International Journal of Engineering and Technology Innovation, vol. 8, no. 3, 2018, pp. 157- 172

Distributed Database Semantic Integration of Wireless Sensor Network to Access the Environmental Monitoring System

Ubaidillah Umar, M. Udin Harun Al Rasyid*, Sritrusta Sukaridhoto

Electronics Engineering Polytechnic Institute of Surabaya (EEPIS), Indonesia

Received 21 May 2017; received in revised form 01 January 2018; accepted 05 February 2018

Abstract

A wireless sensor network (WSN) works continuously to gather information from sensors that generate large volumes of data to be handled and processed by applications. Current efforts in sensor networks focus more on networking and development services for a variety of applications and less on processing and integrating data from heterogeneous sensors. There is an increased need for information to become shareable across different sensors, database platforms, and applications that are not easily implemented in traditional database systems. To solve the issue of these large amounts of data from different servers and database platforms (including sensor data), a semantic sensor web service platform is needed to enable a machine to extract meaningful information from the sensor's raw data. This additionally helps to minimize and simplify data processing and to deduce new information from existing data. This paper implements a semantic web data platform (SWDP) to manage the distribution of data sensors based on the semantic database system. SWDP uses sensors for temperature, humidity, carbon monoxide, carbon dioxide, luminosity, and noise. The system uses the Sesame semantic web database for data processing and a WSN to distribute, minimize, and simplify information processing. The sensor nodes are distributed in different places to collect sensor data. The SWDP generates context information in the form of a resource description framework. The experiment results demonstrate that the SWDP is more efficient than the traditional database system in terms of memory usage and processing time.

Keywords: environment monitoring, semantic database, distributed data, wireless sensor network, semantic web

1. Introduction

Air pollution has attracted many researchers in the past few years. There has been significant public concern surrounding serious community health risks (including heart disease, chronic obstructive pulmonary disease, stroke, and lung cancer) and their strong associations with air pollution. People breathing air of poor quality may suffer from difficulty breathing, coughing, wheezing, and asthma. In addition to human health, air pollution has a major influence on the global environment and economies around the world. Acid rain, haze, and global climate change are caused by air pollution [1]. The activities of a rapidly growing industry have recently resulted in high concentrations of carbon monoxide (CO) and carbon dioxide (CO_2), which are very dangerous for human life. Oxygen (O_2) in the air is carried by red blood cells to the tissues of the body through the respiratory system. When CO gas is absorbed by the red blood cells, the body will lack O_2. Damage to the central nervous and cardiovascular systems occurs when the absorption of CO takes place over a period of time and may result in symptoms such as headache, drowsiness, weakness, dizziness, nausea, and fainting. Higher concentrations of CO can cause increased heart rate, heart failure, coma, and impaired respiratory function. Furthermore, high concentrations of CO_2 in the air have the effect of trapping hot air inside the atmosphere, resulting in a rise in the earth's temperature [2].

* Corresponding author. E-mail address: udinharun@pens.ac.id

Tel.: +6285645065911; Fax: +62315946114

A wireless sensor network (WSN) comprises small sensor devices used to collect and send data about environmental conditions including CO, CO2, O2, temperature, and humidity. The sensor nodes of the WSN are spatially distributed to facilitate the accurate monitoring and controlling of the physical conditions of the environment from remote locations. WSNs use dynamic sensors where the device may appear and disappear in a short period of time and even move to another destination. WSNs continue to gather information that combine to form large sets of sensory data. Different large sensory data obtained have problems too much data but not enough knowledge, lack of integration from different database sources, and communication between these networks [3-5].

Semantic web uses representation languages, such as resource description framework (RDF) and web ontology language (OWL), to provide a formal description of data and knowledge. One of the semantic web goals is to provide intelligent search agents capable of processing and integrating data from heterogeneous resources—such as sensor networks—at a conceptual level. Semantic web is defined as an extension of the current web, where information is presented with well-defined meanings, and allows computers and people to work together. The system is a powerful method for users to filter information and large product space. The semantic web is used as background knowledge to extract data mining features that can improve the results of recommendations. The use of semantic web in sensor networks enables the discovery and analysis of sensor data based on spatial, temporal, and thematic information [6-10].

Many related projects and researches have actively progressed semantic services for sensor data. Rohloff et al. [11] performed a comparison of various triple-store technologies (such as Sesame, Jena, and AllogroGraph) to load data and respond to queries based on ontology, datasets, and standard queries of the Lehigh University Benchmark (LUBM) software tools. They used various metrics (such as cumulative load time, query response time, query completeness, and disk-space requirements) to show the performance. They found that the triple-stores based on the Sesame+DAML DB, Jena+DAML DB, and Sesame+BigOWLIM exhibited the best performance among those tested. Gray et al. [12] proposed a semantic web architecture for integrating different datasets such as sensor data, database, and map layers. Semantic Web service technology is used for querying, accessing, discovering, and integrating datasets of the flood response planning web application. This application (based on semantic sensor web) still needs to be improved to include a tool for selecting two sources of data discovered through the semantic registry and to dynamically request data from the semantic integration service. Moraru et al. [13] proposed a system for publishing data gathered from a sensor network comprised of sensing devices that monitor environmental conditions such as humidity, temperature, pressure, and luminance. They used a single sensor source and added a semantic layer for enriching the obtained sensor data properties by including the location of a sensor and environmental phenomena.

Chenzhou et al. [14] proposed a semantic web of things framework that utilizes multiagent technology to facilitate human-to-human and human-to-machine collaboration. This study takes health as an example domain to describe the details of an active data-based semantic framework. The proposed framework is organized into three layers. The top layer is for presentations that contain active related data on different types of things, actions, and people. The lower layer represents the corresponding real-world objects—including people as active human-centric app participants—and features sensors and smart devices as the IoT infrastructure. The middle layer is used to connect the top and bottom layers, consisting of all the IoT device amplifiers required for data integration and functionality, as well as multiple agents running in a multiagent platform for service activation. Their research shows how technology and architecture significantly enhance the Web of Things capability for human-centric applications. Choi et al. [15] proposed a distributed semantic sensor web architecture called “semantic sensor web platform” to provide a user-centric, context-aware, semantic web service. The goal of the semantic sensor web platform is to separate the processing of the context and service information of environmental data—including sensor data. The semantic sensor web platform contains a sensor/device row data layer, context virtual sensor data layer, and context-based service information layer. Butt et al. [16] evaluated the existing semantic web databases (such as

Jena, Sesame, and AllegroGraph) to learn about their comparative behaviors and scalability trends. They introduced two core performance metrics: “Resource Utilization” and “Success Ratio”; and one derived metric: “Cumulative Query Performance.” Scalability analysis presented the clear idea that the time and resource utilization of each store increases with the number of triples to manage. They consider two important parameters as a measure of its execution cost for each proposed test case: execution time and resource usage. Their research result shows that Sesame is efficient in searching for the predicate of a triple and more effective than other semantic databases for complex queries. Okuno [17] proposed Linked Open Data for aggregation activities and the implementation of community tourism information content. This service offers sightseeing navigation using the “Hakodate Machiaruki Maps” in providing tourist information around the route of Kakodate's official "Hakobura" guide site. Poslad et al. [18] proposed a novel IoT Early Warning System framework used for natural crisis management that addresses semantic challenges such as the need for scalable, time-sensitive data exchange and processing (especially involving heterogeneous data sources); and the need for resilience to changing ICT resource constraints in crisis zones. They proposed validated lightweight semantics and heavyweight semantics with related metrics (such as mean query time) for importing data into a database. Kwon et al. [19] proposed the best sleep pattern using a semantic sleep management service by analyzing data gathered from healthcare sensors including blood pressure, blood sugar, body temperature, snoring, and sleep apnea. Semantic Web technologies were used to detect the number of sleep apnea cases and snoring times.

