Received: January 28, 2023. Revised: April 12, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Publishing and Querying of Spatiotemporal Agriculture Production Data Warehouse on Semantic Web Using QB4MobOLAP

Irya Wisnubhadra[1,2]* **Safiza Suhana Kamal Baharin**[2] **Nurul A. Emran**[2]
Djoko Budiyanto SHR[1]

[1]*Informatics Department, Atma Jaya Yogyakarta University, Indonesia*
[2]*Faculty of Information and Communication Technology, University Teknikal Malaysia Melaka, Indonesia*
* Corresponding author's Email: irya.wisnubhadra@uajy.ac.id

Abstract: Data Warehouse (DW) and on line analytical processing (OLAP) as parts of business intelligence (BI) are proven platform for decision-making support. Over the previous decade, the advent of Web 2.0 technologies has increased the accessibility of the data web across the internet. The use of semantic web (SW) including linked open data, linked open statistical data, and open government data is rising at a breakneck pace, creating a greater pool of machine-readable data that can be shared and analyzed for strategic decision-making. The new DW concept takes a novel approach to merging the availability of these data resources containing the spatiotemporal data. This paper proposed a way to publish agricultural production data in SW using a spatiotemporal DW vocabulary called QB4MobOLAP. The data sources come from the village and rural area information system (SIDeKa), which records agricultural production transactions with spatiotemporal information. This paper also applying a new spatiotemporal data warehousing technique for analyzing spatiotemporal data for agricultural productivity. The experiment uses 2.916.864 triples with temporal type data in a fact table, and 81.914 triples dimension data with spatial and temporal data. This approach offers a practical, simple model, and good performance for enabling executive decisions on agriculture production analysis. The experiment has execution time average below 10 s for spatiotemporal aggregation and less development time compared with DBMS. This approach also has 5-stars open data index. This paper also highlighted opportunities for scaling and fostering the spatiotemporal data warehousing initiative.

Keywords: Spatiotemporal, Data warehouse, Open data, Semantic web, Agriculture production analytics.

1. Introduction

With the support of information technology, the Indonesian government's policy has focused on the development of the agricultural sector during the past decade. In Indonesia, where agricultural land is highly concentrated within villages, agricultural expansion is connected with village growth. Currently, the village community produces the most agricultural commodities. Agriculture contributes significantly to indonesia's gross domestic product (GDP). According to the world bank, the agriculture sector's GDP continued to increase from 2013 to 2021. In 2021, agriculture, forestry, fisheries, and value added accounted for 13.33 percent of Indonesia's total GDP and 139.18 billion USD.

Long-term food safety remains an ongoing challenge for many countries, including Indonesia a predominantly agricultural nation. Safe food refers to both available food and the amount of food that is required. The indonesian statistics central bureau and the agriculture ministry currently oversee and generate all official data and information pertaining to Indonesia's food resources in order to make judgments about how to best satisfy household food needs. However, these institutions lack an efficient system for gathering agricultural data and forecasting food from farming operations. In the context of sustainable food safety, many concerns involving spatial and temporal facts must be answered, such as (a) how much rice was produced

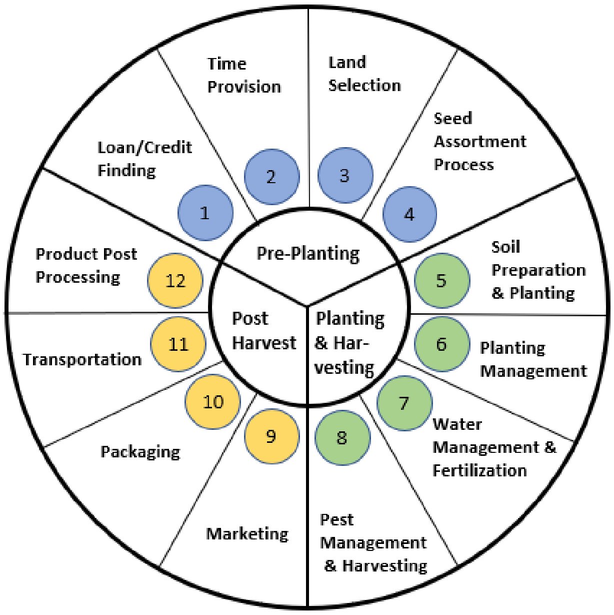

Figure. 1 Farming cycle, adapted from [1]

by farmers in some district during the month of Ramadan this year? How much seed and fertilizer will be required over the next two months? (c) On a sub-district basis, how many soybean agricultural fields failed due to pests in the past year at a given location?

From the farmer's perspective, the farming cycle is divided into three stages, which are as follows: (1) Pre-planting, (2) Planting and harvesting, and (3) Post-harvest. Figure 1. Shows the detail agricultural cycle has been discussed on [1].

The One Data initiative (https://data.go.id) pioneered the integration of agricultural data in Indonesia, although this data lacked particular location and temporal information. This data is rarely complete in terms of agriculture production progress.

Since the 5-star open data concept emerged, the government and researchers have been drawn to the concept of open data. Many governments, including the United States, the United Kingdom, France, Australia, and Indonesia, participate in open data sharing and publication. Open data government provides significant benefits in terms of transparency, citizen service, effectiveness and efficiency, innovation, and economic growth. Indonesia ranks 61st out of 94 countries, with a score of 25% on the global open data index.

Open data has a five-category index, which is as follows: (a) 1-Star: data is stored in a variety of formats, is accessible via the internet, and adheres to open license standards. (b) 2-stars: data is available in structured, machine-readable form (e.g., spreadsheet instead of image scan), (c) 3-stars: data is provided in the same format as 2-Stars and is not proprietary (e.g., CSV instead of Excel), (d) 4-Stars: Data is available as 3-Stars and conforms to the W3C open standard for identifying objects so that persons can point to them; (e) 5-Stars: data is available as 4-Stars and conforms to the Linked Open Data (LOD) standard.

This study proposes the development of a spatiotemporal DW on SW system for agricultural production analytics, which connects many data sources utilizing open data formats with a five-star rating. The DW vocabulary has been represented in an our new QB4MobOLAP, an extension of the QB4OLAP vocabulary.

This research is structured in such a way that section 2 examines related works and the state of the art in spatiotemporal DW in SW. Section 3 details the research methods used in this study. Section 4 discusses the research's discussion and findings, while section 5 concludes the article and discusses future works.

2. Related works

2.1 Data warehouse for government services

BI implementation has expanded dramatically over the previous two decades, owing to the expanding number of BI products available on the market. The DW is the central notion of BI. It serves as an organization's corporate data store. DW is defined as a technology that aides decision-makers in making tactical or strategic decisions and possesses subject-oriented, non-volatile, time-varying, and integrated qualities [2].

