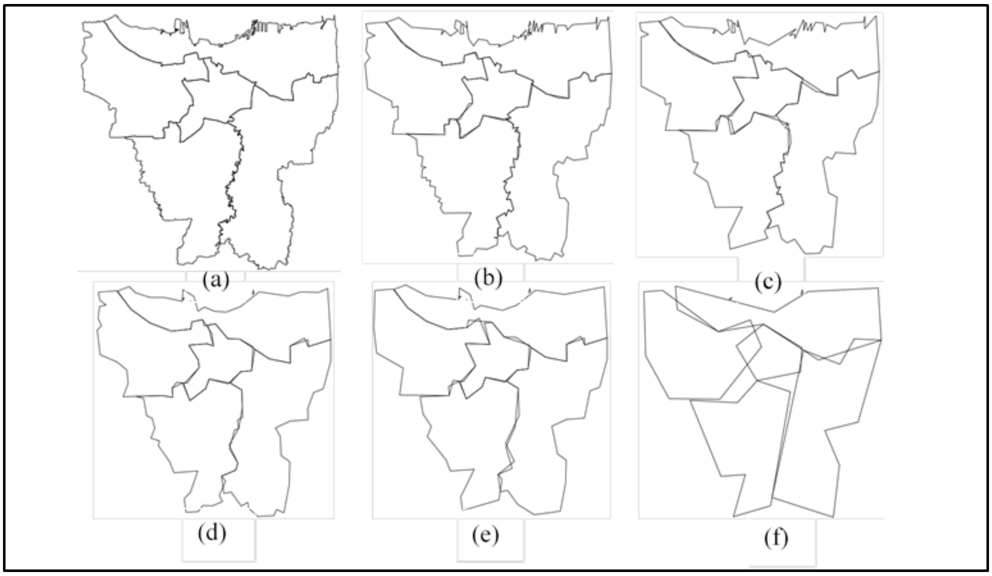

Figure 5: Polygon of DKI Jakarta Province (a) before simplification (b) simplified by Douglas-Peucker algorithm with threshold of 0.00251 (c) simplified by Douglas-Peucker algorithm with threshold of 0.00405 (d) simplified by Visvalingam-Whyatt algorithm with threshold of 0.00003 (e) simplified by Visvalingam-Whyatt algorithm with threshold value of 0.00007 (f) simplified by Visvalingam-Whyatt algorithm with threshold of 0.00058

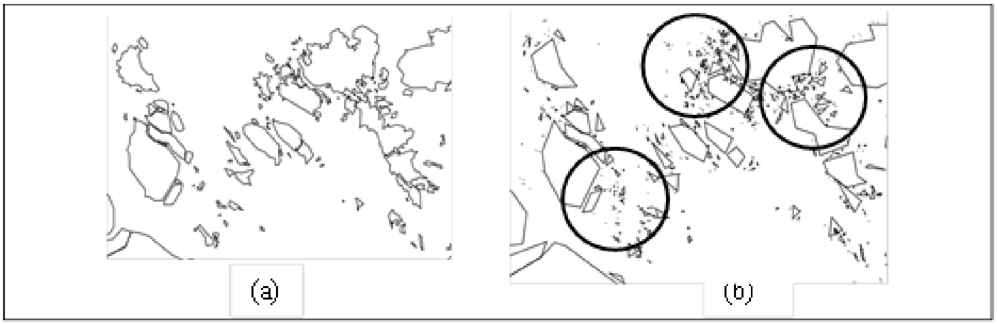

Figure 6: Simplified polygon of Riau Islands (a) simplified by Douglas-Peucker algorithm with threshold value of 0.00405 (b) simplified by Visvalingam-Whyatt algorithm with threshold value of 0.00058

Table 2: Number of polygon coordinates before and after simplification

Polygon level	Number of Coordinate		
	Before simplification	Douglas-Peucker algorithm	Visvalingam-Whyatt algorithm
District	3,427,021	111,953	114,706
Province	2,428,214	68,925	87,326
Island	2,289,355	59,768	81,639