To improve the integration and communication among the various sensor node networks—particularly for environmental monitoring systems—this paper proposes a semantic web data platform (SWDP) for managing distributed data sensors—such as temperature, humidity, carbon monoxide (CO), carbon dioxide (CO2), luminosity (LUM), and noise (MCP) sensors—from different servers based on a semantic database system. This paper has the following contributions: 1) Develop an infrastructure for semantic sensor web with environmental sensor data from various sensor types and nodes to promote communication among different sensor node networks, supporting the decision-making process. 2) Rather than the traditional database, transform the process such that the real sensor data from the sensor node is stored in semantic database ontologies. 3) Integrate the sensor data from heterogeneous resources; extract meaningful, shared information from the sensor’s raw data; and thus better enable smart web applications to find, access, and process the sensor data in order to realize the interconnectedness of sensor networks, achieve faster processing time, and reduce memory usage.

Table 1 Comparison of this paper to others related works

No.	Research	Application domain	Input data	The number of sensor sources	QoS Analysis
1	Bovet et al. [3]	Smart Building	Temperature, humidity	Several Raspberry Pi node	-
2	Bispo et al. [4]	Temperature, osciloscope, and anti Theft application	Temperature, ambient light	Six sensor nodes	Power Consumption, The memory usage.
3	Yadaf et al. [8]	DBPedia	WikiPedia	-	-
4	Ali et al. [10]	Smartphone	Smartphone Sensors Data	Smartphone sensors	Accuracy and consistency of ontology
5	Rohloff et al. [11]	University datasets	LUBM Lehigh University ontology datasets	-	Data load times, Query response times.
6	Gray et al. [12]	Flood monitoring	Live sensor data, historic sensor data, databases, map layers	Sea-state around the coast data sensors	-
7	Moraru et al. [13]	Environment Monitoring	Temperature, humidity, luminance, and pressure	One sensor node	-
8	Chenzhou et al. [14]	Healthcare	Diabetes, blood pressure, glucose, care plan	Smart devices of patients	-
9	Butt et al. [16]	MIT Libraries Barton catalog	Barton library dataset	-	Resource utilization, Success ratio, Cumulative query performance.
10	Okuno [17]	Tourism	Maps, film commission, heritages, travel guide	-	-
11	Poslad [18]	Geologic hazards	Simulated tsunami data	-	Query time, utilization
12	Kwon [19	Healthcare	Acceleration, breath and noise sensor	One sensor node	Sleep satisfaction
13	This reseach: SWDP	Environment Monitoring	CO, CO2, Humidity, temperature, Luminosity (LUM) and Noise (MCP) sensors	Ten sensor nodes	Processing time and memory usage

Table 1 shows the results of our paper compared to other related works in terms of application domain, input data type, the number of sensor sources, and quality of service (QoS) analysis.

2. The SWDP System Design

This section presents the SWDP system design—a semantic web platform for environmental WSN that improves processing time and memory usage, and shares information between different sensor networks. This section explains the semantic web system and includes the case study, system architecture, information processing levels, the components of a system, volume data of distribution data sensor, ontology design, ontology RDF model, Sparql design, namespaces prefix, and relational database.

2.1. Case study

With the growing use of WSNs, some countries have begun to monitor various environmental conditions such as CO, CO2, temperature, humidity, luminosity, and noise. In this research, we refer to a case study (as shown in Fig. 1) in which every country has its own monitoring server and database platform. As such, we need to utilize search engines to find the specific data. For example, a search for the greatest CO value in each location and the time that the data was retrieved is conducted by typing: “location”, “co”, “largest”, and “retrieved,” into the search engine.

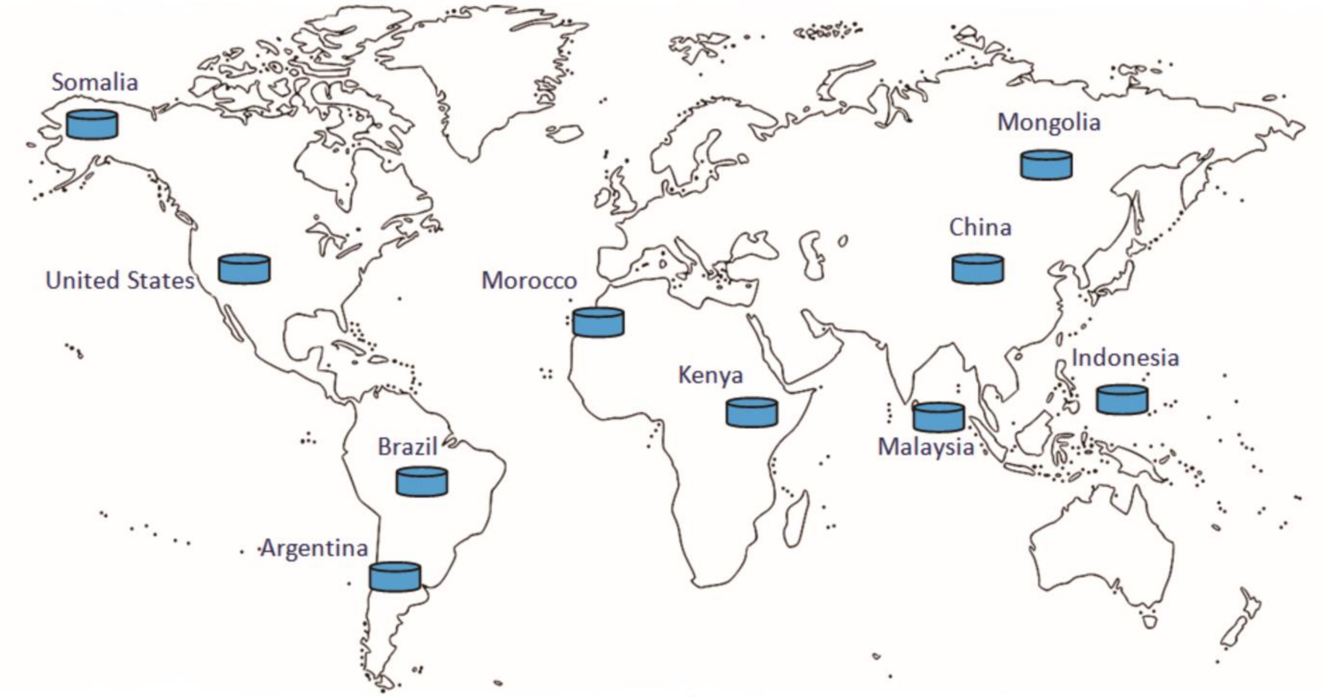

Fig. 1 Semantic web database distribution

Fig. 2 shows a related website of current existing web technology resulting from a search for the largest value of CO data from a CO sensor. If someone needs to know the environmental conditions of other countries, they are required to navigate through the banner link on the front page of the resulting monitoring site in order to access the front page of another country. Ideally, related environmental information should be aggregated and shareable. To achieve this, the implementation of the distributed semantic sensor web database is required to provide shareable and searchable data across country and application boundaries. The integration of data and the facilitation of the search for specific data are done through ‘Direct link,’ which is implemented using the related website of the semantic web, as shown in Fig. 3. As a result, the user can easily obtain the largest and smallest data value, the country location represented by the value, the sensor type, and the location of the sensor within the country.