DW has demonstrated its ability to supply technology that aids in decision-making, notably in governmental institutions. Liu et al. developed the e-GovMon data warehouse to serve as an e-government data repository. The E-GovMon was developed utilizing open source technologies, including a PostgreSQL database management system, operational service data as a data source, and a real-time ETL tool for data acquisition, transformation, and population. Although this data warehouse is distributed, this project does not recognize the data source's shareability as open data[3]. Additionally, Maliappis et al. recommended the use of OLAP data warehouses for agricultural decision-making in Greek. The purpose of this project is to combine a variety of agricultural data sources with varying characteristics, multiple dimensions, granularity, and user needs. The authors of this study devise a method for efficiently producing information through the usage of a data warehouse. The study makes use of pentaho data integration tools for ETL procedures and an open source relational database management system for data storage [4]. Arora and Gupta present an overview of India's data warehouse implementation. The DW might aid the government in increasing efficiency and reducing costs across a range of industries, including agriculture, education, healthcare, and transportation. However, this article continues to employ isolated data and omits the concept of shareability associated with open data [5].

While the implementation of the DW in [2-5] achieves an amazing outcome in terms of providing government support for decision making, all studies still have drawbacks in terms of facilitating data sharing. Even if these studies make use of open source technology, the open notion of the data remains restricted.

2.2 Spatiotemporal data warehouse on semantic web

The SW grows with various data types, including spatiotemporal objects data. Spatiotemporal data grows due to mobile devices, GPS, IoT, network sensors, and location-based applications. The development of the spatiotemporal DW model on the SW starts from the Spatial SW, commonly referred to as Geospatial SW. Perry et al. instated that the web has much spatial and temporal data, and the technology of SW potentially could make the accessibility and usefulness of data better. Because of the lack of ability of SPARQL to query complex spatial and temporal data, Perry et al. proposed a new query language, SPARQL-ST, which is an extension of SPARQL for complex spatiotemporal queries. This research suggested a formal syntax and semantics, added spatial variables and constructs for manipulating temporal triples in SPARQL-ST, and applied them as a prototype built on a commercial DBMS [6]. This approach combines spatiotemporal SW with limited basic operations for spatiotemporal data like topological and temporal relationship. It also cannot be applied for aggregation and analytic query functions. Linked Geospatial Data in the UK government has been successfully deployed, followed by the production of geospatial datasets by the Ordnance Survey Company for areas of the UK that SPARQL Endpoints can access [7]. With the introduction of GeoNames, a linked data dataset that gathers geographical and thematic information for place names in multiple hemispheres and languages, linked data is gaining popularity. It stores longitude, latitude, altitude, population, and administrative information in accordance with the World Geodetic System 1984 (WGS84) regulations. Furthermore, LinkedGeodata is a linked data SW infrastructure converted from OpenStreetMap; this linked data is very useful for integrating and aggregating data related to maps [8] and is readily queryable via SPARQL. This technique drew the attention of several academics as a potential spatially-linked data paradigm, but it did not include temporal data.

The temporal element was then included as a spatiotemporal SW in the construction of the YAGO2 ontology, a continuation of the YAGO knowledge-based ontology in which entities, events, and facts are coupled with information about space and time. By automating Wikipedia, GeoNames, and WordNet-linked data, YAGO2 was created. At now, YAGO2 includes 447 million facts and 9.8 million entities. YAGO2 employs the SPOTL model data (5 tuples), which is SPO's expansion (3 tuples). On YAGO2, entities are given a period, whereas facts are given a time point or period (Hoffart et al., 2013). YAGO2 became a good methodology for enriching large spatiotemporal knowledge bases, but this approach did not have a query extension for this spatiotemporal representation [9].

In addition, GeoSPARQL, a query language that adds geographic aspects to SPARQL, has been created. GeoSPARQL is an SW vocabulary that enables the representation of geographical data. GeoSPARQL consists of two fundamental components: an ontology for storing geographical objects and query methods for analyzing the connections between geospatial items. The ontology is derived from OGC standards that provide expressive ideas and vocabulary. GeoSPARQL's ontology is designed to be compact for ease of comprehension and attachment. There are two primary classes in the GeoSPARQL ontology; feature and geometry. A feature is a spatially-attributed object. Feature examples include a building, statue, castle, lake, etc. Geometry is a form, such as a point, line, triangle, or hexagon, used to depict the spatial placement of a feature. SpatialObject, the third class, is a superclass of both feature and geometry. W3C has adopted GeoSPARQL as a geographical RDF standard for data modeling and querying [10]. GeoSPARQL provides several outstanding characteristics for geospatial implementation on SW. However, it does not support analytical queries and has no extension for representing and querying spatiotemporal and mobility data.

Further, EU research ran the GeoKnow project for three years, from late 2012 to 2015. This project extends the LinkedGeoData project, which develops OpenStreetMap that makes data available as an RDF base. The RDF model and GeoSPARQL standard are the foundation for describing and querying geographic data in this project. GeoKnow contributive findings are (a) the introduction of query optimization methods of geospatial RDF for better performance than existing RDF stores, even still lacking analysis performances compared to relational DBMS, (b) aggregation of geospatial RDF data with fusion [11]. The adoption of the RDF Triple store for semantic geospatial and GeoSPARQL queries developed with the advent of Parliament RDF [10].

The development of stRDF/stSPARQL is an alternative proposal comparable to GeoSPARQL. The stRDF is a data paradigm that extends RDF in order to express dynamic geographical information [12]. stRDF defines spatial and temporal dimensions using spatial and spatial literal data types. Strabon was then created as an RDF store that supports stSPARQL and GeoSPARQL, which are semantic geographic query languages. With the stRDF data model and these query languages, Strabon is expressive. The performance of strategic is scalable to enormous amounts of data and performs well [13]. stRDF/stSPARQL has included many elements of spatial data representation and query implementations, however, it lacks adoption of spatiotemporal data and analytical capabilities.

Zhang et al. suggested an addition to SPARQL for modeling spatiotemporal relationships and querying a quantitative spatiotemporal connection. The extension model adds a time- and space-based event model. The query extension produced 30 new query operators that were able to identify hidden links between items in event ontology with high speed and efficiency [14]. This method provides SPARQL capabilities extensively; however, it does not yet support the spatial-temporal analytical query.

Alternatively, Gur et al. presented QB4SOLAP, a general and extendable language (meta-model) for spatial DW on the SW. QB4SOLAP adds spatial concepts to the QB4OLAP vocabulary. In addition, they provide QB4SOLAP formalization. The essential spatial cube ideas, spatial hierarchies and levels, measurements, aggregation functions, and topological interactions between spatial dimensions and hierarchical level members have been defined in this article. In addition, they established many analytical spatial OLAP operators over QB4SOLAP, their formal semantics, and methods for creating geographically extended SPARQL queries [15]. The QB4SOLAP has a significant result for modeling and querying spatial analytical data, but this work is only limited to spatial data and not to spatiotemporal data with more dynamic characteristics.