Table 3: Number of polygons before and after simplification

Polygon level	Number of Polygon		
	Before simplification	Douglas-Peucker algorithm	Visvalingam-Whyatt algorithm
District	15,798	3,920	15,798
Province	15,344	3,532	14,622
Island	15,321	3,509	14,599

Table 4: Size of GeoJSON file before and after simplification

Polygon level	Size of GeoJSON file (MB)		
	Before simplification	Douglas-Peucker algorithm	Visvalingam-Whyatt algorithm
District	91.30	3.14	3.25
Province	62.40	1.82	2.38
Island	58.40	1.55	2.21

Table 5: Polygon area before and after simplification

Polygon level	Polygon area difference (km^2)	
	Douglas-Peucker algorithm	Visvalingam-Whyatt algorithm
District	64.96	106.94
Province	39.52	66.91
Island	33.97	57.67

Table 6: Access time of visualization module in the SOLAP for Indonesian Agricultural Commodities

Polygon level	Access time (second)		
	Before simplification	Simplification using Douglas-Peucker algorithm	Simplification using Visvalingam-Whyatt algorithm
District			
Minimum	149.16	9.82	8.50
Maximum	284.02	18.08	17.49
Average	197.46	14.68	11.87
Province			
Minimum	86.16	7.99	7.07
Maximum	146.83	10.98	13.60
Average	102.28	9.30	10.23
Island			
Minimum	79.87	8.00	8.62
Maximum	115.94	13.34	14.39
Average	89.61	10.48	10.40

Based on Table 3, after simplification with the Douglas-Peucker algorithm, number of polygons is reduced to 3,920 polygons at the district level. However, the simplification of the Visvalingam-Whyatt algorithm does not reduce the number of polygons. The number of simplified polygons depends on the threshold value when using the Douglas-Peucker algorithm. However, this doesn't apply in using the Visvalingam-Whyatt algorithm. For all threshold values, number of polygons remain the same as the original polygon before simplification. Table 4 shows size of GeoJSON files before and after simplification.

The size of GeoJSON file depends on the number of coordinates that composes the polygon. Based on Table 4, the Douglas-Peucker algorithm results smaller GeoJSON files for all polygon levels than the file from the Visvalingam-Whyatt algorithm. The difference of GeoJSON file size from Douglas-Peucker and Visvalingam-Whyatt algorithms is 0.11 MB at the district level, 0.56 MB at the provincial level, and 0.66 MB at the island level. Reducing number of coordinates on the simplified polygon results different shapes of polygons compared to the original polygon. In addition, area of the polygon has also changed. Table 5 shows the difference between polygon area before and after the simplification. Based on Table 5, in term of polygon area, the Douglas-Peucker algorithm is smaller than using the Visvalingam-Whyatt algorithm. This applies to all polygon levels.

3.6 Implementation of Simplified Polygons

The simplified polygons have been implemented in the SOLAP for Indonesian Agricultural Commodities. Measurement of access time was performed on the visualization module in the SOLAP. Measurement was done at 10 repetitions for each algorithm and polygon level. Table 6 shows access time of the visualization module in the SOLAP with simplified polygons.

Based on Table 6, the average access time for the visualization module at the district level using original polygons differs considerably to the module with simplified polygons. The average access time for visualization module at the district level is 197.46 seconds (or about 3.29 minutes), while using simplified polygons as the result of Douglas-Peucker algorithm, the access time decreases by 13.45 times, which is 14.68 seconds. The simplified polygon using Visvalingam-Whyatt algorithm decreases by 16.64 times, which is 11.87 seconds. The average access time of the module with polygons from Douglas-Peucker algorithm is 2.81 seconds which is larger than those of the Visvalingam-Whyatt algorithm.

The decreasing of average access time also occurs in the visualization module at the provincial and island level. There is decreasing of 11 times and 10 times in the use of polygon simplified with the Douglas-Peucker and Visvalingam-Whyatt algorithms, respectively. While at the island level, the decreasing of access time reaches 8.55 times and 8.62 times for polygons simplified by Douglas-Peucker and Visvalingam-Whyatt algorithms, respectively. Before simplification, access time at the provincial and island level is 1.70 minutes and 1.49 minutes, respectively.