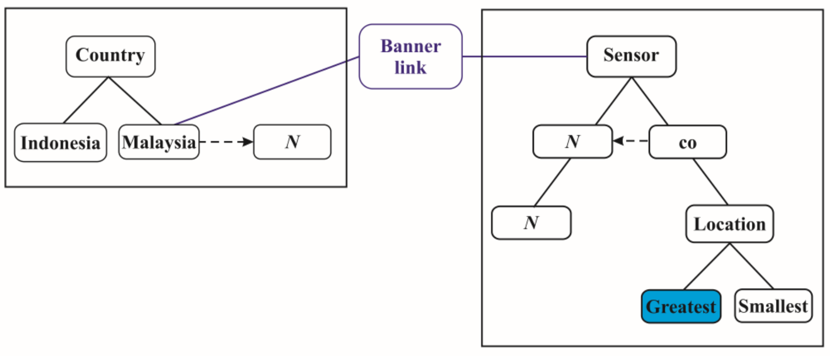

Fig. 2 Related website of current web technology

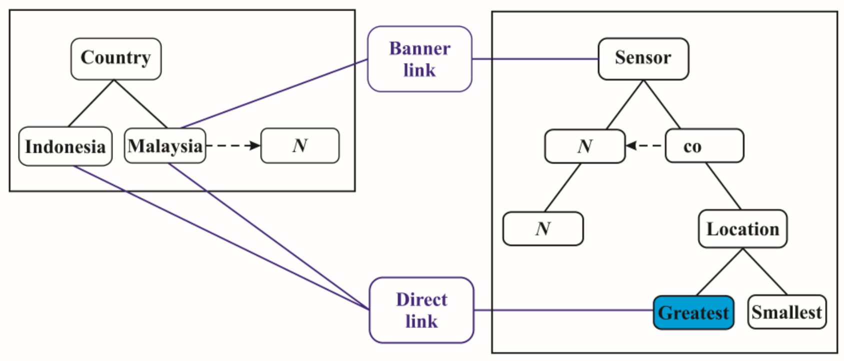

Fig. 3 Related website of semantic web technology

2.2. System Architecture

Currently, we are developing a WSN for the monitoring of environmental conditions in 10 different locations represented by 10 different sensor nodes, as shown in Fig. 4. Every gas sensor node contains a device consisting of a CO sensor (TGS2442), CO2 sensor (TGS4161), temperature sensor (MCP9700A), humidity sensor (808H5V5), luminosity sensor, noise sensor, waspmote gas 2.0 boards, and waspmote PRO 1.2. The gas sensor nodes are distributed and connected to form RDF ontologies as a means of data storage. All data is stored and processed with the SWDP method. The subsequent retrieval of data is conducted by the user with SPARQL.

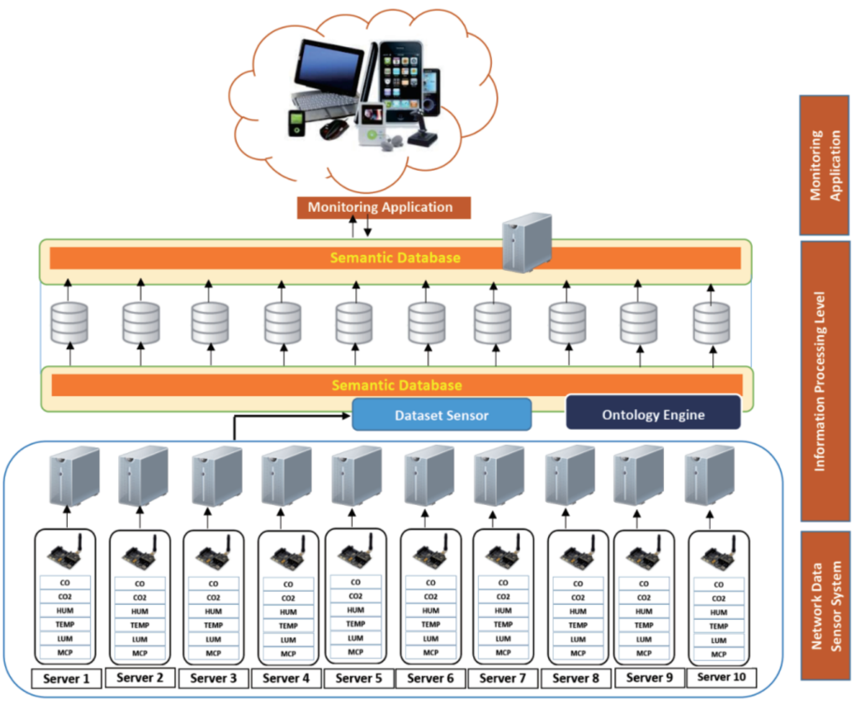

Fig. 4 Proposed system architecture

As presented in Fig. 4, the proposed system architecture is divided into three layers as follows:

2.2.1. Network data sensor system layer

The network data sensor system layer consists of a heterogeneous WSN formed by distributed sensor nodes in different places. Each node has four sensors: one each for temperature, humidity, and carbon monoxide (CO); and one for carbon dioxide (CO2), luminosity, and noise. Data from the WSN are first grouped into different servers, and subsequently stored into one server in the form of a dataset.

We use the TGS2442 sensor for sensing CO conditions, TGS4161 for CO2, and MCP9700A for temperature. The 808H5V5 sensor is used for CO2, luminosity, and noise. These sensors are embedded in the Waspmote Gases 2.0 board which is connected to the Waspmote PRO 1.2 microcontroller.

For communication between nodes and a computer server, we use the XBee module which uses MAC addresses to differentiate between devices. The XBee module is tasked with sending data wirelessly using the IEEE 802.15.4 Zigbee protocol to the server computer where another XBee module is used as the data receiving medium. Data retrieval by the sensor node is done every 60 s and is sent directly to a computer server. Subsequently, the data is processed, categorized, and stored in the semantic database through the information processing layer.

2.2.2. Information processing layer

The information processing layer consists of four levels, as shown in Fig. 5.

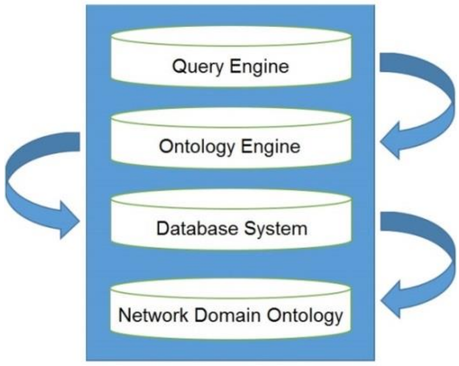

Fig. 5 Information processing levels

(a) Query Engine: The query engine is a system that functions to query the distributed data sensors. This system performs queries on all of the data collected by each of the distributed servers.

(b) Ontology Engine: The ontology engine is a system used to convert the sensor data into RDF form. The system will automatically generate a new relationship based on the data obtained, and some additional information not limited to that pertaining only to the deductive reasoning of the sensor data of each node, but also the combination of the terms, and their relationships.

(c) Database System: The semantic database system is responsible for the interface and method that handles all of the properties, rules, axioms, and relationships for all proposed classes of ontology. Data that has been processed is then stored in a semantic database.

(d) Network Domain Ontology: The network domain ontology serves to collect different uniform resource identifiers (URIs) from ontologies to be used as the overall data source. Each database in a semantic web architecture has a different URI. URIs are used as unique identifiers for concepts in the semantic web.

2.2.3. Monitoring application layer

The monitoring application layer is a medium for displaying data query results from a semantic database. In the monitoring layer, no queries are performed directly on any WSN as presented in the architecture—only on the semantic database. Query results can be from more than one WSN as data from the information processing layer is the result of a combination of multiple distributed WSNs.

The information contained in the semantic database is displayed using PHP programming through the sparqllib library. The information is given in the form of the last value of environmental conditions to facilitate the user's evaluation of the information obtained.

2.3. Component of system

All experiments were performed on sensor nodes, a gateway, and a computer-as-server with specifications as shown in Table 2:

Table 2 The specifications of the hardware and software

Software	
Computer as server	Sensor node and Gateway
a. CPU: i3-4030U CPU @ 1.90ghZ (4 CPUs)	a. Waspmote PRO 1.2
b. Memory: 4096 Mb RAM	b. Waspmote Gases 2.0 board
c. Apache Tomcat	c. CO sensor (TGS2442)
d. Sesame	d. CO2 sensor (TGS4161)
e. Xampp	e. Temperature sensor (MCP9700A)
f. Sparqlib	f. Humidity sensor
g. Protégé	g. Luminosity Sensor
	h. Noise Sensor
	i. Xbee S1 module

Each node installation contains four sensor nodes, including Xbee to connect the node to a computer server. The computer server also has the XBee module used as the data receiving medium. We use Apache server as a web server, Sesame as a semantic data storage media, and SPARQL to query the data from the semantic database.