Table 1 provides an overview of the previous model and query of spatial and new spatiotemporal DW on SW using QB4MobOLAP.

2.3 QB4MobOLAP vocabulary

QB4MobOLAP is the vocabulary for representing spatiotemporal/mobility DW on the SW. QB4MobOLAP using MobilityObject for representing spatiotemporal data. QB4MobOLAP is an extension vocabulary from QB, QB4OLAP, and QB4SOLAP. Fig. 2 depicts the QB4MobOLAP vocabulary. The QB4MobOLAP vocabulary enables DW schemas and DW instances to be defined as RDF triples. QB4MobOLAP expands the QB4OLAP vocabulary with spatiotemporal notions to enable spatiotemporal OLAP operations over RDF data via SPARQL queries. The rectangle with a gray background is a QB and QB4OLAP vocabulary extension class, subclass, and object property. These classes could be used for representing the spatiotemporal/mobility cube class. QB4MobOLAP makes an extension on built-in function QB4OLAP, which is represented with classes `qb4mob:MobilityTemporalAggrFunction` and `qb4mob:MobilitySpatialAggrFunction` in the vocabulary. The `qb4mob:MobilityTemporalAggr Func` class has instances `qb4mob:TAvg`, `qb4mob:TCount`, `qb4mob:TMin`, `qb4mob:TMax`, and `qb4mob:TSum`. The `qb4mob: MobilitySpatial-AggrFunc` class has instances `qb4mob:TCentroid`, `qb4mob:TUnion`, `qb4mob:TConvexHull`, `qb4mob: TIntersect`, and `qb4mob:TMBR`. The QB4MobOLAP also introduced classes `qb4mob:MobilityObject` for representing mobility data types. The `qb4mob:MobilityObject` has subclasses `qb4mob:MobilityBase` and `qb4mob:MobilityGeometry`.

QB4MobOLAP is a viable approach for modeling, publishing, and querying spatiotemporal DW on the SW while taking multidimensional approach into account. In this paper, we implemented QB4MobOLAP with a creation of a spatiotemporal data cube from Indonesian open agriculture production data. Additionally, we demonstrate how to leverage multidimensional spatiotemporal linked data on the SW, which involves not only adding semantics and connecting different data sets on the SW, but also enabling analytical inquiries through the use of spatiotemporal data cubes.

Received: January 28, 2023. Revised: April 12, 2023.

Table 1. Model and query comparison DW on SW

	SPARQL -ST [6]	Linked Geo Data[8]	YAGO2 [9]	Geo SPARQL [10]	stRDF/ stSPARQL [12]	SPARQL Ext. [14]	QB4 SOLAP [15]	QB4Mob OLAP [16]
MODEL								
Semantics and Ontology	√	√	√	√	√	√	√	
Feature and Geometry	√	√		√	√	√	√	
Spatial Fact				√	√		√	
Spatial Measures				√	√		√	
Spatial Hierarchy							√	
Spatiotemporal Fact								√
Spatiotemporal Measures								√
QUERY								
Topological Relationship	√	√		√	√	√	√	
Unary Operations				√	√			
Binary Operations				√	√			
Numeric Operations				√	√			
Azimuthal Relationship				√	√	√		
Temporal Relationship (Before, After, etc)	√	√	√			√		
Spatiotemporal Operations								√
Lifting operations of temporal types								√
Spatial Distributive		√					√	
Spatial Algebraic							√	
Spatial Holistic							√	
Spatial-Rollup							√	
Spatial-Slice							√	
Spatial-Dice							√	
Spatial-Drill-down							√	
Spatiotemporal-Rollup								√
Spatiotemporal -Slice								√
Spatiotemporal -Dice								√

3. Research methodology

The methodology of our research is a constructive research. This constructive research concentrated on the development of methodologies, models, modules, tools, and approaches that were applicable beyond the realistic study that inspired them [22]. The research activities are depicted in Fig. 3.