Simplified polygons from the Douglas-Peucker and Visvalingam-Whyatt algorithms have successfully reduced the access time of the visualization module in the SOLAP to less than 15 seconds from 3.29 minutes. Simplification using the Visalingam-Whyatt algorithm does not reduce the number of polygons, but the number of coordinates of the polygon decreases significantly so that the shape of polygon is not good (Figure 5). Whereas the Douglas-Peucker algorithm results polygons that have the same shape as the original polygon. However number of polygons decreases so that polygons representing small islands are lost (Figure 6). Therefore, simplified polygons as the result of Douglas-Peucker algorithm are selected to be implemented on the SOLAP for Indonesian Agricultural Commodities.

4. Conclusion

Optimization of administrative boundary map in the SOLAP for Indonesian Agricultural Commodity has been successfully performed. Polygons simplification using Douglas-Peucker and Visvalingam-Whyatt algorithms has reduced the access time of the visualization module in SOLAP from 3.29 minutes to less than 15 seconds. Simplification using the threshold value of 0.00251 on the Douglas-Peucker algorithm and 0.00007 on the Visvalingam-Whyatt algorithm produces an optimal shape of polygon and size of GeoJSON files. The simplified polygon from the Douglas-Peucker algorithm is implemented in the SOLAP for Indonesian Agricultural Commodity because the simplified polygons have better shape than polygons resulted from the Visvalingam-Whyatt algorithm. Dynamic merging of spatial objects is implemented in spatial online analytical processing (SOLAP) as an analytics tool for spatial data warehouses. Polygon simplification reduces the cost of computation in OLAP operation execution therefore the performance of SOLAP will increase significantly. In addition to SOLAP and spatial data warehouses, polygon simplification can be applied in spatial data mining to extract hidden patterns on large spatial datasets. Data mining algorithms will run faster on simplified polygons than those on complex polygons. However there is a tradeoff between accuracy and speed of the algorithms. Polygon simplification in data warehousing and data mining should be implemented to optimize either speed or accuracy.

References

Arteaga, M. G., 2013, Historical Map Polygon and Feature Extractor. *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*, 66-71.

Barrilot, X., Hangouet, J. F. and Kadri-Dahmani, H., 2001, *Generalization of the "Douglas and Peucker" Algorithm for Cartographic Applications*. [Online]. Available at http://w.icaci.org/files/ documents/ICC_proceedings/ICC2001/icc2001/file/f13019.pdf.

Benjelloun, M., Merouani, M. E. and Abdelouarit, E. A. A, 2017, Using Snowflake Schema and Bitmap Index for Big Data Warehouse Volume. *International Journal of Computer Applications*, Vol. 180(8), 30-32.

Bhat, M. A., Shah, R. M. and Ahmad, B., 2011, Cloud Computing: A Solution to Geographical Information Systems(GIS). *International Journal on Computer Science and Engineering*, Vol. 3(2), 594-600.

Bimonte, S., Tchounikine, A. and Miquel, M., 2007, Spatial OLAP: Open Issues and a Web Based Prototype. *10th AGILE International Conference on Geographic Information Science 2007*, 11.

Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S. and Schaub, T., 2016, *The GeoJSON Format*. [Online]. Available at https://www.rfc-editor.org/info/rfc7946.

Chevalier, M., Malki, M. E., Kopliku, A., Teste, O. and Tournier, R., 2015, Implementing Multidimensional Data Warehouses Into NoSQL. *17th International Conference on Entreprise Information Systems*.

Crockford, D., 2006, *The Application/json Media Type for JavaScript Object Notation (JSON)*. [Online]. Available at https://www.rfc-editor.org/info/rfc4627.

Di Martino, S., Bimonte, S., Bertolotto, M., Ferrucci, F. and Leano, V., 2011, Spatial Online Analytical Processing of Geographic Data through the Google Earth interface. *Geocomputation, Sustainability and Environmental Planning*, Vol. 348, 163-182.

Douglas, D. H. and Peucker, T. K., 1973, Algorithms for the Reduction of the Number of Points required to represent a Digitized Line or its Caricature. *Canadian Cartographer*, Vol. 10(2), 112–122.