2.4. Volume data of distribution data sensor

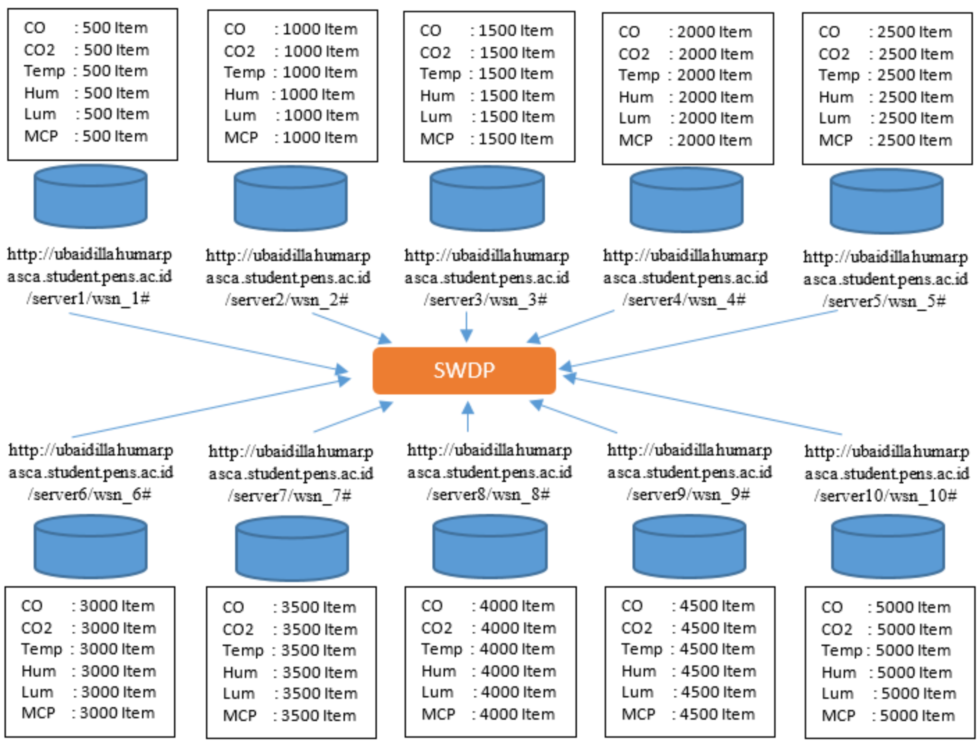

Fig. 6 Volume data of distribution data sensor

Fig. 6 shows the data used to evaluate the proposed semantic query performance. We measured the processing time and memory usage necessary to conduct a search of the contents of 10 semantic databases.

We acquire data from multiple distributed servers, each of which has six sensors–CO, CO2, temperature, humidity, luminosity, and noise. The data is then represented by http://ubaidillahumar.pasca.student.pens.ac.id/server1/wsn_1# for the semantic database on server 1, http://ubaidillahumar.pasca.student.pens.ac.id/server1/wsn_2# for the semantic database on server 2, and likewise up to http://ubaidillahumar.pasca.student.pens.ac.id/server1/wsn_10# for the database on server 10, as shown in table 2.

URI wsn_1 is derived from the database server 1 with the amounts of data at CO=500, CO2=500, temperature=500, humidity=500, luminosity=500, and noise=500 items. The URI wsn_2 is derived from the database server 2 with the amounts of data at CO=1000, CO2=1000, temperature=1000, humidity=1000, luminosity=1000, and noise=1000 items. The URI wsn_3 is derived from the database server 3 with the amounts of data at CO=1500, CO2=1500, temperature=1500, humidity=1500, luminosity=1500, and noise=1500 items. The URI wsn_4 is derived from the database server 4 with the amounts of data at CO=2000, CO2=2000, temperature=2000, humidity=2000, luminosity=2000, and Noise=2000 items. The URI wsn_5 is derived from the database server 5 with the amounts of data at CO=2500, CO2=2500, temperature=2500, humidity=2500, luminosity=2500, and noise=2500 items. The URI wsn_6 is derived from the database server 6 with the amounts of data at CO=3000, CO2=3000, temperature=3000, humidity=3000, luminosity=3000, and noise=3000 items. The URI wsn_7 is derived from the database server 7 with the amounts of data at CO=3500, CO2=3500, temperature=3500, humidity=3500, luminosity=3500, and noise=3500 items. The URI wsn_8 is derived from the database server 8 with the amounts of data at CO=4000, CO2=4000, temperature=4000, humidity=4000, luminosity=4000, and noise=4000 items. The URI wsn_9 is derived from the database server 9 with the amounts of data at CO=4500, CO2=4500, temperature=4500, humidity=4500, luminosity=4500, and noise=4500 items. The last URI, wsn_10, is derived from the database server 10 with the amounts of data at CO=5000, CO2=5000, temperature=5000, humidity=5000, luminosity=5000, and noise=5000 items.

2.5. Ontology design

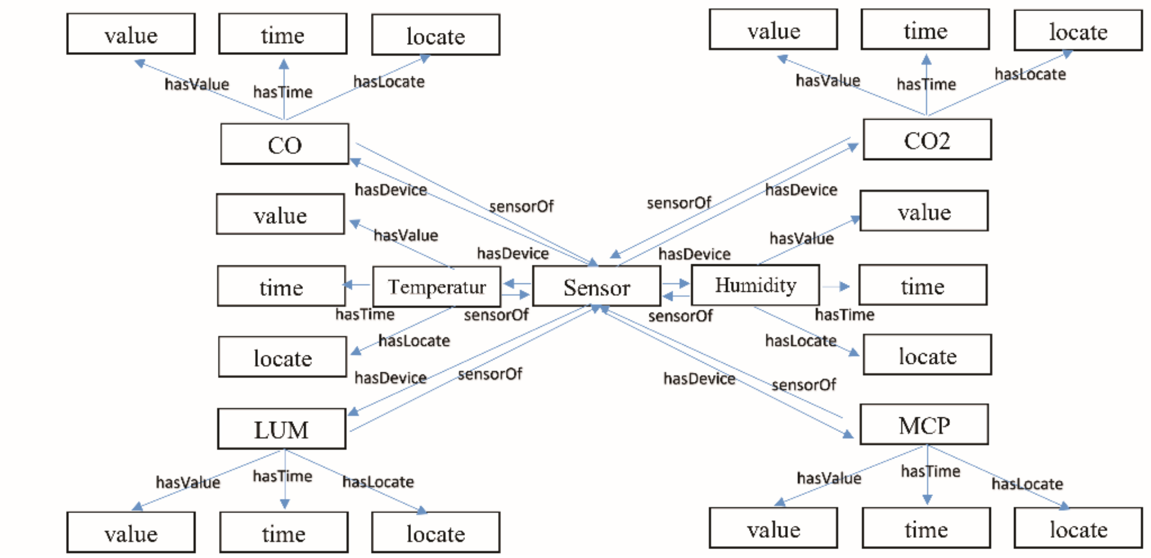

Fig. 7 Proposed ontology design

Ontology is usually expressed in logical language to describe the detailed, accurate, consistent, healthy, and meaningful differences made between classes, properties, and relationships. The ontology of each sensor has a different URI. URIs from each of the different sensors are gathered through network domain ontology. The ontology can be represented in the form of several formats—namely RDF, resource description framework schema, and OWL. Fig. 7 shows the ontology for the domain of environmental health monitoring. In the ontology, there are four main classes that describe the knowledge base data from the sensors about environmental conditions—namely, classes CO2, CO, temperature, humidity, luminosity, and noise. The four classes contain each value, time, and location.

In this ontology, we propose a semantic system that controls the use of the information gathered and manages the data received by the server. In this system, we collect sensor data (such as temperature, humidity, CO, CO2, luminosity, and noise) from the sensor nodes. Subsequently, the data entry is processed, categorized with RDF, and stored in a semantic database. Sensors in this domain ontology are represented as a WSN node that is part of a class of data coverage of all sensors connected to send and receive messages from the sensor nodes.