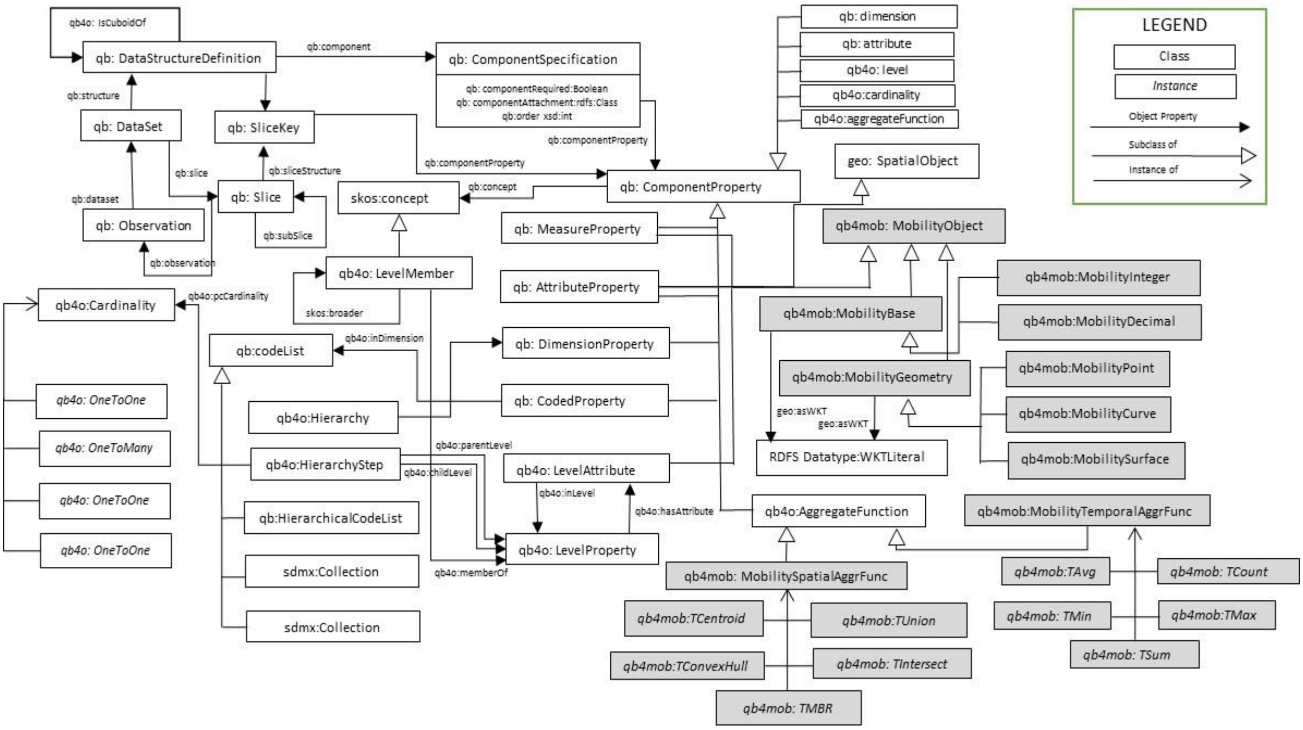

Figure. 2 QB4MobOLAP vocabulary

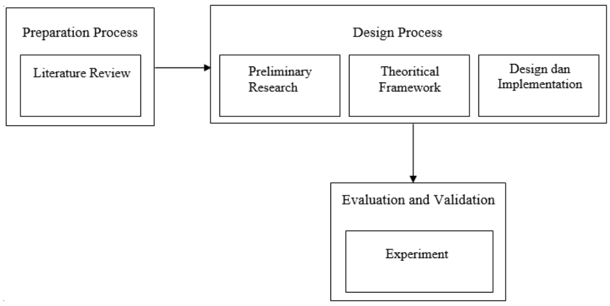

Figure. 3 The research activities

The step of the research are the following:

1. Preparation process: Preparation process held by literature review that reported in [17]
2. Design process: The design process of the research comprised of the step: (a) Preliminary research, (b) Theoretical framework, and (c) Design and implementation. The design and implementation were conducted by developing spatiotemporal DW vocabulary on SW.
3. Evaluation and validation: Evaluation and Validation is the process to validate the model/vocabulary, the query operators, and functions proposed in the previous step. The validation applied by implementing them to a practical use case. A detailed performance study of our implementation also conducted using large synthetic and real-world RDF datasets.

This paper is in the evaluation and validation phase of the research. This phase has a cycle related to development and evaluate the query performance, in the following manner: (1) Requirement analysis including identification, specification, refinement, and prioritizing of user requirements; (2) Design the DW by creating a snowflake/star model DW including fact tables, measurements, and dimensions. Additionally, the design process incorporates spatiotemporal measures and dimensions; (3) Extract, Transform, Load, and Link (ETLL) the data into an RDF DW using the QB4MobOLAP vocabulary; (4) Implement the Spatiotemporal OLAP query for the RDF DW; and evaluate both execution and development time of the spatiotemporal OLAP query.

4. Results and discussions

The creation of the spatiotemporal DW begins with a requirement study. The following steps are involved in the requirement analysis process:

4.1 Requirement analysis

Identification, specification, refinement, and prioritizing of user requirements including the process of identification of user group, user, and goal prioritization. The user and group of users in this case are government, industry, and civil community. The priority goal of the DW implementation is to support decision making for strengthening the food availability and handling of food safety. The decision making could be supported by descriptive and diagnostic analytics with some queries. The detail queries shown in the next section.

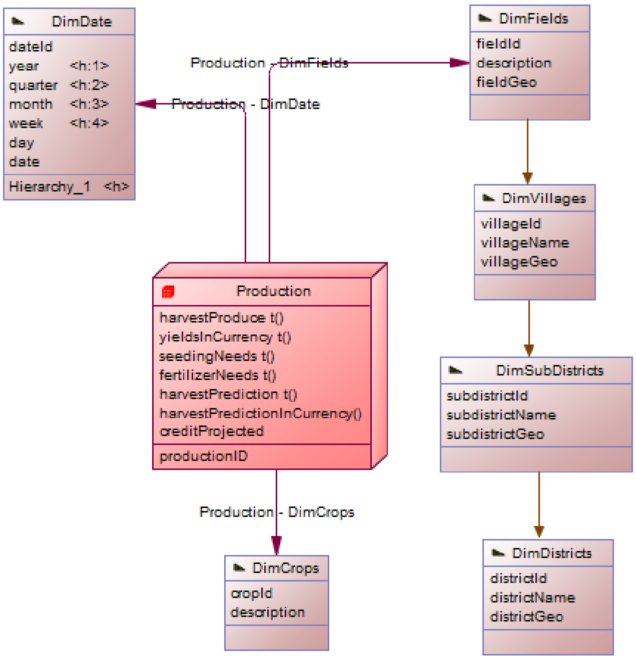

Figure. 4 The AgriProd data warehouse relational schema

4.2 DW design

The spatiotemporal DW design, including facts, measures, and dimension shown in Fig. 4. This design introduces new temporal type measures and spatial dimensions with hierarchy. The _Geo postfix indicates a spatial attribute that contains a collection of location or geographic information.

4.3 The extract, transform, load, and link process

Extract, transform, load, and link (ETLL) is the process for integrating disparate data sources and loading them into enterprise DW.

Data source

The data source is open transactional data from the village information system (SIDeKa), which may be accessed via REST API web services running on top of the SIDeKa server. The web service is written in PHP and is based on the CodeIgniter framework. Each SIDeKa server has a number of unique endpoints. Each endpoint will be responsible for a single function, which will return a value or set of values from a transactional database in SIDeKa via JSON formats.

Cube definition in QB4MobOLAP

QB4MobOLAP vocabulary has modeled a spatiotemporal/mobility DW in RDF terms. DW has observation facts that is central to multidimensional analysis, the DW is called cube. Facts have a set of properties called measures, which have aggregate functions that can be defined. To provide alternative analytic viewpoints, a multidimensional data cube has *n* dimensions (with the attribute of contextual information) on a multidimensional space, where each dimension might have hierarchies with levels. Users can aggregate measures (of facts) at multiple levels of detail by using levels (of dimensions) (a.k.a. granularity). Dimensions and facts are tied to the structure to assist this analysis at various detail levels. Levels also include attributes that describe the basic features of the members of the level.

These multidimensional concepts and their roles and relationships are annotated at the schema level in RDF triple format using the QB4OLAP vocabulary. Schema level (RDF) multidimensional elements correspond to tables and columns (of tables). Fact member with measure values and level members with attribute values are instance-level concepts annotated at the instance level with QB4MobOLAP Vocabulary. These instance level (RDF) cube elements correspond to the actual data rows/records annotated in triple format with QB4MobOLAP.

An RDF triple t is made up of three parts: the subject (s), the predicate (p), and the object (o), which is defined as triple $(s,p,o) \in t = (I \cup Bn) \times I \times (I \cup Bn \cup L)$, where I is the set of IRIs, Bn is the set of blank nodes, and L is the set of literals. The DW using Cube Schema (CS) with element x defined as, $CS(x) \in (I \cup Bn \cup L)$ yielded a set of triples T and denoted by (`x rdf:type ex:Property`).