Gaffuri, J., 2012, Toward Web Mapping with Vector Data. *International Conference on Geographic Information Science*, Vol. 87-101.

Han, J., Kamber, M. and Pei, J., 2012, *Data Mining Concepts and Techniques*. San Fransisco, SF: Elsevier.

Hidayah, R. and Sitanggang, I. S., 2014, Web Based Geographic Information System for Soybean Production in Indonesia Using K-Means Clustering. *Proceedings of National Seminar and Yearly Meeting of MIPA 2014*, 383-391 (in Bahasa).

Houari, M. E, Rhanoui, M. and Asri, B. E, 2017, Hybrid Big Data Warehouse for On-Demand Decision Needs. *3rd International Conference on Electrical and Information Technologies ICEIT'2017*.

Li, W., Song, M., Zhou, B., Cao, K. and Gao, S., 2015, Performance Improvement Techniques for Geospatial Web Services in a Cyberinfrastructure Environment-A Case Study with a Disaster Management Portal. Computers. *Environment and Urban Systems*, Vol. 54, 314-325.

Lightstone, S., Teorey, T. and Nadeau, T., 2006, *Database Modeling and Design: Physical Database Design for Decision Support, Warehousing, and OLAP*. San Francisco, SF: Morgan Kaufmann.

Malinowski, E. and Zimányi, E., 2007, Logical Representation of a Conceptual Model for Spatial Data Warehouses. *Geoinformatica*, Vol. 11(4), 431-57.

Putri, A. I. and Sitanggang, I. S., 2017a, Data Cubes Integration in Spatial OLAP for Agricultural Commodities. *IOP Conference Series: Earth and Environmental Science.* Vol. 58(1).

Putri, D. and Sitanggang, I. S., 2017b, Clustering Module in OLAP for Horticultural Crops Using SpagoBI. *IOP Conference Series: Earth and Environmental Science.* Vol. 58(1).

Salimi, B., Gehrke, J. and Suciu, D., 2018. HypDB: A Demonstration of Detecting, Explaining and Resolving bias in OLAP queries. *Proceedings of the VLDB Endowment*, August 2018.

Shi, W. and Cheung, C., 2006, Performance Evaluation of Line Simplification Algorithms for Vector Generalization. *The Cartographic Journal*, Vol. 43(1), 27-44.

Sitanggang, I. S., Ginanjar, A. R., Syukur, M., Trisminingsih, R. and Khotimah, H., 2017, Integration of Spatial Online Analytical Processing for Agricultural Commodities with OpenLayers. *International Conference on Electrical Engineering and Computer Science (ICECOS)*, 167-170.

Song, J. and Miao, R., 2016, A Novel Evaluation Approach for Line Simplification Algorithms towards Vector Map Visualization. *International Journal of Geo-Information*, Vol. 5(12), 223.

Visvalingam, M., 2016, The visvalingam algorithm: metrics, measures, and heuristic. *Cartographic Journal*, Vol. 53(3), 242-252.

Visvalingam, M. and Whyatt, J. D., 1993, Line Generalization By Repeating Elimination of Points. *Cartographic Journal*, Vol. 30(1), 46-51.

Visvalingam, M. and Williamson, P. J., 1995, Simplification and Generalization of Large Scale Data for Roads: A Comparison of Two Filtering Algorithms. *Cartography and Geographic Information Systems*, Vol. 22(4), 264-275.

Ward, M., Grinstein, G. and Keim, D., 2015, *Interactive Data Visualization: Foundation, Techniques, and Applications*. Boca Raton, FL: CRC Press.

Wijayanti, R. I. and Sitanggang, I. S., 2014, Web Based Geographic Information System for Rice Production in Indonesia using Geoserver. *Proceedings of National Seminar and Yearly Meeting of MIPA 2014* (in Bahasa).

Xiao-li, W. and De, Z., 2010, Selecting Optimal Threshold Value of Douglas-Peucker Algorithm Based on Curve fit. *Networking and Distributed Computing (ICNDC), 2010 First International Conference*, 251-254.