2.6. *Ontology RDF model*

OWL is a family of knowledge representation languages for authoring ontologies. Ontology is a format used to describe the taxonomy and classification network, and it essentially defines the knowledge structure for a variety of domains. It contains nouns, which represent the object classes, and verbs, which represent the relationships between objects. Ontology resembles the class hierarchy in object-oriented programming.

A class hierarchy is meant to represent a structure used in the source code that evolved fairly slowly, while an ontology is meant to represent the information on the Internet and is expected to grow continuously. Similarly, an ontology is usually much more flexible because it is meant to represent the information on the Internet as coming from all sorts of heterogeneous data sources. Alternatively, a class hierarchy is meant to be fairly static, relying on much more diversified sources and more structured data.

Each sensor—CO, CO2, humidity, temperature, luminosity, and noise—on http://ubaidillahumar.pasca.student.pens.ac.id/wsn_1 had three data URI properties that include: http://ubaidillahumar.pasca.student.pens.ac.id/wsn_1#co_locate, to store location data about where the sensor is placed; http://ubaidillahumar.pasca.student.pens.ac.id/wsn_1#co_time, for storing retrieval data time; and http://ubaidillahumar.pasca.student.pens.ac id / wsn_1#co_value, to store sensor data values.

2.7. *Sparql design*

SPARQL language allows us to perform a query consisting of three patterns: conjunctions, disjunctions, and optional patterns. Implementations for multiple programming languages exist. There are tools that allow one to connect and semi-automate the building of a SPARQL query a to SPARQL endpoint (for example, ViziQuer). In addition, tools exist that translate SPARQL queries to other query languages. Code 1 shows a pseudocode SPARQL design to get data CO sensor from semantic.

```
Query_value_time_datashet ()
{
        Step 1: Find all the co values found on datashet wsn_1.
        Step 2: Find all data when the sensor data retrieved in datashet wsn_1.
        Step 3: Save the search results co value found in datashet wsn_1 divariable resultvalue
        Step 4: Save the search results of the co value found in the datashet wsn_1 into the fieldsvalue
                variable in the form of the array.
        Step 5: Save search results data retrieval of sensor data contained in datashet wsn_1 divariabel
                resulttime.
        Step 6: Save the data retrieval results of sensor data contained in datashet wsn_1 into fieldstime
                variable in array form.
}
```

Fig. 8 Code 1: Pseudocode SPARQL design

2.8. *Namespaces prefix*

Namespaces provide a method to avoid conflicting name review elements. We set the OWL to begin with a Namespace declaration, as shown in Fig. 9.

The RDF uses the prefixes “RDF” and “wsn_1” to condense the full address URI. RDF refers to the RDF syntax, while wsn_1 refers to sensor data elements in wsn_1. Both of these prefixes are also called the namespace. In namespace, although we refer to the URI from a schema, the address does not always have to be present as there is no validation activity carried out. This was done to avoid the use of ambiguous elements. The namespace prefix is part of the XML qualified name (QName), which is used to facilitate the writing of RDF. QName consists of a prefix and a local name. For example, wsn_1 is the short name of the URI, http://ubaidillahumar.pasca.student.pens.ac.id/wsn_1#.

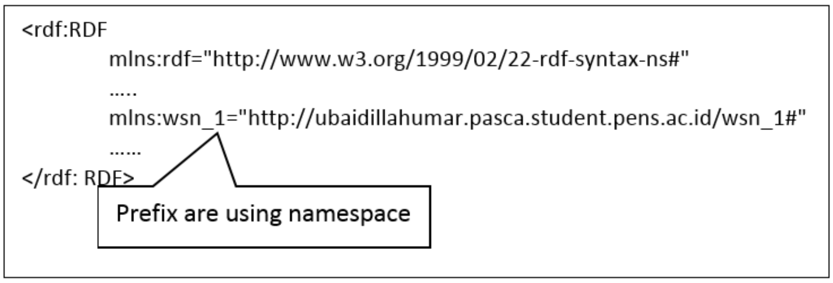

Fig. 9 Namespaces prefix

2.9. *Relational database*

In addition to the ontology RDF model design, we build a relational database to be compared with the semantic database in terms of processing time and memory usage. A relational database is a collection of structured data organized into tables which can be interconnected (related) with one another. This type is a form of conventional database—when someone mentions the word "database," the general public will imagine a relational database (a collection of tables), as shown in Fig. 10.

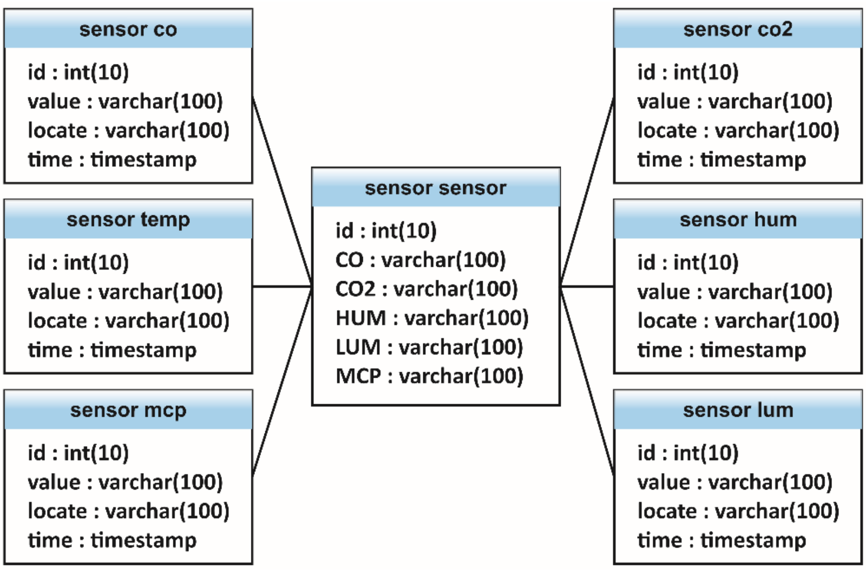

Fig. 10 Relational database

Each table has a primary key in the database. The primary key of a table provides the unique identifying value for a particular row. The relational database above is formed by making comparisons between the primary keys of each sensor table. Relationships between tables define the foreign keys (sensor CO, CO2, temperature, humidity, luminosity, and noise) and imply structural relationships.

Relational databases require that data structures (schemas) are defined before data is entered in order to improve data integrity for application use. Relational databases cannot handle unstructured data which may lead to data storage systems that do not require the definitions of previous data structures [6].

3. Experimental Procedure

The purpose of this experimental evaluation is to compare the semantic and relational databases in terms of the processing time and memory usage required to process the sensor data.

In this section, we describe the test network topology that we use for testing a relational database against a semantic database. Fig. 11 shows the topology used to perform the test of the relational database. In this topology, Apache server is used as a web server, MySQL as the relational database, and SQL query to query the database.

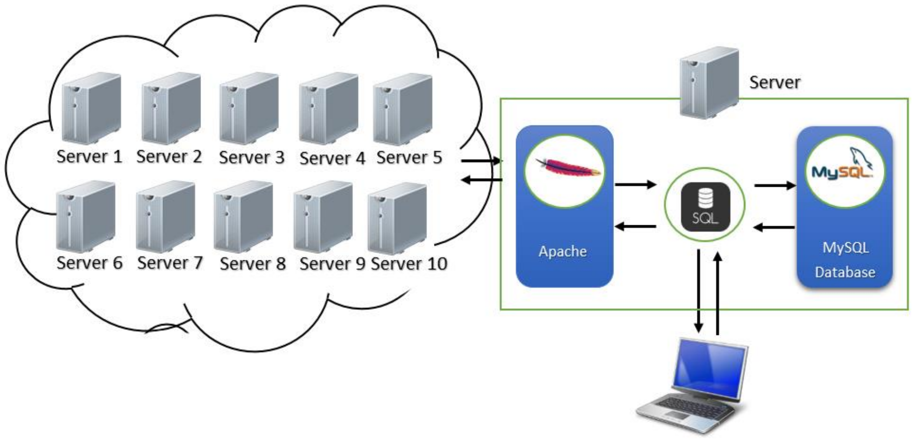

Fig. 11 Relational database topology test

Fig. 12 shows the topology for the test of the semantic network database. In this topology, Apache Tomcat is used as the web server, Sesame as the semantic database, and SPARQL query to query the database.