DW Schema

A data structure definition (DSD) contains information about the schema of a data set (i.e., a data warehouse, an instance of the class `qb:DataSet`). The DSD can be applied to a variety of data sets. A data set's DSD includes dimensions, levels, measures, and attributes, in addition to component properties. As CS = DSD, the DSD is defined by a conceptual MD cube schema CS, which includes a collection of dimension types D, a collection of measure M, and a collection of fact types as CS = (D, M, F). The following definitions apply to the cube schema elements:

Attributes. An attribute $a \in A = \{a_1, a_2, . . . , a_n\}$ has a domain a:dom in the cube schema CS with a set of triples $t_a \in T$ where t_a is represented as (a rdf:type qb:AttributeProperty; rdfs:domain xsd:Schema).

Levels. A level $l \in L = \{l_1, l_2, . . . , l_n\}$ consists of a set of attributes A_l, which is defined by a schema $l(a_1: dom_1, . . . , a_n: dom_n)$, where l is the level, and each attribute a is defined over the domain dom. For each level $l \in L$ in the cube schema CS, a set of triples tl ∈ T represented as (`l rdf:type qb4o:LevelProperty; qb4o:hasAttribute a`).

Dimensions. A set of dimensions $D = \{d_1, d_2, \ldots, d_n\}$ form a cube schema with n dimensions. Each d ∈ D contains a tuple d = (L, H); where L is a Level and H is a hierarchy. In the cube schema CS, each dimension d ∈ D has a set of triples t_d ∈ T, which is defined as (`d rdf:type qb:DimensionProperty; qb4o:hasHierarchy h`). Example 1 gives instances of level, example 2 gives instances of hierarchy, and example 3 gives dimensions example instances in Agriculture Production DW.

Examples 1

```
agri:districts a
qb4o:LevelProperty;
      qb4o:hasAttribute
agri:districtId;
      qb4o:hasAttribute
agri:districtName;
      qb4o:hasAttribute
agri:districtGeo.
```

Examples 2

```
  #Hierarchies:
  agri:LocationGeography   rdf:type
  qb4o:hierarchy;
    qb4o:inDimension
agri:DimDistrict;
  qb4o:hasLevel agri:field,
  agri:village,   agri:subdistrict,
     agri:district.
```

Example 3

```
#Dimensions:
agri:DimDistrict   rdf:type
qb:DimensionProperty;
qb4o:hasHierarchy
agri:LocationGeography.
agri:DimCrops              rdf:type
qb:DimensionProperty .
agri:DimDate               rdf:type
qb:DimensionProperty;
qb4o:hasHierarchy
agri:timeHierarchy.
```

Measures. Facts have a property called measures. A measure $m \in M = \{m_1, m_2, \ldots, m_n\}$ in the cube schema CS is represented as (`m rdf:type qb:MeasureProperty; rdfs:subPropertyOf sdmx-measure:obsValue; rdfs:domain xsd:schema`).

Facts. Facts are associated with values of dimensions and measures. A fact denotes as $f \in F = \{ f_1, f_2, \ldots, f_n\}$. The relation of the facts is described in components in the schema level of the facts cube definition, which is determined by (`F rdf:type qb:DataStructureDefinition; qb:component[qb4o:level l; qb4o:cardinality c]; qb:component[qb:measure m; qb4o:aggregateFunction AF]`). Cardinality, $c \in \{1-1, 1-*, *-1, *-*\}$ specifies the relationship cardinality between facts and level members. The definition of the cube schema requires the specification of measures aggregate functions. QB4OLAP defines typical aggregate functions, for example, AF ∈ {Sum, Avg, Count, Min, Max}. The facts are presented at the instance level, with each fact f having a unique IRI I, an observation. The fact is represented as (`f rdf:type qb:Observation`).

Spatial Attributes. Spatial attributes are specified on a domain level. Each attribute with geometry domain (sa : domgeo ∈ A) is a member of the `geo:Geometry` class and is referred to as a spatial attribute, which is represented in the cube schema CS as (`sa rdf:type qb:AttributeProperty; rdfs:domain geo:Geometry`). In the instances, each spatial attribute type (point, polygon, line, etc.) is assigned using the predicate `rdfs:range`.

Spatial Levels. In the cube schema CS, a spatial level ls ∈ L is defined by a set of triples t_{ls} ∈ T and represented as (`ls rdf:type qb4o:LevelProperty; qb4o:hasAttribute a, as; geo:hasGeometry geo:Geometry`). Spatial levels must be `geo:Geometry` class members and may have spatial attributes. Example 4 shows District level as a spatial level with spatial attribute.

Example 4

```
#Attributes
agri:DistrictId            rdf:type
qb4o:levelAttribute;
   qb4o:inLevel agri:District;
   rdfs:range xsd:Integer .
agri:DistrictName          rdf:type
qb4o:levelAttribute ;
   qb4o:inLevel agri:District;
```

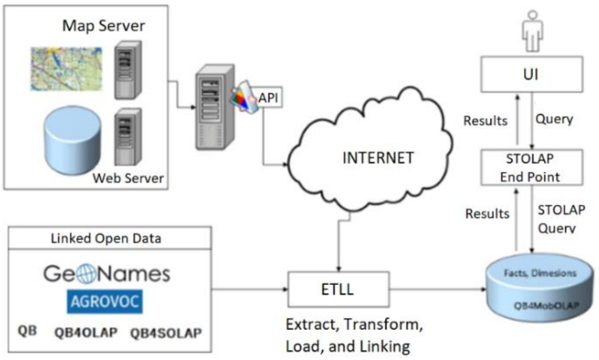

Figure. 5 The deployment diagram of AgriProd data warehouse

```
   rdfs:range xsd:String .
agri:plantationGeo        rdf:type
qb4o:levelAttribute ;
   qb4o:inLevel agri:District;
   rdfs:subPropertyOf
geo:Geometry;
   rdfs:range geo:wktLiteral;
   rdfs:domain geo:Polygon;
   rdfs:subClassOf
geo:SpatialObject;
```

Spatiotemporal Measures. A spatiotemporal measure stm ∈ M is specified in the cube schema CS by a set of triples t_{stm} ∈ T and encoded as (`stm rdf:type qb:MeasureProperty; rdfs:subPropertyOf sdmx-measure:obsValue; rdfs:domain qb4mob:MobilityBase/MobilityGeometry`). The class of the numeric value is given with the property `rdfs:domain` and `rdfs:range` assigns the values from the class `qb4mob:MobilityBase/ mobilityPoint`, i.e., temporal integer, temporal float, temporal point, temporal polygon, etc. at the instance level.

Spatiotemporal Facts. Spatiotemporal facts F_{st} have spatiotemporal nature relates several dimensions either spatial or temporal. Spatiotemporal fact cube has spatiotemporal measures (m_{st}), as its members make it possible to aggregate along with spatiotemporal measures with the aggregation functions M_{agg}. Representation of a complete mobility fact cube at the schema level in RDF is given by a set of triples t_{fs} ∈ T and encoded as (`fs a qb:DataStructureDefinition; qb: component [qb: measure ms, sdmx-measure:obsValue; qb4mob:aggregate-functionMAF]`).