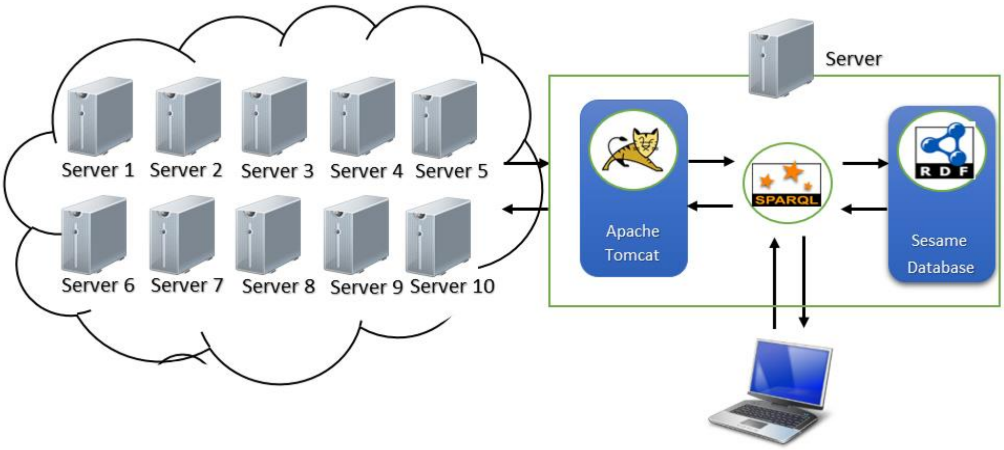

Fig. 12 Semantic database topology test

To evaluate the performance of the proposed semantic database system, we take measurements of the processing time, search speed, and memory usage for the data search process. Measurements were made using two scenarios. The first involves taking measurements by counting the processing time and memory usage of all databases simultaneously, and can be described as follows:

Database 1 + database 2 + database 3 + database 4 + database 5 + database 6 + database 7 + database 8 + database 9 + database 10

The second scenario is performed by measuring the alternate database by adding the previous database, as follows:

1. Database 1
2. Database 1 + database 2
3. Database 1 + database 2 + database 3
4. Database 1 + database 2 + database 3 + database 4
5. Database 1 + database 2 + database 3 + database 4 + database 5
6. Database 1 + database 2 + database 3 + database 4 + database 5 + database 6
7. Database 1 + database 2 + database 3 + database 4 + database 5 + database 6 + database 7
8. Database 1 + database 2 + database 3 + database 4 + database 5 + database 6 + database 7 + database 8
9. Database 1 + database 2 + database 3 + database 4 + database 5 + database 6 + database 7 + database 8 + database 9
10. Database 1 + database 2 + database 3 + database 4 + database 5 + database 6 + database 7 + database 8 + database 9 + database 10

To measure the processing time (*PTime*) required for querying, we use the following Eq. (1):

$$PTime = POpenConnection + PExecuteQuery + PPrintResultSet + PCloseConnection \quad (1)$$

where Processing Time (*PTime*) can be obtained by the addition of the open connection time (*POpenConnection*), execution time (*PExecuteQuery*), query results display time (*PPrintResultSet*), and disconnection time (*PCloseConnection*) of the query process.

Subsequently, to the measure memory usage (*MUsage*) during the query process, we use the following Eq. (2):

$$MUsage = MEnd - MStart \quad (2)$$

where *MUsage* can be obtained from the reduction of the memory usage when the query is executed (*MEnd*) by the memory usage before the query is executed (*MStart*).

Fig. 13 shows the resulting processing times from the relational and semantic databases using scenario 1. The data are taken for as many as 10 iterations where the average processing time is 0.1288 s for the relational database, and 0.0932 s for the semantic database. The processing time of the semantic database is less than that of the relational database. The values on the graph representing the relational database tend to rise and be unstable compared to the semantic database.

Fig. 14 shows the resulting memory usage in scenario 1. The data are taken for as many as 10 iterations. On average, the memory usage required to run the relational database system is 868,160 bytes, while that of the semantic database is 4,936 bytes—this result highlights the vast performance difference between the two.

Then, Fig. 15 shows the resulting processing time using scenario 2. The average processing time required by the relational database for running the query is 3.9054 s, while that of the semantic database is 3.8428 s. In the second scenario, we see that the processing time of the semantic database is slightly lower than that of the relational database.

Processing Time (Second)	
Relational Database	Semantic Database
0,102	0,107
0,137	0,098
0,123	0,102
0,116	0,093
0,102	0,084
0,125	0,078
0,133	0,098
0,129	0,102
0,189	0,089
0,132	0,081

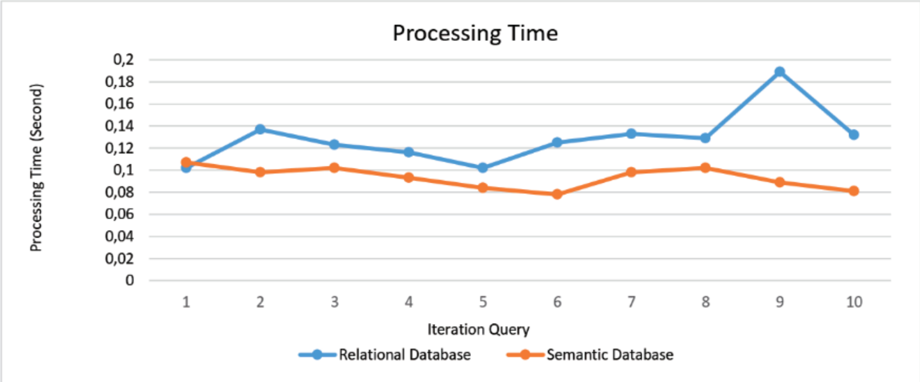

Fig. 13 Processing time in scenario 1

Memory Usage (Byte)	
Relational Database	Semantic Database
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936
868.160	4.936

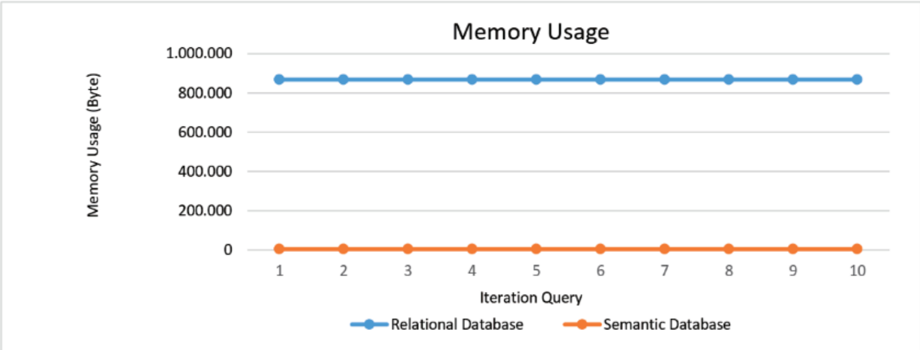

Fig. 14 Memory usage in scenario 1

Processing Time (Second)	
Relational Database	Semantic Database
3,979	3,953
3,952	3,909
3,941	3,892
3,934	3,856
3,926	3,841
3,919	3,832
3,91	3,809
3,89	3,8
3,849	3,791
3,754	3,745

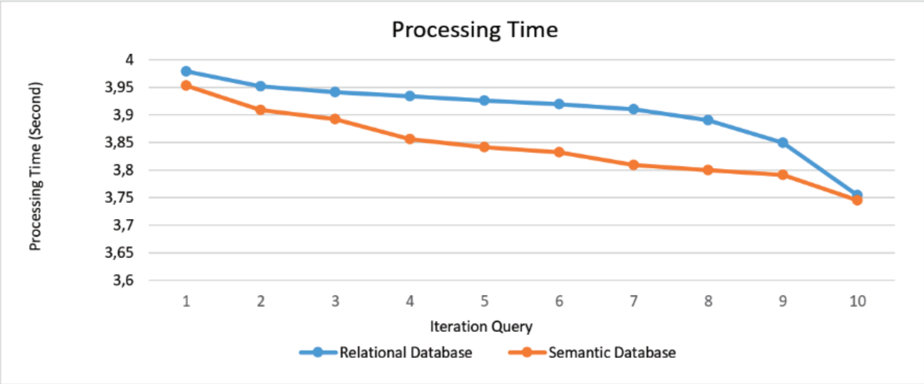

Fig. 15 Processing time in scenario 2

Fig. 16 shows the resulting memory usage in scenario 2. On average, the memory usage required to run the relational database system is 354,883 bytes, while that of the semantic database is 2,605 bytes. This result demonstrates that the memory usage of the semantic database is much less than that of the relational database. In this second scenario, the memory usage of the relational database tends to increase while that of the semantic database exhibits very little change.

Memory Usage (Byte)	
Relational Database	Semantic Database
16.232	344
49.664	1.328
97.992	1.712
162.336	2.088
242.928	2.472
339.264	2.856
451.600	3.240
579.936	3.624
724.272	4.000
884.608	4.384

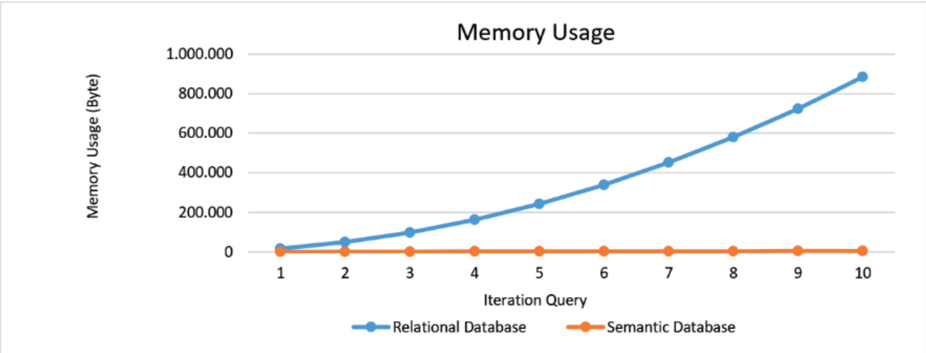

Fig. 16 Memory usage in scenario 2

Processing Usage (Second)	
Relational Database	Semantic Database
0,38	0,112
0,238	0,229
0,256	0,168
0,187	0,041
0,189	0,068
0,217	0,072
0,2	0,043
0,213	0,076
0,277	0,131
0,313	0,16

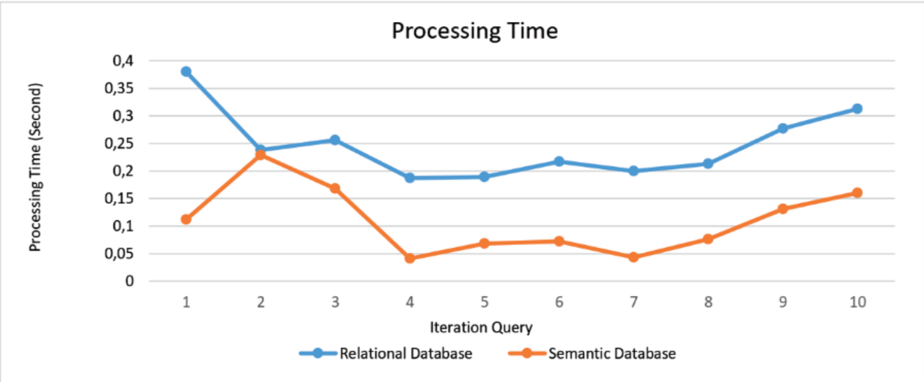

Fig. 17 Processing time on a specific data search process

Fig. 17 shows the resulting processing times of the relational and semantic databases using scenario 1 to search for specific data in a location. Data were taken for as many as 10 iterations where the average processing time was 0.247 s for the relational database and 0.11 s for the semantic database. The processing time of the semantic database is lower than that of the relational database. The values on the graph representing the relational database tend to increase and are unstable compared to those of the semantic database.

Fig. 18 shows the resulting memory usage required to search for specific data. Here, the average memory usage required by the relational database system is 7,848 bytes, while that of the semantic database system is 4,448 bytes. This result shows that the memory usage by the semantic database system is lower than that of the relational database.

Memory Time (Byte)	
Relational Database	Semantic Database
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448
7.848	4.448

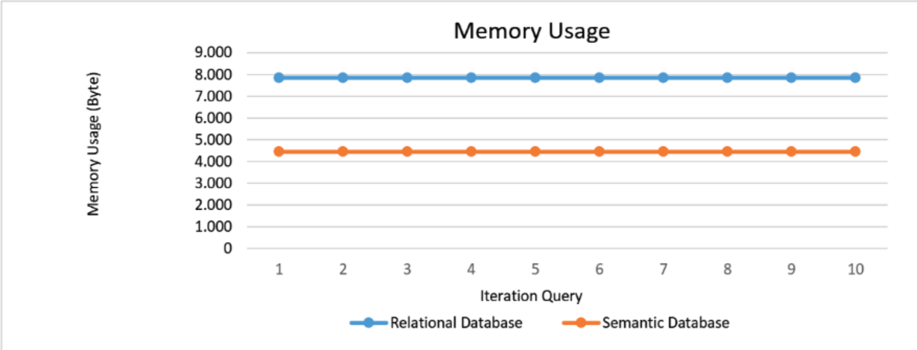

Fig. 18 Memory usage on a specific data search process

Largest data on location 1

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co1_79	lokasi1	8.11	42712.915868055556

Largest data on location 2

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co2_69	lokasi2	9.92	42712.915034722224

Largest data on location 3

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co3_193	lokasi3	8.27	42712.92523148148

Largest data on location 4

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co4_120	lokasi4	8.85	42712.919282407405

Largest data on location 5

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co5_155	lokasi5	8.81	42712.92212962963

Largest data on location 6

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co6_60	lokasi6	8.17	42712.914293981485

Largest data on location 3

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co7_399	lokasi7	8.22	42712.94174768519

Largest data on location 8

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co8_445	lokasi8	8.69	42712.94559027778

Largest data on location 9

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co9_500	lokasi9	8.21	42712.95104166667

Largest data on location 10

Data	Room	value	Time
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co10_461	lokasi10	8.7	42712.946909722225

The Largest CO Data is at

Data	value	Room
http://ubaidillahumar.pasca.student.pens.ac.id/co_all#co2_69	9.92	lokasi2

Fig. 19 Search result from semantic web database

Fig. 19 shows the search results based on the relevance of the integrated data to the user. The user can easily find the locations that have the largest and smallest specific data. For example, Fig. 17 shows that the largest CO value is at location 2, with a value of 9.92 PPM.

This proves far more efficient than the type of relational database that requires entering commands like, “location,” “co,” “largest,” and “Number” into the search form in order to produce this sort of result.

For monitoring applications, we provide a publicly accessible website to display the sensor data stored in the semantic database. A dashboard containing the sensor menu will be displayed when a user accesses the website. We can see the results of data retrieval from the database of semantic web censorship, as shown in Fig. 20. There are four menus in the dashboard to display the sensor results for CO, CO2, humidity, temperature, luminosity, and noise for 10 locations.