QB4MobOLAP extends the built-in functions of QB4OLAP with mobility aggregation functions, which is added with a class `qb4mob:MobilityTemporalAggrFunc` or `qb4mob:MobilitySpatialAggrFunc`. Example 5 display the creation of a spatiotemporal fact instance f_{st} with its relation to measuring values and dimension levels (`agri:harvestProduce`), which is production rate that change by time. Specification of the spatiotemporal/mobility aggregate function for harvestProduce (production rate by time) can be specified as the "Time Average/TAvg".

Example 5

```
#Measures
agri:HarvestProduce        rdf:type
qb:MeasureProperty;
    rdfs:domain
qb4mob:MobilityDecimal;
    rdfs:range geo:wktLiteral;
    rdfs:subClassOf        qb4mob:
MobilityObject .
agri:YieldsInCurrency      rdf:type
qb:MeasurePro-perty;
    rdfs:domain
qb4mob:MobilityDecimal;
    rdfs:range geo:wktLiteral;
    rdfs:subClassOf
qb4mob:Mobility1Object .
agri:CreditProjected       rdf:type
qb:MeasurePro-perty;
    rdfs:range      xsd:decimal.
```

Example 6

```
#Cube Definition
agri:AgriProduction        rdf:type
qb:DataStructureDefinition ;
#Lowest level for each dimension
qb:component           [qb4o:level
agri:fields;cardinality
qb4o:ManyToOne];
qb:component           [qb4o:level
agri:crops;
cardinality qb4o:ManyToOne];
qb:component           [qb4o:level
agri:date      ;     cardinality
qb4o:ManyToOne];
#Cube measures
qb:component           [qb:measure
agri:creditProjected;
qb4o:aggregateFunction qb4o:Sum];
qb:component           [qb:measure
agri:HarvestProduce;
qb4o:aggregateFunction
qb4mob:TSum] ;
```

```
  qb:component          [qb:measure
  agri:YieldsInCurrency;
  qb4o:aggregateFunction
  qb4mob:TSum];
  qb:component          [qb:measure
  agri:harvestPredictionInCurrency;
  qb4o:aggregateFunction
  qb4mob:TSum];        qb:component
  [qb:measure agri: SeedingNeeds ;
  qb4o:aggregateFunction
  qb4mob:TSum ];
```

Linking process

The deployment of the Linked Open Spatiotemporal DW is depicted in Fig. 5. The DW system deploys on GraphDB with QB4MobOLAP vocabulary. QB4MobOLAP could represent temporal and spatiotemporal data types and functions for spatiotemporal analytics.

The ETL process import the data from sources using API, analyzed, linked, and publishing it. Some external vocabulary used in the linking and publishing process like WGS84 for defining spatial objects, Geonames for external linking of place names, and AGROVOC for representing crops. The example of the external linked to represent the place name Sleman district in Indonesia to GeoNames URI of Sleman with `owl:sameAs` predicate.

```
agrii:district_501 a
agri:districts;    qb4o:memberOf
agri:districts; agri:districtId
501; agri:districtName "Sleman";
agri:districtGeo "POLYGON
((109.2299989 -7.3350438,
109.2303208 -
7.3354481,…………………,109.2299989 -
7.3350438)) "^^geo:wktLiteral;
owl:sameAs geonames:1626754 .
```

The instance example of the Production fact is:

```
agrii:prod_315999 a qb:Observation;
  agri:cropId agrii:crop_3;
  agri:startDateId
agrii:date_20190520;
  agri:endDateId
agrii:date_20190830;  agri:fieldId
agrii:field_9;  agri:harvestProduce
"TINT{12@2020-03-02 00:00:00+07,
5.5@2020-03-09 00:00:00+07,
20@2020-03-16
00:00:00+07}"^^geo:wktLiteral;
  agri:yieldsInCurrency
"TINT{12000@2020-03-02 00:00:00+07,
5500@2020-03-09 00:00:00+07,
20000@2020-03-16
00:00:00+07}"^^geo:wktLiteral;
  agri:harvestPredictionInCurrency
"TINT{12000@2020-03-02 00:00:00+07,
5500@2020-03-09 00:00:00+07,
20000@2020-03-16
00:00:00+07}"^^geo:wktLiteral;
agri:creditProjected 5000 .
```

4.4 Implement spatiotemporal OLAP query for the RDF cube

From the user requirement, it could be captured some question for spatiotemporal agriculture production analytics. Those queries are:

Query#1: *Determine the amount of corn, soybeans, and rice produced during the fasting month and Eid al-Fitr in the selected location of sub-districts this year.* This query involves spatotemporal roll-up along the crops dimensions. Then, the dice operation on crops and time dimensions, and spatial slice over certain geographic area. The STOLAP query on the Linked ODW is:

```
SELECT
?desc (AVG(jsfn:TAVG(jsfn:AtPeriod
(?harvestproduce,'2019-05-01',
'2020-06-01')))
as ?tavgharvestproduct)
WHERE {
?obs a qb:Observation ;
    agri:fieldId ?fieldId ;
    agri:cropId ?cropId ;

agri:harvestProduce ?harvestproduce
.
?cropId qb4o:memberOf agri:crops;
    agri:description ?desc .
?fieldId qb4o:memberOf agri:fields;
   agri:fieldGeo ?fieldGeo .
FILTER ((?desc = 'Rice') || (?desc
= 'Corn') || (?desc = 'Soybeans')
&& (geof:sfWithin (?fieldGeo,
"POLYGON ((109.2305407 -7.3356011,
109.2318389 -7.3378251, 109.2334375
-7.3370909, 109.2333838 -7.3367291,
109.2333195 -7.3364099, 109.23318 -
7.3359629, 109.2331049 -7.3355586,
109.2329547 -7.33525, 109.2328796 -
7.3350159, 109.2327508 -7.3348669,
109.2325899 -7.3346222, 109.2305407
-7.3356011))"))) }
GROUP BY ?desc
```

Query#2: *Determine the average crop potential agricultural loss (in monetary terms) incurred by the Sleman district as a result of Mount Merapi's eruption on January 21, 2020.* The impacted area is anticipated to have a radius of 5 kilometers. This query involves spatotemporal roll-up along the crops dimensions. Then, the dice operation on crops and time dimensions, and spatial function within district and field area. The STOLAP query on the Linked ODW is:

```
SELECT ?desc
(SUM((jsfn:TAVG(jsfn:AtPeriod
(?harvestpredictionInCurrency,'2020
-01-21', '2020-03-21')))-
(jsfn:TAvg(jsfn:AtPeriod
(?yieldsInCurrency,'2020-01-21',
'2020-03-21'))))
as ?avgharvestlostprediction)
WHERE {
?obs rdf:type qb:Observation ;
     agri:fieldId ?fieldId ;
     agri:cropId ?cropId ;

agri:harvestPredictionInCurrency ?h
arvestpredictionInCurrency ;

agri:yieldsInCurrency ?yieldsInCurr
ency .
?cropId qb4o:memberOf agri:crops;
    agri:description ?desc .
?fieldId qb4o:memberOf agri:fields;
      agri:fieldGeo ?fieldGeo .
?districtId qb4o:memberOf
agri:districts;

agri:districtGeo ?districtGeo ;
      agri:districtName 'Sleman' .
    FILTER ((?desc = 'Rice') ||
(?desc = 'Corn') || (?desc =
'Soybeans') && (geof:sfWithin
(?fieldGeo, ?districtGeo)) &&
(geof:distance('POINT (110.445671 -
7.540566)', ?fieldgeo) < 5000))}
GROUP BY ?desc
```

Query#3: *Determine how much credit prepared by the bank at per the quarter year 2019 in subdistrict ’Kalasan’?* This query involve roll-up along the time dimensions. Then slice by subdistricts, and spatial function within subdistricts and fields area. The STOLAP query on the Linked ODW is:

```
SELECT ?quarter (SUM
(?creditprojected)
as ?sumcreditprojected)
WHERE {?obs rdf:type
qb:Observation ;
    agri:startDateId ?dateId ;
    agri:fieldId ?fieldId ;
agri:creditProjected ?creditproject
ed .
?fieldId qb4o:memberOf agri:fields;
       agri:fieldGeo ?fieldGeo .
?dateId qb4o:memberOf agri:date;
        agri:quartal ?quarter ;
        agri:year ?year .
?subdistrictId qb4o:memberOf
agri:subdistricts;

agri:subdistrictGeo ?subdistrictGeo
;

agri:subdistrictName ?subdistrictNa
me .
FILTER ((?year = 2019) &&
(?subdistrictName = 'Kalasan') &&
(geof:sfWithin
(?fieldGeo, ?subdistrictGeo)))
}
GROUP BY ?quarter
```

Query#4: *Determine the nearest village that may fill the shortfall in corn, soybean, and rice output for selected communities during 2020's fasting month.*

```
SELECT ?fieldGeo
(jsfn:TWAVG(jsfn:TSUM( jsfn:ATPERIO
D(?harvestproduct, '2020-06-01’,
'2020-06-30')))
as ?avgharvestproduct)
WHERE {?obs rdf:type
qb:Observation ;
agri:cropId ?cropId
     agri:fieldId ?fieldId .
?cropId qb4o:memberOf agri:crops;
        agri:cropdesc ?cropdesc .
?fieldId qb4o:memberOf agri:fields;
       agri:fieldGeo ?fieldGeo .
?subdistrictId qb4o:memberOf
agri:villages;

agri:villageGeo ?villageGeo;

agri:villageName ?villageName .
FILTER ((?villageName =
'Karangtengah') && (geof:sfWithin
(?fieldGeo, ?villageGeo)) &&
(?cropdesc = 'Rice') || (?cropdesc
= 'Corn') || (?cropdesc =
'Soybeans'))
GROUP BY ?fieldGeo
```

Query#5: *Calculate the fertilizer and sowing material requirements for corn, soybean, and rice agriculture in the Bantul district during the first quarter of 2020.*

```
SELECT ?cropdesc
(jsfn:TWAVG(jsfn:TSUM( jsfn:ATPERIO
D(?fertilizerneeds, '2020-01-06’,
'2020-06-06'))
as ?avgfertilizerneeds)
WHERE {?obs rdf:type
qb:Observation ;
      agri:fertilizerneeds ?fertiliz
erneeds ;
agri:dateId ?dateId ;
agri:cropId ?cropId ;
agri:fieldId ?fieldId .
?fieldId qb4o:memberOf agri:fields;
        agri:fieldGeo ?fieldGeo .
?cropId qb4o:memberOf agri:crops;
         agri:cropdesc ?cropdesc .
?subdistrictId qb4o:memberOf
agri:subdistricts;

agri:subdistrictGeo ?subdistrictGeo
;

agri:subdistrictName ?subdistrictNa
me .
FILTER ((?subdistrictName =
'Bantul') && (geof:sfWithin
(?fieldGeo, ?subdistrictGeo) &&
(?cropdesc = 'Rice') || (?cropdesc
= 'Corn') || (?cropdesc =
'Soybeans' ) )
GROUP BY ?cropdesc
```

5. Evaluation and validation

For evaluating the performance of our approach, measurement and presentation of the total time to get similar results from the RDF data in two different non-SW environments. The results are presented under quantitative evaluation. We also compare the open data index as the qualitative evaluation of two different environments. Finally, we discuss the technical lessons and summarize our work.

The implementation scenario is comparing the query run times of the algorithms in QB4MobOLAP with different query platforms ORDBMS tool. Since these tools cannot process RDF data natively, data preparation and load times considers as development costs. Development cost is given in hours, which involves, extracting the data, loading into these (non-SW) environments in their native format. Assumption has been made that the developer has a basic knowledge of the domain, data set, the schema of the data set, and able to extract data with SPARQL queries, can perform SQL queries on ORDBMS to get similar results as in the algorithms. The development cost excludes the preparation (downloading and installing) of the environments. The development cost of QB4MobOLAP is a configuration set-up, where the user should point to the SPARQL endpoint where the instance triples are located and specify the RDF cube schema of the use case data.

Table 2. Queries execution time, development cost, and open data index

	SW			ORDBMS		
Id	Exec. Time (s)	Dev. Cost (min)	Open Data Index	Exec. Time (s)	Dev. Cost (min)	Open Data Index
Q1	6.7	5	5	0.33	60	NA
Q2	10	5	5	0.42	60	NA
Q3	13	5	5	0.55	60	NA
Q4	22	5	5	14	60	NA
Q5	5.2	5	5	0.57	60	NA

The Linked ODW fact contains around 3 million triples with temporal integer types for some triples, 4495 triples of field dimension with geographic data as polygons, 80 triples of village dimension, dozens of triples of the subdistricts, and districts. The DW time dimension contains 3653 triples representing a date for ten years.

The five queries were executed on pentium core i5 8th generation with a 16GB RAM computer. The results of the execution test time of the five queries, development cost, and open data index are shown in Table 2.

Fig. 6 and Fig. 7 show the comparison of execution time and development time in SW and ORDBMS. ORDBMS have less significant execution time compared with SW, but for overall SW has less development time and have the best open data index. The development time equals execution time plus development cost.