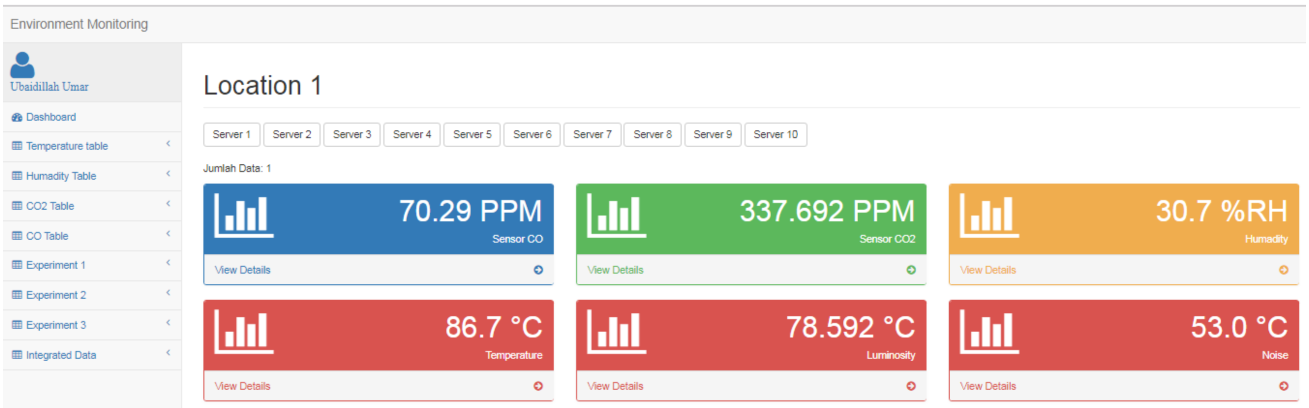

Fig. 20 Monitoring application

To view the overall sensor values stored in the database, we provide a table view for the tenth location of each sensor, as shown in Fig. 21.

CO Table

Server 1 | Server 2 | Server 3 | Server 4 | Server 5 | Server 6 | Server 7 | Server 8 | Server 9 | Server 10

Data	Value	Room	Time
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_1	70.29	Location1	42712.90956018519
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_10	43.757	Location1	42712.910219907404
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_100	70.29	Location1	42712.917604166665
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_101	70.29	Location1	42712.91768518519
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_102	70.29	Location1	42712.91777777778
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_103	46.604	Location1	42712.917858796296
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_104	36.766	Location1	42712.91793981481
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_105	70.29	Location1	42712.918020833335
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_106	46.346	Location1	42712.91811342593
http://ubaidillahumar.pasca.student.pens.id/wsn_1#co_107	70.29	Location1	42712.91819444444

Showing 1 to 10 of 500 entries

Fig. 21 CO sensor data table in the database

4. Experimental Procedure

There are expanding requirements for sensor data to be sharable crosswise over various sensors, database systems, and application boundaries. In the relational database framework, these requirements are difficult to address. In this paper, we have proposed and implemented an SWDP to manage the distribution of sensor data from different sensor nodes, based on the semantic database system. The SWDP framework contains a network data system, information processing level, and monitoring application to both gather sensor information and minimize and simplify the processing of information. The nodes of the SWDP framework utilize temperature, humidity, CO, and CO2 sensors. The SWDP framework is expected to effectively monitor environmental conditions, and produce shareable, searchable sensor data from different servers and database platforms. The results demonstrated that the SWDP framework with the semantic database system performed better than the existing traditional database system in terms of memory usage and processing time.

Acknowledgement

This research was supported in part by Ministry of Research, Technology and Higher Education of Indonesia (KEMENRISTEKDIKTI), under Grant INSINAS Scheme 2017.

References

[1] W. Y. Yi, K. M. Lo, T. Mak, K. S. Leung, Y. Leung, and M. L. Meng, “A survey of wireless sensor network based air pollution monitoring systems,” Sensors, vol. 15, no. 12, pp. 31392-31427, December 2015.

[2] M. U. H. Al Rasyid, I. U. Nadhori, A. Sudarsono, and Y. T. Alnovinda, “Pollution monitoring system using gas sensor based on wireless sensor network,” International Journal of Engineering and Technology Innovation, vol. 6, no. 1, pp. 79-91, January 2016.

[3] G. Bovet and J. Hennebert, “Distributed semantic discovery for web-of-things enabled smart buildings,” 6th International Conf. New Technologies, Mobility and Security (NTMS), April 2014.

[4] K. A. Bispo, N. S. Rosa, and P. R. F. Cunha, “SITRUS: Semantic Infrastructure for Wireless Sensor Networks,” Sensors, vol. 15, no. 11, pp. 27436-27469, November 2015.

[5] M. U. H. Al Rasyid, A. Sayfudin, A. Basofi, and A. Sudarsono, “Development of semantic sensor web for monitoring environment conditions,” Proc. Intelligent Technology and Its Application (ISITIA), IEEE Press, July 2016, pp. 607-612.

[6] S. Zhao and E. Chang, “From database to semantic web ontology: an overview,” Lecture Notes in Computer Science (LNCS), Springer, vol. 4806, pp. 1205-1214, 2007.

[7] M. Koubarakis and K. Kyzirakos, “Modeling and querying metadata in the semantic sensor web: the model stRDF and the query language stSPARQL,” Lecture Notes in Computer Science (LNCS), Springer, vol. 6088, pp. 425-439, 2010.

[8] H. N. Yadav, and R.V. Patel, "DBpedia Based SAWSDL for Semantic Web Services," Proc. 2nd International Conf. on Computing for Sustainable Global Development, IEEE Press, March 2015 pp. 35-39.

[9] P. Ristoski and H. Paulheim, "Semantic web in data mining and knowledge discovery: a comprehensive survey," Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol. 36, pp. 1-22, January 2016.

[10] S. Ali, S. Khusro, I. Ullah, A. Khan, and I. Khan, "Smartontosensor: ontology for semantic interpretation of smartphone sensors data for context-aware applications," Journal of Sensors, vol. 2017, pp. 8790198-1-8790198-26, February 2017.

[11] K. Rohloff, M. Dean, I. Emmons, D. Ryder, and J. Sumner, "An evaluation of triple-store technologies for large data stores," Lecture Notes in Computer Science, Springer, vol. 4806, pp. 1105-1114, 2007.

[12] A. J. G. Gray, J. Sadler, O. Kit, K. Kyzirakos, M. Karpathiotakis, J. P. Calbimonte, K. Page, R. G. Castro, A. Frazer, I. Galpin, A. A. A. Fernandes, N. W. Paton, O. Corcho, M. Koukabaris, D. D. Roure, K. Martinez, and A. G. Perez, "A semantic sensor web for environmental decision support applications," Sensors, vol. 11, no. 9, pp. 8855-8887, September 2011.

[13] A. Moraru, C. Fortuna, and D. Mladenic, "A system for publishing sensor data on the Semantic Web," Proc. 33rd International Conf. on Information Technology Interfaces, IEEE Press, June 2011.

[14] Y. Chenzhou, W. J. Shen, W. Sun, G. Hu, J. Li, L. H. Xu, X. Z. Sun, and Y. Pan, "Active linked data for human centric semantic web of things," Proc. IEEE International Conf. on Green Computing and Communications, IEEE Press, November 2012, pp. 9-16.

[15] H. S. Choi and W. S. Rhee, "Distributed semantic sensor web architecture," Proc. IEEE Region 10 Conf. TENCON, IEEE Press, November 2012, pp. 1-6.

[16] A. S. Butt and S. Khan, "Scalability and performance evaluation of semantic web databases," Arabian Journal for Science and Engineering, vol. 39, no. 3, pp. 1805-1823, March 2014.

[17] T. Okuno, "Aggregation and application of community tourism information contents by using linked open data," Proc. IEEE International Symposium on Soft Computing and Intelligent Systems, IEEE Press, December 2014.

[18] S. Poslad, S. E. Middleton, F. Chaves, R. Tao, O. Necmioglu, and U. Bugel, "A semantic IoT early warning system for natural environment crisis management," Proc. IEEE Transaction on Emerging Topic in Computing, IEEE Press, May 2015.

[19] S. Kwon, D. Park, H. Bang, "Semantic sleep management service in healthcare sensor networks," Proc. IEEE International Conference onConsumer Electronics, IEEE Press, January 2014, pp. 268-269.