Compared to ORDBMS environment, The execution time of the QB4MobOLAP platform on Agriculture Production DW still has limitation. The execution time takes longer time than the ORDBMS environment, as shown Fig. 6. However, the QB4MobOLAP platform has better development time than the ORDBMS platform because the ORDBMS environment must preprocess the data, which entails extracting and loading the data into non-SW environments in their native format. This data preprocessing takes 60 minutes to complete. The QB4MobOLAP platform also has the highest open data index, which is important for the public to make innovations based on that data. Fig. 7 depicts the

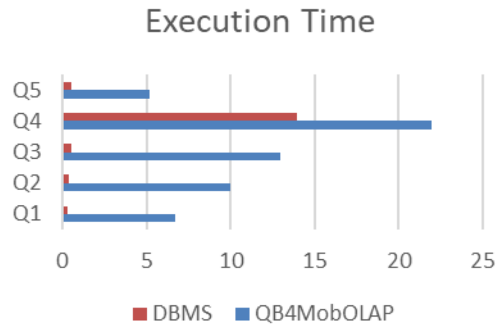

Figure. 6 Query execution time

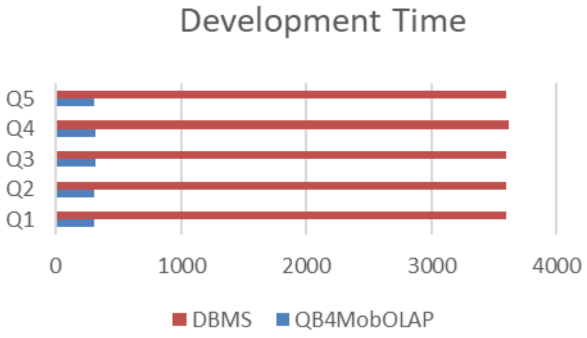

Figure. 7 Query development time

development time, the sum of execution time with development cost.

6. Conclusion and future works

The implementation of this open spatiotemporal data for production analytics was designed utilizing a novel idea of spatiotemporal data warehouse and the Semantic Web's QB4MobOLAP vocabulary. Using spatiotemporal queries for a roll-up, a drill-down, a slice, a dice, a geographical aggregation function, and a temporal aggregation function, this system was able to answer the important issues for agricultural production analysis in a acceptable amount of time. The query result is useful for descriptive and diagnostic analyses of agricultural productivity. The openness of the data source with the 5-stars index provides the chance to integrate and expand the data source with the internet's big data catalog and the semantic web's inference capacity.

Experiments shown that QB4MobOLAP vocabulary for spatiotemporal DW implementation is practical and straightforward. The solution provides important support for spatiotemporal analytics. Representation of DW in QB4MobOLAP vocabulary, integration with another linked data will boost the system's potential with a larger library and standardization of application domain words and ideas [15]. Using this linked data as a data source, the data warehouse may be augmented with the new semantic and method to give more information for reasoning based on logical axioms throughout the full business intelligence and big data warehousing process.

Conflicts of interest

The authors declare no conflict of interest

Author contributions

Conceptualization, methodology, writing—original draft preparation, writing—review and editing, Irya Wisnubhadra; methodology, review, editing, and supervision, Safiza Suhana Kamal Baharin; review and supervision, Nurul Akmar Emran; review and supervision, Djoko Budiyanto SHR;

Acknowledgments

This research is part of a thesis research at University Teknikal Malaysia Melaka.

References

[1] I. Wisnubhadra, S. P. Adithama, S. S. K. Baharin, and N. S. Herman, “Agriculture Spatiotemporal Business Intelligence using Open Data Integration”, In: *Proc. of 2019 2nd International Seminar on Research of Information Technology and Intelligent Systems*, ISRITI 2019, pp. 534–539, 2019.

[2] W. H. Inmon, “Building the Data Warehouse”, *John Wiley & Sons*, 2002.

[3] X. Liu and X. Luo, “a Data Warehouse Solution for E-Government”, *International Journal of Recent Research and Applied Studies*, Vol. 4, No. 1, pp. 101–105, 2010.

[4] M. Maliappis and D. Kremmydas, “An online analytical processing (OLAP) database for agricultural policy data: A Greek case study”, *CEUR Workshop Proc*, Vol. 1498, No. September, pp. 214–225, 2015.

[5] R. K. Arora and M. K. Gupta, “e-Governance using Data Warehousing and Data Mining”, *Int J Comput Appl*, Vol. 169, No. 8, pp. 28–31, 2017.

[6] M. Perry and A. P. Sheth, “SPARQL-ST: Extending SPARQL to Support Spatiotemporal Queries”, *Geospatial Semantics and the Semantic Web*, pp. 61–86, 2011.

[7] K. Bereta, P. Smeros, and M. Koubarakis, “Representation and Querying of Valid Time of Triples in Linked Geospatial Data”, *Lecture Notes in Computer Science*, Vol. 7882, 2013.

[8] C. Stadler, J. Lehmann, K. Höffner, and S. Auer, “LinkedGeoData: A core for a web of spatial

open data", *Semantic Web,* Vol. 3, No. 4, pp. 333–354, 2012.

[9] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2 : A spatially and temporally enhanced knowledge base from Wikipedia", *Artificial Intelligence*, Vol. 194, pp. 28–61, 2013.

[10] R. Battle and D. Kolas, "GeoSPARQL: Enabling a Geospatial Semantic Web", *Semantic Web Journal*, Vol. 0, No. 0, pp. 1–17, 2012, doi: 10.3233/SW-2012-0065.

[11] A. Both, A. G. Rojas, M. Wauer, D. Hladky, and J. Lehmann, "The GeoKnow generator workbench -An integrated tool supporting the linked data lifecycle for enterprise usage", In: *CEUR Workshop Proc.*, Vol. 1481, No. Ld, pp. 92–95, 2015.

[12] I. Linked and G. Data, "Part 2: Spatial and Temporal Data in RDF: stRDF/stSPARQL and GeoSPARQL ESWC 2015 Tutorial Publishing and Interlinking Linked Geospatial Data", 2015, [Online]. Available: https://event.cwi.nl/eswc2015-geo/03-stsparql-geosparql.pdf

[13] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis, "Strabon : A Semantic Geospatial DBMS", *In The Semantic Web – ISWC 2012*, pp. 295–311, 2012.

[14] Y. Zhang and F. Xu, "A SPARQL Extension with Spatial-Temporal Quantitative Query", In: *Proc. of 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 554–559, 2018.

[15] N. Gür, "Modeling, Annotating, and Querying Geo-Semantic Data Warehouses", *Ph.D Thesis, Aalborg University Denmark*, 2020.

[16] I. Wisnubhadra, S. K. Baharin, N. A. Emran, and D. B. Setyohadi, "Qb4mobolap: A vocabulary extension for mobility olap on the semantic web", *Algorithms*, Vol. 14, No. 9, 2021.

[17] I. Wisnubhadra, S. S. K. Baharin, and N. S. Herman, "Modeling and querying spatiotemporal multidimensional data on semantic web: A survey", *J Theor. Appl. Inf. Technol.*, Vol. 97, No. 23, pp. 3608–3633, 2019.